



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**“DISEÑO DE UN SISTEMA EMBEBIDO DE DETECCIÓN Y RASTREO DE MASCOTAS MEDIANTE UNA RED IoT APLICADO EN LA CIUDAD DE IBARRA.”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**AUTOR: FREDY DARIO CHICAIZA CHUQUILLA**

**DIRECTOR: MSC. CARLOS ALBERTO VÁSQUEZ AYALA**

**Ibarra-Ecuador**

**2023**



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD**

**TÉCNICA DEL NORTE**

**IDENTIFICACIÓN DE LA OBRA.**

En cumplimiento del Art. 144 de la Ley Orgánica de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte de manera digital para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

<b>DATOS DEL CONTACTO</b>			
<b>CÉDULA DE IDENTIDAD:</b>	1004070676		
<b>APELLIDOS Y NOMBRES:</b>	Chicaiza Chuquilla Fredy Dario		
<b>DIRECCIÓN:</b>	Calle Jaime Félix y Hugo Guzmán Lara. Cdla La Victoria		
<b>EMAIL:</b>	fredychicaiza97@gmail.com		
<b>TELÉFONO FIJO:</b>		<b>TELÉFONO MÓVIL</b>	0988262949
<b>DATOS DE LA OBRA</b>			
<b>TÍTULO:</b>	Diseño de un Sistema Embebido de detección y rastreo de mascotas mediante una Red IoT aplicado en la ciudad de Ibarra.		
<b>AUTOR:</b>	Chicaiza Chuquilla Fredy Dario		
<b>FECHA:</b>	15 de Febrero de 2023		
<b>PROGRAMA:</b>	Pregrado		
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniería en Electrónica y Redes de Comunicación		
<b>ASESOR/DIRECTOR:</b>	Ing. Carlos Alberto Vásquez Ayala, MSc		

## CONSTANCIAS.

El autor manifiesta que la obra de la presente autorización es original y se la realizo sin violar derechos de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que se asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la universidad en caso de reclamación por parte de terceros

Ibarra, a los 15 días del mes de Febrero de 2023

**EL AUTOR.**



.....

Chicaiza Chuquilla Fredy Dario

**C.I.:1004070676**



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CERTIFICACIÓN**

MAGISTER VÁSQUEZ AYALA CARLOS ALBERTO, Msc, DIRECTOR  
DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente trabajo de Titulación “DISEÑO DE UN SISTEMA  
EMBEBIDO DE DETECCIÓN Y RASTREO DE MASCOTAS MEDIANTE UNA  
RED IoT APLICADO EN LA CIUDAD DE IBARRA” ha sido realizada en su  
totalidad por: Sr. Fredy Dario Chicaiza Chuquilla, bajo mi supervisión:

Es todo en cuanto puedo certificar en honor de la verdad.

.....  
Ing. Carlos Alberto Vásquez Ayala, Msc

Director de Tesis

## **DEDICATORIA**

*Dedico este trabajo:*

*Con todo el amor y cariño a mis padres Fredy y Piedacita, quienes fueron mi principal fuente de apoyo, mi aliento cuando más necesitaba de ellos, quienes me enseñaron que incluso la tarea más grande se puede lograr si se hace un paso a la vez y quienes me motivan día a día en la construcción de mis sueños. Su bendición diaria a lo largo de mi vida me protege y me*

*lleva por el camino del bien.*

*A mis hermanas Erika y Mishell, por su cariño, por sus palabras de aliento y porque siempre fueron una inspiración para poder culminar este Objetivo.*

***Fredy Chicaiza.***

## **AGRADECIMIENTO**

*Siempre agradecido con Dios quien en todo momento está conmigo, por guiar mi camino  
y darme la confianza y fuerza para seguir adelante.*

*Mi eterno agradecimiento a mis padres, hermanas y mi sobrina chabe por que han sido  
el principal cimiento para la construcción de mi vida profesional, con su cariño y amor  
incondicional que me brindan cada día, ayudándome a superar las dificultades que se presentan  
en mi diario vivir.*

*Agradezco a los docentes de la carrera, en especial a mi director, el MSc. Calos Vásquez  
quien compartió su conocimiento y me supo guiar en el desarrollo y culminación de este trabajo  
de titulación.*

*A mis compañeros y amigos de la Universidad Will, Chein, Guano, Max, Macas, Axel,  
Adonis, Vicente, quienes sin esperar nada a cambio compartieron su conocimiento, buenos y  
malos momentos en el día a día de formación académica. A mis amigos de toda la vida Jhon,  
Byron y Jonathan que son como mis hermanos y se hicieron presentes con sus palabras de apoyo  
a lo largo de esta etapa, dándome ánimos para que siga adelante.*

***Fredy Chicaiza.***

## ÍNDICE DE CONTENIDO

IDENTIFICACIÓN DE LA OBRA.....	I
CONSTANCIAS.....	II
CERTIFICACIÓN .....	III
DEDICATORIA .....	IV
AGRADECIMIENTO .....	V
ÍNDICE DE CONTENIDO .....	VI
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE FIGURAS DE ANEXOS .....	XV
ÍNDICE DE TABLAS .....	XVI
ÍNDICE DE ECUACIONES .....	XVII
RESUMEN .....	XVIII
ABSTRACT.....	XIX
1. CAPÍTULO: ANTECEDENTES.....	1
1.1. Tema .....	1
1.2. Problema .....	1
1.3. Objetivos .....	3
1.3.1. Objetivo General.....	3
1.3.2. Objetivos Específicos.....	3
1.4. Alcance .....	3
1.5. Justificación .....	6
2. CAPÍTULO: FUNDAMENTACIÓN TEÓRICA.....	8
2.1. Mascota .....	8

2.2. Sistemas Embebidos .....	8
2.3. Internet de las Cosas (IoT).....	9
2.4. Cloud.....	10
2.5. Global Positioning System(GPS).....	10
2.5.1. Arquitectura .....	11
2.5.2. Funcionamiento.....	12
2.5.3. Pseudodistancia.....	13
2.5.4. Aplicaciones.....	13
2.5.5. Códigos GPS.....	14
2.5.6. Servicios.....	14
2.6. Google Maps .....	15
2.7. Dispositivo GPS utilizados en animales .....	15
2.8. Telefonía Celular .....	16
2.8.1. Red Celular GSM.....	16
2.8.1.1. Arquitectura del sistema GSM.....	17
2.8.1.2. Rango de Frecuencias GSM.....	18
2.8.1.3. Servicios de GSM .....	19
2.8.2. General Packet Radio Service (GPRS).....	20
2.8.2.1. Arquitectura de una red GPRS.....	20
2.8.2.2. Protocolo GPRS .....	22
2.8.2.3. Dispositivos GPRS.....	24
2.9. Diseño del sistema electrónico.....	25
2.9.1. Modelo en V .....	25

2.9.2. Benchmark .....	27
2.9.3. IEEE 29148 .....	27
2.10. Placas de desarrollo y Módulos .....	29
2.10.1. Placa programable eficiente para la compatibilidad de los módulos GPS y GSM.....	29
2.10.2. Módulos ideales para el rastreo de objetos .....	30
3. CAPITULO III Desarrollo Experimental .....	32
3.1. Metodología .....	32
3.2. Análisis de la situación Actual.....	33
3.2.1. Análisis de Resultados .....	34
3.3. Introducción al Desarrollo del Proyecto .....	35
3.3.1. Propósito del Sistema.....	35
3.3.2. Ámbito del Sistema.....	36
3.3.3. Características de los beneficiarios.....	36
3.4. Requerimientos del Proyecto .....	37
3.4.1. Stakeholders .....	37
3.4.2. Construcción de atributos de los requerimientos .....	38
3.4.2.1. Nomenclatura de los requerimientos .....	38
3.4.3. Requerimientos de Stakeholders.....	38
3.4.4. Requerimientos de Sistema.....	41
3.4.5. Requerimientos de Arquitectura .....	44
3.5. Selección de Hardware y Software .....	50
3.5.1. Selección de Hardware .....	50

3.5.1.1. Elección de Placa de desarrollo .....	50
3.5.1.2. Elección del módulo de comunicación inalámbrica GPRS .....	53
3.5.1.3. Elección del módulo GPS .....	56
3.5.2. Selección de Software.....	58
3.5.2.1. Software para la programación de la Placa de desarrollo Arduino Nano.....	59
3.5.2.2. Plataforma de servicio (Cloud) .....	60
3.5.2.3. Software para el desarrollo de la aplicación móvil.....	62
3.6. Recursos.....	63
3.6.1. Recursos Económicos de Hardware.....	63
3.6.2. Recursos Tecnológicos de Software .....	64
3.6.3. Costos de Ingeniería.....	65
3.6.4. Costos varios y de Oficina .....	66
3.6.5. Costo Total del Sistema .....	66
3.7. Diseño del Sistema.....	67
3.7.1. Descripción general del funcionamiento del sistema PETROUTE .....	67
3.7.2. Arquitectura .....	69
3.7.2.1. Diagrama de Flujo del proceso que realiza el Sistema ....	71
3.8. Diseño a detalle del Sistema PETROUTE.....	74
3.8.1. Diseño del bloque de Alimentación.....	74
3.8.1.1. Cálculos de la fuente de Alimentación .....	74
▪ Regulador de Voltaje.....	74

▪	Divisor de Voltaje .....	75
3.8.1.2.	Diseño de la fuente de Alimentación .....	77
3.8.2.	Diseño del bloque de toma de Datos y bloque de Procesamiento .....	77
3.8.2.1.	Bloque de Toma de Datos .....	77
3.8.2.2.	Bloque de Procesamiento .....	80
3.8.3.	Diseño del bloque de Transmisión de Datos .....	82
3.8.3.1.	Uso de Comandos AT .....	83
3.8.4.	Diseño del bloque de Visualización .....	89
3.9.	Implementación .....	91
3.9.1.	Implementación de Hardware .....	92
3.9.1.1.	Diseño de Placa Electrónica .....	92
3.9.1.2.	Construcción de Placa Electrónica .....	93
3.9.1.3.	Estructura Física de la Pechera .....	95
3.9.2.	Implementación de Software .....	96
3.9.2.1.	Bloque de Toma de Datos y Procesamiento .....	96
3.9.2.2.	Geocerca .....	99
3.9.2.3.	Bloque de Transmisión de Datos .....	100
3.9.2.4.	Bloque de Visualización .....	104
•	Conexión con Firebase .....	105
•	Aplicación Móvil de PETROUTE .....	110
4.	CAPITULO IV Pruebas de Funcionamiento y Resultados .....	114
4.1.	Especificación de pruebas a desarrollar .....	114
4.2.	Pruebas de cumplimiento .....	115

4.3.	Pruebas de Funcionalidad .....	117
4.3.1.	Bloque de alimentación.....	117
4.3.1.1.	Tiempo de Vida del Sistema .....	120
4.3.1.2.	Tiempo de Carga y Descarga de la Batería.....	121
4.3.2.	Bloque de toma de datos y procesamiento.....	122
4.3.3.	Bloque de transmisión de datos .....	125
4.3.4.	Bloque de visualización .....	126
4.4.	Pruebas de Implementación .....	129
4.4.	Análisis de Costo del Prototipo.....	133
5.	Conclusiones y Recomendaciones .....	135
5.1.	Conclusiones .....	135
5.2.	Recomendaciones. ....	137
6.	BIBLIOGRAFÍA .....	138
7.	ANEXOS .....	144
	ANEXO1. Formato de encuestas mediante Entrevista .....	144
	ANEXO2. Formato de encuestas mediante Entrevista.....	152
	ANEXO3. Código Fuente de toma de Datos Procesamiento y Transmisión en Arduino .....	156
	ANEXO4. Datasheets .....	160

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Arquitectura GPS .....	12
<b>Figura 2.</b> Arquitectura GSM .....	18
<b>Figura 3.</b> Arquitectura GPRS.....	21
<b>Figura 4.</b> Fases del modelo en V.....	26
<b>Figura 5.</b> Arduino UNO, vista superior. ....	30
<b>Figura 6.</b> SODIAL Zx303, vista superior. ....	31
<b>Figura 7.</b> A9G, vista superior.....	31
<b>Figura 8.</b> Arquitectura del proyecto.....	69
<b>Figura 9.</b> Diagrama de Bloques de PETROUTE. ....	70
<b>Figura 10.</b> Diagrama de Flujo del proceso de PETROUTE. ....	73
<b>Figura 11.</b> Regulador de Voltaje (Arduino Nano). ....	75
<b>Figura 12.</b> Divisor de Voltaje (Arduino Nano – Sim800L). ....	77
<b>Figura 13.</b> Diseño de conexión del circuito de alimentación del sistema PETROUTE. .	77
<b>Figura 14.</b> Diagrama de flujo del bloque de toma de Datos. ....	79
<b>Figura 15.</b> Diagrama de flujo del bloque de Procesamiento.....	81
<b>Figura 16.</b> Diseño de conexión del módulo GPS Neo6M.....	82
<b>Figura 17.</b> Comando AT para verificar el funcionamiento del módulo SIM800L. ....	83
<b>Figura 18.</b> Comando AT para verificar estado de batería.....	84
<b>Figura 19.</b> Comando AT para identificar la intensidad de la señal de red del módulo SIM800L. ....	84
<b>Figura 20.</b> Comando AT para registrar la red del módulo SIM800L.....	85
<b>Figura 21.</b> Comando AT para verificar el estado de Red del módulo SIM800L. ....	85

<b>Figura 22.</b> Comando para verificar redes disponibles. ....	86
<b>Figura 23.</b> Diagrama de flujo del bloque de transmisión de Datos.....	87
<b>Figura 24.</b> Diseño de conexión del módulo SIM 8001.....	88
<b>Figura 25.</b> Diagrama de bloques de la etapa de visualización. ....	89
<b>Figura 26.</b> Diagrama de flujo de la etapa de visualización.....	91
<b>Figura 27.</b> Diagrama esquemático de la conexión del circuito.....	92
<b>Figura 28.</b> Diseño de pistas y conexión del circuito.....	93
<b>Figura 29.</b> PCB (Placa de Circuito Impreso) listo para perforarse y colocar elementos correspondientes. ....	94
<b>Figura 30.</b> PCB (Placa de Circuito Impreso) con sus componentes soldados. ....	94
<b>Figura 31.</b> Estructura de la Pechera (Vista Lateral).....	95
<b>Figura 32.</b> Pechera colocada en la mascota (Vista Frontal).....	95
<b>Figura 33.</b> Código de toma de Datos del sistema PETROUTE.....	97
<b>Figura 34.</b> Módulo GPS NEO6m en estado inactivo, vista superior. ....	97
<b>Figura 35.</b> Módulo GPS NEO6m en estado activo, vista superior. ....	98
<b>Figura 36.</b> Obtención de Datos de coordenadas de ubicación, vista superior. ....	98
<b>Figura 37.</b> Modos de estado del SIM8001, vista superior. ....	101
<b>Figura 38.</b> Código de transmisión de Datos del sistema PETROUTE. ....	102
<b>Figura 39.</b> Coordenadas de ubicación del sistema PETROUTE. ....	103
<b>Figura 40.</b> Coordenadas de ubicación enviadas desde el sistema PETROUTE al Smartphone del dueño.....	104
<b>Figura 41.</b> Proyecto creado con éxito, Firebase.....	105
<b>Figura 42.</b> Creación de la base de datos en Firebase. ....	106

<b>Figura 43.</b> Base de datos en Firebase creada correctamente.....	107
<b>Figura 44.</b> Modificaciones en base de datos de Firebase realizadas correctamente. ....	108
<b>Figura 45.</b> Conexión exitosa a la nube de Firebase. ....	109
<b>Figura 46.</b> Coordenadas de Posición actualizándose. ....	109
<b>Figura 47.</b> Descarga de Flutter SDK.....	110
<b>Figura 48.</b> Agregado de ruta dentro de las variables de entorno. ....	111
<b>Figura 49.</b> Requerimientos de Instalación y configuración correctos. ....	112
<b>Figura 50.</b> Interfaz de PETROUTE. ....	113
<b>Figura 51.</b> Interacción Realtime Database y Aplicativo Móvil. ....	113
<b>Figura 52.</b> Voltaje de entrada para el sistema.....	119
<b>Figura 53.</b> Medición de voltaje en los pines del Arduino nano. ....	120
<b>Figura 54.</b> Línea temporal de voltaje.....	120
<b>Figura 55.</b> Modulo GPS en funcionamiento.....	124
<b>Figura 56.</b> Prueba de conexión a la red móvil mediante comandos AT. ....	124
<b>Figura 57.</b> Datos publicados en el servidor FIREBASE.....	126
<b>Figura 58.</b> Visualización en tiempo real de la ubicación en coordenadas de longitud y latitud. ....	127
<b>Figura 59.</b> Geocerca activa a 100 metros.....	128
<b>Figura 60.</b> Geocerca activa a 200 metros.....	128
<b>Figura 61.</b> Accesorio de portación del sistema. ....	129
<b>Figura 62.</b> Sistema y accesorio montado. ....	130
<b>Figura 63.</b> Sistema PETROUTE montado y listo para su funcionamiento. ....	130
<b>Figura 64.</b> Sistema colocado en la mascota.....	131

<b>Figura 65.</b> Aplicación móvil y data base .....	131
<b>Figura 66.</b> Ubicación de la mascota casi al límite de la geocerca .....	132
<b>Figura 67.</b> Mensaje de Alerta cuando la mascota supera los límites de la geocerca. ....	132

## ÍNDICE DE FIGURAS DE ANEXOS

<b>Ilustración 1.</b> Estadística de la gráfica – Pregunta 1 – Encuesta 1 .....	146
<b>Ilustración 2.</b> Estadística de la gráfica – Pregunta 2 – Encuesta 1 .....	146
<b>Ilustración 3.</b> Estadística de la gráfica – Pregunta 3 – Encuesta 1 .....	147
<b>Ilustración 4.</b> Estadística de la gráfica – Pregunta 4 – Encuesta 1 .....	148
<b>Ilustración 5.</b> Estadística de la gráfica – Pregunta 5 – Encuesta 1 .....	148
<b>Ilustración 6.</b> Estadística de la gráfica – Pregunta 6 – Encuesta 1 .....	149
<b>Ilustración 7.</b> Estadística de la gráfica – Pregunta 7 – Encuesta 1 .....	150
<b>Ilustración 8.</b> Estadística de la gráfica – Pregunta 8 – Encuesta 1 .....	150
<b>Ilustración 9.</b> Estadística de la gráfica – Pregunta 9 – Encuesta 1 .....	151
<b>Ilustración 10.</b> Estadística de la gráfica – Pregunta 10 – Encuesta 1 .....	151
<b>Ilustración 11.</b> Estadística de la gráfica – Pregunta 1 – Encuesta 2 .....	153
<b>Ilustración 12.</b> Estadística de la gráfica – Pregunta 2 – Encuesta 2 .....	154
<b>Ilustración 13.</b> Estadística de la gráfica – Pregunta 3 – Encuesta 2 .....	154
<b>Ilustración 14.</b> Estadística de la gráfica – Pregunta 4 – Encuesta 2 .....	155
<b>Ilustración 15.</b> Estadística de la gráfica – Pregunta 5 – Encuesta 2 .....	156

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Descripción de la Pila de Protocolo GPRS.....	23
<b>Tabla 2.</b> Herramientas para el levantamiento de información de la situación actual.....	33
<b>Tabla 3.</b> Lista de Stakeholders .....	37
<b>Tabla 4.</b> Definición de Abreviaturas .....	38
<b>Tabla 5.</b> Requerimientos de Stakeholders.....	39
<b>Tabla 6.</b> Requerimientos de Sistema.....	41
<b>Tabla 7.</b> Requerimientos de Arquitectura .....	45
<b>Tabla 8.</b> Selección de Placa de Desarrollo.....	51
<b>Tabla 9.</b> Especificaciones Técnicas de Arduino Nano.....	52
<b>Tabla 10.</b> Selección de Módulo de comunicación inalámbrica GPRS .....	53
<b>Tabla 11.</b> Especificaciones Técnicas del módulo GSM GPRS SIM800L .....	55
<b>Tabla 12.</b> Selección de Modulo GPS .....	56
<b>Tabla 13.</b> Especificaciones Técnicas del módulo GPS NEO6Mv2 .....	57
<b>Tabla 14.</b> Selección del Software para la programación de la Placa de desarrollo Arduino Nano.....	59
<b>Tabla 15.</b> Selección de la Plataforma de servicio (Cloud).....	61
<b>Tabla 16.</b> Elección del Software para el desarrollo de la aplicación móvil.....	62
<b>Tabla 17.</b> Costos de Hardware .....	64
<b>Tabla 18.</b> Costos de Software .....	64
<b>Tabla 19.</b> Costos de Ingeniería.....	65
<b>Tabla 20.</b> Costos varios y de Oficina .....	66

<b>Tabla 21.</b> Costo total del Sistema.....	67
<b>Tabla 22.</b> Pines de conexión del módulo GPS NEO6Mv2 con Arduino Nano. ....	82
<b>Tabla 23.</b> Pines de conexión del módulo SIM800L con Arduino Nano. ....	88
<b>Tabla 24.</b> Comandos AT para el uso del módulo SIM800l mediante comunicación Serial .....	102
<b>Tabla 25.</b> Pruebas de Funcionamiento .....	114
<b>Tabla 26.</b> Cumplimiento de requerimientos de Stakeholders .....	115
<b>Tabla 27.</b> Cumplimiento de requerimientos de sistema.....	116
<b>Tabla 28</b> Cumplimiento de requerimientos de arquitectura.....	117
<b>Tabla 29.</b> Test Eléctrico .....	118
<b>Tabla 30.</b> Test de Datos.....	122
<b>Tabla 31.</b> Test de Transmisión de Datos.....	125
<b>Tabla 32.</b> Test de Visualización .....	126
<b>Tabla 33.</b> Costo final del Prototipo a la comunidad Ibarreña .....	133
<b>Tabla 34.</b> Línea de Producción del Sistema PETROUTE .....	134

### ÍNDICE DE ECUACIONES

<b>Ecuación. 1.</b> Cálculo del consumo de los elementos del Sistema.....	74
<b>Ecuación. 2.</b> Ecuación de Divisor de Voltaje (Arduino Nano – Sim800L). ....	76
<b>Ecuación. 3.</b> Fórmula de Haversine .....	99
<b>Ecuación. 4.</b> Promedio del tiempo de Vida Útil del sistema PETROUTE .....	121
<b>Ecuación. 5.</b> Fórmula para calcular el tiempo de descarga de la batería .....	121
<b>Ecuación. 6.</b> Cálculo del tiempo de descarga de la batería .....	122

## RESUMEN

En el presente proyecto de titulación se desarrolla el sistema PETROUTE, este es un sistema GPS Tracker que ayuda a ubicar a las mascotas mediante una red IoT, el objetivo de este sistema es localizar y rastrear en tiempo real a las mascotas para que no incurran en una desaparición por desconcentración del dueño o también hurto por parte de las personas amigas de lo ajeno, sirviendo de apoyo para reducir los índices de animales que se encuentran vagando por las calles de la ciudad de Ibarra en condiciones no óptimas para su salud.

El diseño del sistema se basó en la metodología denominada Modelo en V y se encuentra conformado principalmente por una placa de desarrollo Arduino Nano ya que permite desarrollar sistemas de geolocalización por medio de sus librerías. Además, al sistema se le suma módulos de comunicación inalámbrica y GPS para detectar la ubicación por medio de coordenadas geográficas y así obtener la información para ser enviada a través del módulo de comunicación inalámbrica, con ello, se continua al procesamiento de los datos para determinar el rastreo y monitoreo en tiempo real, pudiendo visualizar gráficamente mediante la interfaz de una aplicación desarrollada en software Open Source.

Dentro del Aplicativo móvil se puede crear la función de geocerca la cual permite que la mascota se mantenga dentro de un rango definido por el usuario y de esta forma precautelar la seguridad de la misma bajo los parámetros expuestos anteriormente, los cuales evitan que la mascota incurra en una desaparición.

## **ABSTRACT**

In this degree project the PETROUTE system is developed, this is a GPS Tracker system that helps to locate pets through an IoT network, the objective of this system is to locate and track pets in real time so that they do not incur in a disappearance due to lack of concentration of the owner or also theft by people who are friends of others, serving as support to reduce the rates of animals that are wandering the streets of the city of Ibarra in suboptimal conditions for their health.

The design of the system was based on the methodology called V-Model and is mainly conformed by an Arduino Nano development board since it allows the development of geolocation systems through its libraries. In addition, wireless communication and GPS modules are added to the system to detect the location by means of geographic coordinates and thus obtain the information to be sent through the wireless communication module, with this, the data processing is continued to determine the tracking and monitoring in real time, being able to visualize graphically through the interface of an application developed in Open Source software.

Within the mobile application you can create the geofence function which allows the pet to be kept within a range defined by the user and thus ensure the safety of the same under the parameters set out above, which prevent the pet to incur in a disappearance.

## **1. CAPÍTULO: ANTECEDENTES**

En este apartado se presenta información con respecto a los antecedentes que se ha considerado para el desarrollo del proyecto de titulación; de igual forma se toma en cuenta la problemática que se ha analizado y plateado, los objetivos tanto generales como específicos, la delimitación del alcance y la correspondiente justificación.

### **1.1. Tema**

Diseño de un Sistema Embebido de detección y rastreo de mascotas mediante una Red IoT aplicado en la ciudad de Ibarra.

### **1.2. Problema**

Según una publicación del diario (El Telégrafo, 2016) por cada 7 habitantes hay un can deambulando por la ciudad de Ibarra. Si en la capital de Imbabura viven 131.856 personas habrían alrededor de 18.836 canes. De acuerdo con los datos publicados por parte de diario (La Hora.2016) Se debía hacer un registro de mascotas en la ciudad de Ibarra. Este proceso era de carácter obligatorio de acuerdo a la ordenanza municipal y el código Orgánico del Ambiente. Dado el caso se debía registrar en el Sistema de Registro Municipal, para tener una base de los animales que existen en el cantón Ibarra, bajo custodia humana, además para evitar así la pérdida o el abandono y que las personas cumplan con una tenencia responsable de animales.

Según una publicación por parte de la página web (Quenoticias, 2019) se proponía la realización de un Registro Nacional de Mascotas del Ecuador el cual permite formalizar la identidad de los animales de compañía a través de la “Pet-Dulación”. El sistema de identificación consiste en la implantación de un microchip que contiene datos importantes como el nombre de la mascota, nombre de su dueño, dirección y teléfonos. En el registro se le entrega también una

placa con código QR y su cédula de identidad. El procedimiento se lo realizaba en clínicas veterinarias afiliadas en Esmeraldas, Quito, Santa Rosa, La Troncal y Loja, los interesados debían llamar a la clínica autorizada para obtener el formulario de inscripción. Sin embargo, desde cualquier parte del país se podía adquirir el kit de "Pet – Dulación" que tiene un costo de 20 dólares e incluye: Cédula, placa QR, microchip de identificación (no es GPS), certificado de implantación, certificado de propietario. El registro informó a través de las redes sociales que el sábado 26 de junio del año 2019 se realizaría una jornada de identificación en la peluquería canina 'Oh My Dog' ubicada en el cantón Ibarra de la provincia de Imbabura y tendrá un costo de 30 dólares.

Hoy en día gracias a las plataformas que se encuentran activamente en redes sociales se ha permitido mediante una búsqueda profunda ver al menos 5 grupos de rescate animal dentro de la ciudad de Ibarra. Los cuales aportan a la ciudadanía a la atención y prioridad de estas mascotas perdidas a través de fotos de los mismos y de esta manera poder lograr llegar a sus dueños. Actualmente se encuentran activos 6 grupos de mascotas perdidas y de igual forma se tiene una organización sin fines de lucro llamada la voz Animal donde al día se realizan un total de 3 publicaciones correspondientes a mascotas perdidas dentro de la ciudad de Ibarra.

Realizando una prospectiva acerca del tema se tiene que hoy en día gracias a la tecnología se pueden desarrollar sistemas electrónicos sofisticados que pueden aportar las mascotas previniendo una desaparición o raptos de los mismos, Todo esto aplicando parámetros o tecnologías emergentes como lo son, IoT y a su vez tecnologías inalámbricas de monitoreo y alerta mediante la implementación de sistemas Embebidos.

## **1.3. Objetivos**

### ***1.3.1. Objetivo General***

Diseñar un sistema electrónico de alerta y monitoreo mediante una red IOT para que establezca la ubicación y rastreo de una mascota perdida en tiempo real dentro de la ciudad de Ibarra

### ***1.3.2. Objetivos Específicos***

- Analizar la base teórica mediante una investigación bibliográfica y documental, con la cual se sustente el proyecto en las áreas de redes IoT para la ubicación y coordenadas de mascotas.
- Desarrollar el sistema basado en el modelo en V para la definición de requerimientos de Stakeholders, Hardware y Software.
- Establecer todos los criterios requeridos para lograr un sistema Experto Completo.
- Verificar el correcto funcionamiento del sistema electrónico en tiempo real mediante pruebas de campo.

## **1.4. Alcance**

En el proyecto de grado descrito en el presente documento se tiene como alcance, diseñar e implementar un sistema electrónico de alerta y monitoreo mediante una red IoT con el fin de evitar la pérdida o hurto de animales dentro de la ciudad de Ibarra. Para lo cual se utilizará la metodología del modelo en V para la realización de este proyecto. Para inicio del presente proyecto se realizará un estudio bibliográfico con el fin de establecer bases teóricas en las que se sustenten un análisis de los componentes más importantes necesarios para la construcción del sistema embebido.

En el modelo en V se establecerá cuáles son los requerimientos de cada una de las etapas del sistema mediante la definición de requerimientos de Stakeholders, Hardware y Software basados en la Arquitectura IoT. El modelo en V mediante selección de Benchmark y el estándar ISO/IEC/IEEE 29148, el cual especifica los procesos necesarios que deben ser implementados para la ingeniería de requisitos para sistemas y productos software (incluidos los servicios) en todo el ciclo de vida, dar pautas para la aplicación de los requisitos y procesos de requisitos.

La etapa de diseño se llevará a cabo basándose en las capas de la arquitectura IoT. Las capas que se van a utilizar para el presente proyecto iniciara desde la capa de Dispositivo (Sensores /Actuadores) donde el funcionamiento de esta capa partirá desde que se emite, acepta y procesa datos a través de la red mediante la tecnología GPS con la cual se va a desarrollar el presente proyecto donde se tiene en primera instancia la función de paridad entre dueño y mascota la misma que se desarrollara mediante una comunicación de alerta entre el tag(smartphone) del dueño y el sistema embebido de la mascota, cuando la mascota se aleje o exista falta de atención dentro de un rango definido. El sistema embebido está construido por una comunicación de sensores que bajo la metodología expuesta anteriormente se definirá cual es la mejor opción para el desarrollo del mismo.

Para la capa de red (Puertas de enlace y adquisición de datos) en esta capa están presentes las puertas de enlace de Internet / red), el funcionamiento de esta capa está ligada al módulo que se elegirá mediante los requerimientos correspondientes estándar ISO/IEC/IEEE 29148 que se establezcan, el cual permita la conexión a la red móvil y que a su vez sea capaz de enviar

grandes cantidades de datos, de esta forma, se necesitaran las puertas de enlace y las redes de alta velocidad para poder realizar la acción de transferencia de los datos.

Para el proceso y almacenamiento de la información se encuentra la capa de procesamiento de datos (Edge IT) que son las puertas de enlace de hardware y software donde se analiza y preprocesan los datos antes de transferirlos a la nube. Si los datos leídos de los módulos y las puertas de enlace no se modifican con respecto a su valor de lectura anterior, entonces no se transfieren a través de la nube, esto guardara los datos utilizados. Los datos recibidos se almacenarán en una base de datos que servirán como apoyo para identificar la información de geolocalización de la mascota. La misma que se elegirá de acuerdo a los requerimientos del estándar ISO/IEC/IEEE 29148.

Finalmente, para la aplicación de usuario final (Dueño de la mascota) se establece la capa aplicación (Centro de datos / nube) en esta capa se plantea hacer uso de la nube donde ocurre la administración de datos los cuales se administran y serán usados por una APP que se implementara para poder de esta forma establecer un monitoreo en tiempo real basándose en puntos de coordenadas bajo un sistema operativo Android

Para la comprobación del correcto funcionamiento se realizará mediante pruebas de arquitectura, tomando en cuenta la primera capa que es donde ocurre la primera alerta basándose en un rango definido por el dueño de la mascota. De igual forma se realizará una prueba de mapeo y geolocalización en tiempo real basándose en la APP brindada al usuario Final.

## 1.5. Justificación

Según (Flores, 2021). “En 2020 recibimos 641 reportes de animales extraviados, principalmente en Guayaquil y Samborondón”, informó la cuenta #AlertaMePerdí, que difunde este tipo de casos. Fueron 536 perros y 105 gatos, indicó la fundación Rescate Animal, que maneja la cuenta. “La mayoría de estos animales jamás regresarán a su hogar”, destaca el mensaje. Piden, por lo tanto, a los dueños de las mascotas que tomen conciencia y que sean responsables con las vidas que tienen a su cargo.

Tomando como referencia a (Maroto-Molina et al., 2019) Las tecnologías para la geolocalización de animales han ido evolucionado considerablemente en los últimos años, tanto así que hoy en día, las soluciones basadas en Sistemas de Posicionamiento Global ya son comercialmente disponibles; sin embargo, los dispositivos existentes tienen varias restricciones principalmente relacionadas con la cobertura de la red inalámbrica, transmisión de los datos y valor adquisitivo.

Las técnicas empleadas para determinar la ubicación de animales han sido varias. (Molapo, 2019), desarrollaron un trabajo cuyo objetivo era transmitir la ubicación y la actividad de los animales, en tiempo real haciendo uso de una WSN (Red de Sensores Inalámbricos).

Otros trabajos adicionales que cabe mencionar son los siguientes: el primero se trata de la implementación de un collar utilizando las tecnologías GPS/GSM con la finalidad de dar seguimiento al ganado (Francis, 2018) el segundo se realizó empleando la tecnología comunitaria de Identificación de Radiofrecuencia (RFID) como métodos alternativos para reducir el robo de ganado (Ibrahim, 2016).

Según (Hernandez, 2018) La Tecnología, mascotas e ideas innovadoras. La combinación de estos tres elementos está empezando a dar sus frutos en un mercado en alza. El sector de los animales de compañía ya mueve 1.000 millones de euros al año en España. Una cifra que sitúa a este país en la quinta posición en gasto dentro de Europa. La preocupación por su salud y bienestar ha aumentado en los últimos años. Así como el número de mascotas, presentes en el 40% de los hogares españoles, según la Asociación Madrileña de Veterinarios de Animales de Compañía (AMVAC). En cifras redondas, en España ya hay 20 millones, de los cuales casi 5,2 millones son perros y más de 2,2 son gatos. Además de los productos destinados a su alimentación e higiene, se está observando un aumento en la adquisición de gadgets. Dispositivos que ayudan a los dueños en el cuidado diario de sus animales de compañía, pero sobre todo en aquellos destinados a su vigilancia y monitorización. Una tendencia que los especialistas en la venta de artículos para mascotas prevén que seguirá aumentando.

Con base a lo ya mencionado este proyecto busca definir la construcción de un prototipo electrónico de detección y rastreo mediante una red IoT de geolocalización orientado a mascotas, con la finalidad de contribuir en la prevención de pérdida de los mismos y ayudar a los dueños a encontrar rápidamente a sus mascotas en la ciudad de Ibarra.

## **2. CAPÍTULO: FUNDAMENTACIÓN TEÓRICA**

El segundo capítulo describe las bases teóricas de los aspectos técnicos y los fundamentos principales de los elementos del sistema Electrónico al igual que el software para el desarrollo de la interfaz de usuario y el monitoreo en tiempo real.

Se establece las principales tecnologías empleadas en el desarrollo del presente proyecto tomando en cuenta los requerimientos tanto de hardware como de software y la arquitectura de red que van a hacer puestos en práctica en el desarrollo del sistema.

### **2.1. Mascota**

Las mascotas son animales domésticos, también conocidos como animales de compañía, y como su nombre indica, sus dueños las mantienen para que se conviertan en parte de la familia, reducir la soledad y traer alegría al hogar. Las mascotas alegran el día del dueño, brindan compañía y, si están bien entrenadas, algunas de ellas pueden incluso ser aliadas en las tareas diarias(Bernal, 2015).

Históricamente, las mascotas han jugado un papel importante en las actividades humanas, brindando compañía, motivación y diversión. La descripción más antigua de animales domésticos involucrados en terapia es de Florence Nightingale, quien en 1860 observó que las mascotas son excelentes compañeros para los pacientes con enfermedades crónicas(Camilo G & Sonia C,2017).

### **2.2. Sistemas Embebidos**

Como sugiere el nombre, Embebido significa algo adjunto a otra cosa. Un sistema integrado se puede considerar como un sistema de hardware de computadora con software

integrado. Los sistemas integrados pueden ser sistemas independientes o parte de un sistema más grande.

Los sistemas integrados son sistemas de hardware informático-basados en microprocesadores con software diseñado para realizar funciones especializadas, ya sea como un sistema independiente o como parte de un sistema más grande. En su núcleo es un circuito integrado diseñado para realizar cálculos para operaciones en tiempo real.

La complejidad varía desde un solo microcontrolador hasta un conjunto de procesadores con periféricos y redes adicionales; desde ninguna interfaz de usuario hasta interfaces gráficas de usuario complejas. La complejidad de los sistemas embebidos varía ampliamente dependiendo de la tarea de su diseño(Lutkevich, 2019).

### **2.3. Internet de las Cosas (IoT)**

IoT es un nuevo cambio de paradigma en TI. La frase "Internet de las cosas", también conocida como IoT para abreviar, se acuñó a partir de estas dos palabras, la primera palabra es "Internet" y la segunda palabra es "Cosas". Internet es un sistema global de redes informáticas interconectadas que utiliza un conjunto de protocolos estándar de Internet (TCP/IP) para atender a miles de millones de usuarios en todo el mundo. Es una red de millones de redes privadas, públicas, académicas, comerciales y gubernamentales que van desde locales hasta globales, conectadas por una amplia gama de tecnologías de redes electrónicas, inalámbricas y ópticas(Lake, 2015).

El término IoT se usa principalmente para dispositivos que generalmente no se espera que tengan una conexión a Internet y pueden comunicarse con la red independientemente de las acciones humanas. Por esta razón, las PC generalmente no se consideran dispositivos IoT, ni los

teléfonos inteligentes, incluso si estos últimos están equipados con sensores. Sin embargo, los relojes inteligentes o las pulseras de fitness u otros dispositivos portátiles pueden contar como dispositivos IoT(Ranger, 2020).

La firma de analistas de tecnología IDC predice que para 2025 habrá un total de 41.600 millones de dispositivos IoT conectados, o "cosas". También muestra que los dispositivos industriales y automotrices representan la mayor oportunidad para conectar "cosas", pero también ven una fuerte adopción de dispositivos portátiles y para el hogar inteligente a corto plazo(Ranger, 2020).

#### **2.4. Cloud**

La computación en la nube es la provisión de servicios informáticos bajo demanda, desde aplicaciones hasta almacenamiento y potencia de procesamiento, generalmente a través de Internet y mediante pago por uso(Ranger, 2022).

El Internet de las cosas y la computación en la nube se complementan entre sí y, a menudo, se mencionan juntos cuando se habla de servicios técnicos para brindar conjuntamente mejores servicios de Internet de las cosas. Sin embargo, existen diferencias importantes entre ellos que hacen de cada uno de ellos una solución técnica eficaz tanto individualmente como en conjunto(McKenna, 2021).

#### **2.5. Global Positioning System(GPS)**

El Sistema de Posicionamiento Global (GPS) es un sistema de navegación que utiliza satélites, receptores y algoritmos para proporcionar datos de posición, velocidad y tiempo sincronizados para viajes por aire, mar y tierra. El sistema de satélites consiste en una constelación de 24 satélites repartidos en seis planos orbitales centrados en la Tierra, cuatro de

cada uno, que orbitan a 13.000 millas (20.000 km) sobre la Tierra a una velocidad de 8.700 millas (14.000 km) por hora.

Aunque sólo se necesite tres satélites para obtener una posición en la superficie de la Tierra, el cuarto satélite suele utilizarse para verificar la información de los otros tres. Un cuarto satélite también lleva a la tercera dimensión y permite calcular la altitud de un dispositivo(Geotab, 2022).

### ***2.5.1. Arquitectura***

La estructura del GPS consta de tres partes o unidades que se detallan a continuación:

- **Segmento espacial GPS:** La unidad espacial consta de 24 satélites activos dotados de enormes paneles solares y baterías recargables como fuente de energía. La función de los satélites en el espacio es encaminar o navegar las señales de radio recibidas de la unidad de control para almacenar y retransmitir información a la unidad receptora correspondiente(Bose, 2019).
- **Segmento de control GPS:** La unidad de control se compone de varios puestos de vigilancia y control. Las estaciones de control supervisan las señales de los satélites GPS. Estas señales se envían a la estación de control principal, donde se comprueban y modifican las especificaciones de funcionamiento antes de transmitir las señales de control a los satélites GPS. Se envían de vuelta a través de una antena de tierra(Capdevila, 2018).
- **Segmento de receptor GPS (usuario):** Las unidades de usuario son todos los receptores GPS, como teléfonos móviles, ordenadores portátiles, PC o cualquier otro dispositivo. Estos dispositivos reciben las señales de los satélites GPS y determinan a qué distancia se encuentra cada uno de ellos(Recio, 2021).



**Figura 1.** Arquitectura GPS

Fuente: Adaptado de (Bose, 2019)

### 2.5.2. Funcionamiento

En la Figura 1 se puede apreciar un ejemplo del funcionamiento partiendo de que cada satélite GPS emite una señal única que permite al receptor GPS decodificar y calcular la posición exacta del satélite. La señal se mueve a la velocidad de la luz y el receptor GPS lo sabe, lo que ayuda a determinar su posición mediante cálculos matemáticos.

El receptor GPS utiliza estos datos para calcular la posición del usuario, que se muestra en el dispositivo electrónico mediante un proceso conocido como "triangulación". Funciona midiendo la distancia a cada satélite por el tiempo total que tarda en recibir la señal transmitida.

Para calcular una posición bidimensional y seguir el movimiento, el receptor GPS debe fijar las señales de radio de al menos tres satélites. Se debe tener en cuenta que cuatro o más satélites permiten al receptor determinar una posición fija tridimensional.

Normalmente, un receptor GPS puede rastrear ocho o más satélites GPS, pero esto depende de su ubicación en la Tierra y de la hora del día(Roberts, 2022).

### ***2.5.3. Pseudodistancia***

Los receptores GPS utilizan relojes de funcionamiento libre basados en osciladores de cristal de bajo coste con errores de sesgo en el reloj estimado durante la estimación de la posición. Como resultado, la medición del tiempo de tránsito aparente está sesgada por los errores en los relojes del usuario y del satélite, lo que se conoce como pseudodistancia(Chang & Kwang, 2017).

### ***2.5.4. Aplicaciones***

Las aplicaciones GPS se dividen generalmente en 5 categorías principales.

- **Posicionamiento:** determinar una ubicación
- **Navegación:** cómo ir de un lugar a otro
- **Seguimiento:** control del movimiento de un objeto o una persona
- **Cartografía:** crear un mapa del mundo
- **Cronometraje:** llevar el cronometraje exacto al mundo

De la minería a la aviación, de la agricultura al mar, del ocio a la defensa, la tecnología GPS tiene muchas aplicaciones. Hoy en día, personas como médicos, científicos, agricultores, soldados, pilotos, excursionistas, repartidores, marineros, pescadores, expedidores, atletas y muchos otros sectores utilizan los sistemas GPS para hacer su trabajo más eficiente, seguro y fácil.

Los tres principales usos industriales del GPS son la agricultura de precisión, la automoción y la defensa. En cada ejemplo, el GPS resuelve un problema de posicionamiento(NovAtel, 2020).

### **2.5.5. Códigos GPS**

Cada satélite GPS transmite dos señales, L1 y L2, en frecuencias diferentes. La trilateración es un método sencillo para encontrar la posición de un receptor GPS (latitud, longitud y altitud). Mediante este método, la posición de un punto desconocido puede medirse a partir de tres puntos conocidos(Oxley, 2017).

A continuación, se presentan los dos tipos de códigos GPS.

- Código de adquisición gruesa o código C/A
- Código preciso o código P

La señal L1 se modula con una secuencia de bits pseudoaleatoria de 1,023 Mbps. Este código se denomina código de adquisición gruesa o código C/A y es el utilizado por el público.

La señal L2 se modula con una secuencia de bits pseudoaleatoria de 10,23 Mbps. Este código se denomina código preciso o código P y se utiliza en los sistemas de posicionamiento militar. Generalmente, este código P se transmite en un formato encriptado y se denomina código Y.

El código P proporciona una mayor precisión de medición en comparación con el código C/A, ya que la tasa de bits del código P es mayor que la del código C/A(Oxley, 2017).

### **2.5.6. Servicios**

A continuación, se detallan los dos tipos de servicios que ofrece el GPS.

- Servicio de Posicionamiento Preciso (PPS)
- Servicio de posicionamiento estándar (SPS)

El receptor PPS rastrea el código C/A y el código P de las señales L1 y L2. el código Y se descifra en el receptor para obtener el código P.

El receptor SPS sólo rastrea el código C/A de la señal L1(Laveti et al., 2014).

## 2.6. Google Maps

Google Maps es una de las innovaciones más buscadas en la historia de la tecnología. Google Maps permite a las personas navegar y encontrar la ruta más corta y conveniente a su destino deseado. Según un estudio reciente, Google Maps ha ganado casi 64 millones de usuarios. Además, incluye nuevas funciones como Street View, la localización de hospitales, cafeterías estaciones de policía y muchas otras funciones útiles.

Los algoritmos, la tecnología y las técnicas utilizadas por Google son punteros y muy avanzados. El equipo de ingenieros de Google ha guardado y analizado innumerables conjuntos de datos, tanto históricos como en tiempo real, lo que hace que Google Maps sea tan progresivo y preciso. Modelos predictivos Función de valores continuos, es decir, predicción de nuevos valores. Además, los modelos de ML pueden utilizarse y aplicarse a las técnicas informáticas tradicionales para mejorar la precisión existente(Mehta et al., 2019).

## 2.7. Dispositivo GPS utilizados en animales

A continuación, se citan algunos ejemplos de Dispositivo GPS utilizados en animales.

- **Vertex Plus:** Es un collar el cual esta construido sobre un concepto modular, consiste en un GPS estándar y varios sensores que le permite elegir la configuración que mejor se adapte al diseño de investigación.
- **Waterproof Pet Training:** Es un collar de entrenamiento impermeable diseñado para el seguimiento de mascotas, el cual este compuesto de un GPS y funciona con una tarjeta SIM GSM.

- **Airtag:** El AirTag es un pequeño dispositivo de rastreo con forma de botón y con un frontal blanco brillante que puede personalizarse con una base grabada y plateada. Las airTags están diseñadas en torno a una pila interna CR2032 y requieren accesorios adicionales para fijarlas al artículo. Apple ha incluido un chip U1 en cada AirTag para que se pueda ver su ubicación exacta en interiores o exteriores si se está cerca, o su última ubicación conocida si no lo estás (Clover, 2022).

## **2.8. Telefonía Celular**

El teléfono móvil ha pasado de ser un teléfono utilizado para simples llamadas entre dos partes a un ordenador completo con muchas funciones. Hoy en día, no sólo se utiliza para hacer llamadas, sino sobre todo para enviar mensajes sin coste alguno.

Las redes sociales han experimentado un gran avance gracias a los teléfonos móviles, que ofrecen una forma fácil y rápida de conectarse con otras personas a través de las redes sociales. Los selfies, las actualizaciones de estado y las noticias rápidas se pueden difundir y compartir fácilmente dentro de las aplicaciones.

Gracias a las múltiples prestaciones de los teléfonos móviles y a la posibilidad de descargar aplicaciones de terceros, se ha llegado a un punto de desarrollo en el que el teléfono móvil es una necesidad en la sociedad. Se utiliza para las gestiones bancarias, la identificación en línea, los pagos, el contacto con los médicos y mucho más (Vasseur, 2019).

### **2.8.1. Red Celular GSM**

El GSM (Sistema Global de Comunicaciones Móviles) es una red de telefonía móvil digital ampliamente utilizada por los usuarios de teléfonos móviles en la mayor parte del mundo. El GSM utiliza una variante del Acceso Múltiple por División de Tiempo (TDMA) y es la más

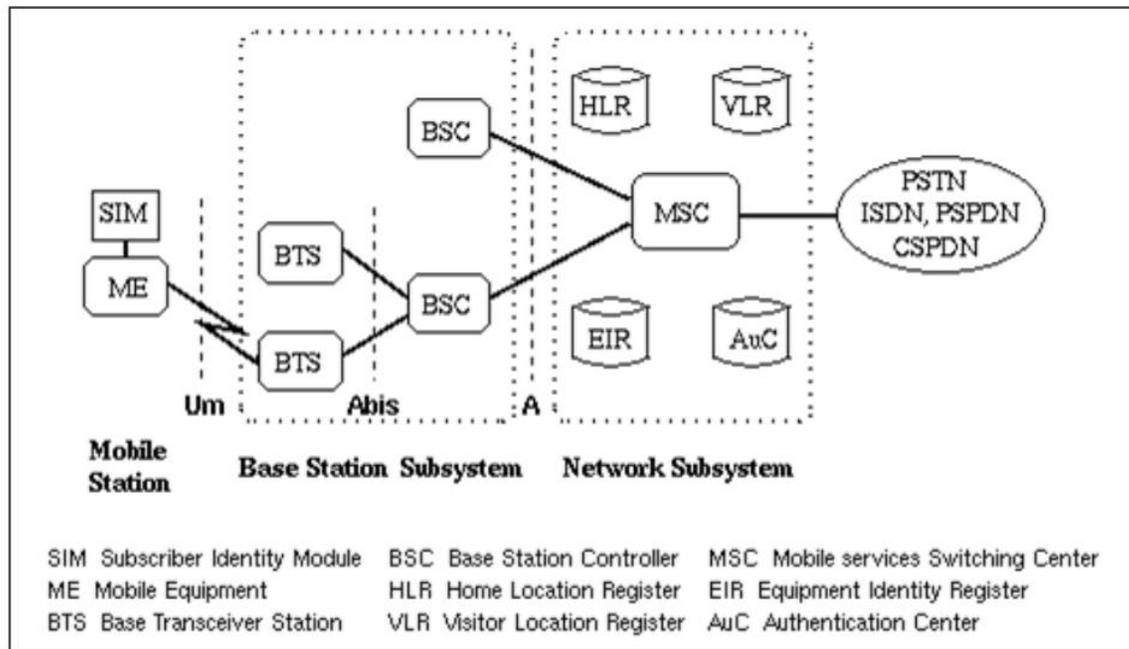
utilizada de las tres tecnologías de radiotelefonía digital: TDMA, GSM y Acceso Múltiple por División de Código (CDMA). El GSM digitaliza y comprime los datos y luego envía, junto con otros dos flujos de datos de usuario en un canal, cada uno con su propia franja horaria. Funciona en la banda de frecuencias de 900 megahercios (MHz) o 1.800 MHz(Ndungu, 2021).

### **2.8.1.1. Arquitectura del sistema GSM**

La arquitectura de la red GSM consta de tres subsistemas principales:

- **La estación móvil (MS):** La estación móvil se comunica con el transceptor del SRS en la misma célula en la que se encuentra la unidad móvil de abonado a través de la interfaz aérea. Si la interfaz aérea se comunica con el SRS, la EM se comunica con el usuario para obtener información y la modifica al protocolo de transmisión(Ramina, 2017).
- **El subsistema de estación base (BSS):** Un subsistema de estación base está formado por un controlador de estación base y una o varias estaciones transceptoras de base. Cada estación transceptora base define una célula. El radio de una célula puede ser de entre 100 m y 35 km, según el entorno(Wright, 2021).
- **El subsistema de red y conmutación (NSS):** El SNE es responsable del funcionamiento de la red. Proporciona el enlace entre la red celular y la red pública conmutada de telecomunicaciones (RTPC o RDSI o red de datos)(Rwanda, 2021).
- **La interfaz de radioenlace entre la MS y El transceptor de estación base (BTS), que forma parte del BSS:** Muchas BTS están controladas por un Controlador de Estación Base (BSC) que está conectado al Centro de Conmutación Móvil (MSC), que forma parte del NSS. Como se puede apreciar en

la figura 2, se muestra los elementos funcionales clave de la arquitectura de la red GSM(Chen & Guo, 2016).



**Figura 2.** Arquitectura GSM

Fuente: Adaptado de (Ndungu, 2021)

### 2.8.1.2. Rango de Frecuencias GSM

La banda de frecuencias es la parte más fundamental de cualquier red de radio para la tecnología celular. Cuando se iniciaron las redes GSM de segunda generación (2G), se limitaron inicialmente a la banda de 900 MHz, pero se han ampliado para operar en diversas bandas de frecuencia(Teltech, 2020).

Las redes GSM utilizan varias bandas de frecuencias, como 900 MHz, 1800 MHz, 850 MHz y 1900 MHz. Las combinaciones de 900 MHz/1800 MHz se utilizan principalmente en

Europa, Asia, África, Oriente Medio y Australia, mientras que las de 850 MHz/1900 MHz se emplean sobre todo en América del Norte y del Sur(Ghayas, 2019).

La red móvil de Ecuador y los operadores utilizan 2 bandas GSM. A continuación, se presenta las bandas en las que trabajan las operadoras más conocidas en Ecuador.

- Claro: 850 MHz (Banda 5)
- CNT: 1900 MHz (Banda 2)
- Movistar: 850 MHz
- Tuenti: 850 MHz

### **2.8.1.3. Servicios de GSM**

Los servicios GSM se dividen a su vez en servicios portadores y Teleservicios

- **Los servicios portadores:** son servicios de telecomunicaciones que garantizan la transmisión de señales entre puntos de acceso en las redes de telecomunicaciones. Se utiliza exclusivamente para la transmisión y define los servicios que ofrece la red, como la transferencia de datos entre la red corporativa y ordenadores portátiles o la descarga de datos de la web(Jörg et al., 2015).
- **Teleservicios:** Los servicios de telecomunicaciones son servicios y funciones de telecomunicaciones que permiten la comunicación entre usuarios y se basan en protocolos establecidos por el administrador de la red. Los servicios de telecomunicaciones afectan al último eslabón del proceso de comunicación: el usuario final. Estos incluyen servicios telefónicos, llamadas de emergencia, servicios de mensajes cortos, correo electrónico y fax(Castillo et al., 2019).
- **Los servicios complementarios:** modifican o complementan los servicios básicos de telecomunicaciones. Sólo se ofrecen en combinación con un servicio portador o

telefónico. Los mismos servicios complementarios pueden aplicarse a una serie de servicios de telecomunicaciones. La mayoría de estos servicios se heredan directamente de la red fija, pero con algunas concesiones en cuanto a la adaptación a la red de radio móvil. Los servicios complementarios incluyen, por ejemplo, la identificación de líneas y el desvío de llamadas. Los servicios de valor añadido dependen del proveedor de servicios o del operador de red al que se ofrecen. No están predefinidos en el pliego de condiciones e incluyen, por ejemplo, servicios de reserva o información (Ramina, 2017).

### ***2.8.2. General Packet Radio Service (GPRS)***

El GPRS (General Packet Radio Services) es un nuevo tipo de servicio de transmisión de datos por paquetes.

Se basa en la red GSM existente con actualizaciones adicionales de hardware y software para formar una nueva entidad de red lógica, y utiliza protocolos de red de datos IP basados en la tecnología de conmutación de paquetes para permitir que la red GSM existente proporcione servicios de datos a una velocidad máxima de 9,6 kbit/s y hasta 170 kbit/s. Los usuarios pueden utilizar una variedad de servicios de datos de alta velocidad en movimiento a través de GPRS, incluidos el correo electrónico, la navegación por Internet y otros servicios IP (Jian & Ya, 2015).

#### **2.8.2.1. Arquitectura de una red GPRS**

GPRS añade una nueva entidad de red a la red GSM para implementar servicios de datos en paquetes, La arquitectura de red se muestra en la Figura 3.

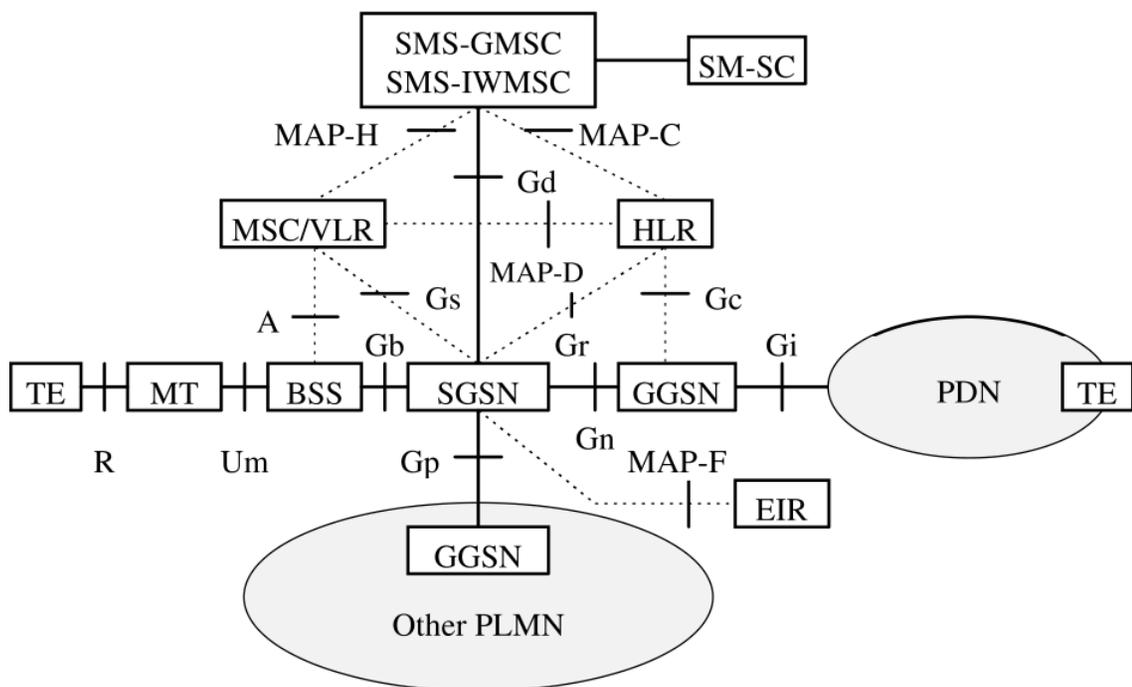


Figura 3. Arquitectura GPRS

Fuente: Adaptado de (Niemelä & Kari, 2016)

A continuación, se detalla los elementos que integran la Arquitectura de red GPRS.

GSN es el componente de red más importante en la red GPRS, hay dos tipos de SGSN y GGSN.

- **SGSN (nodo de soporte de servicio GPRS):** El papel principal del SGSN es registrar la información de la ubicación actual de la MS, proporcionar servicios como la gestión de la movilidad y el enrutamiento, y completar el envío y la recepción de datos de paquetes móviles entre la MS y el GGSN(Thakur, 2019).
- **GGSN (Gateway GPRS Support Node, nodo de soporte de la pasarela GPRS):** La GGSN se utiliza como pasarela para convertir los paquetes de la red GSM en protocolos y enviarlos a la red TCP/IP o X.25.
- **PCU (unidad de control de paquetes):** La PCU se encuentra en el BSS para manejar los servicios de datos y separar los servicios de datos de los servicios de

voz GSM. La PCU añade la funcionalidad de agrupación para controlar el enlace de radio y permite que varios usuarios ocupen los mismos recursos de radio(Jang, 2018).

- **BG(puerta de entrada fronteriza):** La BG se utiliza para interconectar las redes troncales GPRS entre los PLMN, completando principalmente las funciones de enrutamiento y de gestión de la seguridad entre los SGSN y los GSN pertenecientes a diferentes redes GPRS, además de añadir las funciones pertinentes según los acuerdos de itinerancia entre los operadores(García, 2017).
- **PC(Puerta de enlace de carga, puerta de enlace de carga):** El CG se encarga principalmente de recopilar, fusionar y preprocesar los tickets de varios GSN y sirve de interfaz de comunicación entre el GPRS y el centro de facturación.
- **DNS(Servidor de nombres de dominio):** En las redes GPRS existen dos tipos de DNS. Uno es el DNS entre el GGSN y la red externa, cuya función principal es analizar el nombre de dominio de la red externa, desempeñando el mismo papel que el DNS habitual en Internet, y el otro es el DNS en la red troncal GPRS, determinado según el proceso de activación del contexto PDP. APN (Access Point Name ) resuelve el nombre del GSN dirección IP, durante el proceso de actualización del área de enrutamiento entre SGSNs, basado en el número de área de enrutamiento original n del puerto de conexión de red de la dirección IP(Vejlgaard et al., 2017)

#### **2.8.2.2. Protocolo GPRS**

A continuación, en la Tabla 1, se presenta la pila de protocolos GPRS de acuerdo con el modelo OSI.

**Tabla 1.**

Descripción de la Pila de Protocolo GPRS.

<i><b>PILA DE PROTOCOLO GPRS</b></i>		
<i><b>Numero de Capa</b></i>	<i><b>Nombre de Protocolo</b></i>	<i><b>Descripción</b></i>
<b>6. Presentación</b>	<b>IP/x25</b>	Protocolo del BackBone de GPRS Encaminamiento de datos de usuario y control de señalización
<b>5. Sesión</b>	<b>SNDCP</b>	El SNDCP es transparente para el subsistema de estación base (BSS) y es la capa más alta de la propia pila de protocolos GPRS. Proporciona interfaces a capas superiores que son específicas del dispositivo o de la aplicación, incluyendo IP y PPP. Existe una estrecha conexión y dependencia entre SNDCP y LLC, que son capas adyacentes en el protocolo GPRS. SNDCP depende totalmente de LLC para la recuperación de errores y las capacidades de transmisión.
<b>4. Transporte</b>	<b>LLC</b>	Los datos superiores de última hora se encapsulan en tramas LLC. LLC también es responsable de la corrección de errores y proporciona diferentes niveles de servicio a través del protocolo de acceso al enlace en el canal G (LAPG).
<b>3. Red</b>	<b>RLC</b>	En la pila del protocolo GPRS, las capas RLC y MAC están casi combinadas en una sola, a menudo denominada capa RLC/MAC.
<b>2. Enlace de Datos</b>	<b>MAC</b>	
<b>1. Física</b>	<b>Radiofrecuencia GSM</b>	La función GSM RF constituye la primera capa de la pila de protocolos GPRS. Esta capa de la pila de protocolos controla el funcionamiento de todos los datos físicos para transmitir y recibir datos de carga útil, y también controla el control del enlace.

---

La pila de protocolos GPRS es una entidad en el terminal GPRS, aunque sea software y no un elemento de hardware definido.

---

Fuente: Adaptado de (Ortega & Molano, 2016) Elaborado por: Chicaiza F. 2022

### **2.8.2.3. Dispositivos GPRS**

Los dispositivos GPRS son muy versátiles. Estos dispositivos permiten al propietario utilizarlos para muchas cosas diferentes. Cualquier dispositivo con capacidades GPRS permitirá al usuario no sólo la comunicación móvil, sino también el acceso a Internet, como el correo electrónico y la navegación por Internet(SSR, 2019).

Algunos dispositivos pueden incluso utilizarse como medio de comunicación entre dos personas, como un padre y un hijo o un familiar y una persona mayor. Esto se consigue manteniendo una parte del dispositivo con cada parte. Incluso puede ajustar varias configuraciones en el dispositivo para hacer un seguimiento del pensamiento, como los cambios de temperatura, el sonido y el movimiento. Los dispositivos GPRS también tienen la capacidad de ampliar su funcionalidad permitiendo a los usuarios añadir cosas como cámaras y receptores GPS(Huete, 2019).

El GPRS es una tecnología que muchos dispositivos de localización GPS utilizan para obtener la información más actualizada a través del seguimiento. Una vez que el dispositivo GPS ha registrado los datos, éstos pueden transmitirse a través de GPRS a otra ubicación central, como un ordenador o por correo electrónico. Es la tecnología GPRS la que permite que los sistemas de localización GPS se actualicen en tiempo real. Esta conexión GPRS directa es la que proporciona a los usuarios de los sistemas GPS los datos más fiables que existen actualmente en el mercado(Acá Cio Id et al., 2022).

A continuación, se citan algunos ejemplos de Dispositivos que usan tecnología GPRS.

- **Tracker gadget EV-07 GPRS/GPS:** Mini localizador global en tiempo real para adultos mayores de la tercera edad. Dispositivo de seguimiento GPRS / GPS con botón SOS rastreador gps.
- **PAX s910 GPRS:** El terminal POS móvil S910 de PAX ha sido diseñado para ofrecer un rendimiento inalámbrico superior, integrado en un factor de forma resistente, múltiples opciones de comunicación como SIM simple o SIM dual GPRS, CDMA, WiFi y 3G, así como una gran memoria y una batería recargable de iones de litio de alta capacidad(Alibaba, 2022).
- **Amitek Vehicle Car:** Localizador de Coche GPS GSM SMS GPRS Dispositivo de seguimiento. Admite formas de posicionamiento de estación GPS / GSM. Establece transferencia de datos de red SMS/GPRS/Internet. Tecnología GPRS en línea y la línea se vuelve a conectar automáticamente. Soporte de trabajo basado en la red GSM/GPRS existente(Amazon, 2015).

## **2.9. Diseño del sistema electrónico**

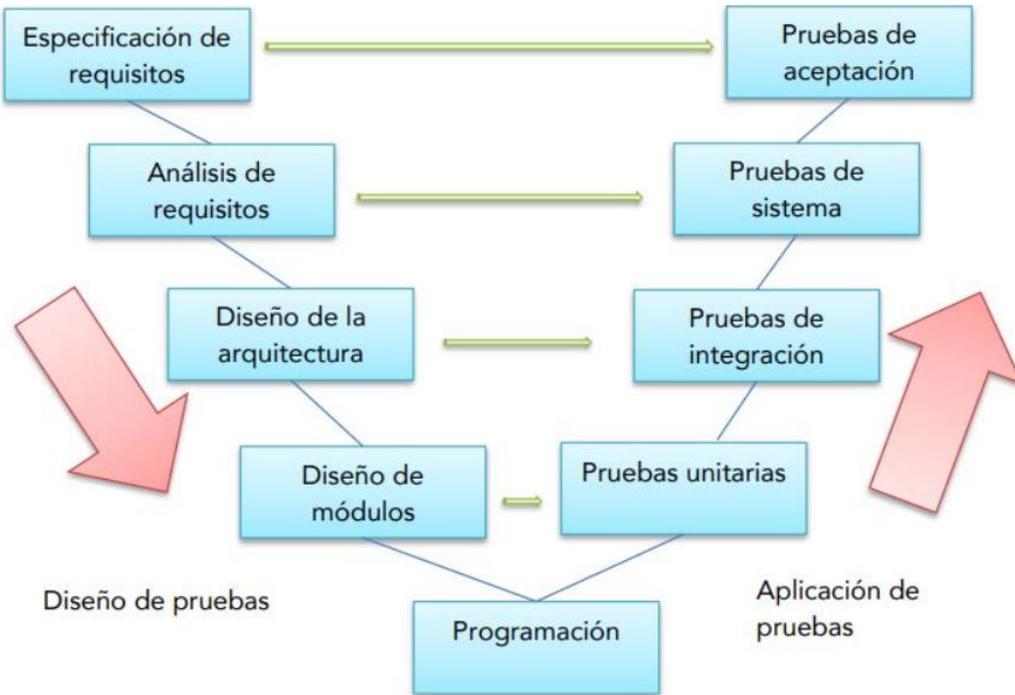
Para el diseño del sistema Electrónico es muy importante tener en cuenta las metodologías y normas que se van a usar por lo cual una vez analizadas y comprendidas se procede a seguir las indicaciones paso a paso para obtener un proyecto eficiente. De esta forma se a escogido utilizar la metodología del modelo en V y la normativa IEEE 29148 para lograr un sistema Electrónico de acuerdo con los objetivos planteados en el presente proyecto.

### **2.9.1. Modelo en V**

Para (Hamilton, 2022), el modelo V es actualmente uno de los procesos de desarrollo de software más utilizados. La introducción del modelo V justifica de hecho la realización de

pruebas a partir de la fase de requisitos. Además, es un modelo regulado en el que hay una fase de pruebas paralela a cada fase de desarrollo. el modelo V es una extensión del modelo de cascada en el que las pruebas de cada fase se realizan en paralelo al desarrollo de forma secuencial y se lo conoce como modelo de validación o verificación.

A continuación, en la Figura 4, se muestran las fases del modelo en V.



**Figura 4.** Fases del modelo en V.

Fuente: Adaptado de <https://grhsoftware.com/wp-content/uploads/2021/01/modelo-v-aplicado-en-la-verificacion-del-software.png>

A continuación, se definen las funciones de cada fase.

- Fase 1: En esta fase se recogen, analizan y estudian los requisitos
- Fase 2: En esta fase se construye el diseño del software. Se estudia e investiga cómo se van a aplicar los requisitos. También se investiga la viabilidad técnica de los requisitos y se propone los módulos/dependencias y los requisitos de hardware/software que se crearán(Childs, 2019).

- Fase 3: En esta fase se crea la arquitectura del sistema a partir de un diseño. Los módulos, sus relaciones y dependencias, los diagramas de arquitectura y los detalles técnicos se finalizan en esta fase
- Fase 4: En esta fase, se realiza la verificación a cada módulo de Hardware y Software de esta forma se verifica el correcto funcionamiento de cada módulo.
- Fase 5: Se integra y se combinan los módulos que compone el sistema y se Implementa el sistema en el entorno respectivo.

### **2.9.2. Benchmark**

Según (Fisher, 2021), un Benchmark es una prueba que mide el rendimiento del hardware, el software o los ordenadores. Estas pruebas ayudan a comparar la calidad de un producto con la de otros. Cuando se comparan los puntos de referencia, cuanto más alto es el valor del resultado, más rápido es el componente, el software o el hardware en su conjunto.

En el caso del presente trabajo de titulación se establece un listado de productos que se encuentran actualmente dentro del mercado y lo siguiente a realizar será un análisis comparativo el cual se lo interpretará en tablas. De esta forma las tablas ayudaran a determinar mediante los requerimientos que se establezca, para seleccionar al que mejor cumpla con los requisitos antes planteados y de esta forma poder realizar correctamente el Benchmark para poder satisfacer a los objetivos que se plantearon al inicio.

### **2.9.3. IEEE 29148**

Esta norma internacional ofrece un tratamiento unificado de los procesos y productos relacionados con los requisitos de ingeniería de todo el ciclo de vida del sistema y del software.

- Conceptos

- Fundamentos de los requisitos: En este apartado habla acerca de la ingeniería de requisitos la cual implica el descubrimiento, elicitación, desarrollo, análisis, verificación (incluidos los métodos y estrategias de verificación), validación, comunicar, documentar y gestionar los requisitos.
  - Consideraciones prácticas: Informa las formas de aplicación del proceso - iterativa y recursiva – las cuales son esenciales y útiles para aplicar los procesos definidos.
  - Elementos de información sobre los requisitos: describe la relación entre el proceso de requisitos y el proyecto de información de requisitos, ilustrando los estilos típicos de aplicación en el proyecto
- Procesos
  - Proceso de requisitos: describe los procesos relacionados con los requisitos con el fin de proporcionar una guía adicional de planificación e implementación para los usuarios.
  - Proceso de definición de requisitos de usuario (Stakeholders): define los requisitos de las Stakeholders para un sistema que pueda proporcionar las capacidades que necesitan los usuarios en un entorno definido.
  - Proceso de definición de requisitos del sistema: Define los límites funcionales del sistema [sistema o elemento de software], es decir, el comportamiento y las propiedades que deben proporcionarse. y atributos que se deben proporcionar.
  - Actividades de ingeniería de requisitos en otros procesos técnicos: Define las interfaces e interacciones entre los elementos del sistema con los sistemas externos.

- Gestión de requisitos: Seleccionar las tareas del sistema de que se identifican de forma exclusiva como elementos de configuración, gestión de información y medición de requisitos.
- Elementos de Información
  - Documento de especificación de requisitos de Stakeholders.
  - Documento de especificación de requisitos del sistema.
  - Documento de especificación de requisitos de software.

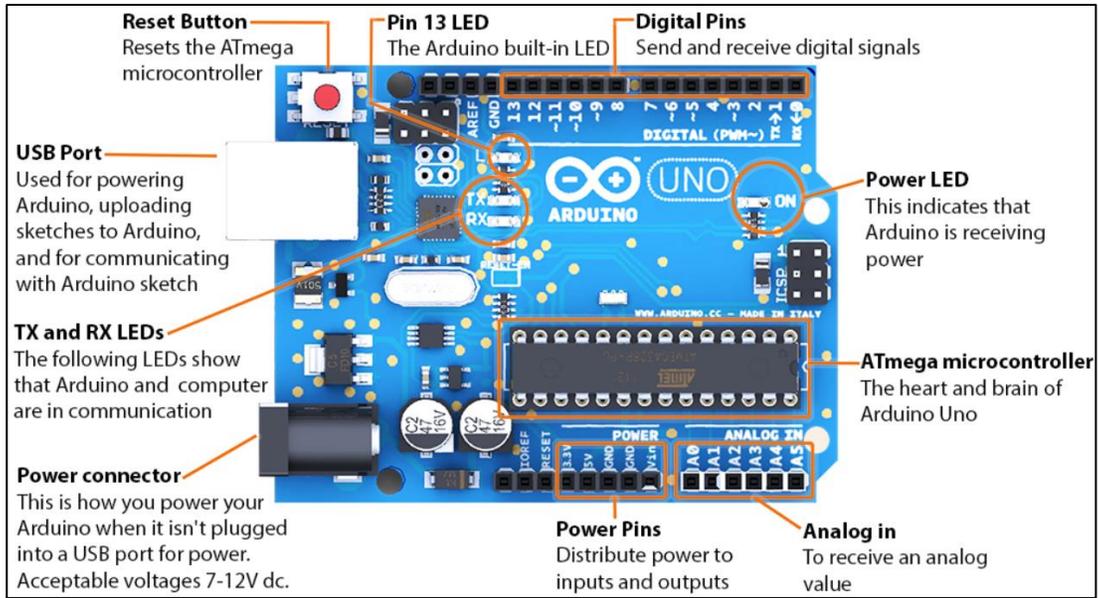
## **2.10. Placas de desarrollo y Módulos**

Las placas de desarrollo son placas de circuito impreso con un microcontrolador /microprocesador dentro de ellas y algunos otros componentes de hardware. Las placas de desarrollo están diseñadas para que los diseñadores de sistemas se familiaricen con la programación de un procesador integrado y también para desarrollar y probar proyectos de manera efectiva y eficiente(Mujahid, 2020).

### ***2.10.1. Placa programable eficiente para la compatibilidad de los módulos GPS y GSM.***

#### **Arduino**

Arduino es una plataforma electrónica de código abierto basada en un hardware y un software fáciles de usar. La placa Arduino es capaz de leer entradas -luz en un sensor, un dedo en un botón o un mensaje de Twitter y convertirlas en salidas -activar un motor, encender un LED, publicar algo en Internet. Prácticamente se puede decir a la placa lo que debe hacer enviando un conjunto de instrucciones al microcontrolador de la placa. Para ello, se utiliza el lenguaje de programación Arduino (basado en Wiring) y el software Arduino (IDE) basado en Processing(Arduino, 2018).



**Figura 5.** Arduino UNO, vista superior.

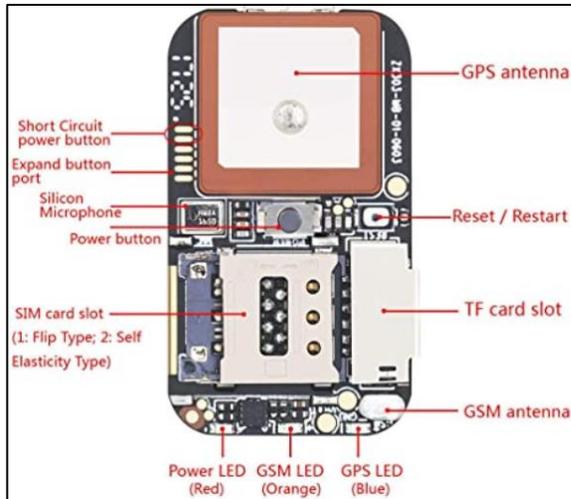
Fuente: Adaptado de <https://arduinooint.com/wp-content/uploads/Components-of-Arduino-UNO-Board.jpg>

En la Figura 5, se detalla los pines de conexión del Arduino al igual que se permite observar el microchip ATmega que se está usando. Este compuesto de un Botón que sirve para resetear la placa en caso de que se presente errores, y de igual manera cuenta con un led integrado el cual parpadea cuando se conecta a una fuente de alimentación de 5v.

### **2.10.2. Módulos ideales para el rastreo de objetos**

#### **SODIAL Zx303**

Es un módulo GPS GSM WiFi Localizador Lbs el cual pose un modo de posicionamiento múltiple y actualización de hora automática mediante el servicio de hora GPS preciso. En la Figura 6, se muestra gráficamente los módulos y se detalla los pines de conexión.

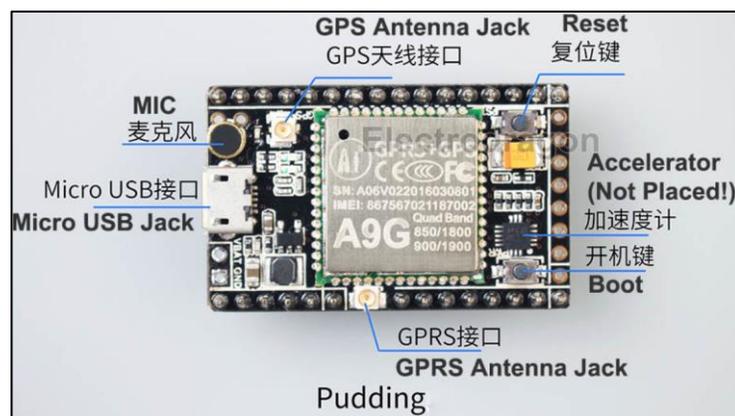


**Figura 6.** SODIAL Zx303, vista superior.

Fuente: Adaptado de [https://m.media-amazon.com/images/I/61X-Ksp3DVL.\\_AC\\_SX466\\_.jpg](https://m.media-amazon.com/images/I/61X-Ksp3DVL._AC_SX466_.jpg)

## A9G

Es un módulo que permite realizar y recibir llamadas de voz, recepción y envío de mensajes SMS y también permite conectarse a Internet mediante GPRS y poder acceder al Internet de las cosas (IoT). Soporta GSM en las bandas de 850/900/1800/1900MHz. Incluye socket para tarjeta micro-SIM. En la Figura 7, se muestra gráficamente el módulo y se detalla los pines de conexión.



**Figura 7.** A9G, vista superior.

Fuente: Adaptado de [https://www.openhacks.com/uploadsproductos/a9g-gsmgprsgpsbds-module-dev\\_6v.jpg](https://www.openhacks.com/uploadsproductos/a9g-gsmgprsgpsbds-module-dev_6v.jpg)

### **3. CAPITULO III Desarrollo Experimental**

En el presente capítulo se procede a realizar el análisis de la situación actual, definición de requerimientos y la metodología que se va a utilizar para el desarrollo del presente proyecto, a su vez se realizará la elección de hardware y software para lo cual se hará uso de la metodología en V y el estándar IEEE29148 los cuales serán ejes importantes en la descripción del ciclo de vida del proyecto y el desarrollo del sistema.

#### **3.1. Metodología**

Es fundamental seguir un conjunto de principios y procesos para gestionar un proyecto el cual ayude a definir una estructura, las directrices del proceso, las actividades de prueba y los resultados. De esta forma permitan obtener un enfoque conciso de lo que se desee obtener y como se lo pueda lograr.

Los pasos metodológicos se basan en la forma en que se ejecuta el método, el cual emplea una serie de indicaciones de forma empírica, todo este procedimiento sigue un orden de ejecución y determinan un sistema el cual asegura que la finalidad de la investigación logre ser alcanzada.

La metodología que se utiliza para el desarrollo del presente proyecto es el denominado “Modelo en V” o también conocido como modelo de cuatro niveles el cual cumple con el procedimiento de creación del proyecto, esta metodología es esencial para proyectos pequeños y medianos, donde los requisitos están estrictamente predefinidos. Además, trabaja con el estándar IEEE29148 para poder definir las especificaciones y requisitos del sistema, esta norma internacional ofrece un tratamiento unificado de los procesos y productos relacionados con los requisitos de ingeniería de todo el ciclo de vida del sistema y del software.

### 3.2. Análisis de la situación Actual

En este apartado se procede a presentar el análisis de la situación Actual, donde se aborda temas de importante relevancia como lo son las encuestas aplicadas hacia personas que sean dueñas de las mascotas dentro de la ciudad de Ibarra, de esta forma una vez aplicada la encuesta, servirá para poder esclarecer las preguntas que se tiene en relación con el diseño del sistema Embebido. Además, esta sección hace hincapié en la descripción general de la metodología que se va a implementar en el presente proyecto como lo es el modelo en V para la definición de requerimientos de Stakeholders, Hardware y Software.

Mediante el análisis de la situación actual que busca recopilar información para responder las preguntas referentes al diseño del sistema y lograr de esta manera sustentar el desarrollo del proyecto, la información se obtendrá a través de entrevistas a personas que posean mascotas dentro de la ciudad de Ibarra.

A continuación, en la Tabla 2 se presenta la información relacionada a la situación actual definiendo el método y formato a usarse para el desarrollo del presente proyecto.

**Tabla 2.**

Herramientas para el levantamiento de información de la situación actual.

---

<i>Situación Actual</i>	
<b>Método</b>	El método efectuado que se establece para el desarrollo del presente proyecto será el método de Investigación aplicada tecnológica para la recolección de información mediante una encuesta. Este método se selecciona ya que sirve para generar conocimientos que se puedan poner en

---

---

práctica en el desarrollo del diseño del sistema embebido, con el fin de impulsar un impacto positivo en la vida cotidiana de las personas.

---

**Formato**

El tipo de encuesta que se ejecutará será mediante la encuesta de tipo analítico ya que esta encuesta permite realizar preguntas directamente al dueño de la mascota, para así obtener una aproximación a lo que piensa, siente o ha vivido, de esta forma se podrá obtener varias respuestas en relación con el diseño del sistema embebido.

El tipo de preguntas que se aplicó fueron preguntas de respuesta cerrada.

Las preguntas cerradas tienen respuestas muy distintas, se pueden usar estas respuestas asignando un valor a cada respuesta. Esto facilita la comparación de las respuestas de diferentes personas, lo que, a su vez, permite el análisis estadístico de los resultados de la encuesta. Las preguntas realizadas fueron en base a el paseo habitual de las mascotas, la seguridad de las mascotas; si se conoce o no tecnología para evitar la pérdida de las mascotas, el conocimiento de aplicaciones de monitoreo y rastreo de mascotas; por lo cual se elige este tipo de preguntas para que la persona a la que se realiza la encuesta se encuentre en una total libertad de responder en su rutina habitual al cuidado de su mascota.

---

### ***3.2.1. Análisis de Resultados***

Una vez aplicada la encuesta a las personas que concurren por los diferentes parques de la urbe, se obtuvo resultado que aportan un avance significativo al desarrollo del proyecto. A

continuación, se plantea las conclusiones con respecto a la tabulación de las encuestas las cuales se especifican en el Anexo 1.

Al aplicar las encuestas a usuarios dentro de la ciudad de Ibarra permite adquirir sus requerimientos con respecto a la implementación del sistema, de esta forma, se puede aplicar un plan de estrategia para complacer las necesidades del usuario. En relación con el sistema embebido, se debe implementar una solución mediante modulo GPS para indicar la ubicación mediante coordenadas de latitud y longitud que se establezca gracias a la ayuda de este módulo. También para la etapa de alerta por notificaciones los usuarios describen que la mejor forma de alertar sobre su mascota será a través de mensajes de texto mediante la red móvil hacia su Smartphone ya que esto permite ser más cómodo para el usuario al traer siempre consigo su teléfono móvil todo el tiempo.

Finalmente, la manera en que el usuario espera que se presente el monitoreo en tiempo real será mediante su smartphone con la ayuda de una aplicación que le permita acceder de manera inmediata a la ubicación de su mascota.

### **3.3. Introducción al Desarrollo del Proyecto**

Hoy en día los sistemas embebidos se han convertido en un elemento muy valioso en la vida cotidiana de las personas y por si fuera poco de igual forma lo han hecho en la vida de las mascotas ya que estos sistemas están diseñados para cubrir necesidades específicas; por lo cual las aplicaciones que se les pueda dar a estos sistemas son ilimitadas.

#### ***3.3.1. Propósito del Sistema***

El desarrollo del presente proyecto está enfocado a el rastreo en tiempo real de las mascotas, mediante una red IoT el cual ayude a evitar la desaparición y hurto de mascotas dentro de la ciudad de Ibarra. La información del rastreo en tiempo real se podrá visualizar en una aplicación mediante la API de Google Maps, de esta forma el dueño pueda sentirse lo más cómodamente posible con la seguridad de su mascota mediante el sistema implementado.

De igual forma es importante mencionar que el sistema a desarrollar se denominara **PETROUTE**.

### ***3.3.2. Ámbito del Sistema***

Gracias al uso del módulo GPS y el módulo GPRS se puede obtener la ubicación en tiempo real y notificar al smartphone del dueño de la mascota sobre el posicionamiento de su mascota.

También se plantea que el sistema posea una interfaz de visualización amigable con el usuario que en este caso corresponde al dueño de la mascota; bajo este concepto se reducirá la sobrepoblación de animales callejeros que se encuentran deambulando por las calles de la ciudad de Ibarra.

### ***3.3.3. Características de los beneficiarios***

Los usuarios beneficiados indirectamente serán los dueños de las mascotas, el sistema funcionara en base a la seguridad de prevención de hurto o perdida de mascotas. El diseño del sistema presenta características amigables para el usuario, es decir el dueño de la mascota a través de la interfaz gráfica de la aplicación podrá acceder a la API de Google Maps y monitorear el posicionamiento de su mascota en tiempo real.

Los usuarios beneficiados directamente serán las mascotas ya que mediante la implementación de este sistema se espera que se reduzca en gran parte la sobrepoblación de animales callejeros que existen en la ciudad de Ibarra.

### **3.4. Requerimientos del Proyecto**

Una vez analizado el proyecto y la información adquirida para ejecutar el desarrollo del prototipo es posible definir los elementos que serán necesarios y que puedan cumplir con las necesidades presentadas.

Por lo tanto, a través de un listado, se pone a una evaluación los requerimientos de usuario, requerimientos de sistema y requerimientos de arquitectura.

#### **3.4.1. Stakeholders**

Hace referencia a las personas involucradas en el desarrollo del sistema, tanto quienes lo desarrollan como hacia quienes va dirigido, a continuación, en la Tabla 3 se especifica un listado de los Stakeholders que integran el equipo de trabajo en la investigación y desarrollo del proyecto.

**Tabla 3.**

Lista de Stakeholders

<b>LISTA DE STAKEHOLDERS</b>
<b>1. Comunidad Ibarreña</b>
<b>2. Personas propietarias de Mascotas</b>
<b>3. MsC. Carlos Vásquez (Director del trabajo de Titulación)</b>
<b>4. MsC. Fabián Cuzme (Asesor del trabajo de Titulación)</b>

### ***3.4.2. Construcción de atributos de los requerimientos***

En este apartado se hace referencia a la construcción de atributos de los requerimientos, los cuales se llevan a cabo en base a tres requerimientos, que son: Stakeholders, Sistema y Arquitectura; de esta forma cada requerimiento deberá efectuarse de manera exitosa para poder cumplir con el objetivo del proyecto.

#### **3.4.2.1. Nomenclatura de los requerimientos**

A continuación, se presenta la abreviatura para el diseño de tablas que contienen la información relevante relacionada al proyecto; de esta forma se permite un mejor manejo de datos. En la Tabla 4 se define la nomenclatura utilizada.

**Tabla 4.**

Definición de Abreviaturas

<b>Descripción del requerimiento</b>	<b>Acrónimo</b>
<b>Stakeholders</b>	<b>StSR</b>
<b>Sistema</b>	<b>SySR</b>
<b>Arquitectura</b>	<b>SRSR</b>

### ***3.4.3. Requerimientos de Stakeholders***

El proceso de definición de requisitos de Stakeholders traduce las necesidades de capacidad en un conjunto de requisitos técnicos. Este proceso ayuda a garantizar que se

comprendan los requisitos, las expectativas y las limitaciones percibidas de cada Stakeholders desde la perspectiva de adquisición.

En la Tabla 5 se analizan concretamente los requerimientos operacionales y de usuario que tienen que ver con la interrelación directa del sistema y usuario.

**Tabla 5.**

Requerimientos de Stakeholders

<b>StSR</b>					
#	REQUERIMIENTO	PRIORIDAD			RELACIÓN
		Alta	Media	Baja	
<b>REQUERIMIENTOS OPERACIONALES</b>					
<b>StSR1</b>	El sistema debe tener alimentación eléctrica mediante baterías(portables).	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<b>StSR2</b>	El prototipo debe ser lo más reducido posible para la comodidad de la mascota.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<b>StSR3</b>	El sistema debe tener conexión a la red inalámbrica para él envío de notificaciones.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

<b>StSR4</b>	Integración de módulos de Red Telefónica y GPS.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------	---	-------------------------------------	--------------------------	--------------------------

### REQUERIMIENTOS DE USUARIO

<b>StSR5</b>	La ubicación de la mascota se lo podrá visualizar a través de una aplicación móvil.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------	---	-------------------------------------	--------------------------	--------------------------

<b>StSR6</b>	La aplicación móvil debe ser amigable y perspicaz con el usuario.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------	---	-------------------------------------	--------------------------	--------------------------

<b>StSR7</b>	La aplicación móvil debe contener un registro de usuario.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
--------------	---	--------------------------	-------------------------------------	--------------------------

<b>StSR8</b>	Las notificaciones de alerta se enviarán mediante SMS al teléfono móvil del dueño (Usuario).	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------	--	-------------------------------------	--------------------------	--------------------------

<b>StSR9</b>	La información que se muestra en la aplicación móvil debe estar en constante actualización siempre que exista un nuevo cambio de posición (Tiempo Real).	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------	--	-------------------------------------	--------------------------	--------------------------

Fuente: Encuestas en los parques de la ciudad de Ibarra.

Los requerimientos especificados han sido obtenidos a través de la indagación mediante encuestas en los diferentes parques más conocidos y populares de la ciudad de Ibarra donde se observa bastantes personas que salen a pasear con su mascota. De tal forma que mediante la ayuda de sus puntos de vista se ha logrado adquirir toda la información posible en referencia a los requerimientos que hacen posible la relación de usuario y del sistema conciliándose de esta forma una idea general del sistema en entorno de operabilidad.

#### 3.4.4. *Requerimientos de Sistema*

Los requerimientos de Sistema (SySR) son un conjunto de información que describe las características y el comportamiento de un sistema que intentan definir la funcionalidad esperada para satisfacer a sus distintos usuarios, se analizan y consideran requerimientos de uso, interfaz, performance, estados y físicos. En la Tabla 6 se analiza los requerimientos del sistema. Este apartado guarda una estrecha relación con los requerimientos de Stakeholders ya que se describe la funcionalidad del sistema en base a la aplicación de acuerdo con la importancia del requerimiento.

**Tabla 6.**

Requerimientos de Sistema

SySR					
#	REQUERIMIENTO	PRIORIDAD			RELACIÓN
		Alta	Media	Baja	
<b>REQUERIMIENTOS DE INTERFAZ</b>					
SySR1	La placa de desarrollo del sistema debe presentar una interfaz	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

	serial para la conexión del módulo GPRS.			
<b>SySR2</b>	El sistema presenta conexión a internet para la transmisión de datos.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>SySR3</b>	La placa de desarrollo debe contar con pines, tanto digitales como seriales para la conexión de módulos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SySR4</b>	El sistema debe interactuar con todos los módulos conectados a la placa de desarrollo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### REQUERIMIENTOS DE USO

<b>SySR5</b>	El sistema Electrónico se energiza mediante baterías.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SySR6</b>	Los datos del sistema de monitoreo deben ser obtenidos rápidamente.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>SySR7</b>	El sistema presenta una ejecución de rastreo de la mascota en tiempo real.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SySR8</b>	El registro de ubicación debe	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

	almacenarse en tiempo real.			
<b>SySR9</b>	Para la ubicación de coordenadas el sistema electrónico debe presentar tecnología GPS.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### REQUERIMIENTOS DE PERFORMANCE

<b>SySR10</b>	El sistema deberá funcionar a un rango menor a los 65000 Hz ya que esta sería la capacidad máxima de los canes para percibir sonidos de alta frecuencia.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SySR11</b>	La aplicación móvil siempre debe estar en funcionamiento cuando el usuario lo requiera.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SySR12</b>	Visualización grafica de coordenadas a través de la aplicación móvil.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>SySR13</b>	La tasa de recepción y transferencia de datos se lo realiza mediante el módulo GPRS.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

---

<b>SySR14</b>	Obtención de Coordenadas GPS en tiempo real del prototipo electrónico.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
---------------	---	-------------------------------------	--------------------------	--------------------------

---

### REQUERIMIENTOS DE MODO Y ESTADO

---

<b>SySR15</b>	El sistema debe permanecer activo siempre que la mascota salga de su hogar.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
---------------	---	-------------------------------------	--------------------------	--------------------------

---

### REQUERIMIENTOS FÍSICOS

---

<b>SySR16</b>	La ubicación del dispositivo se lo colocara en el cuello de la mascota para su comodidad.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
---------------	---	-------------------------------------	--------------------------	--------------------------

---



---

<b>SySR17</b>	El dispositivo electrónico cuenta con un tamaño reducido para que pueda cumplir sus funciones.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
---------------	--	-------------------------------------	--------------------------	--------------------------

---

#### ***3.4.5. Requerimientos de Arquitectura***

Los requerimientos de arquitectura hacen referencia a los requerimientos que el sistema necesita para su funcionamiento por lo cual en este apartado se definen los requerimientos de

Software, Hardware, Lógicos, Diseño y Eléctricos. A continuación, en la Tabla 7 se interpretan los requerimientos de Arquitectura.

**Tabla 7.**

Requerimientos de Arquitectura

<b>SRSH</b>				
<b>PRIORIDAD</b>				
#	REQUERIMIENTO	RELACIÓN		
		Alta	Media	Baja
<b>REQUERIMIENTO LÓGICOS</b>				
<b>SRSH1</b>	El sistema debe tener compatibilidad con una placa embebida.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH2</b>	El sistema requiere el uso de un Sistema Operativo, el cual ayude a la ejecución de acciones del sistema.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH3</b>	Se requiere que el software sea compatible con librerías y bibliotecas de Geoposicionamiento.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH4</b>	Aplicación móvil para tener conexión al sistema embebido.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>REQUERIMIENTO DE DISEÑO</b>				

<b>SRSH5</b>	El sistema debe tener comunicación con la base de datos y la aplicación móvil.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH6</b>	El sistema debe ser accesible para el dueño de la mascota.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>SRSH7</b>	Unión del módulo de comunicación con la placa de procesamiento en un circuito único.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH8</b>	El dispositivo electrónico debe presentar un case con parámetros de resistencia para evitar daños a la placa interna.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>SRSH9</b>	El software de la placa de desarrollo debe presentar librerías de compatibilidad con el módulo de comunicación inalámbrica	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>REQUERIMIENTOS DE HARDWARE</b>				
<b>SRSH10</b>	El módulo de Comunicación	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Inalámbrica deberá tener un slot para poder insertar una tarjeta SIM.			
<b>SRSH11</b>	El sistema deberá contar con un módulo GPS para obtener las coordenadas de posicionamiento.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH12</b>	El dispositivo debe tener un tamaño compacto para comodidad de la mascota.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>SRSH13</b>	Se requiere una placa de desarrollo accesible.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH14</b>	La placa de desarrollo debe contar con entradas y salidas análogas y/o digitales para manejo de módulos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH15</b>	La placa de desarrollo debe contar con el adecuado número de pines para la conexión de sus elementos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH16</b>	El módulo de comunicación inalámbrica requiere	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

---

una alimentación  
mayor a 2A y un  
voltaje de 3.7 voltios  
para su correcto  
funcionamiento.

---

### REQUERIMIENTO DE SOFTWARE

---

**SRSH17** El Lenguaje de programación debe ser compatible con la placa de desarrollo y el módulo de comunicación inalámbrica.

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-------------------------------------	--------------------------	--------------------------

---

**SRSH18** Se requiere software que permita ejecutar las coordenadas de geoposicionamiento del sistema embebido en tiempo real.

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-------------------------------------	--------------------------	--------------------------

---

**SRSH19** Se requiere librerías que sean compatibles con los módulos del sistema embebido.

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-------------------------------------	--------------------------	--------------------------

---

**SRSH20** El software de la placa de desarrollo debe permitir programación por comandos AT para la conexión del módulo inalámbrico de comunicación.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
--------------------------	-------------------------------------	--------------------------

---

<b>SRSH21</b>	La programación de módulos se la realizara bajo software Open Source.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH22</b>	La base de datos del sistema debe ser Open Source.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH23</b>	La aplicación móvil tiene que ser intuitiva y amigable con el usuario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>SRSH24</b>	La aplicación móvil deber ser realizada en software Open Source	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### REQUERIMIENTO ELÉCTRICOS

<b>SRSH25</b>	Regulador de voltaje DC-DC STEP-DOWN o STEP-UP	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH26</b>	Alimentación mediante Bateria de 3,7 V a 2000mAh.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>SRSH27</b>	Fuente de Alimentación externa para recargar la Bateria	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Los requerimientos analizados anteriormente se han establecido bajo una relación con las necesidades e ideas que han planteado los usuarios con el sistema, de tal forma que se establezca

un sistema sólido bajo los criterios y características efectivas tanto de hardware como de software.

### **3.5. Selección de Hardware y Software**

En este apartado se seleccionará el Hardware y Software idóneos para el desarrollo del proyecto, la selección tiene que ser efectiva y acertada, por lo cual se utilizará algunas metodologías que permiten elegir los mejores elementos para tener una elección más técnica, tomando a consideración estos puntos de vista mediante la metodología de Benchmark se seleccionará los elementos correspondientes al hardware del desarrollo del proyecto basándose en las hojas de datos (datasheets) de cada elemento y para seleccionar los elementos correspondientes al software se utilizará la integración de los requerimientos de Stakeholders, Sistema y Arquitectura. De esta manera una vez seleccionado a los mejores candidatos de cada tipo se procede a realizar un análisis comparativo para la evaluación de cada uno y finalmente escoger al mejor candidato y por ende la mejor opción factible para el desarrollo del proyecto.

#### ***3.5.1. Selección de Hardware***

La selección de Hardware se lleva a cabo tomando en cuenta la metodología de Benchmark y los requerimientos de Hardware que se establecieron en la tabla 6 la cual hace referencia a los requerimientos de Arquitectura. Puntualmente se seleccionará la placa de desarrollo y los módulos que vayan a usarse para la ubicación y la conexión inalámbrica de red.

##### ***3.5.1.1. Elección de Placa de desarrollo***

Para la elección de la placa en desarrollo se estableció 6 opciones, las cuales poseían características relacionadas a las necesidades del proyecto. A continuación, se presenta la forma de evaluación que se realizó para poder calificar a los requerimientos que se establecieron anteriormente, tomando en cuenta el requerimiento de Stakeholders **StSR4**, los requerimientos

de sistema: **SySR1**, **SySR3** y los requerimientos de arquitectura **SRS7**, **SRS12**, **SRS13**, **SRS14**, **SRS15**. En la Tabla 8 se detallan los aspectos valorados donde el numero 1 significa que cumple el requerimiento y el numero 0 indica que no cumple.

**Tabla 8.**

Selección de Placa de Desarrollo

<b>PLACA DE DESARROLLO</b>									
<b>HARDWARE</b>	<b>REQUERIMIENTO</b>								<b>VALOR</b>
	<b>SySR4</b>	<b>SySR1</b>	<b>SySR3</b>	<b>SRS7</b>	<b>SRS12</b>	<b>SRS13</b>	<b>SRS14</b>	<b>SRS15</b>	<b>TOTAL</b>
<b>Arduino Uno</b>	1	1	1	1	0	1	1	1	7
<b>Arduino Nano</b>	1	1	1	1	1	1	1	1	8
<b>SODIALZx303</b>	1	1	0	1	1	0	0	0	4
<b>A9G</b>	1	1	1	1	0	1	1	0	6
<b>ESP8266</b>	1	1	1	1	0	1	1	1	7
<b>ESP32</b>	1	1	1	1	0	1	1	1	7
<b>Cumple "1"</b>									
<b>No cumple "0"</b>									

Elección: La placa de desarrollo que ha permitido cumplir con los requerimientos planteados para este proyecto ha sido la placa de desarrollo Arduino Nano ya que es una placa compacta con capacidad de procesamiento ideal, su tamaño a diferencia del Arduino Uno facilita la interconexión con otros módulos en tamaño reducido lo que lo hace esencial para el diseño del prototipo, posee una entrada mini-USB por la cual se puede subir el código fuente para ejecutar el programa y está basado en el microcontrolador ATmega328.

A continuación, en la Tabla 9 se presenta las características técnicas que presenta el hardware seleccionado, en este caso el Arduino Nano ya que es la mejor elección para el desarrollo del proyecto.

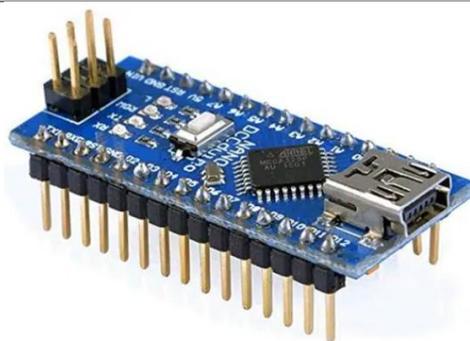
**Tabla 9.**

Especificaciones Técnicas de Arduino Nano

---

**ARDUINO NANO**

---



ESPECIFICACIÓN	PROPIEDADES
<b>Microcontrolador</b>	ATmega328
<b>Pines de e/s digitales</b>	22 (6 de los cuales son PWM)
<b>Arquitectura</b>	AVR
<b>Tensión de funcionamiento</b>	5 voltios
<b>Salida pwm</b>	6
<b>Memoria flash</b>	32 KB de los cuales 2 KB utilizados por el gestor de arranque
<b>Consumo de energía</b>	19mA
<b>Sram</b>	2 KB
<b>Velocidad de reloj</b>	16 MHz

---

<b>Tamaño de placa de circuito impreso</b>	18x45mm
<b>Pines de entrada analógica</b>	8
<b>Eeprom</b>	1 KB
<b>Peso</b>	7g
<b>Corriente cc por pines de e/s</b>	40 mA (pines de E/S)
<b>Voltaje de entrada</b>	7-12V

Fuente: Adaptado de (Arduino, 2021) Elaborado por: Chicaiza F. 2022

### ***3.5.1.2. Elección del módulo de comunicación inalámbrica GPRS***

El módulo que se encargará de la transmisión de los datos entre el microcontrolador y la red inalámbrica será el módulo GPRS. Como el sistema se ejecutará bajo la acción del teléfono móvil del dueño es fundamental que el módulo funcione bajo la red telefónica móvil por lo cual se estableció 4 opciones, las cuales se alineaban a las necesidades del proyecto. A continuación, se presenta la forma de evaluación que se realizó para poder calificar a los requerimientos que se establecieron anteriormente, tomando en cuenta el requerimiento de Stakeholders **StSR4, StSR8**, los requerimientos de sistema: **SySR1, SySR4, SySR13** y los requerimientos de arquitectura **SRS7, SRS16, SRS17, SRS19**.

**Tabla 10.**

Selección de Módulo de comunicación inalámbrica GPRS

---

## **MÓDULO DE COMUNICACIÓN INALÁMBRICA GPRS**

---

<b>HARDWARE</b>	<b>REQUERIMIENTO</b>
-----------------	----------------------

---

	S <sub>t</sub> SR <sub>4</sub>	S <sub>t</sub> SR <sub>8</sub>	S <sub>y</sub> SR <sub>1</sub>	S <sub>y</sub> SR <sub>4</sub>	S <sub>y</sub> SR <sub>13</sub>	S <sub>R</sub> SH <sub>7</sub>	S <sub>R</sub> SH <sub>16</sub>	S <sub>R</sub> SH <sub>17</sub>	S <sub>R</sub> SH <sub>19</sub>	VALOR TOTAL
<b>SIM 808</b>	1	1	1	1	1	0	1	1	1	8
<b>SIM 868</b>	1	1	1	1	1	0	0	1	1	7
<b>SIM 900</b>	1	1	1	1	1	0	0	1	1	7
<b>SIM 800L</b>	1	1	1	1	1	1	1	1	1	9
Cumple "1"										
No cumple "0"										

**Elección:** El Módulo de comunicación inalámbrica GPRS que ha permitido cumplir con los requerimientos planteados para este proyecto ha sido el Módulo GSM GPRS SIM800L ya que es un módulo celular en miniatura que permite la transmisión GPRS, el envío y recepción de SMS. El bajo costo y el tamaño pequeño y el soporte de frecuencia de banda cuádruple hacen que este módulo sea el seleccionado para el desarrollo del proyecto el cual requiere conectividad de largo alcance. Se requiere un microcontrolador para controlarlo, de esta forma se usará el Arduino Nano que se seleccionó en la tabla 8, ya que con el UART que este posee se puede enviar y recibir comandos a través de los pines RX/TX.

A continuación, en la Tabla 11 se presenta las características técnicas que presenta el hardware seleccionado, en este caso el Módulo GSM GPRS SIM800L ya que tiene las mejores características de funcionalidad y su tamaño compacto hacen que sea la mejor opción para el desarrollo del proyecto.

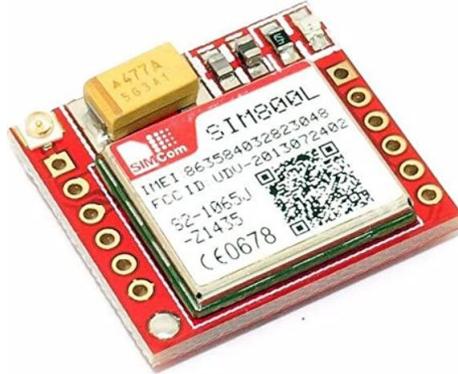
**Tabla 11.**

Especificaciones Técnicas del módulo GSM GPRS SIM800L

---

**MÓDULO GSM GPRS SIM800L**

---



<b>ESPECIFICACIÓN</b>	<b>PROPIEDADES</b>
<b>Voltaje de Operación</b>	3.4V - 4.4V DC
<b>Nivel Lógico</b>	3V a 5V
<b>Consumo de corriente (max)</b>	500 mA
<b>Consumo de corriente (sleep)</b>	0.7 mA
<b>Interfaz</b>	Serial UART
<b>Quad-band</b>	850/900/1800/1900MHz
<b>Receptor</b>	FM
<b>Controlado por Comandos</b>	AT (3GPP TS 27.007, 27.005 y SIMCOM enhanced AT Commands)
<b>Tamaño de la SIM</b>	Micro SIM
<b>Velocidades de transmisión serial</b>	desde 1200bps hasta 115200bps
<b>Soporta A-GPS</b>	SI
<b>Soporta Reloj en tiempo real (RTC)</b>	SI

---

Fuente: Adaptado de(ElectroPro, 2020) Elaborado por: Chicaiza F. 2022

### 3.5.1.3. Elección del módulo GPS

El módulo que se encargará de la transmisión de los datos de las coordenadas de ubicación con el microcontrolador será el módulo GPS. Este módulo es fundamental ya que será el encargado de ubicar mediante las coordenadas de latitud y longitud la ubicación de la mascota, por lo cual se estableció 4 opciones, las cuales se alineaban a las necesidades del proyecto. A continuación, se presenta la forma de evaluación que se realizó para poder calificar a los requerimientos que se establecieron anteriormente, tomando en cuenta el requerimiento de Stakeholders **StSR2**, los requerimientos de sistema: **SySR4**, **SySR8**, **SySR9**, **SySR14** y los requerimientos de arquitectura **SRS11**, **SRS18**, **SRS19**, **SRS21**.

**Tabla 12.**

Selección de Módulo GPS

<b>MÓDULO DE COMUNICACIÓN INALÁMBRICA GPRS</b>										
<b>HARDWARE</b>	<b>REQUERIMIENTO</b>									<b>VALOR</b>
	<b>StSR2</b>	<b>SySR4</b>	<b>SySR8</b>	<b>SySR9</b>	<b>SySR14</b>	<b>SRS11</b>	<b>SRS18</b>	<b>SRS19</b>	<b>SRS21</b>	<b>TOTAL</b>
<b>GPS NEO 7M</b>	0	1	1	1	1	1	1	1	1	8
<b>GPS NEO 6M</b>	1	1	1	1	1	1	1	1	1	9

v2

<b>GPS 14030</b>	1	0	1	1	1	1	1	1	1	8
<b>SPARKFUN</b>										
<b>GPS GT-U7</b>	1	1	1	1	1	1	1	1	1	9
Cumple "1"										
No cumple "0"										

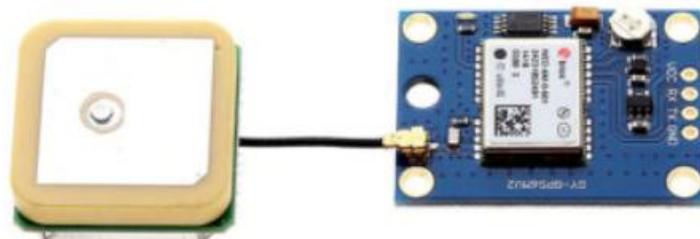
**Elección:** Para la elección del Módulo GPS se ha seleccionado el análisis de 2 módulos GPS, el módulo GPS NEO 6M v2 y el módulo GPS GT-UT los cuales muestran características muy parecidas dentro del análisis. Debido a las características de consumo, alimentación y de precio se ha decidido incorporar como solución de módulo GPS, el módulo GPS NEO 6M v2, además se elige este módulo tomando en cuenta el cumplimiento de los requerimientos y la disponibilidad dentro del País. Por otra parte, es una excelente alternativa de precisión y costo beneficio. Es de un tamaño reducido por lo que es perfecto para el desarrollo del presente proyecto, en cuanto a su funcionalidad cuenta con una antena de gran potencia, posee una memoria EEPROM para guardar datos y una batería para respaldar la configuración del módulo, lo que le permite recibir las señales de los satélites que están alrededor de la tierra.

A continuación, en la Tabla 13 se presenta las características técnicas que presenta el hardware seleccionado, en este caso el GPS NEO 6M v2.

**Tabla 13.**

Especificaciones Técnicas del módulo GPS NEO6Mv2

### MÓDULO GPS NEO6Mv2



<b>ESPECIFICACIÓN</b>	<b>PROPIEDADES</b>
<b>Dimensiones antena</b>	25 mm X 25 mm
<b>Dimensiones módulo</b>	36 mm X 25.9 mm
<b>Voltaje de alimentación mínimo</b>	3.3V
<b>Voltaje de alimentación máximo</b>	5V
<b>Corriente de funcionamiento</b>	45 mA
<b>Memoria</b>	EEPROM
<b>Pines de módulo</b>	VCC, RX, TX y GND
<b>Velocidad de comunicación</b>	9600 bps
<b>Sensibilidad de captura</b>	148 dBm
<b>Sensibilidad de rastreo</b>	161 dBm
<b>Máxima altura medible</b>	18000
<b>Máxima velocidad</b>	515 m/s
<b>Exactitud</b>	1 microsegundo
<b>Tiempo de conexión por primera vez</b>	Mínimo 5min o Hasta casi 10min
<b>Tiempo de conexión por segunda vez</b>	Mínimo 40 segundos o Hasta 1 minuto

Fuente: Adaptado de (Electronilab, 2019) Elaborado por: Chicaiza F. 2022

### **3.5.2. Selección de Software**

Una vez que se haya realizado la selección de hardware se procede a realizar la selección de Software, el software seleccionado se lo realiza en base a los atributos que se encuentran inmersos en las tablas 5 y 6, las cuales hacen referencia a los requerimientos de sistema y

arquitectura. A continuación, se presenta una evaluación del software para las diferentes fases del proyecto. En esta etapa se toman en cuenta algunas alternativas en cuanto a la codificación del sistema para la tecnología de comunicación inalámbrica, tecnología GPS, base de datos y desarrollo de aplicación móvil.

### 3.5.2.1. Software para la programación de la Placa de desarrollo Arduino Nano

Para la elección del software que ayudara en la programación de la placa de desarrollo se estableció 4 opciones, las cuales poseían características relacionadas a las necesidades del proyecto. A continuación, se presenta la forma de evaluación que se realizó para poder calificar a los requerimientos que se establecieron anteriormente, tomando en cuenta los requerimientos de arquitectura **SRSH3, SRSH9, SRSH17, SRSH18, SRSH19, SRSH20, SRSH21**. En la Tabla 14 se presenta la mejor selección del software para la realización del programa y de igual forma el lenguaje de programación con el que se va a trabajar.

**Tabla 14.**

Selección del Software para la programación de la Placa de desarrollo Arduino Nano

PLACA DE DESARROLLO								
SOFTWARE	REQUERIMIENTO							VALOR TOTAL
	SRSH3	SRSH9	SRSH17	SRSH18	SRSH19	SRSH20	SRSH21	
<b>Arduino (IDE)</b>	1	1	1	1	1	1	1	7
<b>C ++</b>	0	0	1	0	0	0	1	2
<b>Java Script</b>	1	0	0	0	0	0	1	2

<b>Python</b>	1	1	1	1	1	1	1	7
---------------	---	---	---	---	---	---	---	---

---

**Cumple “1”**

---

**No cumple “0”**

---

**Elección:** Para la elección del Software para la programación de la Placa de desarrollo Arduino Nano se ha seleccionado el análisis de 2 candidatos, el software Arduino IDE y el software Raspberry (Python), los cuales cumplen con los requerimientos de arquitectura dentro del análisis. Debido a que el Lenguaje de programación debe ser compatible con la placa de desarrollo y el módulo de comunicación inalámbrica a su vez contar con librerías que sean compatibles con los módulos del sistema embebido, se ha decidido incorporar como solución de software para la programación de la Placa de desarrollo Arduino Nano el software de Arduino IDE, además se selecciona este software ya que es de código abierto (IDE) y a su vez facilita la escritura de código y la carga en la placa del Arduino Nano ya que al ser de la misma Marca son compatibles por lo cual resulta muy beneficioso para el desarrollo del proyecto.

---

### ***3.5.2.2. Plataforma de servicio (Cloud)***

Es muy importante contar con una plataforma de servicio para el desarrollo de este proyecto ya que de esta forma existe mayor flexibilidad en la creación de la aplicación móvil y base de datos en tiempo real, tomando en cuenta la calidad y herramientas de trabajo de primer nivel. Se estableció 4 opciones, las cuales poseían características relacionadas a las necesidades del proyecto. A continuación, se presenta la forma de evaluación que se realizó para poder calificar a los requerimientos que se establecieron anteriormente, tomando en cuenta los requerimientos de arquitectura **StSR7** el cual detalla que la aplicación móvil debe contener un registro de usuario; con relación al **SySR6** los datos del sistema de monitoreo deben ser

obtenidos rápidamente, además, **SySR7** el sistema presenta una ejecución de rastreo de la mascota en tiempo real y su registro de ubicación almacenarse en tiempo real como especifica el **SySR8**. En la Tabla 15 se presenta la mejor selección del software para el servicio de almacenamiento en la nube.

**Tabla 15.**

Selección de la Plataforma de servicio (Cloud)

<b>PLACA DE DESARROLLO</b>					
<b>SOFTWARE</b>	<b>REQUERIMIENTO</b>				<b>VALOR TOTAL</b>
	<b>SySR7</b>	<b>SySR6</b>	<b>SySR7</b>	<b>SySR8</b>	
<b>Back4App</b>	1	0	1	0	2
<b>Parse</b>	1	0	1	1	3
<b>Firebase</b>	1	1	1	1	4
<b>Firehose</b>	0	1	0	0	1
<b>Cumple "1"</b>					
<b>No cumple "0"</b>					

**Elección:** Para la elección de la Plataforma de servicio (Cloud), Firebase es la seleccionada por sus prestaciones en la integración de varios servicios los cuales incluyen cosas como análisis, autenticación, bases de datos, configuración, almacenamiento de archivos, mensajería push y entre otros. Los servicios están alojados en la nube y escalan con poco o ningún esfuerzo por parte del desarrollador. De igual forma para la compatibilidad con el desarrollo de la aplicación móvil existe una biblioteca llamada FirebaseUI (Android , iOS , web )

que proporciona una serie de herramientas útiles para facilitar aún más el desarrollo con Firebase.

### 3.5.2.3. Software para el desarrollo de la aplicación móvil

La elección del software que ayudará en la programación para el desarrollo de la aplicación móvil deberá realizarse en un lenguaje abierto y de manera clara para poder cumplir con los requerimientos planteados, De este modo en la tabla 16 se selecciona algunos requerimientos como **StSR5** el cual establece que la ubicación de la mascota se podrá visualizar a través de una App móvil, en el **StSR6** indica que la aplicación móvil debe ser amigable y perspica con el usuario, además el **StSR7**, especifica que la App móvil debe contener un registro de usuario, también el **StSR9** el cual detalla que la App móvil debe estar en constante actualización siempre que exista un nuevo cambio de posición(Tiempo Real) y el **SySR12** donde indica que la visualización grafica de coordenadas se la observara a través de la aplicación móvil, por último el **SRSH24** el cual indica que la App móvil deber ser realizada en software Open Source.

**Tabla 16.**

Elección del Software para el desarrollo de la aplicación móvil

PLACA DE DESARROLLO							
SOFTWARE	REQUERIMIENTO						VALOR
	StSR5	StSR6	StSR7	StSR9	SySR12	SRSH24	TOTAL
<b>Microsoft Visual Studio</b>	1	1	1	1	0	1	5

<b>Qt Creator</b>	1	1	1	1	0	0	4
<b>Android Studio</b>	1	1	1	1	1	1	6
<b>Flutter</b>	1	1	1	1	1	1	6

**Cumple “1”**

**No cumple “0”**

**Elección:** En la elección del Software para el desarrollo de la aplicación móvil se ha seleccionado el análisis de 2 candidatos, el software Android Studio y el software Flutter, los cuales cumplen con los requerimientos dentro del análisis. Debido a que la programación para el desarrollo de aplicaciones Android implica contar con una buena interfaz gráfica que sea amigable e intuitiva con el usuario, lo cual el software de Flutter proporciona una variedad de funciones para poder diseñar de la manera más efectiva posible, a su vez Flutter es compatible con Google y Firebase. Se ha decidido seleccionar el software de Flutter ya que la compatibilidad de funciones de IDE lo hace la mejor opción.

### 3.6. Recursos

En esta sección se hace referencia a los recursos que se usan para el desarrollo del proyecto de titulación denominado PETROUTE los cuales son indispensables para alcanzar los objetivos que se plantearon al inicio. Por lo tanto, a continuación, en las siguientes tablas se detallan los recursos Humanos, Económicos y Tecnológicos.

#### 3.6.1. Recursos Económicos de Hardware

Los recursos económicos permiten establecer un costo aproximado del prototipo final, a continuación, se detalla los gastos que se realizan para la elaboración del diseño de este proyecto los cuales se muestran en la Tabla 17 que se expone a continuación.

**Tabla 17.**

Costos de Hardware

<b>COSTOS DE HARDWARE</b>			
<b>CANTIDAD</b>	<b>DESCRIPCIÓN</b>	<b>COSTO UNITARIO(USD)</b>	<b>COSTO TOTAL(USD)</b>
1	Placa de desarrollo Arduino Nano	10	12
1	Modulo GPRS SIM 800L	15	15
1	Modulo GPS NEO 6mv2	15	15
1	Baterías	10	10
1	Pchera	6,50	6,50
<b>TOTAL</b>			<b>58,5</b>

**3.6.2. Recursos Tecnológicos de Software**

A continuación, se especifica el software posible que se va a utilizar para el desarrollo del presente proyecto, esto se puede observar en la Tabla 18 a continuación.

**Tabla 18.**

Costos de Software

<b>COSTOS DE SOFTWARE</b>		
<b>SOFTWARE</b>	<b>COSTO(USD)</b>	<b>OBSERVACIÓN</b>
OFFICE 365	0	El costo es de 0\$ por licencia académica Universitaria (Universidad Tecnica del Norte)
ARDUINO IDE	0	Open-Source software
FRITZING	0	Open-Source software

EAGLE	0	El costo es de 0\$, por plan de descarga de uso educativo, valido por 1 año.
PROTEUS	248,00	
FIREBASE	0	Open-Source software
FLUTTER	0	Open-Source software
<b>TOTAL</b>	<b>248.00\$</b>	

### 3.6.3. Costos de Ingeniería

En costos de Ingeniería, se toma en cuenta el tiempo de diseño, programación y construcción empleado para el desarrollo del sistema, como dato importante según el (Ministerio de Trabajo,2023), los haberes mínimos que se acreditan a un Ingeniero Electrónico Especialista de Mantenimiento mensualmente son de \$484.05, la Tabla 19 muestra los costos de Ingeniería en relación a las actividades ejecutadas en un tiempo determinado.

**Tabla 19.**

Costos de Ingeniería

<b>COSTOS DE INGENIERÍA</b>			
<b>ACTIVIDAD</b>	<b>NRO. DE DIAS</b>	<b>VALOR DIARIO(USD)</b>	<b>VALOR TOTAL(USD)</b>
Diseño de Circuito del Sistema en FRITZING	2	16,13	32,26
Diseño PCB en EAGLE	2	16,13	32,26
Construcción de PCB	1	16,13	16,13
Programación en ARDUINO	7	16,13	112,91
Programación en FLUTTER	10	16,13	161,30

Integración de Componentes Electrónicos	1	16,13	16,13
<b>TOTAL</b>	<b>23</b>		<b>370.99\$</b>

### 3.6.4. Costos varios y de Oficina

En este apartado, se presentan los recursos de oficina usados, así también como costos adicionales que surgieron al realizar prueba tras prueba en base al funcionamiento del sistema, esto se puede visualizar en la Tabla 20 a continuación.

**Tabla 20.**

Costos varios y de Oficina

<b>COSTOS VARIOS Y DE OFICINA</b>			
<b>DESCRIPCIÓN</b>	<b>CANTIDAD</b>	<b>COSTO UNITARIO(USD)</b>	<b>COSTO TOTAL(USD)</b>
Laptop	1	800	800
Internet (Mensual)	6	25	150
Tarjeta SIM	1	4,50	4,50
Datos Móviles (Paquete de Datos Móviles)	1	10	10
Prueba errada con modulo SIM800L	1	15	15
Prueba errada con modulo fuente STEP-DOWN DC-DC LM2596	1	2,25	2,25
		<b>TOTAL</b>	<b>981,75</b>

### 3.6.5. Costo Total del Sistema

**Tabla 21.**

Costo total del Sistema

<b>COSTO TOTAL DEL SISTEMA</b>	
<b>DESCRIPCIÓN</b>	<b>COSTO TOTAL (USD)</b>
Costos de Hardware	58,50
Costos de Software	248
Costos de Ingeniería	370,99
Costos varios y de Oficina	981,75
<b>TOTAL</b>	<b>1659.24</b>

La Tabla 21 muestra que el costo invertido para el desarrollo del sistema es de 1659,24 USD, cabe recalcar que este valor elevado se lo considera tomando en cuenta los costos de Ingeniería ya que al ser un prototipo electrónico suma el tiempo de investigación, diseño y construcción del proyecto dando un valor estimado de 370.99 USD, además resulta conveniente resaltar que el precio neto del sistema es de 58.50 USD lo cual puede llevarse a un ámbito comercial tomando en cuenta otros dispositivos que se encuentran en el mercado actual.

### **3.7. Diseño del Sistema**

Una vez que se ha definido los elementos y requerimientos más importantes para el desarrollo del sistema, la siguiente acción que se presenta es la de delimitar los lineamientos del diseño del producto, en ese contexto se presenta los criterios que se obtuvieron, como eje fundamental para el desarrollo, funcionamiento e implementación del sistema de detección y rastreo de mascotas.

#### **3.7.1. Descripción general del funcionamiento del sistema *PETROUTE***

El funcionamiento que presenta este sistema se basa en la detección y rastreo de mascotas de la ciudad de Ibarra a través de una red IoT.

La operatividad de PETROUTE consta de 4 fases que se presenta a continuación:

La primera fase consiste es la fase donde el usuario que en este caso es el dueño de la mascota accede a ejecutar la aplicación móvil sea por A o B situación en la cual su mascota no se encuentre a su lado, seguidamente aparecerá el mapa y un marcador de color rojo especificando la ubicación en tiempo real de su mascota.

En segunda fase se empieza a adquirir los datos de coordenadas tanto de latitud como de longitud para establecer una ubicación optima lo más precisamente posible a la ubicación de la mascota, esto se logra mediante el módulo GPS el cual recoge datos de los satélites que se encuentran en órbita y los transmite al módulo de comunicación que se encuentra en el circuito del sistema embebido desarrollado. Por otra parte, para que los datos lleguen hacia el dueño de la mascota se debe tener una comunicación inalámbrica por lo tanto el módulo GPRS juega un papel fundamental y se consigue transmitir los datos obtenidos por el módulo GPS hacia el dueño mediante el uso de comandos AT.

La tercera fase consiste en la transmisión de datos desde el módulo GPRS hacia la nube esto se lo realiza gracias a la ayuda de la configuración del APN y de comandos AT que se encuentran codificados dentro de la placa de desarrollo. Una vez que se tenga listo todos los datos en este caso se hace referencia a los parámetros de latitud y longitud, los cuales son datos obtenidos en la segunda fase, entonces se realiza la conexión a la nube y se empiezan a subir los datos.

La cuarta y última fase consiste en la alerta mediante un mensaje de la aplicación móvil del Smartphone del dueño cuando este ha establecido un rango de perímetro y la mascota haya

salido de dicho rango, de igual forma esta etapa hace referencia a la interacción entre el usuario y la aplicación móvil donde el mismo podrá visualizar a detalle la imagen del mapa con la información de ubicación de la mascota en tiempo real, estos parámetros se muestran en función de longitud y latitud plasmados en el mapa de ubicación y establecidos por el prototipo electrónico.

### 3.7.2. Arquitectura

En este apartado se presenta la Arquitectura del Proyecto tomando en cuenta su funcionamiento en una Arquitectura IoT partiendo desde la etapa de obtención de datos hasta terminar en la etapa de Visualización por parte del usuario. En la figura 8 se presenta la Arquitectura IoT considerada para el desarrollo del presente Proyecto.

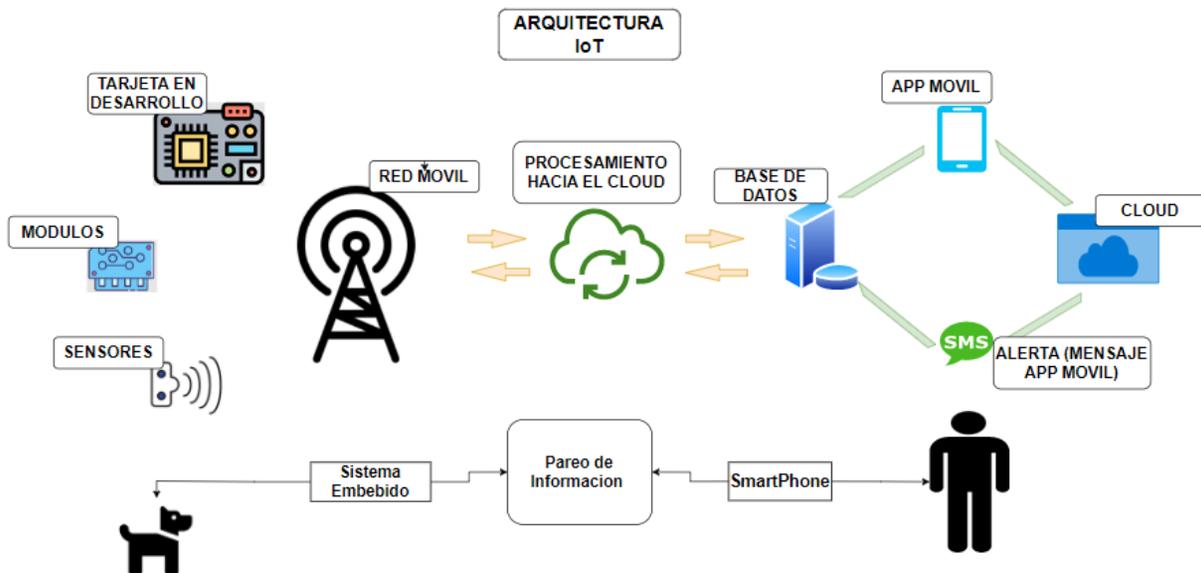


Figura 8. Arquitectura del proyecto.

Para entender la arquitectura del proyecto se presenta el funcionamiento del sistema a través de un diagrama de bloques como se muestra en la figura 8 donde constan los bloques por etapas del sistema PETROUTE.

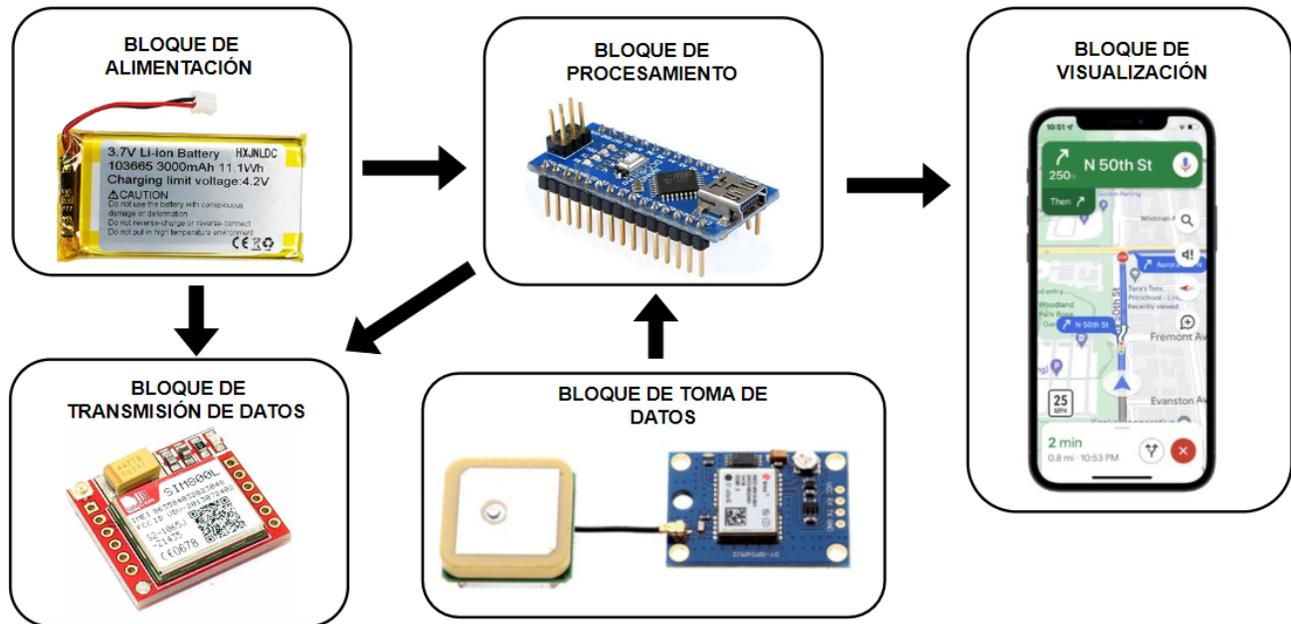


Figura 9. Diagrama de Bloques de PETROUTE.

La arquitectura del sistema PETROUTE consta de 5 bloques los cuales cumplen con funciones propias por cada bloque, los cuales se interpretan a continuación.

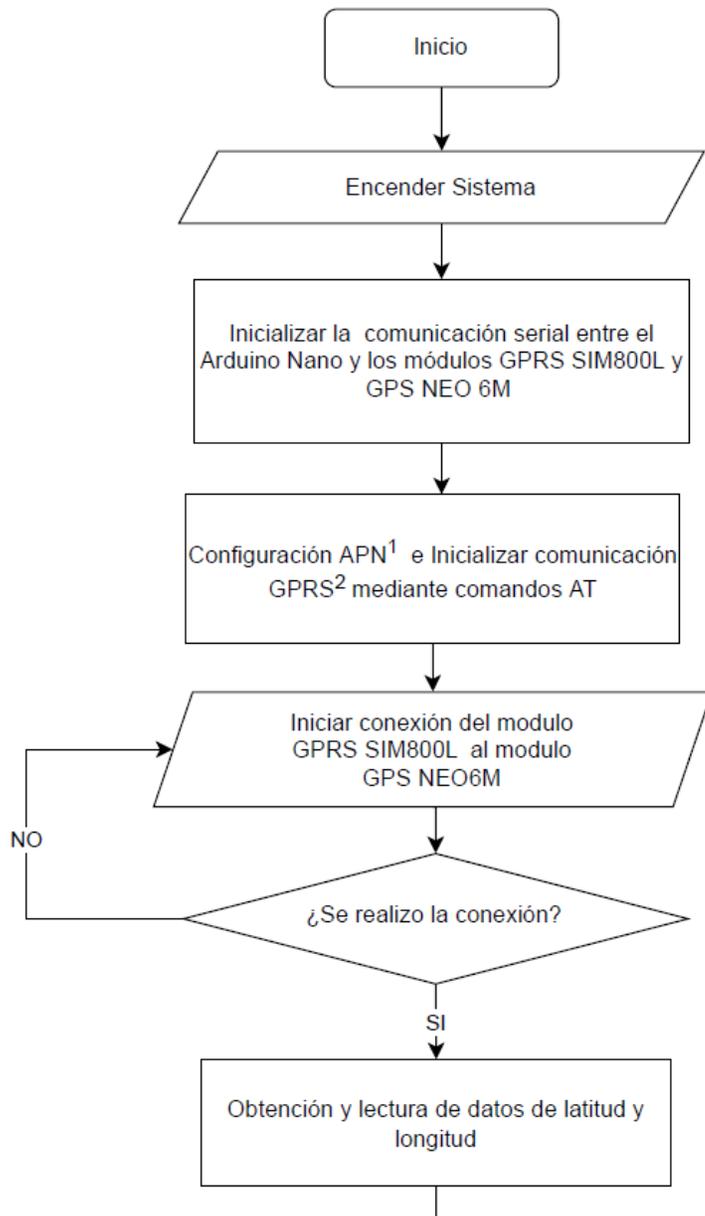
- **Bloque de alimentación:** Este bloque representa la alimentación del circuito el cual necesita un voltaje para poder funcionar. La alimentación se basa en suministrar un voltaje y corriente a los módulos que se encuentran conectados a la batería, en este caso se usa una batería Lipo de 3.7v /3000mah recargable para alimentar todo el circuito.
- **Bloque de toma de datos:** Es importante la obtención de datos de ubicación, por ello con la ayuda del módulo GPS se logra obtener los datos de ubicación que transmiten los satélites al módulo para su correspondiente análisis.

- **Bloque de Transmisión de datos:** En este bloque los datos de ubicación obtenidos que se encuentran ya analizados en el microcontrolador, son transmitidos hacia el módulo de comunicación inalámbrica GPRS mediante comandos AT para luego ser subidos a la nube.
- **Bloque de Procesamiento:** Este bloque es el pilar fundamental del programa ya que es donde se procesa la información y se programa las distintas acciones para la comunicación entre módulos del prototipo.
- **Bloque de Visualización:** Finalmente se tiene el bloque donde el usuario en este caso el dueño de la mascota, mediante el uso de la aplicación móvil puede detectar y rastrear a su mascota desde un mapa para su mejor visualización, esto se lo logra mediante el software de Flutter y los servicios de Firebase.

#### ***3.7.2.1. Diagrama de Flujo del proceso que realiza el Sistema***

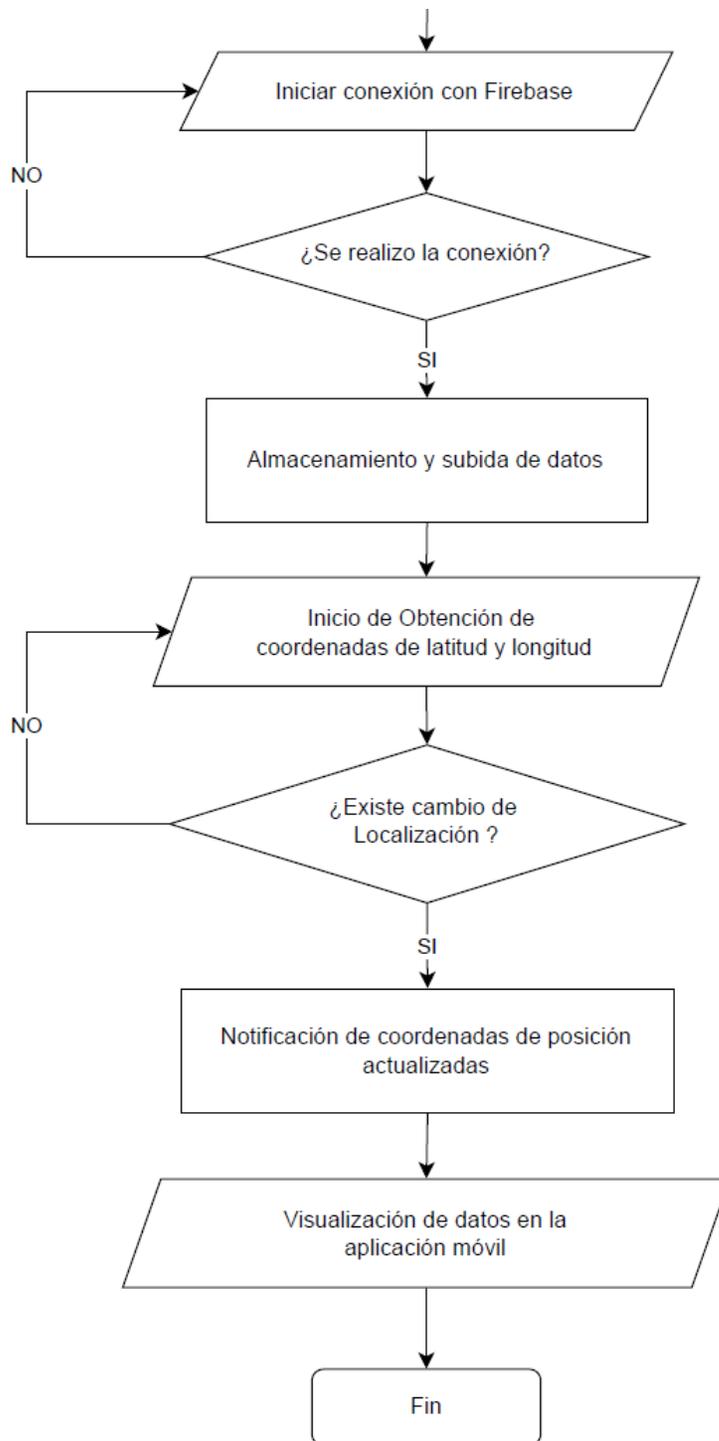
El proceso que ira ejecutando paso a paso el sistema para la obtención de datos que se obtiene a través de los módulos, se lo representa mediante un diagrama de flujo, el cual va describiendo paso a paso el proceso que realiza el sistema para detectar y rastrear mediante el prototipo embebido a la mascota y ser visualizado por el usuario final.

Posteriormente, en la figura 10 se puede observar el diagrama de flujo



<sup>1</sup> Access Point Name, se trata de la configuración que especifica por qué camino virtual se debe conectarse a Internet.

<sup>2</sup> General Packet Radio Service y corresponde el método de transmisión de datos en las redes de telefonía móvil 2G.



**Figura 10.** Diagrama de Flujo del proceso de PETROUTE.

### **3.8. Diseño a detalle del Sistema PETROUTE**

En el presente apartado se detalla el diseño para la construcción de cada uno de los bloques del sistema PETROUTE tomando en cuenta los requerimientos que se expusieron anteriormente en las tablas 5 y 6 respectivamente.

#### ***3.8.1. Diseño del bloque de Alimentación***

##### **3.8.1.1. Cálculos de la fuente de Alimentación**

En este apartado se especifican los cálculos de la fuente de alimentación que se usa para su correcto funcionamiento, tomando a consideración los datos del consumo de energía que vienen definidos por el datasheets de cada módulo que integra el sistema.

El consumo de los elementos a utilizar en el sistema son los siguientes:

- Arduino Nano =19mA
- Modulo GPS NEO 6m V2 =45mA
- Modulo SIM800L =350mA

El consumo total de energía se calcula mediante la Ecuación 1.

$$\text{Consumo Total} = 19\text{mA} + 45\text{mA} + 350\text{mA}$$

$$\text{Consumo Total} = 414\text{mA}$$

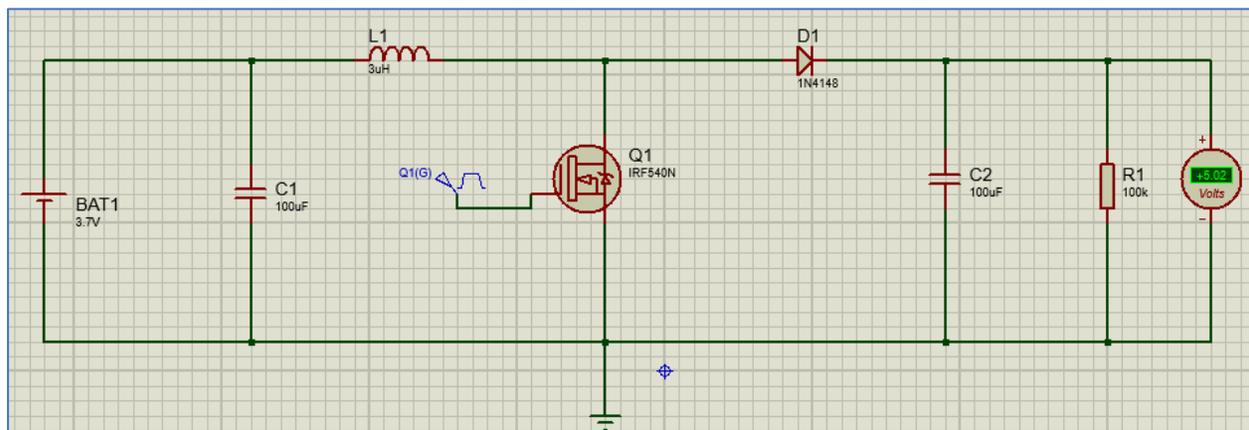
**Ecuación. 1.** Cálculo del consumo de los elementos del Sistema

De esta forma se puede apreciar de la Ecuación 1 el consumo total de energía del sistema es de 414mA para su funcionamiento, por lo cual se necesita una fuente de 5v a 1A.

- ***Regulador de Voltaje***

Sabiendo que receptor GPS Neo 6Mv2, el módulo GSM SIM800L y la placa Arduino Nano tienen un voltaje de operación de 3.3V, 4V y 5V respectivamente fue necesario implementar un regulador de voltaje para que la alimentación del dispositivo pueda abastecer la demanda de los módulos.

Regulador para placa Arduino Nano. Se implementó un regulador de 5V haciendo uso del regulador de tensión MT3608. A continuación, se presenta como está compuesto este regulador utilizando la fuente de alimentación de 3.7v del sistema como se muestra en la Figura 11.



**Figura 11.** Regulador de Voltaje (Arduino Nano).

- ***Divisor de Voltaje***

Los divisores de voltaje son útiles para proporcionar diferentes niveles de voltaje a partir de un voltaje de suministro común. No se puede conectar los pines Rx del módulo directamente a los pines digitales del Arduino porque el Arduino Nano utiliza GPIO (Pin genérico de Entrada/Salida) de 5V y el módulo SIM800L utiliza lógica de nivel de 3,3V por lo cual no es resistente a los 5V, esto significa que la señal Tx del Arduino Nano tiene que bajar a 3,3V para

no dañar el módulo SIM800L. Por lo tanto, se utiliza un divisor de resistencias. El uso de una resistencia de 20K entre SIM800L Rx y Arduino y de una resistencia de 10K entre SIM800L Rx y GND. También se necesitará un componente que permita almacenar energía y proporcionarla muy rápidamente cuando el SIM800L la necesite, para ello se utiliza un capacitor electrolítico de 100µF 5 V en paralelo al pin VCC y GND del módulo SIM800L. Esto se lo realiza para evitar problemas que tengan que ver con el pico de consumo de energía al realizar operaciones de radio.

El divisor de voltaje se lo realiza mediante la Ecuación 2.

$$V_{R(x)} = V_S \left( \frac{R_x}{R_T} \right)$$

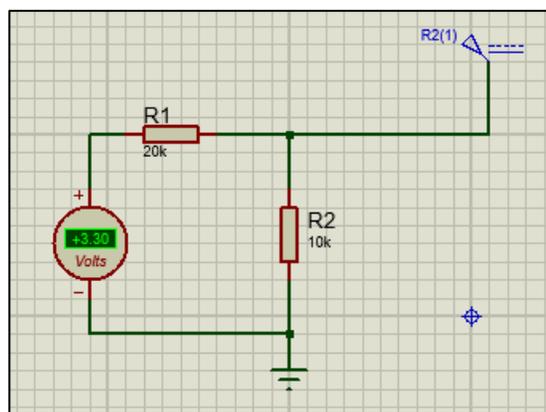
$$R_T = R_1 + R_2 = 20k\Omega + 10k\Omega = 30k\Omega$$

$$V_{R1} = V_S \left( \frac{R_1}{R_T} \right) = 5 \left( \frac{20k\Omega}{30k\Omega} \right) = 3.3v$$

$$V_{R1} = 3.3v$$

**Ecuación. 2.** Ecuación de Divisor de Voltaje (Arduino Nano – Sim800L).

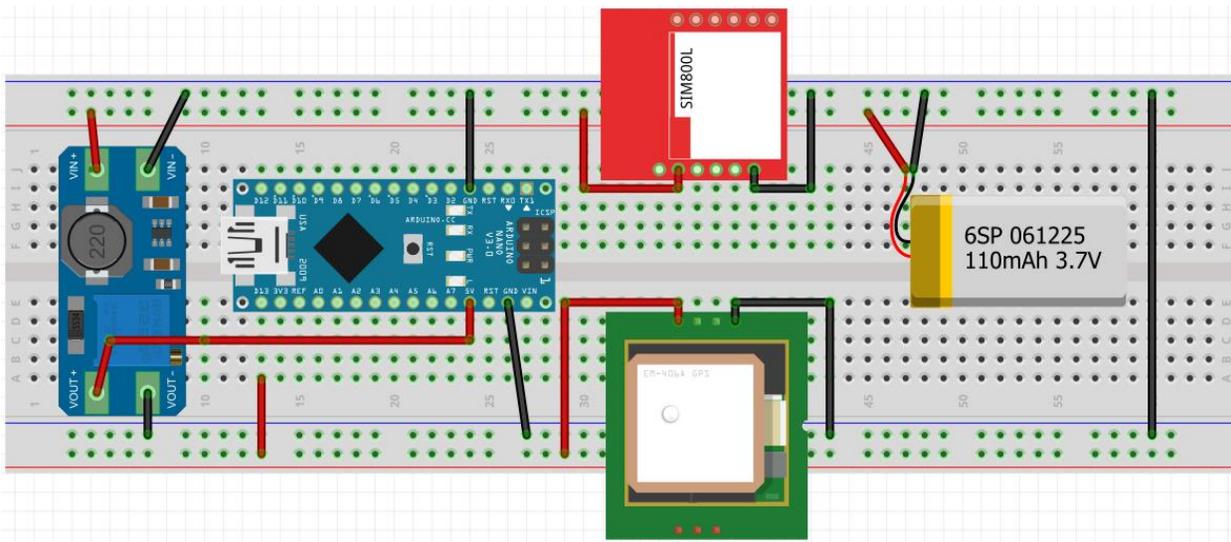
De esta forma se puede apreciar de la Ecuación 1 el consumo total de energía del sistema es de 414mA para su funcionamiento, por lo cual se necesita una fuente de 5v a 1A.



**Figura 12.** Divisor de Voltaje (Arduino Nano – Sim800L).

### 3.8.1.2. Diseño de la fuente de Alimentación

El diseño de alimentación se basa en los requerimientos que se establecieron anteriormente, donde se toma en cuenta que se necesita un voltaje de 3.7v/2A para el correcto funcionamiento del módulo GPRS SIM800L, la batería de Litio recargable es la encargada de suministrar estas características técnicas de energía y a su vez el módulo DC-DC Step Up MT3608 ayuda a la elevación de voltaje para el funcionamiento del Arduino Nano, para finalmente obtener la alimentación del módulo GPS NEO6M mediante el voltaje de 3.3v que suministra el Arduino Nano.

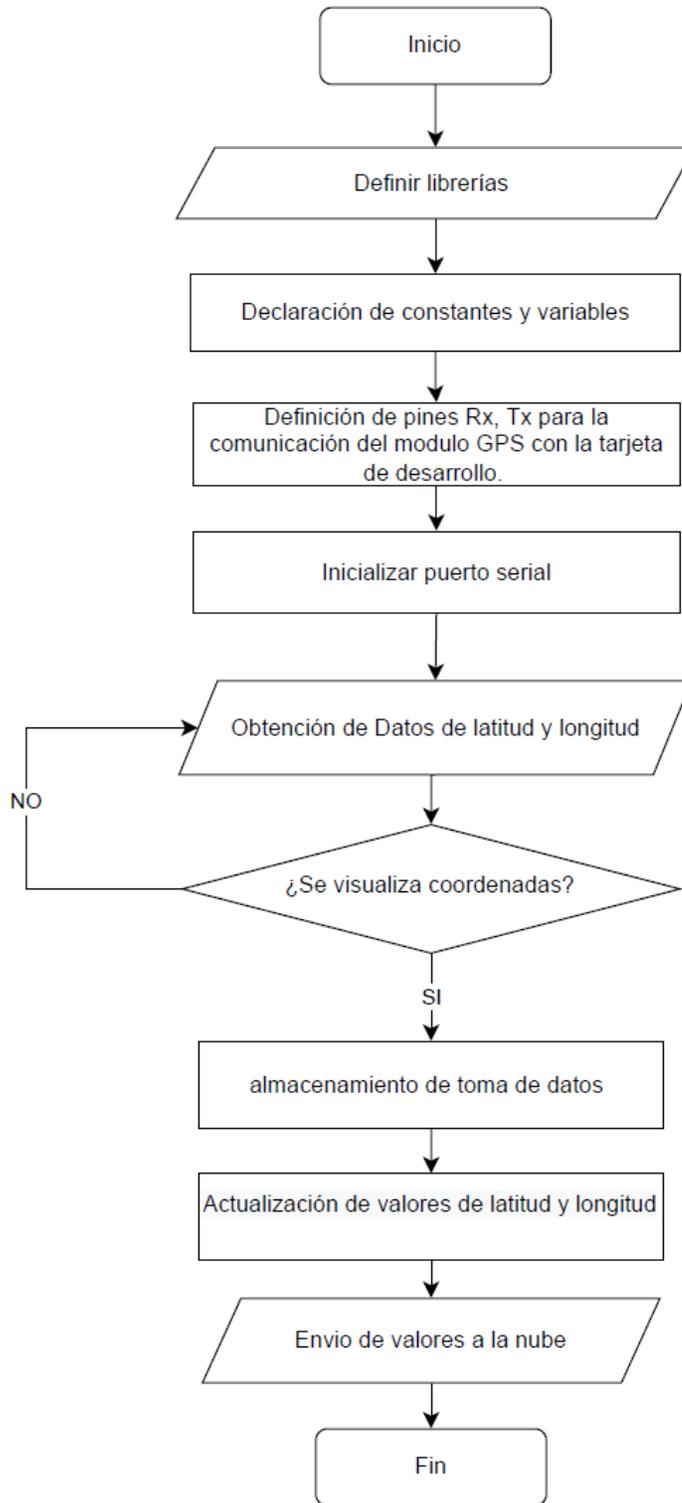


**Figura 13.** Diseño de conexión del circuito de alimentación del sistema PETROUTE.

### 3.8.2. Diseño del bloque de toma de Datos y bloque de Procesamiento

#### 3.8.2.1. Bloque de Toma de Datos

En este bloque se utiliza el módulo idóneo para poder recolectar coordenadas de latitud y longitud mediante la red GPS, en la cual se encuentra operando un total de 27 satélites y orbitando sobre la tierra. En este punto es donde actúa el módulo GPS Neo6M que gracias al seguimiento del protocolo NMEA(National Marine Electronics Association), que son sentencias estandarizadas para la recepción de datos GPS, se puede receptor dichos datos hacia la placa de desarrollo en este caso en referencia al Arduino Nano. En la Figura 14 se presenta el diagrama de flujo donde se analiza el proceso de toma de datos de geoposicionamiento, partiendo desde la inicialización de variables, hasta llegar a la obtención de coordenadas de ubicación.

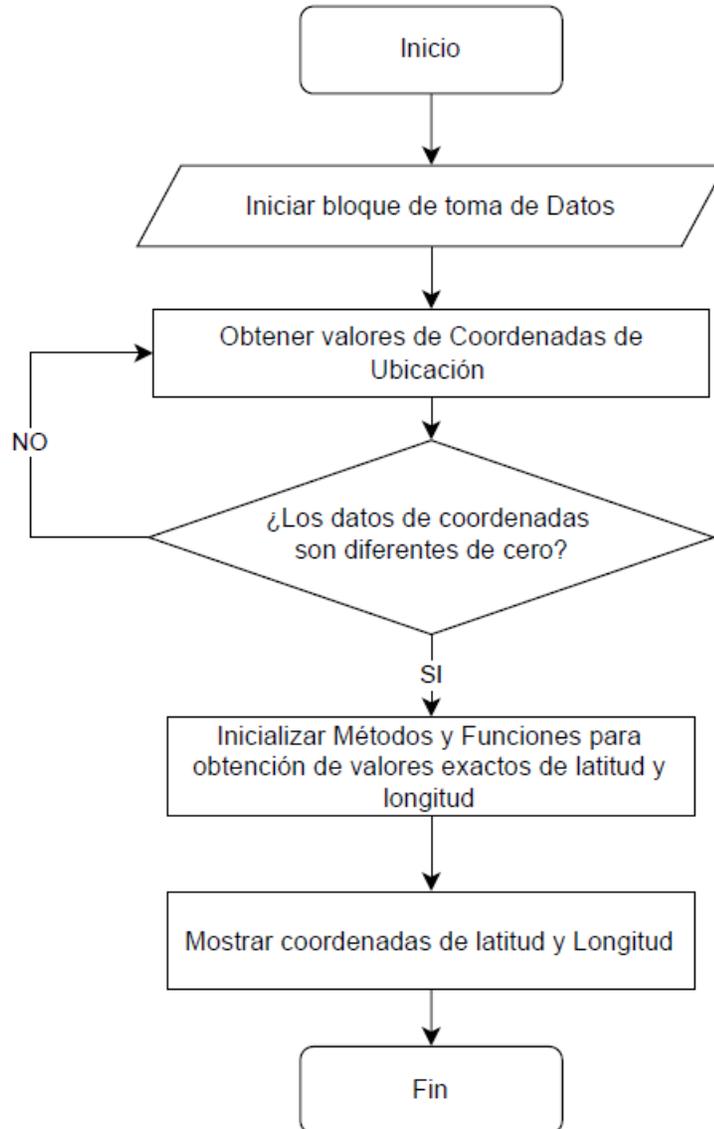


**Figura 14.** Diagrama de flujo del bloque de toma de Datos.

### **3.8.2.2. Bloque de Procesamiento**

Una vez que se obtiene los datos receptados por el módulo GPS se continua con el procesamiento de la información donde se analiza los valores de coordenadas de ubicación mediante la librería TinyGPS plus la cual se utiliza para analizar flujos de datos NMEA proporcionados por el módulo GPS, de esta forma se extraen datos arbitrarios de cualquiera de las innumerables sentencias NMEA que existe dando como resultado la posición básica. El funcionamiento que se describe en el presente proyecto con respecto a esta librería toma en cuenta los submódulos de la librería TinyGPS Plus donde se debe canalizar repetidamente los caracteres desde el módulo GPS utilizando el método `encode()`. Tomando en cuenta la conexión dentro del circuito que está conectado a los pines (RX) y (TX), la función que realiza la obtención de latitud se describe con `gps.location.lat()` y `gps.location.lng()` para obtener los valores de longitud.

El proceso que realiza la placa de desarrollo para analizar los valores de coordenadas se puede observar a través de la Figura 15.



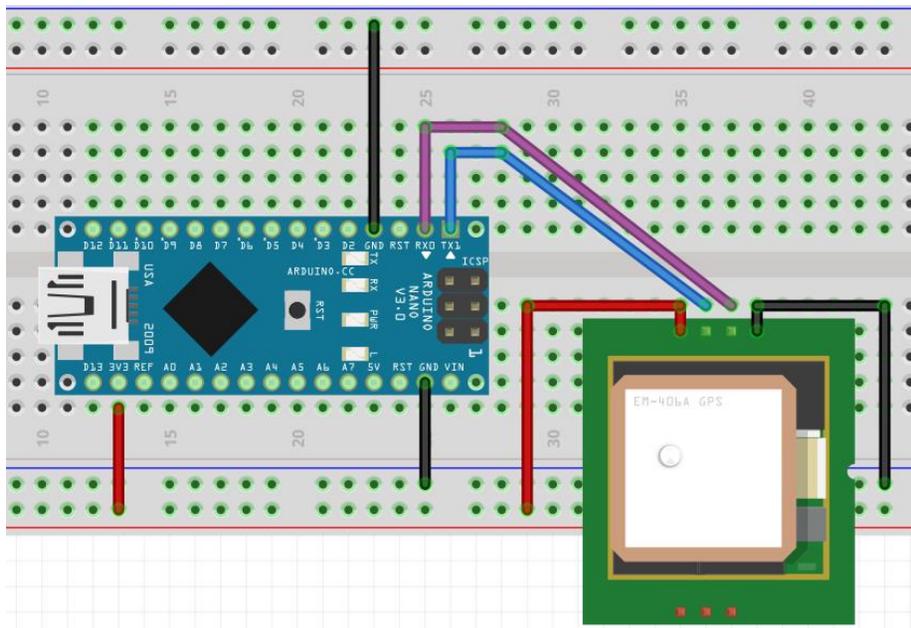
**Figura 15.** Diagrama de flujo del bloque de Procesamiento.

El diagrama de conexión del módulo GPS neo6Mv2 hacia la placa de desarrollo Arduino Nano se puede visualizar en la Figura 16, donde se presenta las conexiones de los pines Recepcion y Transmisión del módulo a los pines del Arduino Nano.

**Tabla 22.**

Pines de conexión del módulo GPS NEO6Mv2 con Arduino Nano.

GPS NEO 6Mv2	Arduino Nano
1: GND	Pin GND
2: TX	Pin RX
3: RX	Pin TX
4: VCC	Pin 3v3



**Figura 16.** Diseño de conexión del módulo GPS Neo6M.

### **3.8.3. Diseño del bloque de Transmisión de Datos**

Este bloque es muy fundamental porque es donde los datos de coordenadas de ubicación se transmiten hacia la red móvil para posteriormente ser subidos a la nube. El módulo que trata de ejecutar las acciones de transmisión de datos es el módulo SIM8001, el cual gracias a sus funciones de comunicación por comandos AT establece una conexión con la placa de desarrollo

para poder transmitir los datos de coordenadas de ubicación hacia la nube a través del servicio HTTPS que presente este módulo. El acceso a internet de este módulo se lo realiza a través de la configuración del APN donde se introduce el usuario y contraseña de la operadora móvil para facilitar el acceso a internet.

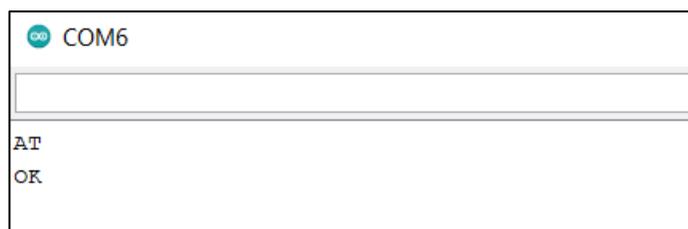
### 3.8.3.1. Uso de Comandos AT

El uso de comandos AT<sup>3</sup> (de attention) es muy importante para el funcionamiento de este módulo.

Lo primero a realizar será abrir Arduino IDE después de esto, se seleccionará el puerto COM, posteriormente a esto se deberá abrir el monitor serial e ingresar los siguientes comandos AT el código en su totalidad se lo puede observar en el anexo 2.

- **AT**

AT: Si el módulo Sim800l funciona correctamente, Sim800l responderá OK después de recibir este comando.



**Figura 17.** Comando AT para verificar el funcionamiento del módulo SIM800l.

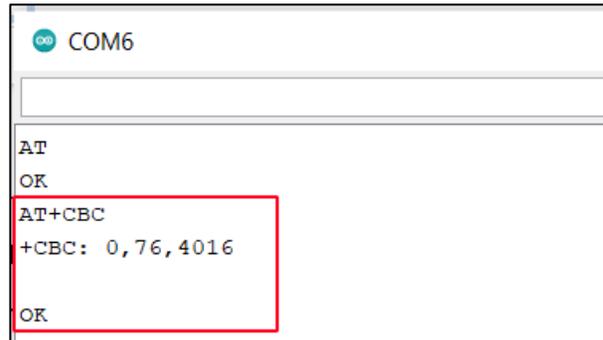
/

---

<sup>3</sup> Denominados así por la abreviatura de **attention**, su función principal es la comunicación con modems, la telefonía móvil GSM ha incorporado como estándar este lenguaje para poder comunicarse con sus terminales.

- **AT+CBC**

Este comando se utiliza para verificar el voltaje de la batería.

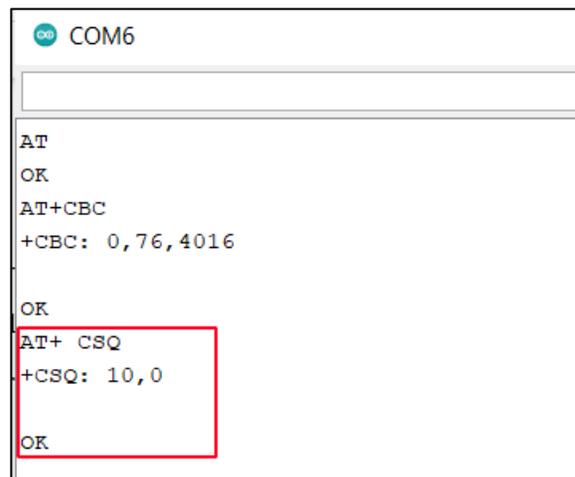


```
COM6
AT
OK
AT+CBC
+CBC: 0,76,4016
OK
```

**Figura 18.** Comando AT para verificar estado de batería.

- **AT+CSQ**

El módulo Sim800l indica la intensidad de la señal después de recibir este comando.



```
COM6
AT
OK
AT+CBC
+CBC: 0,76,4016
OK
AT+CSQ
+CSQ: 10,0
OK
```

**Figura 19.** Comando AT para identificar la intensidad de la señal de red del módulo SIM800l.

- **AT+CREG**

Se puede usar este comando AT para registrar la red si el módulo SIM800l pierde la red abruptamente y tarda mucho tiempo en escanear la red.

```
COM6
AT+CBC
+CBC: 0,76,4016
OK
AT+ CSQ
+CSQ: 10,0
OK
AT+CREG
ERROR
```

**Figura 20.** Comando AT para registrar la red del módulo SIM8001.

En este caso se puede observar que el módulo ya se encuentra registrado a una red por lo cual el módulo responde con un mensaje de error. A continuación, en el siguiente comando se presenta sobre el estado de red y se constata que el módulo ya se encuentra registrado a la red.

- **¿AT+COPS?**

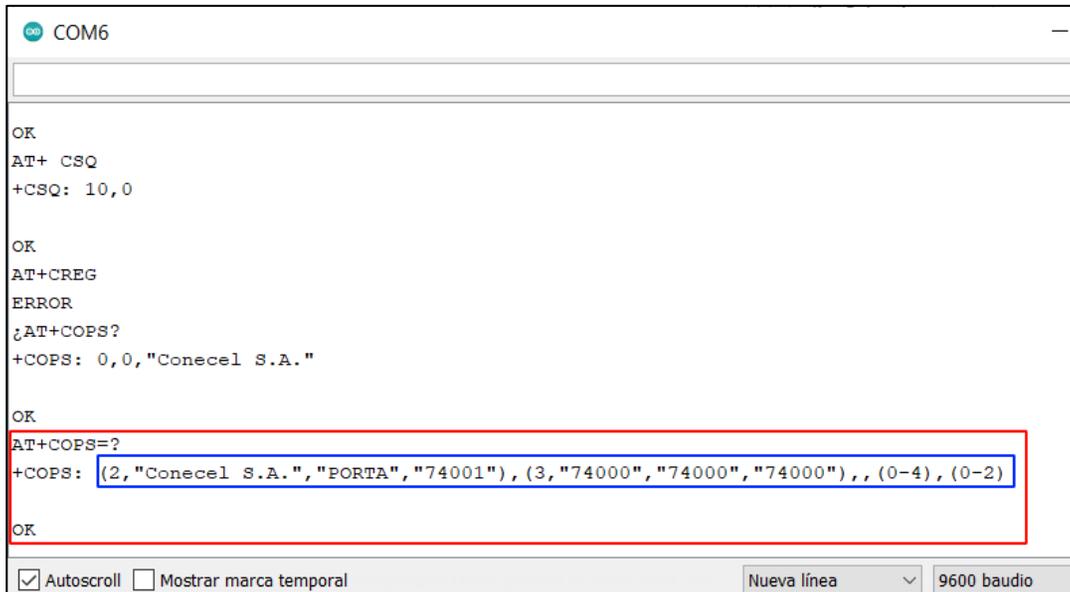
Si se coloca este comando, el módulo SIM8001 le informará sobre el estado de su red.

```
COM6
+CBC: 0,76,4016
OK
AT+ CSQ
+CSQ: 10,0
OK
AT+CREG
ERROR
¿AT+COPS?
+COPS: 0,0,"Concecel S.A."
OK
```

**Figura 21.** Comando AT para verificar el estado de Red del módulo SIM8001.

- **AT+COPS=?**

El módulo SIM800l muestra las redes disponibles después de recibir este comando AT.

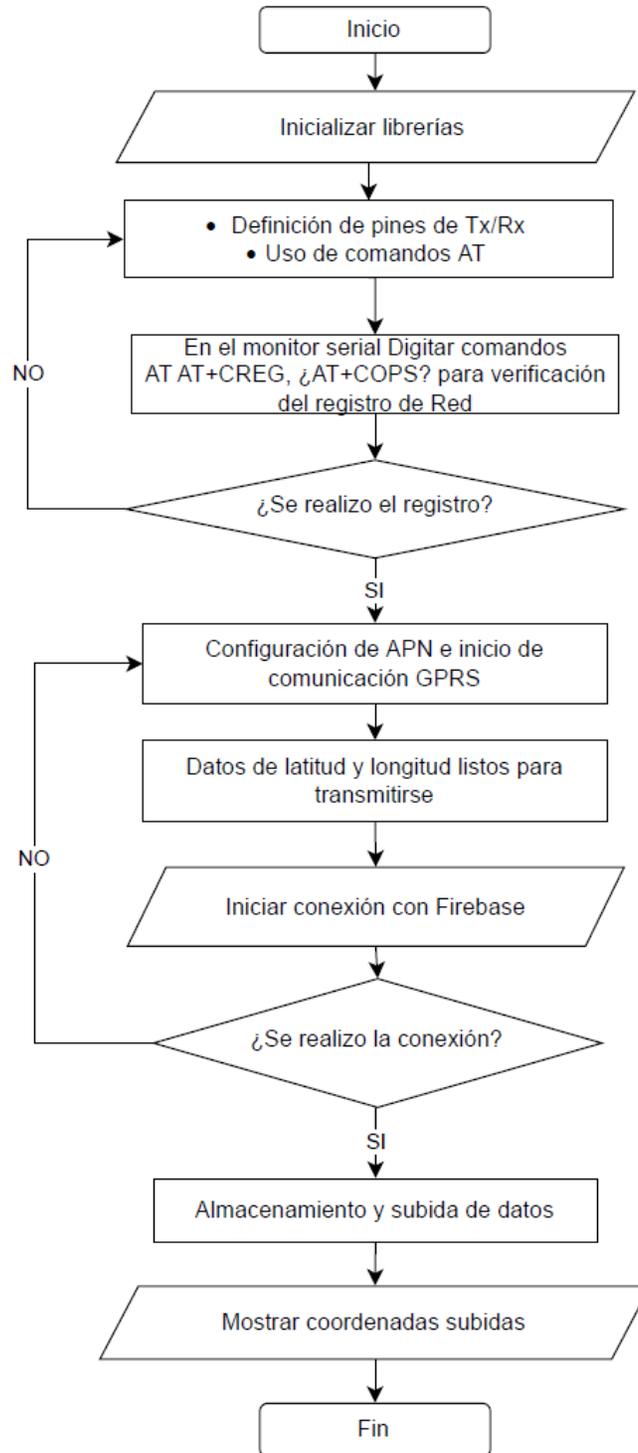


```
COM6
OK
AT+ CSQ
+CSQ: 10,0
OK
AT+CREG
ERROR
¿AT+COPS?
+COPS: 0,0,"Concecel S.A."
OK
AT+COPS=?
+COPS: (2,"Concecel S.A.,"PORTA","74001"),(3,"74000","74000","74000"),,(0-4),(0-2)
OK
```

Autoscroll      Mostrar marca temporal    Nueva línea    9600 baudio

**Figura 22.** Comando para verificar redes disponibles.

Una vez que se haya probado el funcionamiento de registro de red del módulo se realiza la configuración APN y se realiza el inicio de comunicación GPRS, en este punto los datos obtenidos por el módulo GSM se encuentran listos para transmitirse por la red inalámbrica y ser subidos al Cloud por lo cual se usa el servicio HTTPS del módulo SIM800l mediante el uso comandos AT. De esta forma se establece la conexión al Cloud y se suben los datos de latitud y longitud a la nube, esto se puede visualizar en el diagrama de flujo de la Figura 23.



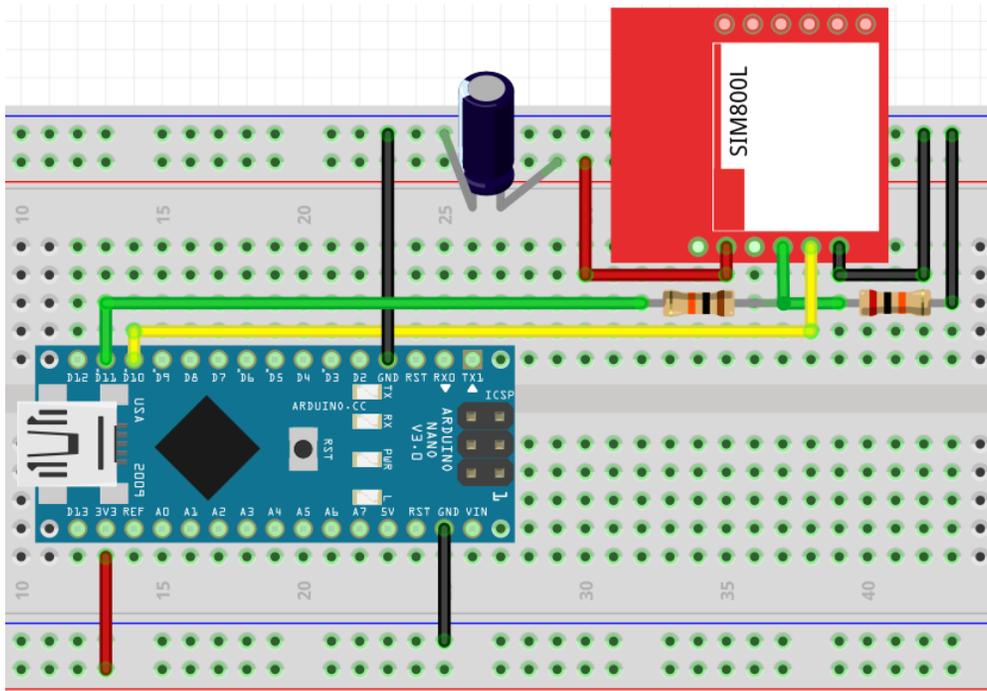
**Figura 23.** Diagrama de flujo del bloque de transmisión de Datos.

La interconexión del módulo se presenta en la Figura 24, en donde se aprecia el módulo SIM 800L conectado al Arduino Nano, las acciones que ejecuta el módulo SIM800L se encuentra definidas de acuerdo con sus pines de conexión.

**Tabla 23.**

Pines de conexión del módulo SIM800L con Arduino Nano.

SIM800L	Arduino Nano	Batería
2: VCC		Pin VCC
4: RXD	Pin 11	
5: TXD	Pin 10	
6: GND	GND	Pin GND



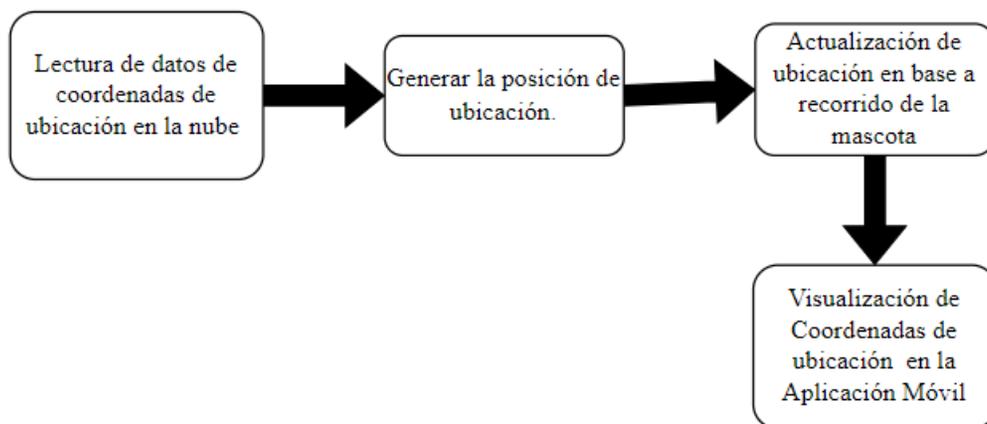
**Figura 24.** Diseño de conexión del módulo SIM 800L.

Se debe tomar en cuenta las siguientes especificaciones con respecto al PinOut del módulo:

- **DTR:** Este pin se refiere como el pin de habilitación. Este pin juega un papel importante en el ahorro de energía. Si se hace que este pin sea alto, el módulo entrará en modo de reposo y deshabilitará la comunicación serial, si lo hace en estado bajo, el módulo se encenderá.
- **NET:** Aquí, se puede conectar la antena en espiral.
- **VCC y GND:** Se puede utilizar estos pines para alimentar el SIM8001, pero el voltaje no debe superar los 3,3V y la corriente no debe ser inferior a 1A.
- **RX and TX:** Estos pines se utilizan para la comunicación en serie, Rx para recibir comandos de la placa de desarrollo y Tx para enviar datos.

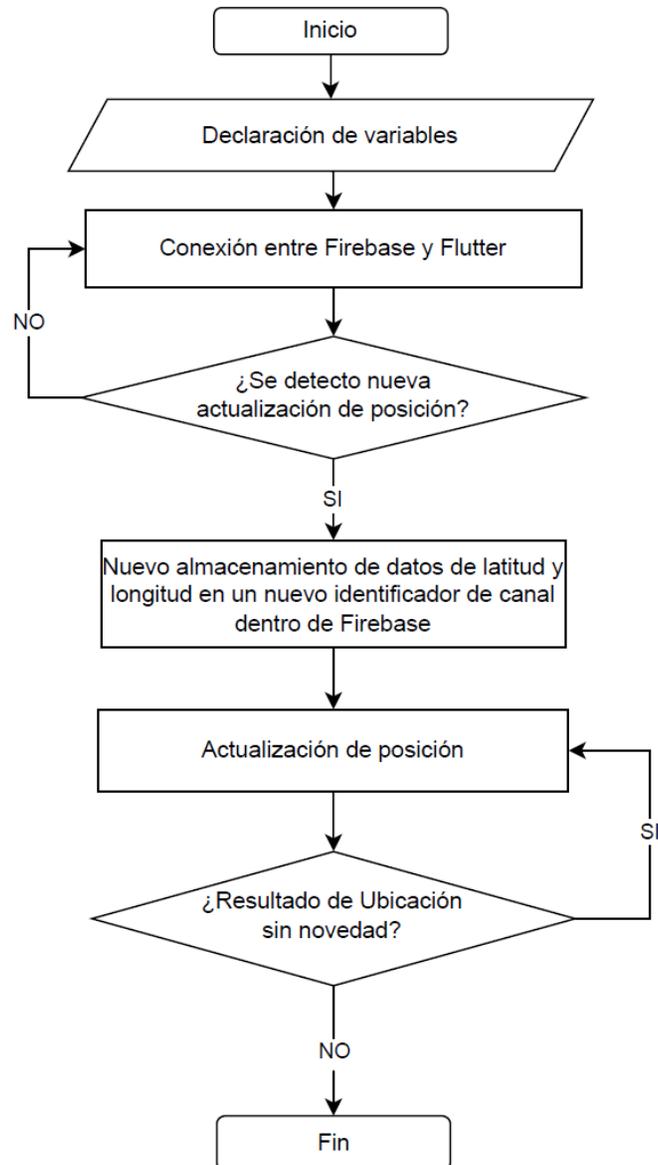
#### 3.8.4. Diseño del bloque de Visualización

El desarrollo de este bloque se lleva a cabo gracias a la ayuda del software Flutter ya que fue seleccionado en el apartado 3.5.2.3. El desarrollo del funcionamiento del aplicativo grafico se lo lleva a cabo de acuerdo con el diagrama de bloques de la Figura 25.



**Figura 25.** Diagrama de bloques de la etapa de visualización.

- **Lectura de datos:** Los datos se obtienen a través del bloque de procesamiento, son transmitidos a través del módulo SIM800l debido a que las coordenadas de ubicación se logran obtener en tiempo real.
- **Genera posición de Ubicación:** La posición de ubicación se genera a través de los datos que emite el módulo GPS NEO6m.
- **Almacenamiento de Datos:** Una vez que los datos de ubicación se genera esta información se va actualizando en tiempo real dentro de la base de datos del Firebase.
- **Visualización:** La visualización de ubicación se lleva a cabo dentro de la aplicación móvil donde aparece un indicativo haciendo referencia a la posición en tiempo real de la mascota que lleva colocado el PETROUTE, esto se logra gracias a la API de Google Maps y el software Flutter.



**Figura 26.** Diagrama de flujo de la etapa de visualización.

### 3.9. Implementación

En este apartado se realiza la implementación del sistema PETROUTE basándose en dos partes, que son la implementación de hardware y software.

### 3.9.1. Implementación de Hardware

#### 3.9.1.1. Diseño de Placa Electrónica

El diseño de la placa Electrónica se lo realizo en el Software de Eagle en su versión 7.6.0 donde una vez realizada la conexión de los elementos se procede a realizar el ruteo de pistas tomando en cuenta como factor clave que no existan cortocircuitos ni tampoco puentes ya que el sistema debe de ser lo más reducido posible. A continuación, en la Figura 27 se puede observar cómo se realiza la conexión de los elementos en el software de Eagle.

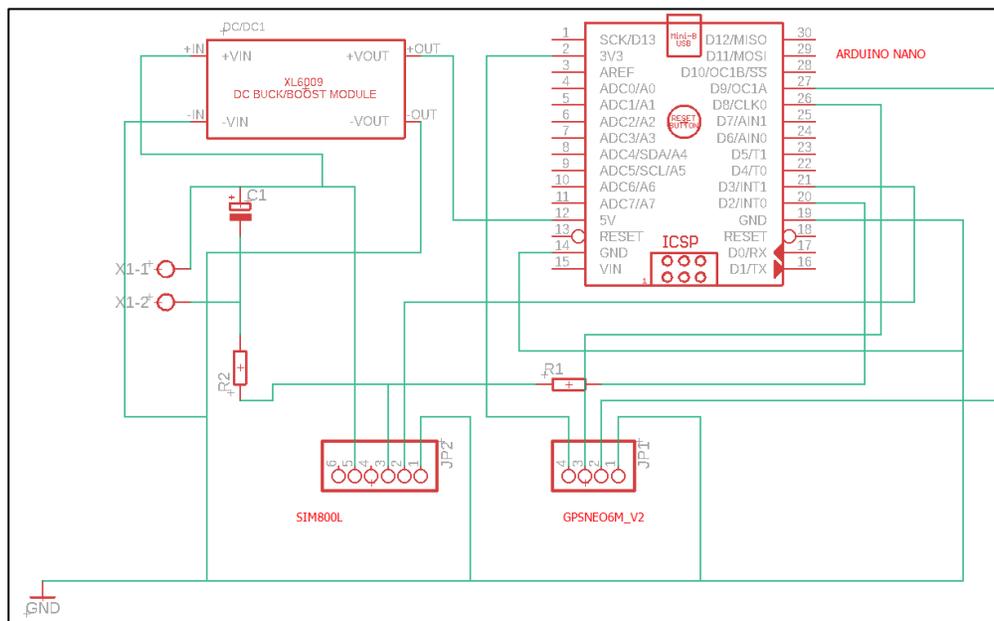
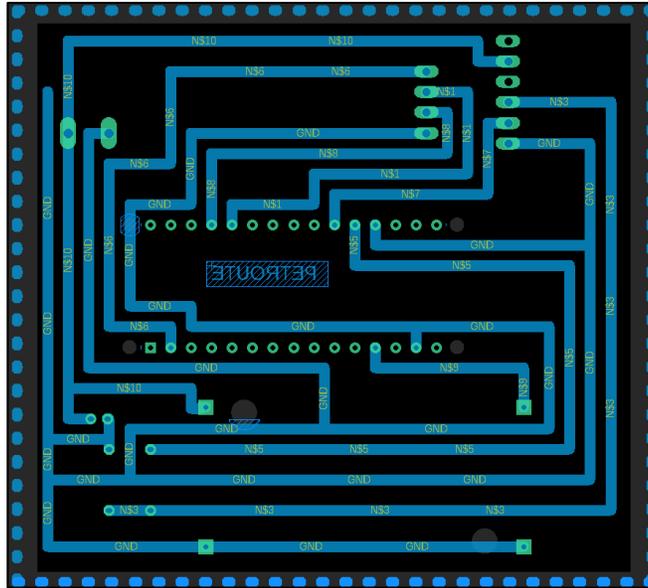


Figura 27. Diagrama esquemático de la conexión del circuito.

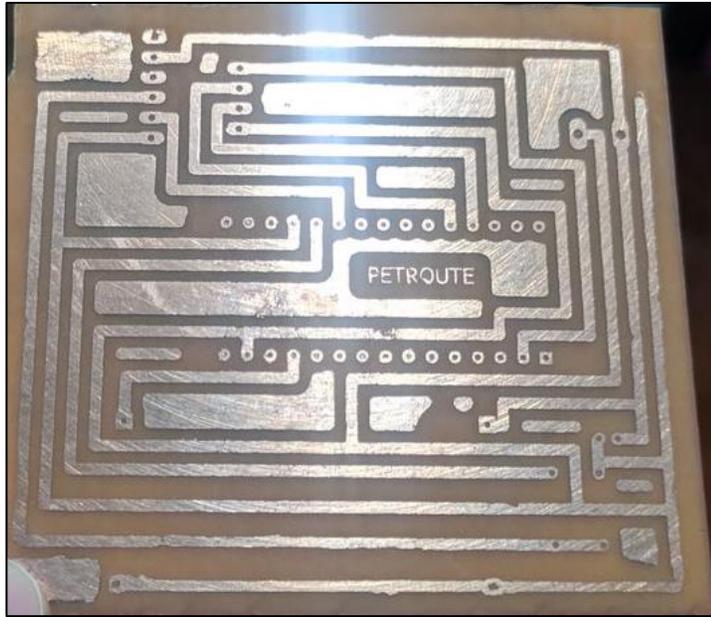
Una vez que se realiza la conexión de los elementos se pasa a realizar el ruteo de pistas tomando en cuenta que las pistas no deben de ser muy extensas para que, al momento de la fabricación de la placa, esta no quede muy grande. En la Figura 28 se puede Observar cómo se realizó el diseño del ruteo de pistas previo a la fabricación de la placa Electrónica.



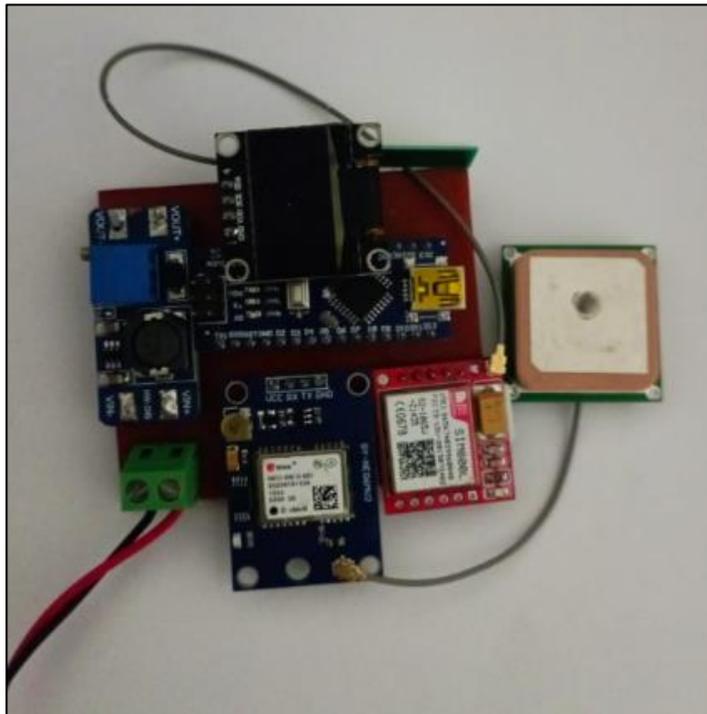
**Figura 28.** Diseño de pistas y conexión del circuito.

### 3.9.1.2. Construcción de Placa Electrónica

Para la fabricación de la placa electrónica, se lo realizo de forma tradicional mediante el método de Planchado el cual es muy útil y de bajo costo. Este método consistió en la transferencia térmica del toner de una impresión del circuito en papel de foto a una placa de cobre(baquelita), después se lo paso a ácido férrico tomando en cuenta que el ácido férrico debe de cubrir en su totalidad a la baquelita, una vez que el cobre haya salido en su totalidad, se procedió a sacar la baquelita del depósito para lavarlo con agua y luego dejarlo secar. Después se procede a perforar todos los orificios con ayuda de un mini taladro y por último se procede a soldar mediante estaño todos los elementos del circuito. En la Figura 29 se puede observar cómo se encuentra impresa la placa electrónica y en la figura 30 compuesta por todos sus elementos visualizando desde una vista frontal.



**Figura 29.** PCB (Placa de Circuito Impreso) listo para perforarse y colocar elementos correspondientes.



**Figura 30.** PCB (Placa de Circuito Impreso) con sus componentes soldados.

### 3.9.1.3. Estructura Física de la Pechera

La estructura física donde estará contenida la placa de circuito impreso será en una pechera de componentes de tela gruesa fabricado a mano bajo las medidas adecuadas, ya que se tomó en cuenta el espacio necesario para los elementos de la placa electrónica, sus dimensiones son de 11cm de largo, 9cm de ancho y 3cm de alto, a continuación en la Figura 31 se puede observar una vista lateral de la pechera y en la Figura 32 una vista frontal de la pechera ya colocada en la mascota.



**Figura 31.** Estructura de la Pechera (Vista Lateral).



**Figura 32.** Pechera colocada en la mascota (Vista Frontal).

### 3.9.2. Implementación de Software

En esta etapa se hace referencia a la programación del sistema, tomando en cuenta los bloques de desarrollo antes mencionados tales como el bloque de toma de datos, procesamiento, transmisión y aplicación móvil.

#### 3.9.2.1. Bloque de Toma de Datos y Procesamiento

En esta sección lo que se realiza es la programación en referencia al módulo GPS NEO 6m para poder obtener los datos que después servirán para establecer un geoposicionamiento en tiempo real para lo cual lo primero a realizar será establecer una comunicación entre el módulo y el Arduino. En la Figura 33 se visualiza parte de la programación que se utiliza, el código en su totalidad se lo presenta en el Anexo 3.

```
#include <TinyGPS++.h>
TinyGPSPlus gps;
double latitud, longitud;

#include <SoftwareSerial.h>
SoftwareSerial SIM800L(10,11);

String response;
int lastStringLength = response.length();

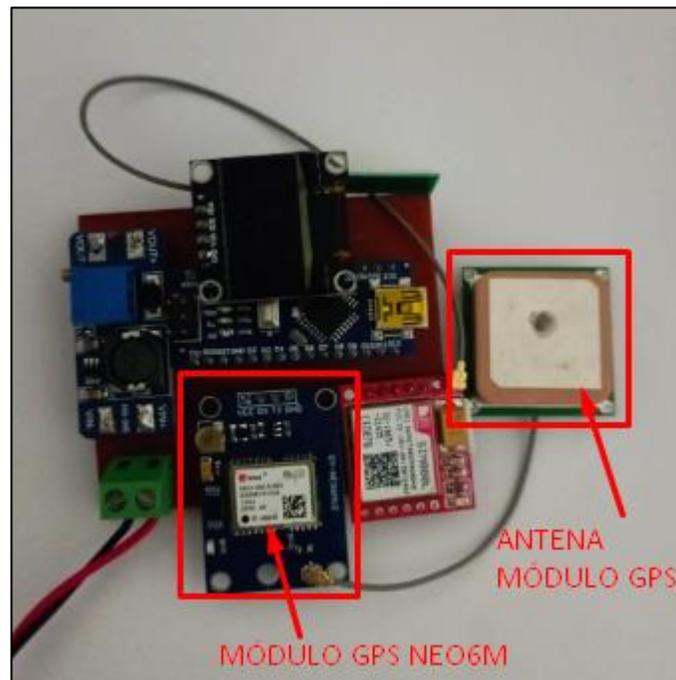
String link;

void setup() {
  Serial.begin(9600);
  Serial.println("Encontrar a mi Mascota");
}

void GPS(){
  if(Serial.available()) {
    gps.encode(Serial.read());
  }
  if(gps.location.isUpdated()) {
    latitud = gps.location.lat();
    longitud = gps.location.lng();
    link = "www.google.com/maps/place/" + String(latitud, 6) + "," + String(longitud, 6) ;
    Serial.println(link);
  }
}
```

**Figura 33.** Código de toma de Datos del sistema PETROUTE.

En la Figura 34 se puede observar como el módulo se conecta al Arduino Nano y tarda un tiempo en encender ya que, en este momento se encuentra estableciendo conexión con los satelites para receptor los datos de coordenadas por lo tanto el led indicador del módulo no se prendera.



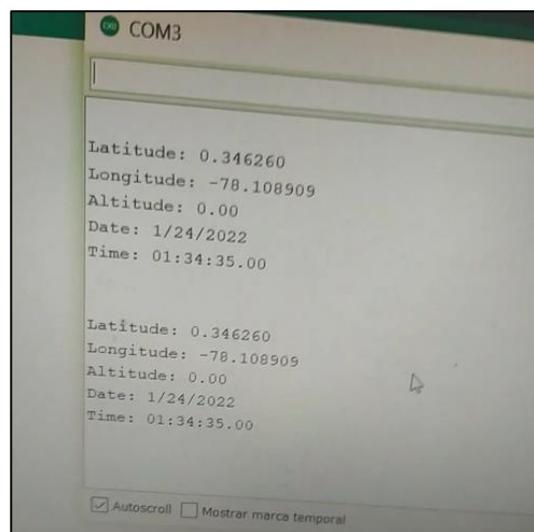
**Figura 34.** Módulo GPS NEO6m en estado inactivo, vista superior.

En la Figura 35 se puede observar que una vez que establece conexión con los satelites comienza a operar, por lo tanto el led indicador se prende y el modulo comienza a adquirir los datos de coordenadas de latitud y longitud.



**Figura 35.** Módulo GPS NEO6m en estado activo, vista superior.

En la Figura 36 se muestra el terminal del IDE de Arduino, donde por medio de la librería TinyGPS se logra analizar flujos de datos NMEA proporcionados por el módulo GPS, de esta forma se extraen datos arbitrarios de cualquiera de las innumerables sentencias NMEA que existe dando como resultado la posición básica, la altitud, la hora y la fecha.



**Figura 36.** Obtención de Datos de coordenadas de ubicación, vista superior.

### 3.9.2.2. Geocerca

Geocerca es un límite geográfico virtual, o una valla, definida por la tecnología GPS.

Permite que el software active una alerta cuando un objeto entra o sale de un área en particular.

En el presente proyecto, se usa la técnica de geocerca circular. Esta técnica trabaja sobre la posición inicial (punto de partida) y la distancia máxima (radio).

La geocerca resultante se ve como una forma circular. El proyecto calcula continuamente la distancia actual desde la posición inicial del objeto. y luego verifica si la distancia actual se excede de la distancia máxima. y si ese es el caso, entonces se enviará un mensaje dentro del mismo aplicativo hacia el dueño de la mascota.

El método usado para la creación de la geocerca se basa en la fórmula de Haversine el cual calcula la distancia entre dos puntos generando una circunferencia a partir de sus latitudes y longitudes(Azdy & Darnis, 2020).

$$haversine\left(\frac{d}{r}\right) = haversin e(\phi_2 - \phi_1) + \cos(\phi_1) \cos(\phi_2) haversine(\lambda_2 - \lambda_1)$$

donde:

- $d$ = es la distancia entre los dos puntos a lo largo de la circunferencia.
- $r$ = es el radio de la esfera.
- $\phi_2 - \phi_1$  son la latitud del punto 2 y la latitud del punto 1.
- $\lambda_2 - \lambda_1$  son la longitud del punto 2 y la longitud del punto 1.

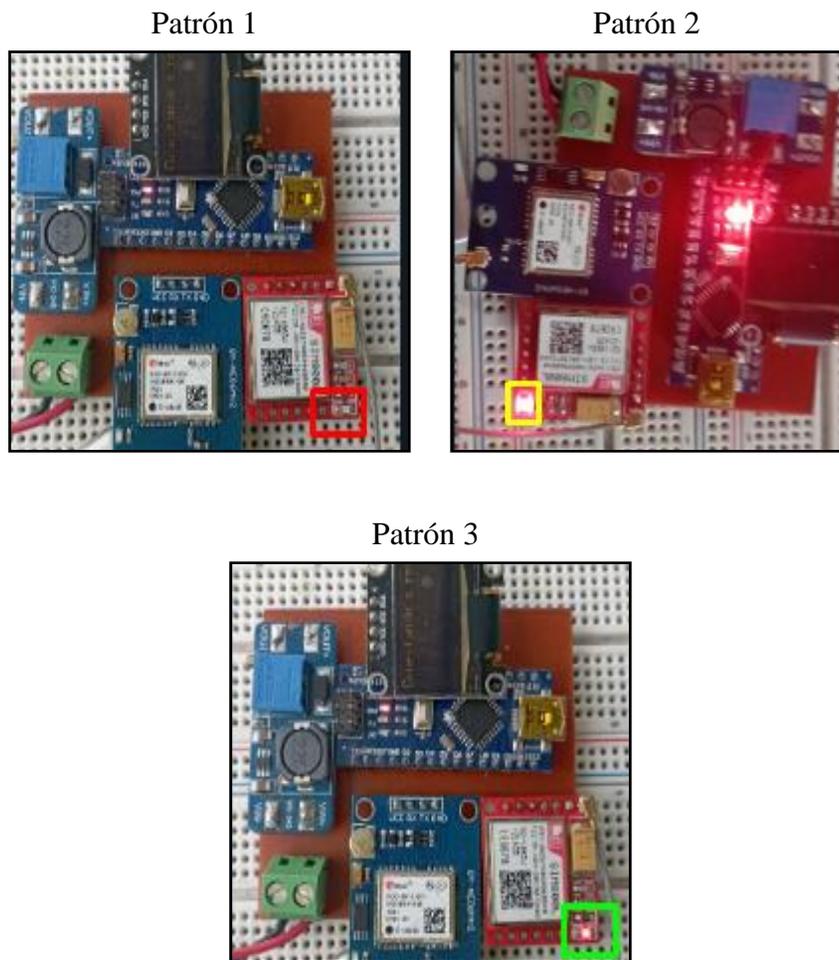
**Ecuación. 3.** Fórmula de Haversine

### 3.9.2.3. Bloque de Transmisión de Datos

Este procedimiento se realiza en conjunto con la programación del módulo GPRS SIM800L para poder transmitir los datos correspondientes a las coordenadas de ubicación en tiempo real, entonces partiendo de la etapa de bloque de procesamiento lo primero a realizar será establecer una comunicación entre el módulo SIM 800l y el Arduino. Cuando se realiza la conexión del módulo con el Arduino, hay que tomar en cuenta 3 patrones fundamentales.

- Patrón 1: el LED parpadea después de cada 1 s, lo que significa que el SIM800l está encendido, pero no ha establecido la conexión.
- Patrón 2: el LED parpadea después de cada 3S. Significa que sim800l ha establecido con éxito la conexión.
- Patrón 3: el LED parpadea después de cada 2S. Significa que la función GPRS está activada.

A continuación, en la figura 37 se presenta la conexión entre la placa de desarrollo y el módulo SIM 800l tomando en cuenta a los patrones que se hace referencia en los puntos mencionados anteriormente.



**Figura 37.** Modos de estado del SIM800L, vista superior.

Tomando en cuenta la programación de este módulo se lo realiza en el software de Arduino IDE al igual que el módulo GPS NEO6m, en la figura 38 se visualiza la declaración de pines a utilizar, se inicializa la comunicación serial, se envía mensajes de éxito cuando el módulo SIM 800L esté preparado para emitir mensajes al smartphone del dueño a través de la red móvil, el código en su totalidad se lo presenta en el Anexo 3.

```

#include <TinyGPS++.h>
TinyGPSPlus gps;
double latitud, longitud;

#include <SoftwareSerial.h>
SoftwareSerial SIM800L(10,11);

String response;
int lastStringLength = response.length();

String link;

void setup() {
  Serial.begin(9600);
  Serial.println("Encontrar a mi Mascota");

  SIM800L.begin(9600);
  SIM800L.println("AT+CMGF=1");
  Serial.println("SIM800L empezo a 9600");
  delay(1000);
  Serial.println("Instalación completa! SIM800L está listo!");
  SIM800L.println("AT+CNMI=2,2,0,0,0");
}

```

**Figura 38.** Código de transmisión de Datos del sistema PETROUTE.

En este punto los comando AT juegan un papel muy fundamental ya que serán encargados de la configuración y acciones que el módulo necesite para poder transmitir los datos de coordenadas y de igual forma atender a las peticiones del usuario final cuando lo necesite. A continuación, en la Tabla 24 se presenta las acciones más importantes que realizan los comandos AT para su ejecución.

**Tabla 24.**

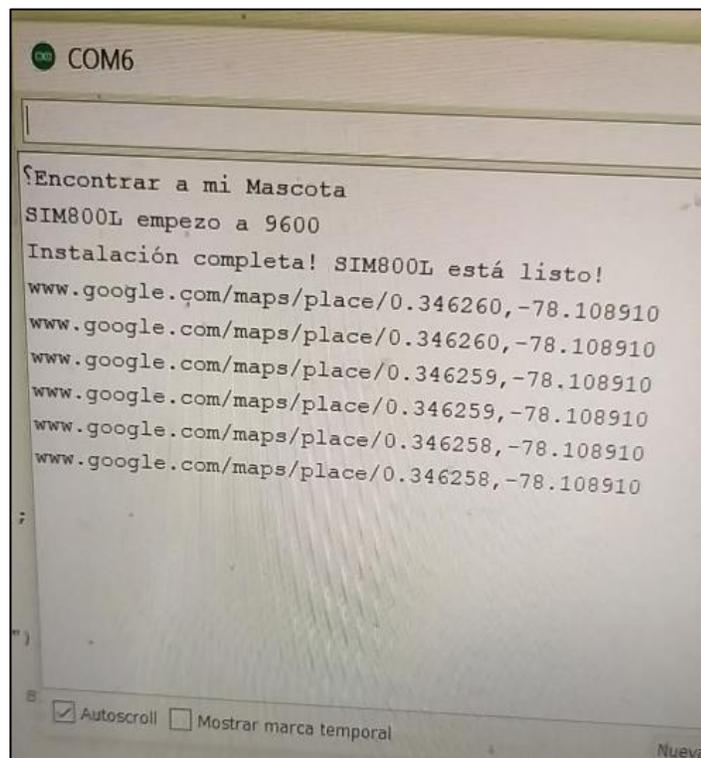
Comandos AT para el uso del módulo SIM800l mediante comunicación Serial

Comando AT	Descripción
AT	Si el módulo SIM800l funciona correctamente, Sim800l responderá OK después de recibir este comando.
AT+CBC	Este comando se utiliza para verificar el voltaje de la batería.

<b>AT+ CSQ</b>	El módulo SIM800l indica la intensidad de la señal después de recibir este comando.
<b>AT+CREG</b>	Se utiliza este comando para registrar la red si el módulo SIM800l pierde la red abruptamente y tarda mucho tiempo en escanear la red.
<b>¿AT+COPS?</b>	Si se utiliza este comando el SIM800l informará sobre el estado de la red.
<b>¿AT+COPS=?</b>	El módulo Sim800l especifica las redes disponibles después de recibir este comando AT

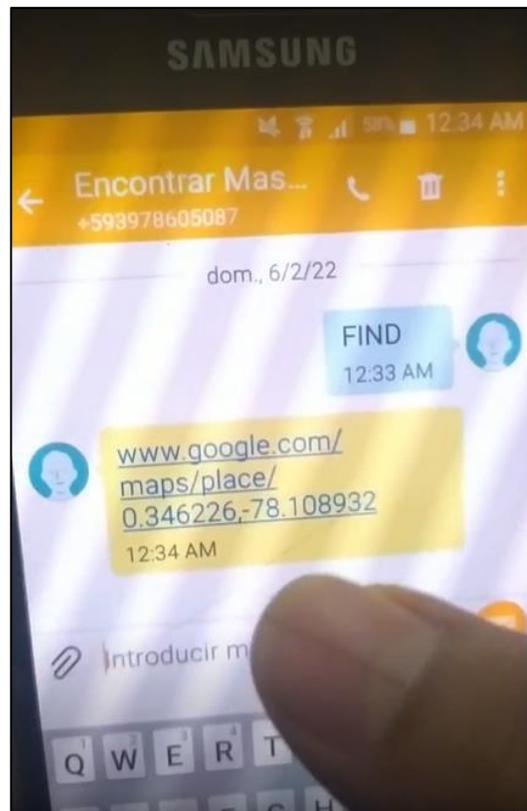
Fuente: Adaptado de (SIMCom, 2015) Elaborado por: Chicaiza F. 2022.

En la figura 39 se observa las coordenadas de Ubicación que se envían desde el monitor Serial del Arduino hacia el Smartphone del Dueño.



**Figura 39.** Coordenadas de ubicación del sistema PETROUTE.

En la figura 40 se observa que las coordenadas de ubicación llegan a través de un SMS. Esta acción se la lleva a cabo al digitar la palabra FIND desde el smartphone del dueño mediante un SMS a la tarjeta SIM del módulo, lo cual el módulo SIM 8001 atiende a esta petición y seguidamente realiza la acción de enviar las coordenadas al Smartphone del dueño, las mismas coordenadas que se encontraban anteriormente visualizadas en el monitor Serial del Arduino Nano.



**Figura 40.** Coordenadas de ubicación enviadas desde el sistema PETROUTE al Smartphone del dueño.

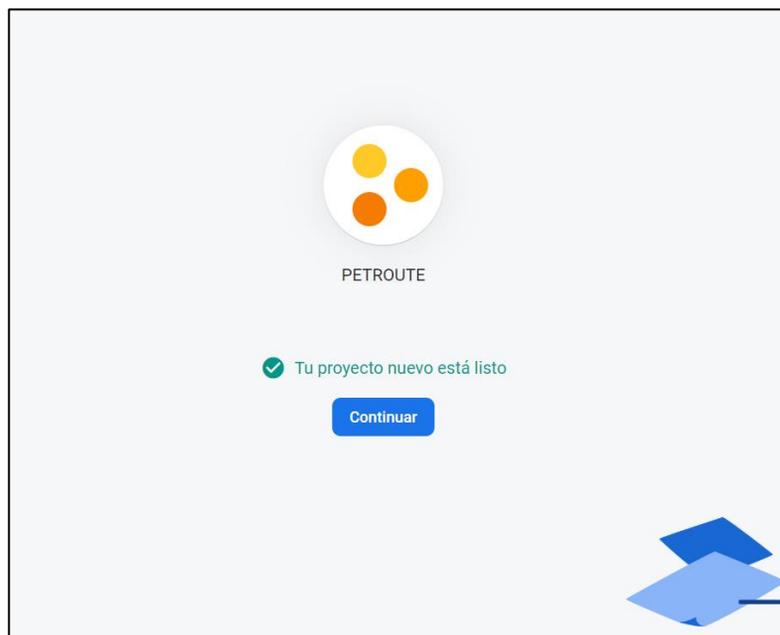
#### **3.9.2.4. Bloque de Visualización**

Finalmente se tiene el bloque de visualización y uno de los bloques más importante ya que será el encargado de contribuir al desarrollo de la programación de la base de datos en

tiempo real a través del Cloud y también el aplicativo móvil para el usuario final a mediante el software Flutter.

- ***Conexión con Firebase***

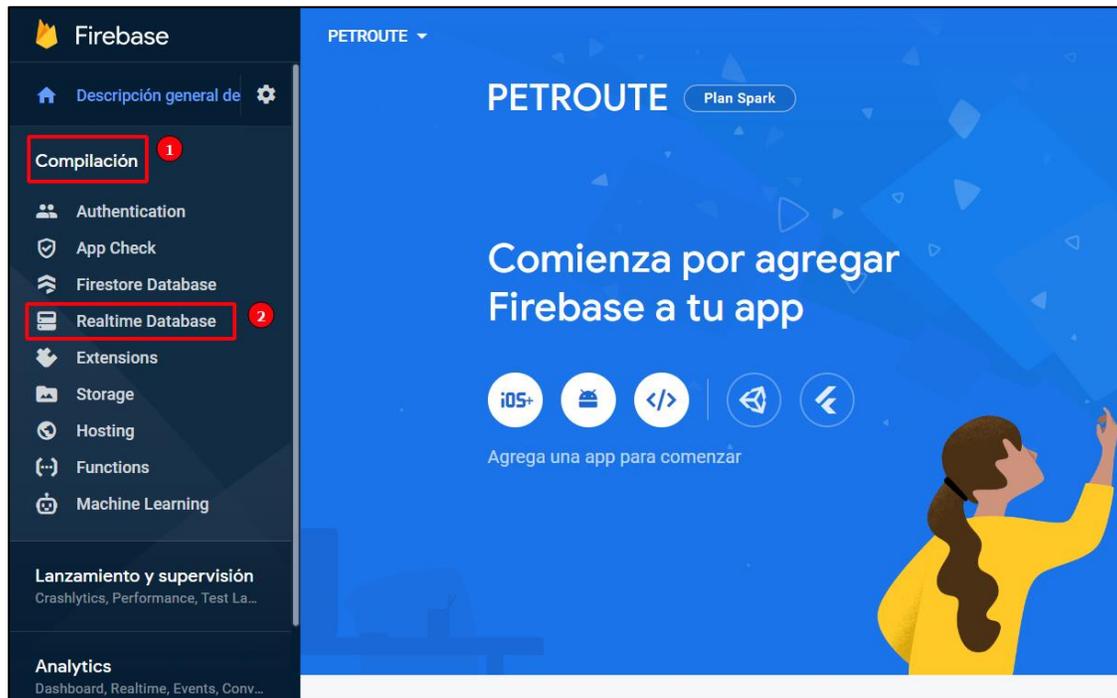
Lo primero que se realizará será crear una cuenta de Google Firebase. Luego, una vez que se complete el registro de la cuenta, se debe crear un proyecto en Google Firebase, realizando clic en Ir a Consola→Agregar proyecto, finalmente se procede a asignarle un nombre, y de esta forma, en la Figura 41 se puede observar la pantalla que debe aparecer si se realizó correctamente el proceso.



**Figura 41.** Proyecto creado con éxito, Firebase.

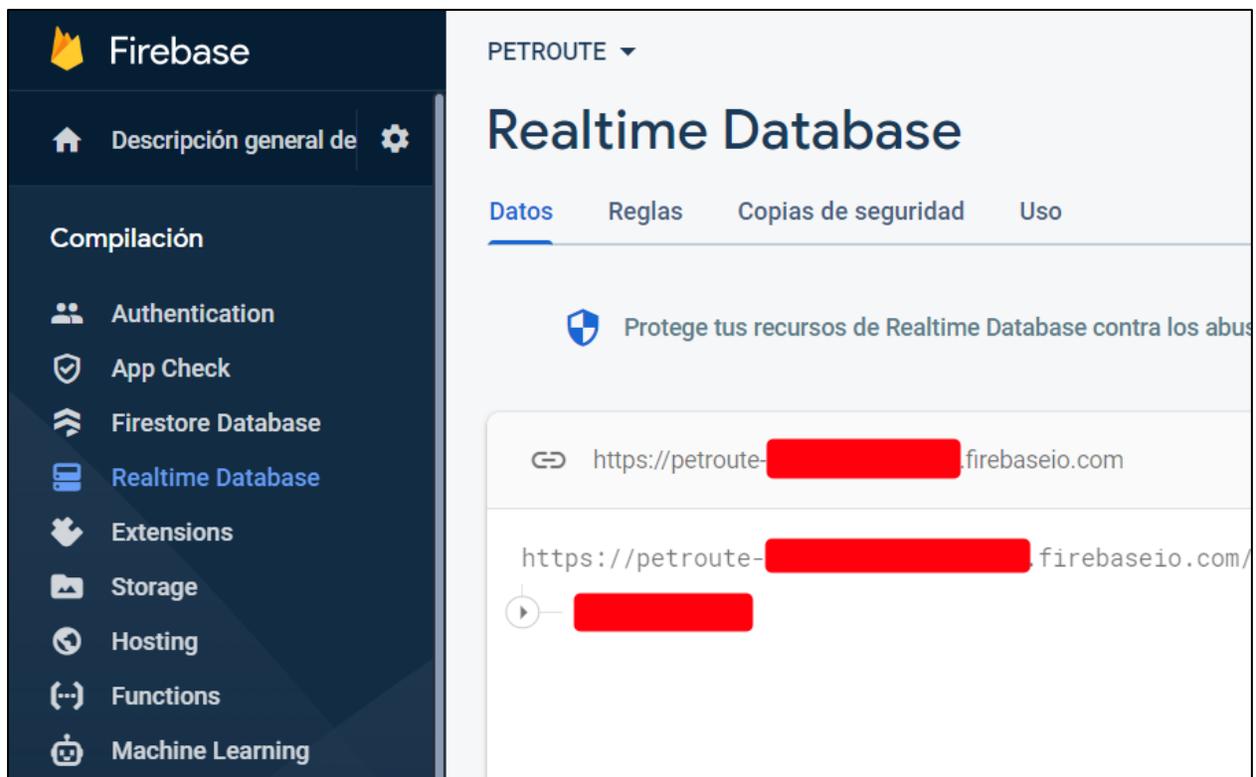
Se deberá dar click en el botón de continuar. Lo siguiente a realizar será crear una base de datos en tiempo real para lo cual en la pestaña del menú superior izquierdo llamado “compilación”, se deberá escoger el menú "Base de datos", de esta forma se seleccionará "Base

de datos en tiempo real" y luego se deberá dar click en crear base de datos. En la figura 42 se visualiza el procedimiento descrito.



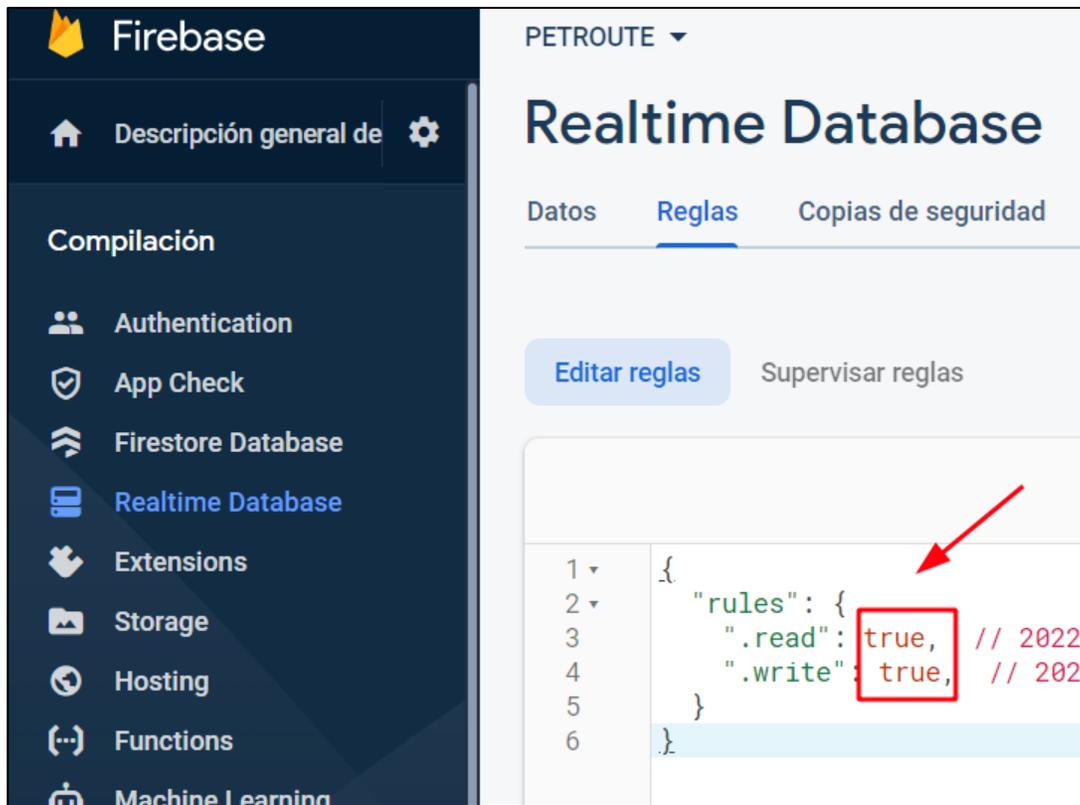
**Figura 42.** Creación de la base de datos en Firebase.

Después de hacer clic en Crear base de datos, aparecerá una regla de seguridad. luego se dará clic en habilitar. De esta forma en la figura 43 se puede visualizar como se ha creado la base de datos correctamente.



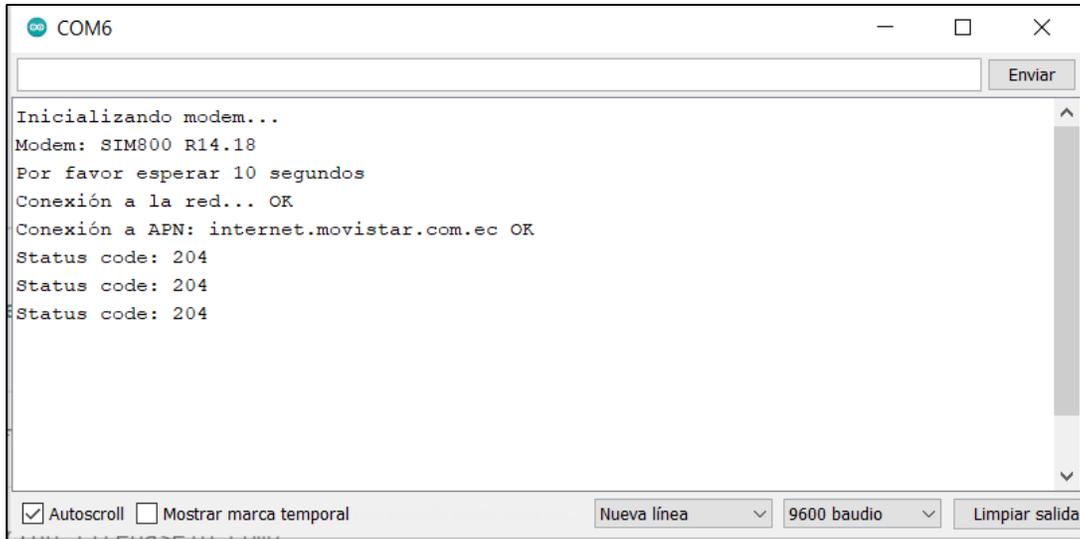
**Figura 43.** Base de datos en Firebase creada correctamente.

Posteriormente se procede a modificar las reglas de la base de datos en tiempo real, para lo cual se debe dirigir a la pestaña de reglas y modificar el parámetro de “write” → “true” de esta forma se podrá obtener los datos de coordenadas de ubicación en tiempo Real a través del módulo GPS. En la figura 44 se puede visualizar como se ha modificado las reglas de la base de datos correctamente.



**Figura 44.** Modificaciones en base de datos de Firebase realizadas correctamente.

Una vez creada la base de datos en tiempo real se procede a realizar la conexión del sistema embebido con Firebase mediante el servicio HTTP del módulo SIM800L. De esta forma en la figura 45 se puede observar a través del monitor serial como se establece la conexión hacia la APN de la operadora móvil.



**Figura 45.** Conexión exitosa a la nube de Firebase.

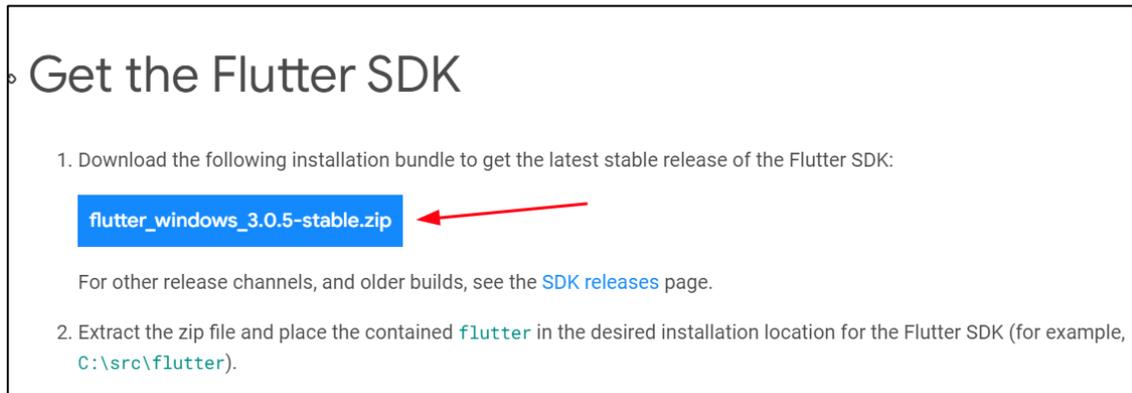
De esta forma los datos de coordenadas son enviados a la base de datos en tiempo real del Cloud de Firebase y se puede observar como las coordenadas de GPS se van actualizando constantemente en tiempo real cuando el sistema cambia de posición. En la figura 46 se puede visualizar como ocurre el cambio de coordenadas y como modifica Firebase estos datos.



**Figura 46.** Coordenadas de Posición actualizándose.

- **Aplicación Móvil de PETROUTE**

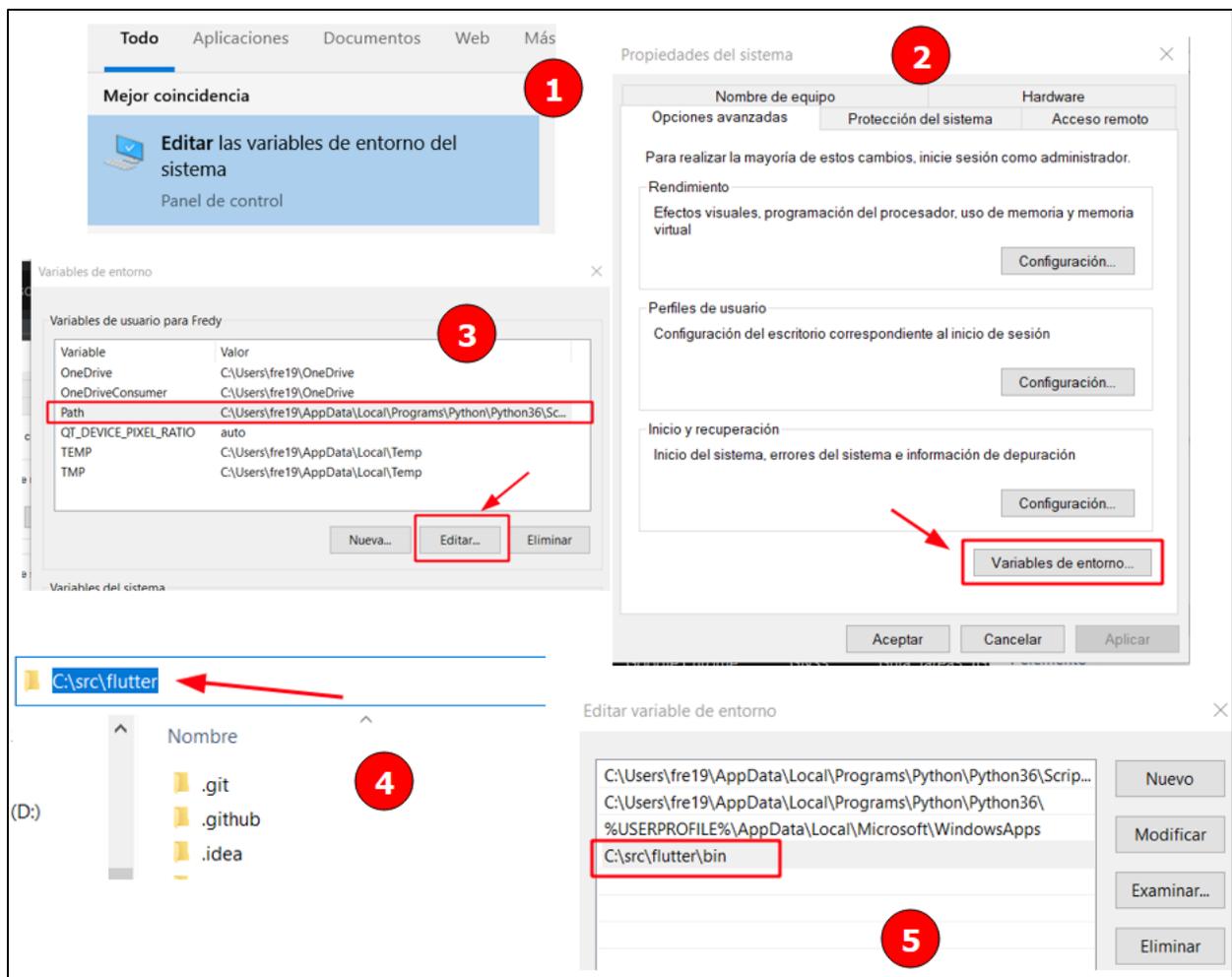
Lo primero que se realizará será obtener el SDK de Flutter para poder programar la aplicación móvil. Por esta razón se deberá dirigir a la página <https://docs.flutter.dev/get-started/install/windows> y dar clic en la pestaña de color celeste como se puede observar en la Figura 47.



**Figura 47.** Descarga de Flutter SDK.

Una vez realizada esta acción se deberá editar las variables de entorno para poder añadir la ruta del archivo de Flutter descargado anteriormente. A continuación, en las Figura 48 se puede observar cómo se ejecutan las acciones, tomando en cuenta los pasos que se deben de seguir para que no existan inconvenientes a la hora de programar la Aplicación:

- El primer paso será abrir la edición de variables de entorno del sistema.
- El segundo paso será dar clic en la pestaña de variables de entorno.
- El tercer paso será ubicar el Path y dar clic en la pestaña de Editar.
- El cuarto paso será copiar la ruta donde se encuentra el archivo de Flutter descargado anteriormente.
- Y por último se agregará dicha ruta al Path de edición abierto anteriormente.



**Figura 48.** Agregado de ruta dentro de las variables de entorno.

Seguidamente se verifica que la instalación y configuración de Flutter, que se encuentre este en buen estado, para realizar dicha verificación se usa el comando “flutter doctor” dentro del Powershell de Windows. En la Figura 49 se puede observar que todos los requerimientos se encuentran en perfecto estado lo cual quiere decir que la instalación y configuración es la correcta.

```
PS C:\Users\fre19> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.0.5, on Microsoft Windows [Version 10.0.19044.1826], locale es-EC)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Community 2019 16.9.5)
[✓] Android Studio (version 2021.2)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

• No issues found!
PS C:\Users\fre19>
```

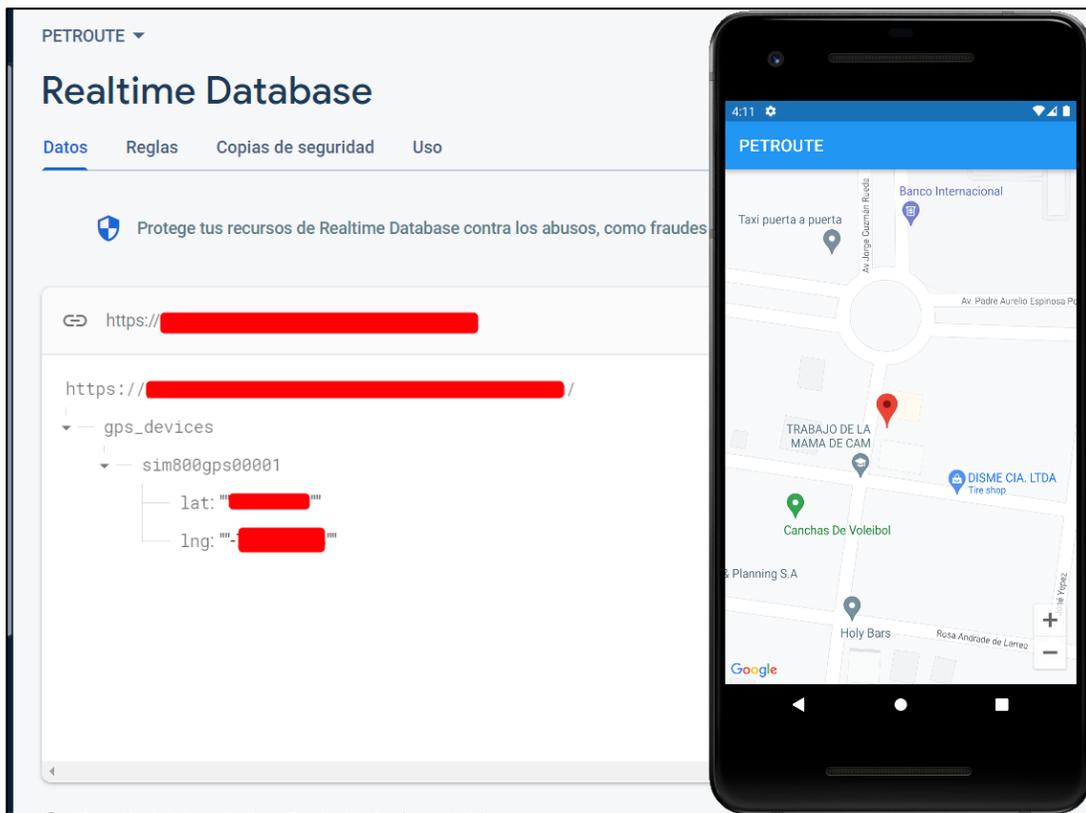
**Figura 49.** Requerimientos de Instalación y configuración correctos.

De esta forma se procede a realizar el desarrollo de la aplicación móvil a continuación se observa cómo opera la aplicación, la cual cada vez que ocurre un cambio de posición (actualización de coordenadas geográficas), dentro de la base de datos, comienza el marcador de la aplicación de igual forma a cambiar su posición en tiempo real dentro de la pantalla del mapa. En la Figura 50 se puede observar la interfaz del mapa.



**Figura 50.** Interfaz de PETROUTE.

Cada vez que exista un cambio de posición del sistema de igual forma existe dentro de la base de datos, por lo tanto, el marcador de la aplicación comienza a cambiar su posición en tiempo real dentro de la pantalla del mapa. En la Figura 51 se puede observar el formato que se da mediante actualización de coordenadas entre la base de datos en tiempo real y el aplicativo móvil.



**Figura 51.** Interacción Realtime Database y Aplicativo Móvil.

#### 4. CAPITULO IV Pruebas de Funcionamiento y Resultados

En este capítulo se presenta las pruebas de funcionamiento del sistema embebido de detección y rastreo de mascotas. Estas pruebas establecen la parte final de la metodología del modelo en V, en este apartado se toma en cuenta la validación del sistema implementado, arquitectura, funciones, requisitos y especificaciones que se han establecido en el diseño. Con esta etapa se da por concluida la integración de cada fase del proceso metodológico para representar la información de forma ordenada

##### 4.1. Especificación de pruebas a desarrollar

La ejecución de pruebas a realizar, se componen para cada nivel de pruebas del modelo en V, a continuación, en la Tabla 25 se presenta el descriptivo de cada fase y los resultados que se desea obtener y documentar en el desarrollo del presente capítulo.

**Tabla 25.**

Pruebas de Funcionamiento

<b>Pruebas de Funcionamiento</b>		
<b>Tipo de Prueba</b>	<b>Descripción</b>	<b>Resultado previsible</b>
<b>Fase 1. Pruebas de Cumplimiento</b>	Estas pruebas corresponden a donde se realiza una lista de comprobación de las características del prototipo electrónico establecidas en los requerimientos de usuario y operacionales.	En la primera fase se espera verificar que se cumpla la lista de requisitos de Stakeholders
<b>Fase 2. Pruebas de Funcionalidad</b>	Estas pruebas tienen la finalidad de verificar cómo se establece el	En esta fase se espera comprobar la lista de requerimientos de sistema

funcionamiento de los módulos que integran el sistema.

<b>Fase 3. Pruebas de Implementación</b>	Estas pruebas se realizan en campo con el usuario correspondiente.	En esta fase final se espera comprobar que el sistema funcione para establecer la ubicación en tiempo real de la mascota.
--	--	---

#### 4.2. Pruebas de cumplimiento

En estas pruebas se busca recopilar la información con respecto a los requerimientos de Stakeholders para verificar su existencia y establecer la lista de los requisitos más importantes que se cumple en el desarrollo de la etapa de pruebas de especificaciones.

En la Tabla 26 se muestran los requerimientos más relevantes en el desarrollo de este sistema.

**Tabla 26.**

Cumplimiento de requerimientos de Stakeholders

<b>Requerimientos de Stakeholders</b>	<b>StSR1</b>	<b>StSR2</b>	<b>StSR3</b>	<b>StSR4</b>	<b>StSR5</b>	<b>StSR6</b>	<b>StSR7</b>	<b>StSR8</b>	<b>StSR9</b>
<b>Cumple</b>	SI	X		X	X	X	X	X	X
	NO		X						

En los requerimientos de Stakeholders existe un requerimiento que en su totalidad no se logró cumplir, debido a que los equipos electrónicos ocupan un espacio considerable, por lo cual el sistema debe ser integrado en mascotas de raza grande.

En la Tabla 27 se especifican los requerimientos de sistema, donde se seleccionó a los requerimientos que poseen un aporte significativo al sistema.

**Tabla 27.**

Cumplimiento de requerimientos de Sistema

<b>Requerimientos de sistema</b>		<b>SySR2</b>	<b>SySR4</b>	<b>SySR7</b>	<b>SySR8</b>	<b>SySR12</b>	<b>SySR16</b>	<b>SySR17</b>
<b>Cumple</b>	<b>SI</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>		
	<b>NO</b>						<b>X</b>	<b>X</b>

Para requerimientos de sistema es importante tomar en cuenta abarcar todos los requerimientos tanto los que cumple al 100% como los que en su totalidad no poseen un cumplimiento del 100%, por lo cual los requerimientos que poseen relación a la comunicación móvil y comunicación entre módulos del sistema poseen un cumplimiento aceptable, no obstante los requerimientos de aspectos físicos no cumple debido a que el tamaño del sistema no puede ser adaptado en el cuello de la mascota, por lo cual se debe implementar en otra parte de la mascota.

En la Tabla 28 de muestra los requerimientos de arquitectura, donde se seleccionó a los requerimientos que poseen un aporte significativo al sistema.

**Tabla 28**

Cumplimiento de requerimientos de arquitectura

Requerimientos de arquitectura	SRSH3	SRSH4	SRSH5	SRSH8	SRSH10	SRSH14	SRSH15	SRSH16	SRSH17	SRSH20	SRSH22	SRSH23	SRSH25	SRSH26	SRSH27
	Cumple	X	X	X	X	X	X	X	X	X	X	X	X	X	X
NO															

Para los requerimientos de arquitectura al ser requerimientos que se establecen y cumplen por el desarrollador del proyecto y al ser los requerimientos de Software, Hardware, Lógicos, Diseño y Eléctricos, se logra cumplir satisfactoriamente los requerimientos.

### **4.3.Pruebas de Funcionalidad**

Dentro de las pruebas de funcionalidad se aborda la verificación de cada uno de los bloques que corresponden al sistema logrando así demostrar el correcto funcionamiento de cada uno de los elementos que forman parte del sistema.

#### **4.3.1. Bloque de alimentación**

Para el bloque de alimentación se procede a realizar un test eléctrico con la finalidad de verificar el correcto funcionamiento de la batería y las líneas de alimentación, tanto para el módulo SIM800L, GPS Neo6M y ARDUINO NANO, donde se procede a verificar mediante la prueba de continuidad en las líneas de alimentación y medición de voltaje creando una línea de

tiempo el voltaje de la entrada al sistema durante un periodo de dos horas, para la verificación de su funcionamiento y si existen caídas de voltaje, en la Tabla 29 se muestra el test a realizar.

**Tabla 29.**

Test Eléctrico

<b>Test Eléctrico</b>	
<b>Bloque de prueba</b>	<b>Bloque de alimentación</b>
<b>Descripción</b>	Verificación de funcionamiento para bloque de alimentación.
<b>Prerrequisitos:</b>	
<ul style="list-style-type: none"> <li>✓ Adquisición de batería de Lipo de 4600mAh 3.7 V</li> <li>✓ Adquisición de módulo SIM800L.</li> <li>✓ Adquisición de módulo GPS Neo6M.</li> <li>✓ Adquisición de Arduino nano.</li> <li>✓ Adquisición de modulo estabilizador y elevador de voltaje Mt3608.</li> <li>✓ Placa PCB con líneas de para VCC y GND.</li> </ul>	
<b>Pasos:</b>	
<ul style="list-style-type: none"> <li>✓ Verificar valor de voltaje de la batería 3.7 voltios.</li> <li>✓ Verificar la entrada al módulo SIM800L</li> <li>✓ Verificar la entrada de voltaje al Arduino nano.</li> <li>✓ Verificar continuidad en placa PCB en líneas VCC y GND mediante el uso de multímetro.</li> <li>✓ Crear una línea de temporal del voltaje suministrado.</li> </ul>	
<b>Resultados esperados:</b>	
Bloque de alimentación alimenta correctamente al sistema con 3.7 voltios al módulo SIM800L, y mediante el elevador de voltaje 5 voltios al Arduino nano.	

**Resultado obtenido:** tras realizar las diferentes mediciones en las líneas de alimentación del sistema, se obtiene que la batería entrega un voltaje de 3.8 voltios lo cual entra en el rango soportado por el módulo SIM800L que soporta un voltaje de 3.3 hasta 4.4 voltios, en las líneas

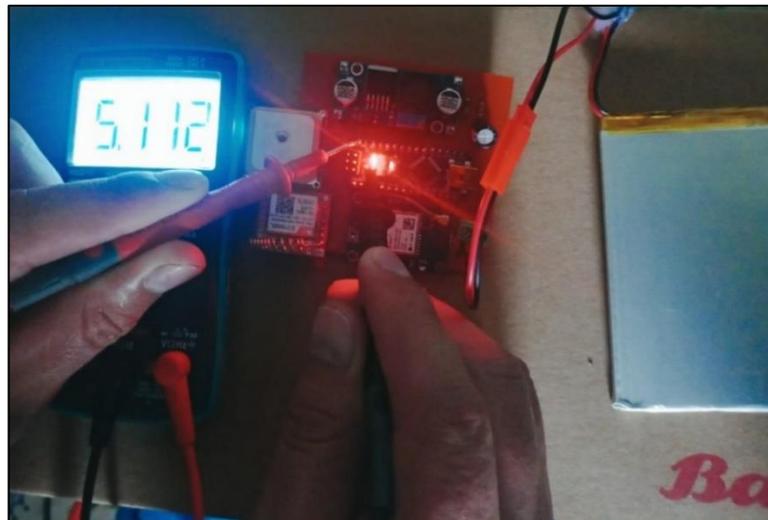
de alimentación del Arduino nano existe una medición de 5.1 voltios el cual entra en el rango del voltaje soportado por el Arduino nano que soporta  $5 \pm 0.1$  voltio. Este voltaje es proporcionado por el elevador de voltaje.

En la Figura 52 se muestra la medición en las borneras de entrada las cuales van conectadas a la batería que posee el sistema, con una medición de 3.897 voltios.



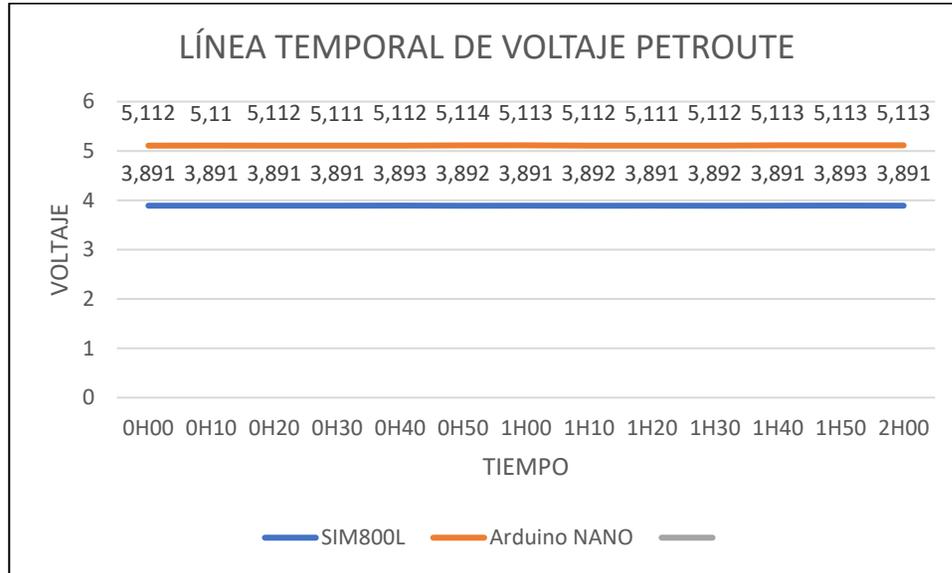
**Figura 52.** Voltaje de entrada para el sistema

En la Figura 53 se muestra la medición de voltaje en los pines de entrada del Arduino nano, con una medición de 5.112 voltios.



**Figura 53.** Medición de voltaje en los pines del Arduino nano.

La Figura 54 es un histograma temporal del voltaje suministrado al sistema, capturado a la entrada del sistema donde se diseñó dos líneas enérgicas la cual suministra en una línea con 3.7 voltios a modulo SIM800L y una línea de 5 voltios al Arduino nano.



**Figura 54.** Línea temporal de voltaje

#### 4.3.1.1. Tiempo de Vida del Sistema

Para calcular el tiempo de vida útil del sistema se debe verificar en las hojas técnicas de cada módulo.

En el caso de los módulos MT3608, SIM800L y GPS NEO 6M v2 el tiempo de vida útil es mayor o igual a 5 años

Para el caso del Arduino nano se debe verificar el número de ciclos de escritura y borrado de las memores no volátiles, para el Arduino nano tenemos: Write/erase cycles: 10,000 flash/100,000 EEPROM, como las dos memorias son dependientes una del otro, entonces se debe basar al número de ciclos menor, en este caso sería el valor de la memoria flash 10000

ciclos, si le garbamos 2 veces, se tendría  $10000/2= 5000$  días, se tendría un tiempo de vida del microcontrolador de  $\cong 13$  años (ATMEL,2020).

A continuación, se puede sacar un promedio total del tiempo de Vida útil de todo el sistema, como se visualiza en la ecuación 4.

$$13 \text{ años} + 5 \text{ años} + 5 \text{ años} + 5 \text{ años} = 28 \text{ años}$$

$$\textit{Promedio de Vida Util} = \frac{28 \text{ años}}{4}$$

$$\textit{Promedio de Vida Util} = 7 \text{ años}$$

**Ecuación. 4.** Promedio del tiempo de Vida Útil del sistema PETROUTE

#### **4.3.1.2. Tiempo de Carga y Descarga de la Batería.**

Según (COELECTRIXP,2022) para calcular el tiempo de descarga se utilizará las siguientes formulas, que ayudaran con el cálculo del tiempo de durabilidad de la batería antes de descargase.

$$Vb \times Ib = Wb$$

$$Vb \times Ic = Wc$$

$$\frac{Wb}{Wc} = \textit{Tiempo\_Bateria}$$

**Ecuación. 5.** Fórmula para calcular el tiempo de descarga de la batería

Donde, (Vb) es el Voltaje del batería multiplicado por (Ib) intensidad de la batería, (Vb) voltaje del batería multiplicado por (Ic) la corriente consumida.

Se tiene una batería de 3000mAh a 3.7 Voltios y de acuerdo con la sección 3.8.1.1. se realizó el cálculo de consumo de todo el sistema donde se tiene 414mA. Se procede a calcular

$$W_b = 3.7V \times 3000mAh = 111000VmAh$$

$$W_c = 3.7V \times 414mA = 1531.8VmA$$

$$\frac{W_b}{W_c} = \text{Tiempo\_Bateria}$$

$$\frac{111000VmAh}{1531.8VmA} = \text{Tiempo\_Bateria}$$

$$72.46\text{Horas} = \text{Tiempo\_Bateria}$$

**Ecuación. 6.** Cálculo del tiempo de descarga de la batería

Para la carga de batería se tiene una batería de 3000mAh lo que es igual a 3Ah, en este caso se utiliza un cargador a 2 A, por lo cual 3Ah dividido entre 2 A se tiene 1,5 horas, que es el tiempo de carga de una batería de 3000mAh con un cargador a 2A.

#### 4.3.2. *Bloque de toma de datos y procesamiento*

Para el bloque de toma de datos y procesamiento se realiza un test de datos donde se procede a verificar la toma, procesamiento y envío de datos, en la Tabla 30 se muestra el test a realizar.

**Tabla 30.**

Test de Datos

<b>Test de Datos</b>	
<b>Bloque de prueba</b>	Bloque de toma de datos y procesamiento

---

<b>Descripción</b>	Prueba de funcionamiento para bloque de toma de datos y procesamiento
--------------------	---

---

**Prerrequisitos:**

- ✓ Modulo SIM800L
- ✓ Modulo GPS Neo6M.
- ✓ Arduino nano.
- ✓ Dependencias y librerías para cada módulo en un sketch de Arduino.

**Pasos:**

- ✓ Cargar sketch al Arduino nano.
- ✓ Verificar compilación correcta del sketch.
- ✓ Abrir serial monitor de Arduino
- ✓ Verificar toma de datos GPS.
- ✓ Verificar envío de datos mediante le comunicación 2.5G.

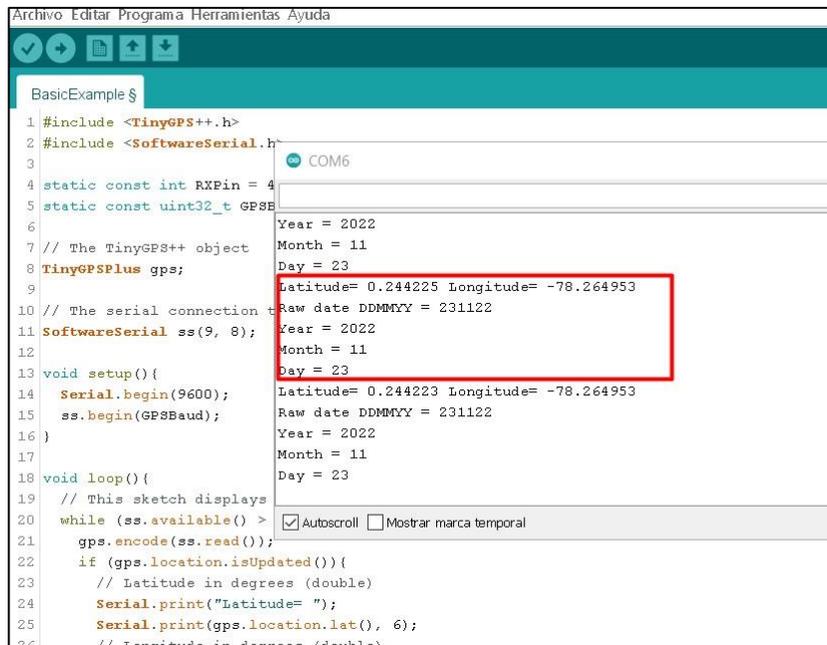
**Resultados esperados:**

Toma de datos correctas en tiempo real de las coordenadas latitud y longitud de la ubicación del sistema y respuesta correcta de comando AT y correcta conexión a la APN mediante el módulo sim800l.

---

**Resultados obtenidos:** los módulos funcionan correctamente se lo puede visualizar en el serial monitor de Arduino tanto para las coordenadas GPS y la respuesta correcta del módulo sim800l al aplicar comando AT.

Se verifica la captura de la ubicación en tiempo real de la mascota mediante le módulo GPS Neo6M, en la Figura 55 se muestra mediante la comunicación serial de Arduino y el serial monitor, la resultante al momento de poner a funcionamiento el módulo.



**Figura 55.** Modulo GPS en funcionamiento

Para la verificación del módulo sim8001 se procede a verificar conexión a la APN mediante el uso de comandos AT, en la Figura 56 se muestra la respuesta a cada uno de los comandos AT.



**Figura 56.** Prueba de conexión a la red móvil mediante comandos AT.

### 4.3.3. Bloque de transmisión de datos

Para el bloque de transmisión de datos se realiza un test de transmisión de datos para verificar el funcionamiento correcto del módulo SIM800L y la correcta conexión a la red móvil.

En la Tabla 31 se muestra el test a realizar.

**Tabla 31.**

Test de Transmisión de Datos

<b>Test de Transmisión de Datos</b>	
<b>Bloque de prueba</b>	Bloque de transmisión de datos
<b>Descripción</b>	Prueba de funcionamiento para bloque de transmisión de datos.
<b>Prerrequisitos:</b>	
<ul style="list-style-type: none"><li>✓ Datos GPS capturados.</li><li>✓ Conexión a red móvil 2.5G.</li><li>✓ Servidor Firebase activo.</li></ul>	
<b>Pasos:</b>	
<ul style="list-style-type: none"><li>✓ Verificar conexión a red móvil 2.5G.</li><li>✓ Envío de datos a servidor Firebase.</li><li>✓ Verificación de envío, recepción y almacenamiento de datos.</li></ul>	
<b>Resultados esperados:</b>	
Datos gps en tiempo real y conexión estable del módulo sim800l a la red móvil.	

**Resultados obtenidos:** los datos capturados por el módulo gps Neo6M poseen una exactitud de  $\pm 8$  metros, además el módulo sim800l mantiene una conexión estable con la red móvil 2.5G, lo cual los datos se reflejan en el servidor Firebase correctamente.

En la Figura 57 se muestra los datos enviados y publicados en el servidor Firebase y comparados con los valores reflejados en el serial monitor de Arduino.

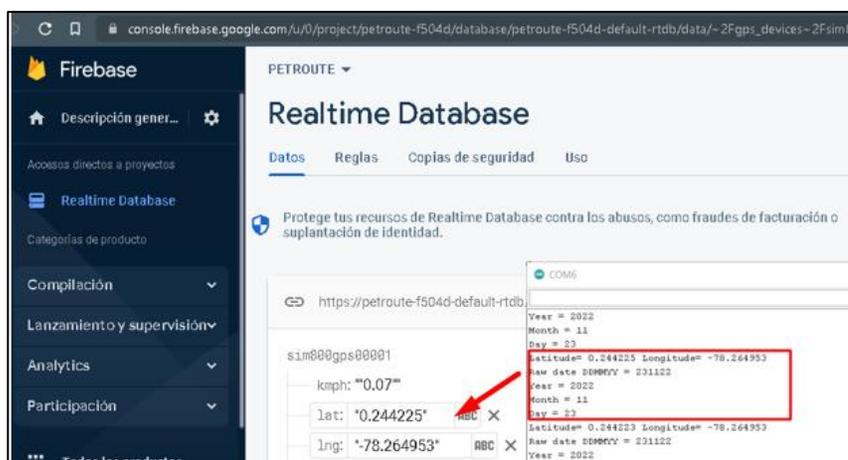


Figura 57. Datos publicados en el servidor FIREBASE

#### 4.3.4. Bloque de visualización

Para el bloque de visualización se realizar un test de visualización de datos, donde se precede a verificar que los datos sean visibles en tiempo real tanto en el servidor como en la aplicación móvil. En la Tabla 32 se muestra el test a realizar.

Tabla 32.

Test de Visualización

Test de Visualización	
<b>Bloque de prueba</b>	Bloque de visualización de datos.
<b>Descripción</b>	Prueba de funcionamiento para bloque de visualización de datos.
<b>Prerrequisitos:</b>	
<ul style="list-style-type: none"> <li>✓ Modulo GPS activo y capturando datos.</li> <li>✓ Modulo SIM 800L con conexión estable a red móvil 2.5G.</li> <li>✓ Servidor Firebase activo.</li> <li>✓ Aplicación móvil activa.</li> </ul>	
<b>Pasos:</b>	
<ul style="list-style-type: none"> <li>✓ Verificar de captura de datos GPS.</li> <li>✓ Verificar conexión a red móvil 2.5G.</li> </ul>	

- 
- ✓ Envío de datos a servidor Firebase.
  - ✓ Verificación de envío, recepción y almacenamiento de datos.
  - ✓ Verificación de datos registrados en servidor.
  - ✓ Verificación de ubicación en aplicación móvil.

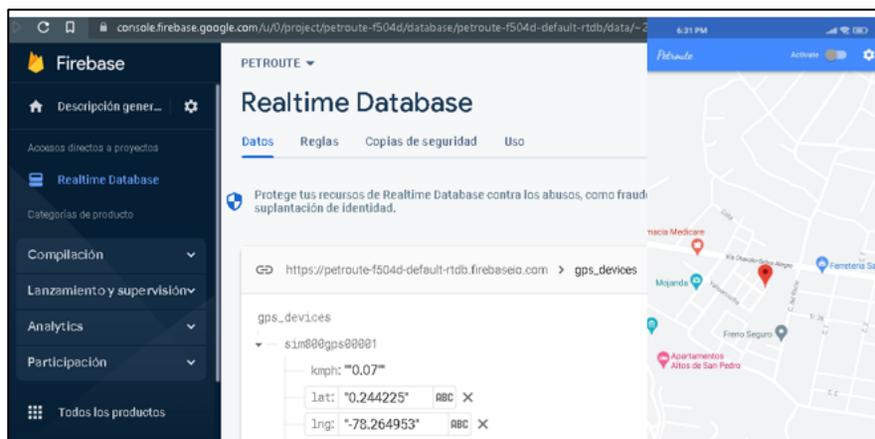
### Resultados esperados:

Sistema funcionando correctamente capturando y enviando los datos de la ubicación al servidor para que los datos de la ubicación se reflejen en la aplicación móvil.

---

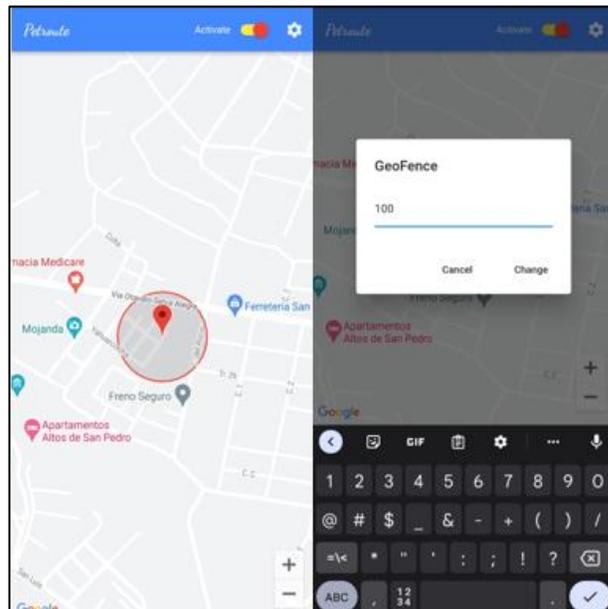
**Resultados obtenidos:** El sistema funciona correctamente esto se refleja en los datos adquiridos y que se actualizan en tiempo real, lo cual es visible en la base de datos y en la aplicación móvil que muestra la ubicación del sistema actualizando constantemente de acuerdo con el movimiento en las coordenadas de latitud y longitud.

En la Figura 58 se muestra la ventana de servidor de base de datos en tiempo real, donde muestra los datos capturados de longitud y latitud, además se puede visualizar en la parte derecha la ubicación en Google Maps del dispositivo.

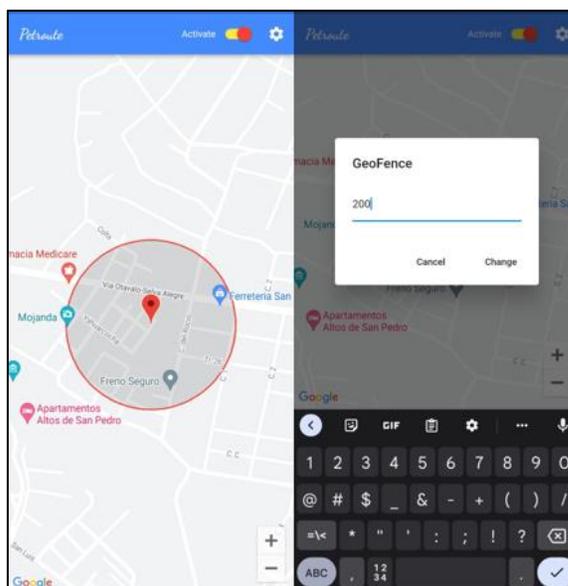


**Figura 58.** Visualización en tiempo real de la ubicación en coordenadas de longitud y latitud.

Además, en la aplicación móvil poseen una geocerca que se puede modificar el rango de cobertura en metros, dependiendo de la necesidad de rastreo de la mascota, en la Figura 59 se muestra la asignación de cobertura a una Geocerca activa de 100 metros y en la Figura 60 se muestra una Geocerca activa a 200 metros).



**Figura 59.** Geocerca activa a 100 metros

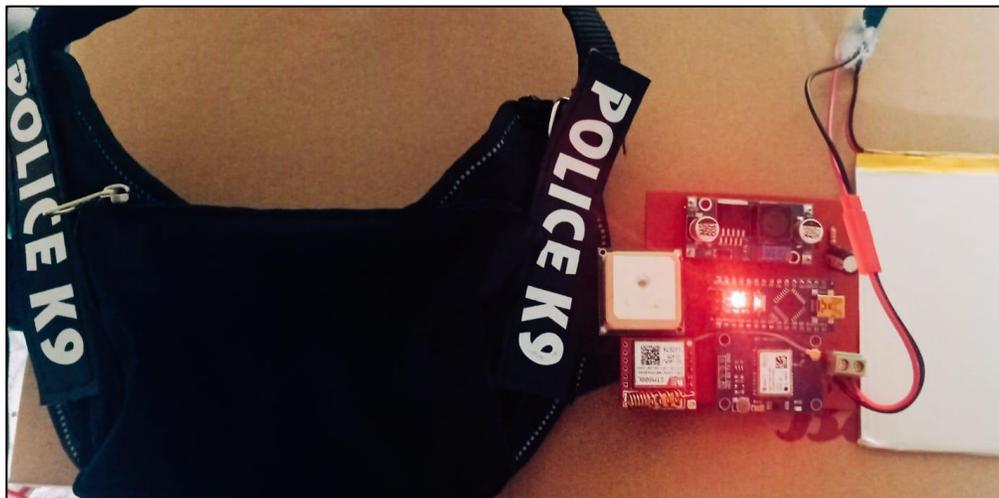


**Figura 60.** Geocerca activa a 200 metros

#### 4.4. Pruebas de Implementación

Estas pruebas se llevan a cabo cuando se verifica que la integración de las anteriores pruebas actúa correctamente, por lo tanto, se comprueba la funcionalidad y a su vez tienen la finalidad de verificar cómo se comporta el dispositivo electrónico tomando como referencia al usuario final y su interacción con el sistema.

En la Figura 61 se muestra el sistema final funcionado y accesorio que se utilizara para colocar el sistema en la mascota, este accesorio se diseñó con el fin de no ocasionar mal estar en la mascota o interfiera con el funcionamiento eléctrico y radiofrecuencia del sistema o daños físicos en el sistema.



**Figura 61.** Accesorio de portación del sistema.

En la Figura 62 se muestra la adaptación del accesorio con el sistema, el cual encaja satisfactoriamente, logrando un diseño que no cause algún mal estar en la mascota, por su tamaño y diseño.



**Figura 62.** Sistema y accesorio montado.

En la Figura 63 se muestra el sistema final listo para la colocación en la mascota.



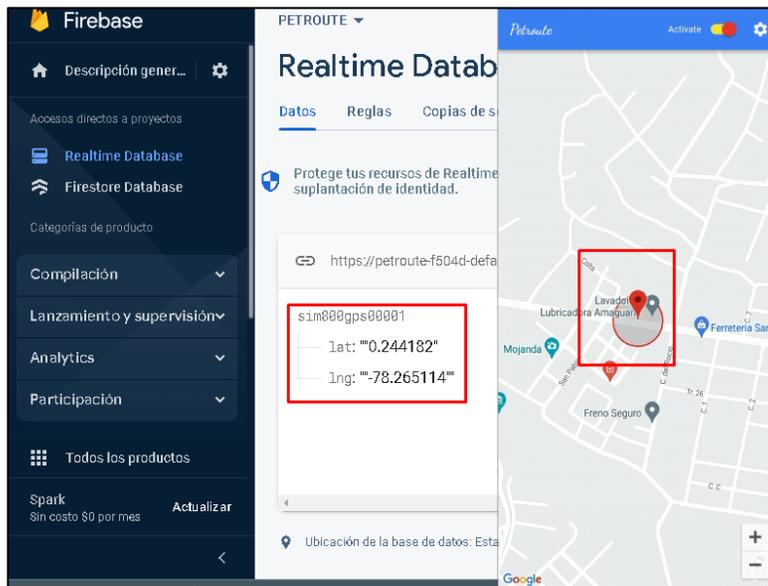
**Figura 63.** Sistema PETROUTE montado y listo para su funcionamiento.

En la Figura 64 se muestra a la mascota a la cual se le ha colocado el accesorio juntamente con el sistema y se procede a observar el comportamiento de la mascota y el comportamiento del sistema ya en funcionamiento



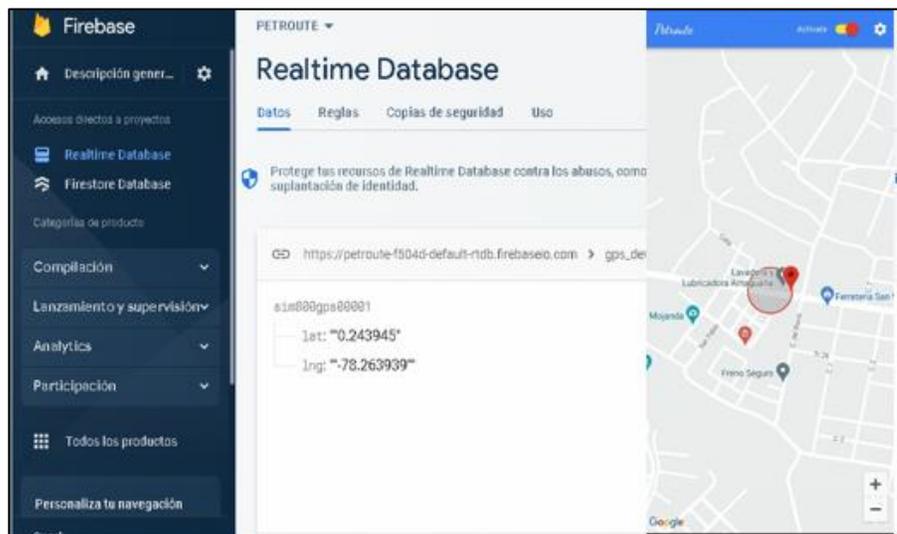
**Figura 64.** Sistema colocado en la mascota

En la Figura 65 se muestra en la aplicación móvil la ubicación de la mascota y la trayectoria que realiza además se activó la geocerca para verificar cuando la mascota sobrepase los límites de esta.



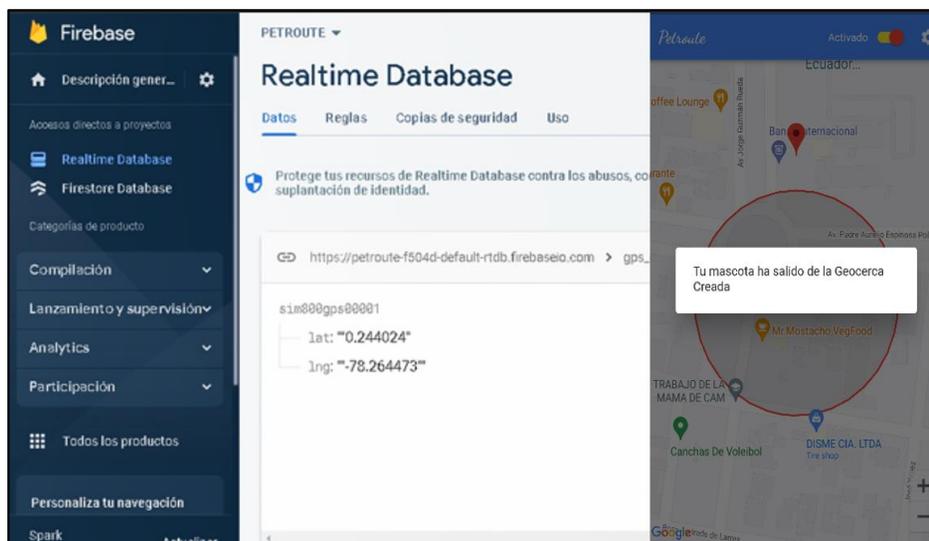
**Figura 65.** Aplicación móvil y data base

En la Figura 66 se puede observar como la mascota está por llegar al límite de la geocerca.



**Figura 66.** Ubicación de la mascota casi al límite de la geocerca

En la Figura 67 se muestra cuando la mascota ya sobrepasa la geocerca, por lo cual se refleja en la aplicación móvil una notificación que la mascota ha sobrepasado los límites establecidos.



**Figura 67.** Mensaje de Alerta cuando la mascota supera los límites de la geocerca.

#### 4.4. Análisis de Costo del Prototipo

El presente apartado se enfoca en analizar el costo de los elementos utilizados para la construcción del sistema tomando en cuenta el producto final para la construcción de varios o más prototipos, a su vez verificar el tiempo en el que se recupera la inversión del sistema implementado.

A continuación, en la tabla 33 se presenta el costo de los elementos en relación a la inversión realizada para la construcción del prototipo final a la comunidad Ibarreña.

**Tabla 33.**

Costo final del Prototipo a la comunidad Ibarreña

<b>COSTO FINAL DEL PROTOTIPO A LA COMUNIDAD IBARREÑA</b>			
<b>CANTIDAD</b>	<b>DESCRIPCIÓN</b>	<b>COSTO UNITARIO(USD)</b>	<b>COSTO TOTAL(USD)</b>
1	Placa de desarrollo Arduino Nano	10	12
1	Modulo GPRS SIM 800L	15	15
1	Modulo GPS NEO 6mv2	15	15
1	Baterías	10	10
1	Pechera	6,50	6,50
1	Aplicativo Móvil	18,50	18,50
1	Mantenimiento	11	11
<b>TOTAL</b>			<b>88,00</b>

Si la comunidad Ibarreña desea adquirir el sistema PETROUTE tendrá que invertir un monto de \$88,00 tomando en cuenta solo la parte de Hardware y Software, de esta forma

permitiendo utilizar el sistema para localizar en tiempo real a su mascota y visualizando su posición a través del aplicativo móvil.

Para recuperar el capital invertido inicialmente que fue de \$1659,24 en referencia a la Tabla 21 del costo total de sistema, se toma en cuenta un valor de ganancia de \$18.50 del aplicativo móvil ya que solo se desarrolla una vez y además se suma una cantidad adicional de 12 dólares, dando un total de \$100 que se vendería en su fase final, el sistema PETROUTE a la comunidad Ibarreña, a continuación en la Tabla 34 se presentan datos en función de 1 año de tiempo, que se usaría para ver el tiempo de recuperación invertida y el tiempo de ganancia a favor al completar el año, en periodos de 4 meses de producción.

**Tabla 34.**

Línea de Producción del Sistema PETROUTE

<b>LINEA DE PRODUCCIÓN DEL SISTEMA PETROUTE</b>			
<b>INVERSIÓN INICIAL</b>		<b>FLUJO DE CAJA</b>	
<b>MES 0</b>	<b>MES 4</b>	<b>MES 8</b>	<b>MES 12</b>
<b>Unidades Producidas (0)</b>	<b>Unidades Producidas (45)</b>	<b>Unidades Producidas (45)</b>	<b>Unidades Producidas (45)</b>
- \$1659,24	\$1372,50	\$1372,50	\$1372,50
<b>Retorno del Capital</b>	-286.74	\$1085.76	\$2458.26

Realizando el análisis se puede concluir que el retorno de inversión se obtiene bajo una producción de 110 prototipos los cuales son producidos entre el octavo y doceavo mes, tomando en cuenta el valor de ganancia que se usa del aplicativo móvil, ya que solo se desarrolla una sola vez y más los \$12 de ganancia adicional, dando un total de \$30.50, después al finalizar el año se

analiza una ganancia neta de \$799.02 que se obtendría aparte de lo que se invirtió en un principio.

## **5. Conclusiones y Recomendaciones**

### **5.1. Conclusiones**

El sistema embebido de detección y rastreo de mascotas logró cumplir con los requerimientos de Stakeholders, sistema y arquitectura, donde el diseño físico cumplió con las expectativas del usuario final, logrando un producto con un tamaño adecuado para que la mascota no sufra ningún tipo de lesión.

El uso de módulos de comunicación móvil GPRS son una gran opción al momento de envío de datos en áreas metropolitanas, además que trabajan en frecuencias de 850 MHz donde la señal puede atravesar varios obstáculos sin atenuarse, además que el sistema no necesita gran ancho de banda, por lo que 2.5G se adapta al sistema con los 14.4 kbps que ofrece.

Al usar una base de datos Realtime en la nube se logra tener los datos disponibles para acceder a ellos desde cualquier lugar y desde cualquier dispositivo que posea los accesos a la misma, por lo cual la aplicación móvil diseñada puede ser distribuida a todos los miembros de la familia que deseen saber la ubicación de la mascota en tiempo real.

Al usar placas de desarrollo Open Source como lo es Arduino, se presentan grandes prestaciones al desarrollo de proyectos de innovación, ya que el desarrollador logra adaptar la

placa al propósito que desee, de igual forma añadir módulos de comunicación inalámbrica y a su vez enviar los datos desde cualquier punto con acceso a internet.

Al verificar cada uno de los bloques que posee el sistema mediante los test de funcionamiento se logra establecer un sistema robusto que cumple con el estándar ISO/IEC 30141 Internet de las Cosas (IoT) - Arquitectura de Referencia, logrando así ser un producto final fiable para el usuario.

Considerando los datasheets de los fabricantes el tiempo de vida útil del sistema es de aproximadamente 8 años o 70080 horas en las que el prototipo electrónico puede mantener un funcionamiento estable sin que haya inconvenientes con los módulos del sistema ni tampoco el microcontrolador Arduino.

Si la comunidad Ibarreña desea adquirir el sistema PETROUTE tendrá que invertir un monto de \$88,00 permitiendo utilizar el sistema para localizar en tiempo real a su mascota y a su vez poder visualizar su posición a través del aplicativo móvil. De igual forma gracias a la función de Geocerca el sistema PETROUTE puede ser incorporado para personas que sufren de Alzheimer, ayudando a que se alerte a su familiar en el momento en el que la persona salga de la geocerca creada.

## **5.2.Recomendaciones.**

Verificar la cobertura mediante los mapas que cada operadora presenta en sus páginas oficiales así se puede verificar que la zona donde se desplegara el producto no posea inconvenientes de acceso a la red.

Verificar las hojas técnicas de cada módulo a implementar verificando así el voltaje de trabajo, para así mejorar la vida útil del módulo.

Probar la continuidad de las líneas de la placa para no ocasionar cortos en el circuito, ya que los módulos son muy sensibles y se pueden echar a perder.

Si se desea cambiar de tecnología para la etapa del bloque de transmisión de datos, se debe verificar que el módulo que se vaya a implementar sea compatible con el microcontrolador Arduino NANO ya que algunos módulos no son compatibles o necesitarían más memoria de almacenamiento dentro del microcontrolador.

Para futuro se recomienda utilizar un módulo 4G LTE como mejora para el bloque de transmisión de datos ya que hoy en la actualidad existe una mayor cobertura de señal en todas las operadoras con respecto a esta tecnología y de esta forma tener una conexión a Internet con datos móviles más estable.

## 6. BIBLIOGRAFÍA

- Acá Cio Id, M., Atkinson, P. W., Silva, J. P., & Franco, A. M. A. (2022). Performance of GPS/GPRS tracking devices improves with increased fix interval and is not affected by animal deployment. *PLOS ONE*, *17*(3), e0265541.  
<https://doi.org/10.1371/JOURNAL.PONE.0265541>
- Alibaba. (2022). *Large Quantity Second Hand Used Good Condition Pax S910 Gprs Pos Terminal Wireless Pos Device - Buy Pax S910,Pax Pos Terminal,S910 Pos Terminal Product on Alibaba.com*. [https://www.alibaba.com/product-detail/large-quantity-second-hand-used-good\\_62152999497.html?spm=a2700.7724857.normal\\_offer.d\\_image.20672760Wt848E](https://www.alibaba.com/product-detail/large-quantity-second-hand-used-good_62152999497.html?spm=a2700.7724857.normal_offer.d_image.20672760Wt848E)
- Amazon. (2015). *Amitek Vehicle Car GPS GSM SMS GPRS Tracking Device(Black)* : *Amazon.in: Electronics*. <https://www.amazon.in/Amitek-Vehicle-Tracking-Device-Black/dp/B00NV1032U>
- Arduino. (2018, February 5). *What is Arduino? | Arduino*.  
<https://www.arduino.cc/en/Guide/Introduction>
- Arduino, 2021. (2021). *Arduino Nano — Tienda Oficial Arduino*.  
<https://store.arduino.cc/products/arduino-nano>
- Azdy, R. A., & Darnis, F. (2020). Use of Haversine Formula in Finding Distance between Temporary Shelter and Waste End Processing Sites. *Journal of Physics: Conference Series*, *1500*(1). <https://doi.org/10.1088/1742-6596/1500/1/012104>
- Bose, A. (2019, January 31). *Global Positioning System (GPS) - Architecture, Applications, Advantages*. <https://electricalfundablog.com/global-positioning-system-gps/>

- Botero Bernal, T. L. (n.d.). *Foro: TENENCIA RESPONSABLE DE MASCOTAS / Revista Colombiana de Ciencia Animal*. Retrieved April 4, 2022, from <http://revistas.ut.edu.co/index.php/ciencianimal/article/view/167>
- Camilo G, A., & Sonia C, O. (n.d.). *La influencia de las mascotas en la vida humana*. Retrieved April 4, 2022, from [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=s0120-06902007000300016](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=s0120-06902007000300016)
- Capdevila, J. (2018). *El Sistema de Posicionamiento Global*.
- Castillo, J., Jiménez, R., & Silva López, R. (2019). *Redes de datos: Contexto y evolución SAMSARA*.
- Chang, M., & Kwang, S. (2017). *Mathematical Algorithms for Two-Dimensional Positioning Based on GPS Pseudorange Technique*.  
<https://www.koreascience.or.kr/article/JAKO201011949346183.pdf>
- Chen, X., & Guo, D. (2016). Public Safety Broadband Network with Rapid-deployment Base Stations. *Wireless Public Safety Networks 2: A Systematic Approach*, 173–198.  
<https://doi.org/10.1016/B978-1-78548-052-2.50006-2>
- Childs, P. R. N. (2019). Design. *Mechanical Design Engineering Handbook*, 1–47.  
<https://doi.org/10.1016/B978-0-08-102367-9.00001-9>
- Clover, J. (2022, February 16). *AirTags: Apple's Item Trackers - Everything We Know - MacRumors*. <https://www.macrumors.com/guide/airtags/>
- Electronilab. (2019). *Módulo GPS Ublox NEO-6M v2 con memoria EEPROM - Electronilab*.  
<https://electronilab.co/tienda/modulo-gps-ublox-neo-6m-v2-con-memoria-eprom/>

ElectroPro. (2020). *MÓDULO GSM SIM800L*.

<http://electropro.pe/image/data/imgProductos/140.%20M%C3%B3dulo%20GSM%20SIM800/SIM800L.pdf>

Fisher, T. (2021, September 27). *What does it mean to benchmark something?*

<https://www.lifewire.com/what-is-a-benchmark-2625811>

García, G. (2017, April 4). *Monitoring lidia cattle with GPS-GPRS technology; a study on grazing behaviour and spatial distribution*. [https://www.medigraphic.com/cgi-](https://www.medigraphic.com/cgi-bin/new/resumenI.cgi?IDREVISTA=39&IDARTICULO=77729&IDPUBLICACION=7523)

[bin/new/resumenI.cgi?IDREVISTA=39&IDARTICULO=77729&IDPUBLICACION=7523](https://www.medigraphic.com/cgi-bin/new/resumenI.cgi?IDREVISTA=39&IDARTICULO=77729&IDPUBLICACION=7523)

Geotab, T. (2022, May 22). *What Is GPS? Fleet Management Experts Answer | Geotab*.

<https://www.geotab.com/blog/what-is-gps/>

Ghayas, A. (2019, November 14). *GSM frequency bands: What frequencies do GSM networks use? – Commsbrief*. <https://commsbrief.com/gsm-frequency-bands-what-frequencies-do-gsm-networks-use/>

Hamilton, T. (2022, April 16). *Modelo V en pruebas de software*. <https://www.guru99.com/v-model-software-testing.html>

Huete, M. (2019). *1 GPRS Ejemplo de Red de Conmutación de Paquetes. - ppt descargar*.

<https://slideplayer.es/slide/14212542/>

Jang, L. (2018). *General Packet Radio Service (GPRS)*.

Jian, W., & Ya, H. (2015). *The Vehicle Terminal Based on GPRS & CDMA Communication*.

*2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable*

*Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*,  
1657–1660. <https://doi.org/10.1109/UIC-ATC-SCALCOM-CBDCOM-IOP.2015.301>

Jörg, E., Christian, B., Hans-Joerg, V., & Christian, H. (2015, March 4). *GSM - Architecture, Protocols and Services - Jörg Eberspächer, Christian Bettstetter, Hans-Joerg Vögel, Christian Hartmann - Google Libros*.

[https://books.google.com.ec/books?id=v6eN1tt9CEUC&pg=PA211&lpg=PA211&dq=GS  
M+Services&source=bl&ots=LFTSJYGYLV&sig=ACfU3U24XMxRdj5PNFK\\_ay6w57P\\_  
mtsTMQ&hl=es&sa=X&ved=2ahUKEwjzMSMi6P3AhUyTTABHWQkBPwQ6AF6BAg  
1EAM#v=onepage&q&f=false](https://books.google.com.ec/books?id=v6eN1tt9CEUC&pg=PA211&lpg=PA211&dq=GS+Services&source=bl&ots=LFTSJYGYLV&sig=ACfU3U24XMxRdj5PNFK_ay6w57P_mtsTMQ&hl=es&sa=X&ved=2ahUKEwjzMSMi6P3AhUyTTABHWQkBPwQ6AF6BAg1EAM#v=onepage&q&f=false)

Lake, V. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, 03(05), 164–173. <https://doi.org/10.4236/JCC.2015.35021>

Laveti, G., Rao, G. S., & Naline, A. (2014). Performance evaluation of standard positioning service GPS receiver with LS and LQ estimators over southern region of Indian sub continent. *2014 International Conference on Science Engineering and Management Research, ICSEMR 2014*. <https://doi.org/10.1109/ICSEMR.2014.7043672>

Lutkevich, B. (2019, November 24). *¿Qué es un Sistema Embebido?*

<https://www.techtarget.com/iotagenda/definition/embedded-system>

McKenna, N. (2021, May 5). *The Relationship Between IoT, Big Data and the Cloud*.

<https://www.mckennaconsultants.com/relationship-between-iot-big-data-and-the-cloud/>

Mehta, H., Kanani, P., & Lande, P. (2019). Google Maps. *Google Maps Article in International Journal of Computer Applications*, 178(8), 975–8887.

<https://doi.org/10.5120/ijca2019918791>

- Ndungu, S. (2021, March 15). *GSM (Global System for Mobile communication)*.  
<https://www.techtarget.com/searchmobilecomputing/definition/GSM>
- Niemelä, T., & Kari, H. (2016). *GPRS - Daoko Baba*. <https://www.doc88.com/p-2743099035883.html?r=1>
- NovAtel. (2020, April 9). *Uses of GPS: What are GPS systems used for? - NovAtel Blog*.  
<https://blog.novatel.com/what-are-gps-systems-used-for/>
- Ortega, A., & Molano, R. (2016). *Sistema prototipo de monitoreo de vehículos mediante GPS y sensores inerciales con soporte en la red GSM/GPRS*.  
<http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1485>
- Oxley, A. (2017). GPS Modernization. *Uncertainties in GPS Positioning*, 71–80.  
<https://doi.org/10.1016/B978-0-12-809594-2.00005-8>
- (PDF) *Embedded Development Boards for Edge-AI: A Comprehensive Report*. (n.d.). Retrieved January 23, 2023, from  
[https://www.researchgate.net/publication/344067021\\_Embedded\\_Development\\_Boards\\_for\\_Edge-AI\\_A\\_Comprehensive\\_Report](https://www.researchgate.net/publication/344067021_Embedded_Development_Boards_for_Edge-AI_A_Comprehensive_Report)
- Ramina, V. (2017). *GSM Architecture*. <https://studylib.net/doc/25394541/gsm-architecture>
- Ranger, S. (2020, February 3). *¿Qué es el IoT? Todo lo que necesitas saber sobre el Internet de las Cosas ahora mismo / ZDNet*. <https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/>
- Ranger, S. (2022, February 25). *¿Qué es la computación en la nube? Todo lo que necesitas saber sobre la nube explicado / ZDNet*. <https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/>

- Recio, L. (2021, December 19). *what are the three basic segments of the gps system - Lisbdnet.com*. <https://lisbdnet.com/what-are-the-three-basic-segments-of-the-gps-system/>
- Roberts, T. (2022, March 1). *How GPS Works (Step-by-Step) | Trakkit*.  
<https://trakkitgps.com/how-gps-works/>
- Rwanda, M. (2021, February 12). *Manager, Network Switching System (NSS) - NDANGIRA*.  
<https://www.ndangira.net/manager-network-switching-system-nss/>
- SIMCom. (2015). *SIM800 Series\_AT Command Manual\_V1.09* .  
[https://www.elecrow.com/wiki/images/2/20/SIM800\\_Series\\_AT\\_Command\\_Manual\\_V1.09.pdf](https://www.elecrow.com/wiki/images/2/20/SIM800_Series_AT_Command_Manual_V1.09.pdf)
- SSR. (2019). *CAP CAP Í Í TULO 5 TULO 5--2 2 GPRS GPRS*.
- Teltech, I. (2020, July 29). *Understanding GSM Frequency Band*.  
<https://teltechinsight.blogspot.com/2020/07/understanding-gsm-frequency-band.html>
- Thakur, D. (2019, April). *What is General Packet Radio Service (GPRS)? - Computer Notes*.  
<https://ecomputernotes.com/computernetworkingnotes/network-technologies/general-packet-radio-service>
- Vasseur, J.-P. (2019). What Are Smart Objects? *Interconnecting Smart Objects with IP*, 3–20.  
<https://doi.org/10.1016/B978-0-12-375165-2.00001-6>
- Vejlgaard, B., Lauridsen, M., Nguyen, H., Kovacs, I. Z., Mogensen, P., & Sorensen, M. (2017). Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT. *IEEE Vehicular Technology Conference, 2017-June*. <https://doi.org/10.1109/VTCSPRING.2017.8108666>
- Wright, G. (2021, March 25). *What is a base station?*  
<https://www.techtarget.com/whatis/definition/base-station>

## 7. ANEXOS

### ANEXO1. Formato de encuestas mediante Entrevista

#### ENCUESTA DIRIGIDA A LAS PERSONAS QUE TRANSITAN CON SU MASCOTA POR LOS DIFERENTES PARQUES DE LA CIUDAD DE IBARRA

La presente encuesta está dirigida a las personas que pasean a sus mascotas en los parques más transitados de la ciudad de Ibarra, con el fin de obtener información sobre los requerimientos de usuario para el desarrollo del sistema PETROUTE.

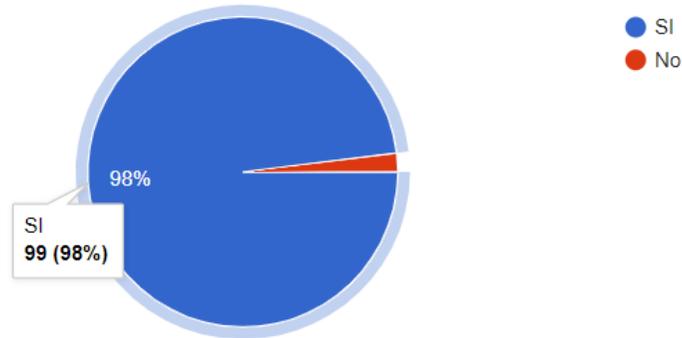
1. ¿Considera usted que se ha incrementado el número de mascotas perdidas dentro de la ciudad de Ibarra?
  - a) Si
  - b) No
2. ¿Por qué medios de información usted conoce casos de mascotas perdidas dentro de la ciudad?
  - a) Redes Sociales
  - b) Periódicos
  - c) Carteles
3. ¿Qué acciones toma cuando se reporta una mascota perdida?
  - a) Compartir la información en Redes Sociales
  - b) Compartir la información en Páginas Web
  - c) Avisar a las autoridades
4. ¿Qué acciones recomienda usar a los dueños de mascotas para evitar esta situación?
  - a) Dar más atención a sus mascotas
  - b) Utilizar una cadena para que no se aparte de su lado.

- c) Tener una placa con su número de teléfono en caso de desaparición
5. ¿Usted conoce de algún dispositivo tecnológico que ayude a rastrear a las mascotas en caso de desaparición?
- a) Si
  - b) No
6. ¿Usted considera útil un dispositivo electrónico que ayude a localizar a las mascotas perdidas dentro de la ciudad?
- a) Si
  - b) No
7. ¿Usted considera que el dispositivo electrónico debe funcionar por medio de GPS?
- a) Si
  - b) No
8. ¿Usted considera que el dispositivo electrónico debe tener indicadores de uso?
- a) Si
  - b) No
9. ¿Usted considera que el aplicativo (aplicación móvil) del dispositivo electrónico debe tener información clara e intuitiva?
- a) Si
  - b) No
10. ¿Usted considera que la pchera del dispositivo electrónico debe ser de bajo costo?
- a) Si
  - b) No

## ▪ Tabulación de Resultados

1. ¿Considera usted que se ha incrementado el número de mascotas perdidas dentro de la ciudad de Ibarra?

101 respuestas



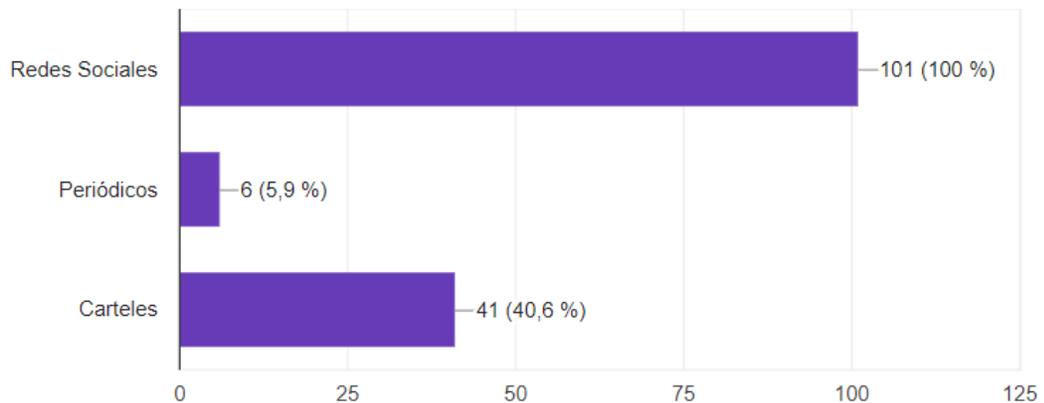
**Ilustración 1.** Estadística de la gráfica – Pregunta 1 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 1 indica que las respuestas por parte de la ciudadanía, se ha incrementado un 98% el número de mascotas perdidas dentro de la ciudad de Ibarra.

2. ¿Por qué medios de información usted conoce casos de mascotas perdidas dentro de la ciudad?

101 respuestas



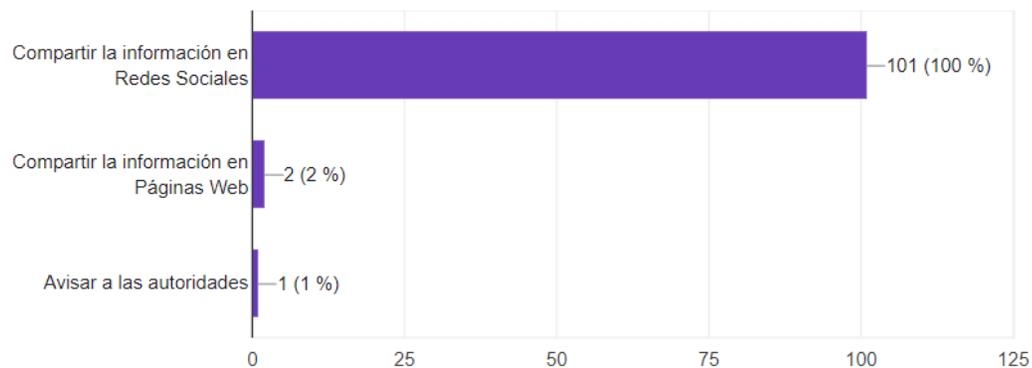
**Ilustración 2.** Estadística de la gráfica – Pregunta 2 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 2 indica que las respuestas por parte de la ciudadanía, un 100 % de los casos de mascotas perdidas se conoce por redes sociales, 5,9% en Periódicos, y un 40,6% en carteles.

### 3. ¿Qué acciones toma cuando se reporta una mascota perdida?

101 respuestas



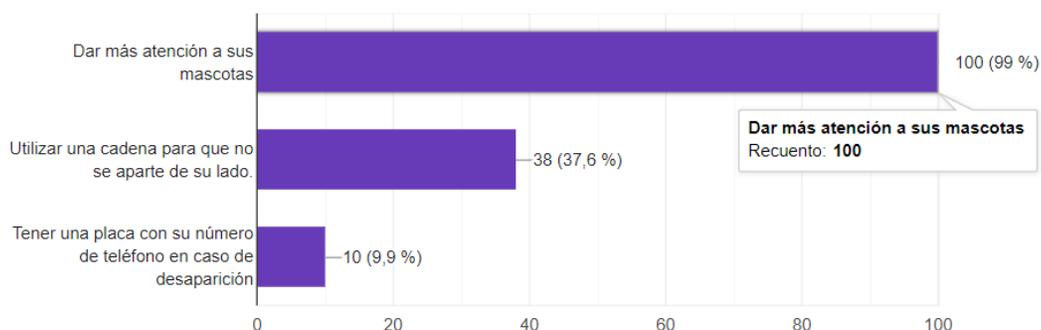
**Ilustración 3.** Estadística de la gráfica – Pregunta 3 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 3 indica que las respuestas por parte de la ciudadanía, un 100 % comparte información en redes sociales cuando se reporta una mascota perdida, un 2% comparte la información en paginas web y un 1% avisa a las autoridades.

4. ¿Qué acciones recomienda usar a los dueños de mascotas para evitar esta situación?

101 respuestas



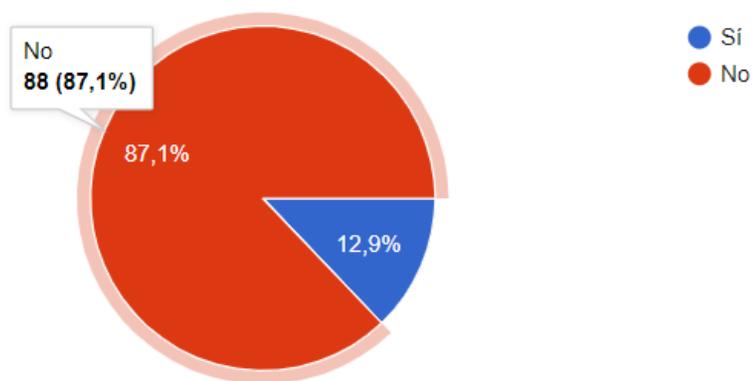
**Ilustración 4.** Estadística de la gráfica – Pregunta 4 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 4 indica que las respuestas por parte de la ciudadanía, un 99 % recomienda a los dueños de mascotas dar más atención a sus mascotas, un 37,6% utiliza una cadena para que no se aparte de su lado y un 9,9% recomienda tener una placa con un número de teléfono en caso de desaparición.

5. ¿Usted conoce de algún dispositivo tecnológico que ayude a rastrear a las mascotas en caso de desaparición?

101 respuestas



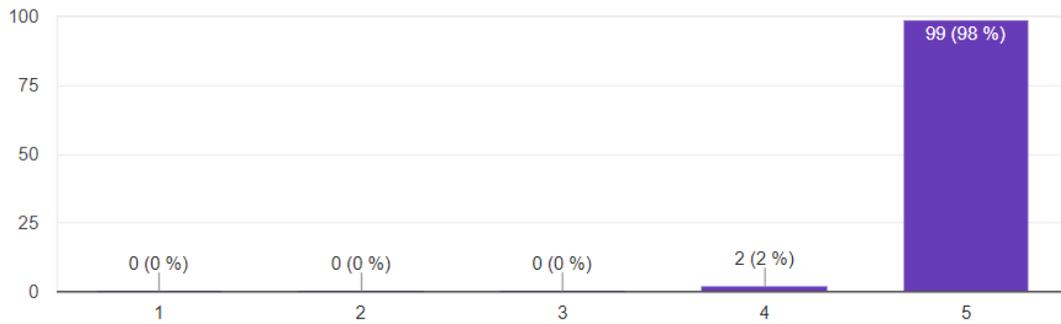
**Ilustración 5.** Estadística de la gráfica – Pregunta 5 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 5 indica que las respuestas por parte de la ciudadanía, un 87,1 % NO conoce de algún dispositivo tecnológico que ayude a rastrear a las mascotas en caso de desaparición y un 12,9% SI conocen.

6. ¿Usted considera útil un dispositivo electrónico que ayude a localizar a las macotas perdidas dentro de la ciudad?

101 respuestas



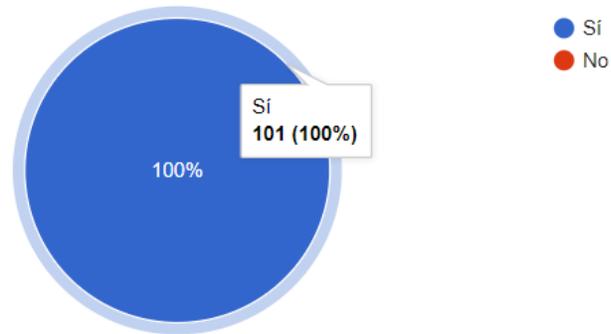
**Ilustración 6.** Estadística de la gráfica – Pregunta 6 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 6 indica que las respuestas por parte de la ciudadanía, dentro de una escala siendo 5 lo más alto un 98 % considera útil un dispositivo electrónico que ayude a localizar a las macotas perdidas dentro de la ciudad y un 2% escogió las escala de 4.

7. ¿Usted considera que el dispositivo electrónico antes mencionado, debe funcionar por medio de GPS(Sistema de localización de objetos)?

101 respuestas



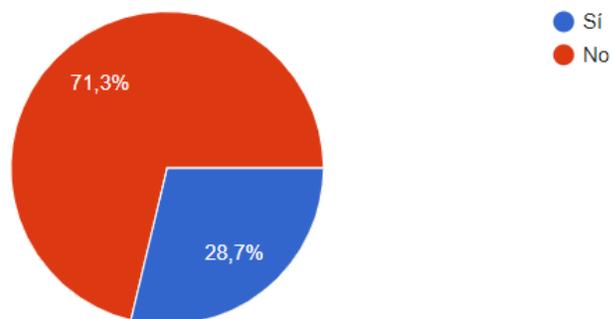
**Ilustración 7.** Estadística de la gráfica – Pregunta 7 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 7 indica que las respuestas por parte de la ciudadanía, el 100 % considera que el dispositivo electrónico, debe funcionar por medio de GPS.

8. ¿Usted considera que el dispositivo electrónico debe tener indicadores de uso?

101 respuestas



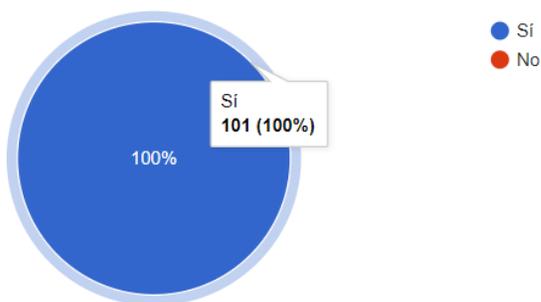
**Ilustración 8.** Estadística de la gráfica – Pregunta 8 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 8 indica que las respuestas por parte de la ciudadanía, el 71,3 % considera que el dispositivo electrónico, NO debe tener indicadores de uso y un 28,7% SI.

9. ¿Usted considera que el aplicativo (aplicación móvil) del dispositivo electrónico debe tener información clara e intuitiva?

101 respuestas



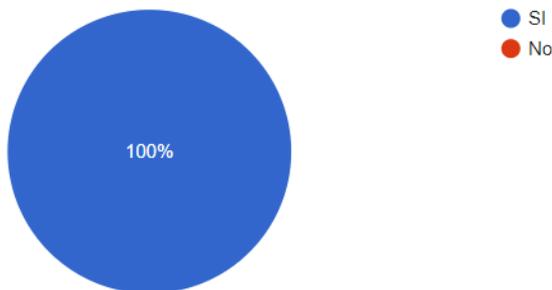
**Ilustración 9.** Estadística de la gráfica – Pregunta 9 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 9 indica que las respuestas por parte de la ciudadanía, el 100 % considera que el aplicativo (aplicación móvil) del dispositivo electrónico debe tener información clara e intuitiva.

10. ¿Usted considera que la pechera del dispositivo electrónico debe ser de bajo costo?

101 respuestas



**Ilustración 10.** Estadística de la gráfica – Pregunta 10 – Encuesta 1

**Fuente:** Autoría

El resultado de la pregunta 10 indica que las respuestas por parte de la ciudadanía, el 100 % considera que la pechera del dispositivo electrónico debe ser de bajo costo.

## **ANEXO2. Formato de encuestas mediante Entrevista**

### **ENCUESTA DIRIGIDA A LOS PROPIETARIOS DE LOS DIFERENTES “PESHOP’S” DE LA CIUDAD DE IBARRA**

La presente encuesta está dirigida a los propietarios de los diferentes Petshop’s de la ciudad de Ibarra, con el fin de obtener información sobre los requerimientos de usuario para el desarrollo del sistema PETROUTE.

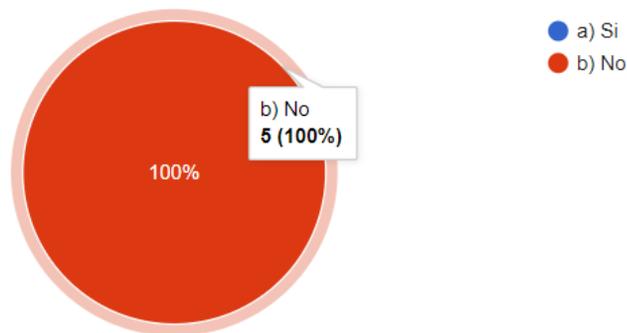
1. ¿Conoce algún sistema o aplicativo web para la geolocalización de mascotas?
  - a) Si
  - b) No
  
2. ¿Conoce de alguna iniciativa que se encuentre implementada o esté en funcionamiento sobre algún sistema dentro de la ciudad de Ibarra para la geolocalización de mascotas?
  - a) Si
  - b) No
  
3. ¿Considera que un aplicativo móvil que permita geolocalizar a una mascota aliviaría la preocupación de los dueños de extraviarlas?
  - a) Si
  - b) No
  
4. ¿Si se desarrollase un sistema electrónico con aplicativo web con características de geolocalización que es lo que esperaría del sistema?

- a) Amigable con el usuario
  - b) Fácil de Usar
  - d) Que solo permita ver donde se ubican las mascotas
  - e) Que el sistema informe si la mascota se ha perdido.
5. ¿Usted consideraría que el sistema de geolocalización, a través de una geocerca definida, le alerte mediante mensajes de texto si su mascota ha salido de un rango de distancia?
- a) Si
  - b) No

▪ **Tabulación de Resultados**

1. ¿Conoce algún sistema o aplicativo web para la geolocalización de mascotas?

5 respuestas



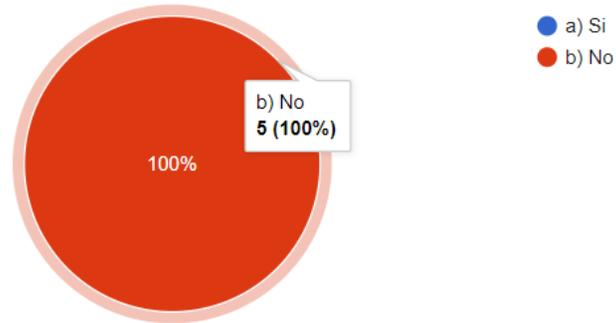
**Ilustración 11.** Estadística de la gráfica – Pregunta 1 – Encuesta 2

**Fuente:** Autoría

El resultado de la pregunta 1 indica que las respuestas por parte de los PET SHOPS de la ciudad de Ibarra, el 100 % NO conoce algún sistema o aplicativo web para la geolocalización de mascotas.

2. ¿Conoce de alguna iniciativa que se encuentre implementada o esté en funcionamiento sobre algún sistema dentro de la ciudad de Ibarra para la geolocalización de mascotas?

5 respuestas



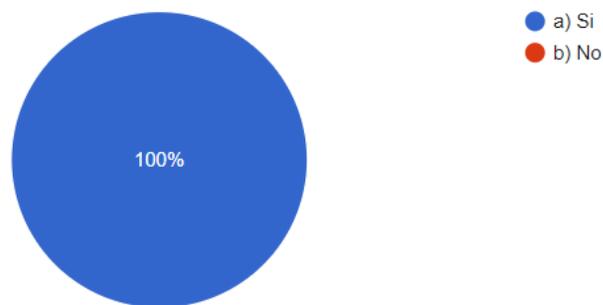
**Ilustración 12.** Estadística de la gráfica – Pregunta 2 – Encuesta 2

**Fuente:** Autoría

El resultado de la pregunta 2 indica que las respuestas por parte de los PET SHOPS de la ciudad de Ibarra, el 100 % NO conoce de alguna iniciativa que se encuentre implementada o esté en funcionamiento sobre algún sistema dentro de la ciudad de Ibarra para la geolocalización de mascotas.

3. ¿Considera que un aplicativo móvil que permita geolocalizar a una mascota aliviaría la preocupación de los dueños en el caso que se extravíen?

5 respuestas



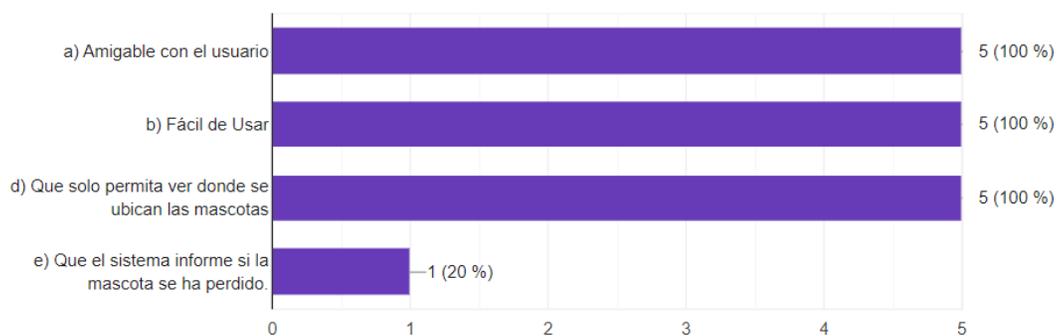
**Ilustración 13.** Estadística de la gráfica – Pregunta 3 – Encuesta 2

**Fuente:** Autoría

El resultado de la pregunta 3 indica que las respuestas por parte de los PET SHOPS de la ciudad de Ibarra, el 100 % considera que un aplicativo móvil que permita geolocalizar a una mascota aliviaría la preocupación de los dueños en el caso de que se extravíen las mascotas.

4. ¿Si se desarrollase un sistema electrónico con aplicativo web con características de geolocalización que es lo que esperaría del sistema?

5 respuestas



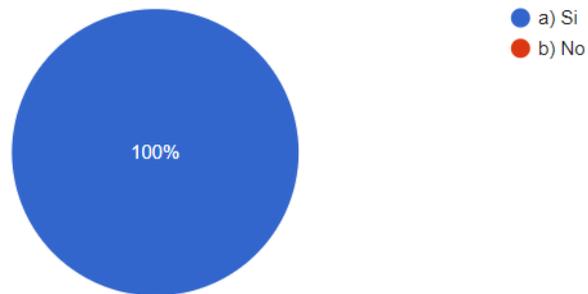
**Ilustración 14.** Estadística de la gráfica – Pregunta 4 – Encuesta 2

**Fuente:** Autoría

El resultado de la pregunta 4 indica que las respuestas por parte de los PET SHOPS de la ciudad de Ibarra, un 100 % espera que sea amigable con el usuario, un 100% espera que sea fácil de usar, un 100% de igual forma espera que se permita ver donde se ubican las mascotas y un 20% espera que el sistema informe si la mascota se ha perdido.

5. ¿Usted consideraría que el sistema de geolocalización, a través de una geocerca definida, le alerte mediante mensajes de texto si su mascota ha salido de un rango de distancia?

5 respuestas



**Ilustración 15.** Estadística de la gráfica – Pregunta 5 – Encuesta 2

**Fuente:** Autoría

El resultado de la pregunta 5 indica que las respuestas por parte de los PET SHOPS de la ciudad de Ibarra, el 100 % considera que el sistema de geolocalización, a través de una geocerca definida, le alerte mediante mensajes de texto si su mascota ha salido de un rango de distancia

### **ANEXO3. Código Fuente de toma de Datos Procesamiento y Transmisión en Arduino**

```
//                                UNIVERSIDAD TECNICA DEL NORTE
//                                FICA - CIERCOM
//                                PROYECTO DE TITULACIÓN
//                                FREDY CHICAIZA

#define TINY_GSM_MODEM_SIM800
#define TINY_GSM_SIM800

/*****Librerias del modulo GPS*****/
#include <TinyGPS++.h>
/*****Librerias del modulo GSM*****/
#include <TinyGsmClient.h>
#include <ArduinoHttpClient.h>
/*****Libreria para la comunicación entre pines*****/
#include <SoftwareSerial.h>

/*****Distribución de Parámetros y Configuración*****/
```

```

#define SerialMon Serial//Establecer serial para la consola de depuración (al
monitor serie, velocidad predeterminada)
#define DEVICE_ID "sim800gps00001"
#define modemBAUD 9600
#define gpsBAUD 9600
SoftwareSerial SerialAT(3, 2); //Comunicación por comandos AT
SoftwareSerial ss(9, 8); // Comunicación Serial

/*****Parámetros para el APN*****/
const char apn[] = "internet.movistar.com.ec"; // APN
const char user[] = "movistar";// Usuario del APN
const char pass[] = "movistar";// Contraseña del APN

/*****Parámetros de Firebase*****/
const char server[] = "petroute6-a0592-default-rtdb.firebaseio.com";//
Dirección del Servidor
const int port = 443;// Numero de Puerto(HTTPS)
const String UPDATE_PATH = "gps_devices/"+ String(DEVICE_ID); // Tabla raiz de
firebase

/*****Asignación de variables globales*****/
int count=0;
String fireData="";

/*****Definir la consola serie para impresiones de depuración
*****/
#ifdef DUMP_AT_COMMANDS//Ver todos los comandos AT, si se lo desea
#include <StreamDebugger.h>//Librería de Depuración, para mejora de
comunicación
StreamDebugger debugger(SerialAT, SerialMon);// Depuración, comunicación AT
y Monitor Serie
TinyGsm modem(debugger);
#else
TinyGsm modem(SerialAT);//Creación de una solicitud de modem Tiny GSM
#endif

TinyGsmClientSecure client(modem);//Creación de una solicitud, Cliente seguro
HttpClient https(client, server, port);//Solicitud HTTPS hacia el servidor web

TinyGPSPlus gps;//Objeto gps

void setup() {
/*****Establecer la velocidad de transmisión de la consola *****/
SerialMon.begin(9600);
delay(10);
initializeModem(); // comprobar la comunicación del módem
prueba(); // probar conexión a internet
ss.begin(gpsBAUD); // abrir el serial Gps
}

/*****Escaneo de coordenadas GPS *****/
void loop() {

if(ss.isListening()){
while(fireData.equals("")){

```

```

        scan();}
    }
    else{
        fireData="";
        ss.begin(gpsBAUD); //Comunicacion Serial abierta
    }
}

/*****Envio de datos a Firebase *****/
void sendData(const char* method, const String & path , const String & data,
Httpclient* http) {

    http->connectionKeepAlive(); // Conexión persistente HTTP
    String url;
    if (path[0] != '/') {
        url = "/";
    }
    url += path + ".json";
    url += "?print=silent";
    url += "&x-http-method-override=";
    url += String(method);
    String contentType = "application/json";
    http->post(url, contentType, data);

    /*****Lectura del código de estado y el cuerpo de la respuesta *****/
    int statusCode = http->responseStatusCode();
    http->responseBody();//http respuesta
    Serial.print("Status code: ");
    Serial.println(statusCode);//Estado del codigo
    if (!http->connected()) {
        Serial.println();
        http->stop();// Apagar
        Serial.println("HTTP POST disconnected");
        SerialAT.begin(modemBAUD);//Puerto Serie AT abierto
        prueba();//Conexión de Prueba
        fireData="";
        ss.begin(gpsBAUD);//Comunicación serial Abierta
    }
}

/*****Escanear la ubicación GPS *****/
void scan(){
    while (ss.available() > 0)
        if (gps.encode(ss.read()))
            displayInfo();
}
void displayInfo()
{
    if (gps.location.isValid())
    {
        data();
    }
    else
    {
        data();
    }
}

```

```

    SerialAT.begin(modemBAUD);
    updateData();
}

/*****Conexión a Internet *****/
void prueba(){
    test:
    SerialMon.print(F("Conexión a la red..."));
    if (!modem.waitForNetwork()) {
        SerialMon.println("Not");
        delay(300);
        goto test;
    }
    SerialMon.println("OK");
    SerialMon.print(F("Conexión a APN: "));
    SerialMon.print(apn);
    if (!modem.gprsConnect(apn, user, pass)) {
        SerialMon.println("Not");
        delay(300);
        goto test;
    }
    SerialMon.println("OK");
}

/*****Actualizar base de Datos *****/
void updateData(){
    infoModem();
    sendData("PUT", UPDATE_PATH, fireData, &https);
}

/*****Inicializacion del Modem SIM800L *****/
void initializeModem(){
    SerialAT.begin(modemBAUD); //Puerto Serie AT abierto
    delay(3000);
    SerialMon.println(F("Iniciando modem..."));
    modem.restart(); //Reinicio

    String modemInfo = modem.getModemInfo(); //Obtención de los datos del Modulo
    SerialMon.print(F("Modem: "));
    SerialMon.println(modemInfo); //Información del Modulo
    if (!modem.hasSSL()) {
        SerialMon.println(F("SSL no es compatible con este módulo"));
        while(true) { delay(1000); }
    }
    Serial.println("Por favor esperar 10 segundos");
    delay(10000);
}

/*****Obtención de Datos GPS *****/
void data(){
    fireData="";
    fireData += "{";
    fireData += " \"lat\" : \"\\\"\" + String(gps.location.lat(), 6)+"\\\"\", ";
    fireData += " \"lng\" : \"\\\"\" + String(gps.location.lng(), 6)+"\\\"\", ";
}

/*****Obtención de Datos del Módulo GSM *****/

```



### 3. Package Information

#### 3.1. Pin out Diagram

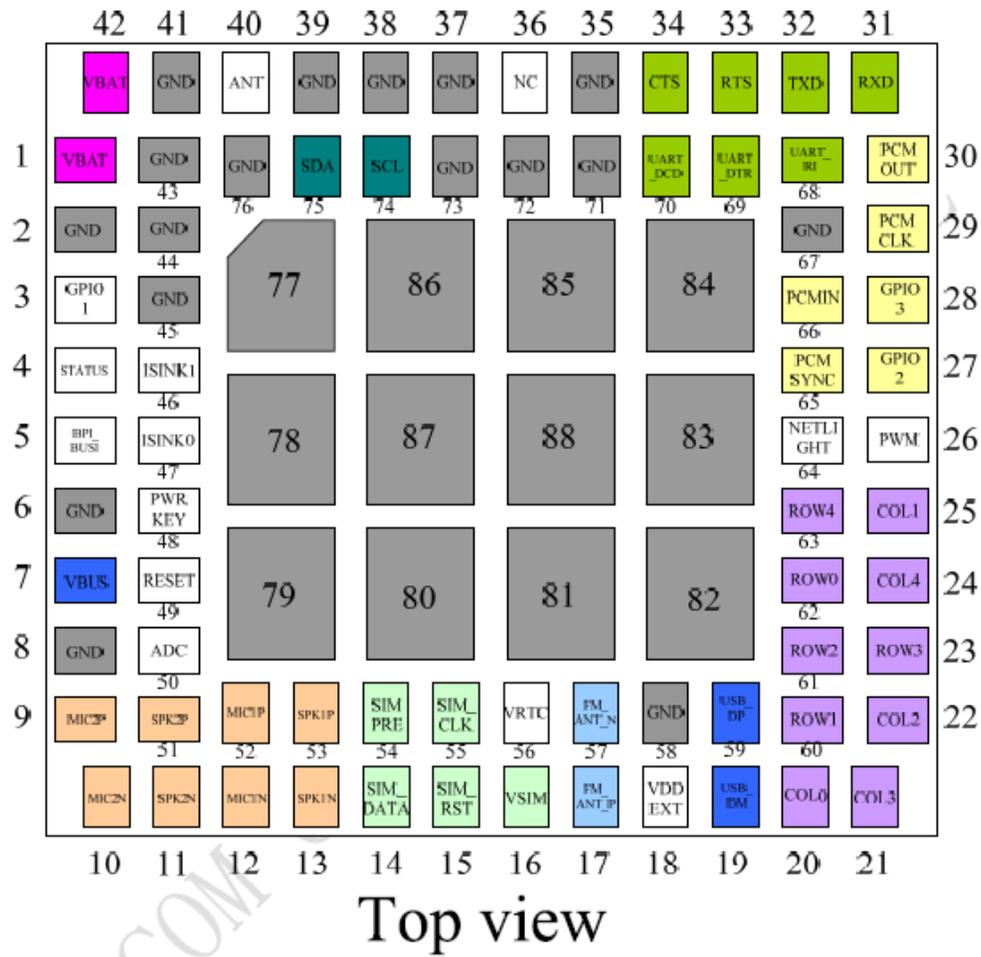


Figure 2: SIM800L pin out diagram (Top view)

### 3.3. Package Dimensions

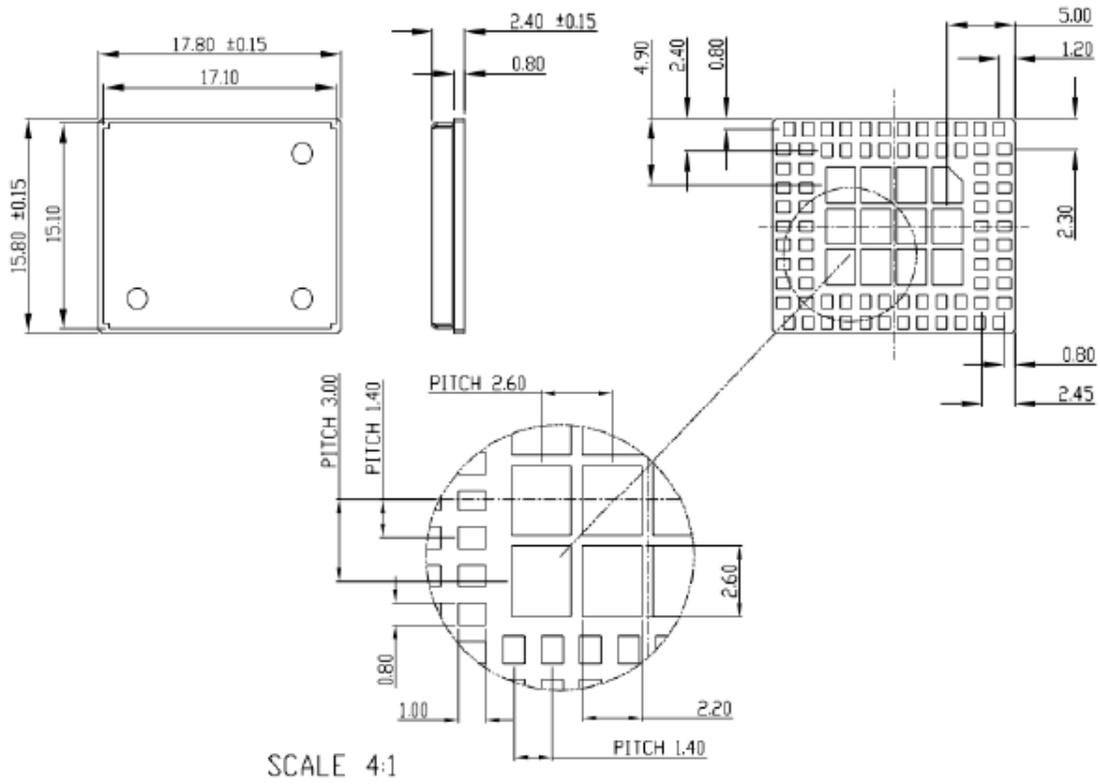


Figure 3: Dimensions of SIM800L (Unit: mm)

SIMCOM CONFIDENTIAL

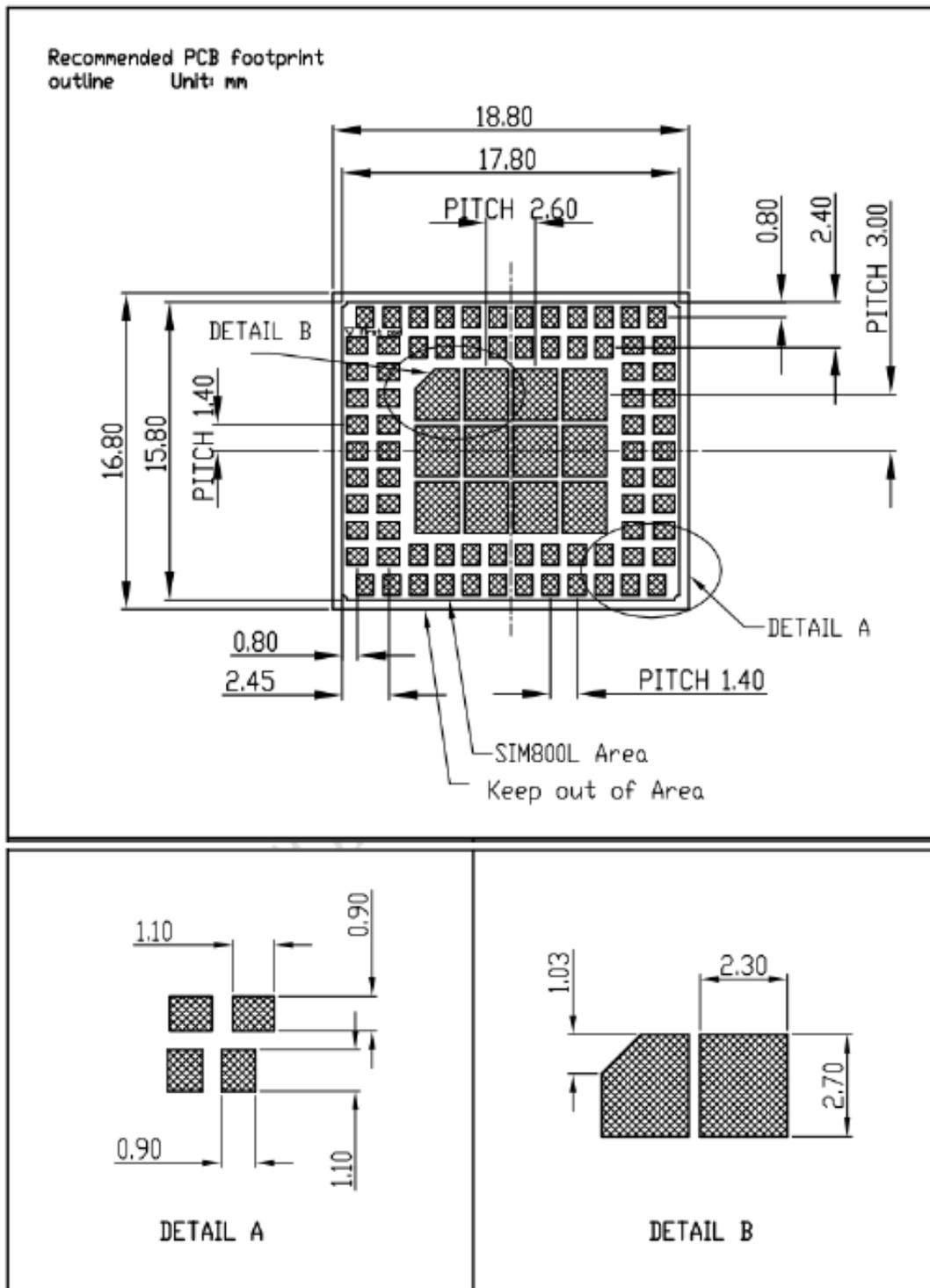


Figure 4: Recommended PCB footprint outline (Unit: mm)

▪ **MODULO GPS NEO6M V2**



**1.3 GPS performance**

Parameter	Specification		
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS, GAGAN		
Time-To-First-Fix <sup>1</sup>		NEO-6G/Q	NEO-6M
	Cold Start (Autonomous)	29 s	32s
	Warm Start (Autonomous)	29 s	32s
	Hot Start (Autonomous)	<1 s	<1 s
	Aided Starts <sup>2</sup>	<1 s	<3 s
Sensitivity <sup>3</sup>		NEO-6G/Q	NEO-6M
	Tracking & Navigation	-160 dBm	-160 dBm
	Reacquisition	-160 dBm	-160 dBm
	Cold Start (Autonomous)	-147 dBm	-146 dBm
Maximum Navigation update rate		5Hz	
Horizontal position accuracy <sup>4</sup>	Autonomous	2.5 m	
	SBAS	2.0 m	
Configurable Timepulse frequency range		0.1 Hz to 1 kHz	
Velocity accuracy		0.1m/s	
Heading accuracy		0.5 degrees	
Operational Limits	Dynamics	≤ 4 g	
	Altitude <sup>5</sup>	50,000 m	
	Velocity <sup>5</sup>	500 m/s	

**Table 2: NEO-6 GPS performance**

<sup>1</sup> All satellites at -130 dBm  
<sup>2</sup> Dependent on aiding data connection speed and latency  
<sup>3</sup> Demonstrated with a good active antenna  
<sup>4</sup> Under good GPS signal conditions  
<sup>5</sup> Assuming Airborne <4g platform

## 2 Pin Definition

### 2.1 Pin assignment

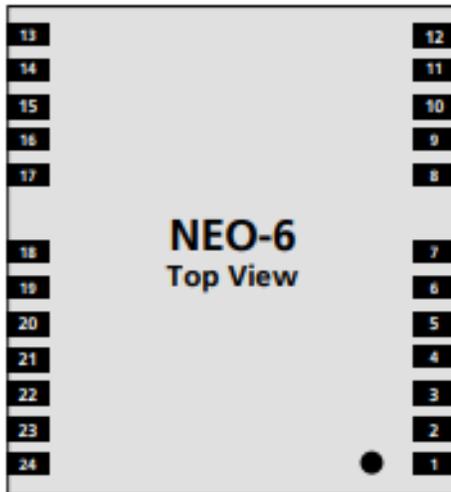


Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SPI Slave Select
3	All	TIMEPULSE	O	Time pulse (1PPS)
4	All	EXTINTD	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage RF section Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MOSI/CFG_COM0	O/I	SPI MOSI / Configuration Pin. Leave open if not used.
15	All	MISO/CFG_COM1	I	SPI MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPS0SCK	I	Power Mode Configuration Pin / SPI Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	DDC Data
19	All	SCL2	I/O	DDC Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1