

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



TESIS DE GRADO

**“PROTOTIPO DE COLLAR PARA MONITOREO DE
MASCOTAS USANDO GEOLOCALIZACIÓN BASADA EN
TECNOLOGÍAS GPS”**

**Tesis de Grado para obtener el Título de Licenciatura en Informática
Mención Ingeniería de Sistemas Informáticos**

POR: REYNEL SANCHEZ CHUQUIMIA

TUTOR: M.SC. GROVER ALEX RODRIGUEZ RAMIREZ

LA PAZ – BOLIVIA

2021



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y
NATURALES CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

Esta tesis va dedicado a mi padre Rene Sanchez, por haber sacrificado gran parte de su vida para formarme y educarme, por todo el cariño, amor y comprensión que siempre me brindo, por el apoyo constante hasta el último día de su vida.

A mi madre Luisa Chuquimia, por los impresionantes esfuerzos que siempre realizo. Me ha educado, me has brindado cuidados, atención y apoyo la mejor madre del mundo, no me encontraría aquí de no ser por tu ayuda, tu compañía y tu amor.

A mi hermano Marco, quien a lo largo de mi vida ha sido como un padre más para mi he aprendido mucho de ti, sin tu apoyo y cariño no estaría aquí.

A mi cuñada Patty, quien se portó como una hermana para mí agradezco el apoyo brindado así mí.

AGRADECIMIENTOS

Agradezco a Dios quien es el guía de mi camino y está en cada paso que doy, por su infinita bondad y haberme dado salud para lograr mis objetivos, por darme también la oportunidad de concluir mi carrera universitaria y permitirme dar un paso más en mi vida profesional.

Por supuesto a mi familia, son ustedes un modelo de constancia y de lucha, gracias por estar siempre apoyándome, quiero hacerles saber que son y siempre serán mi principal motivación.

Agradecer a mi tutor M.Sc. Grover Alex Rodríguez Ramírez, por la dedicación y colaboración en el desarrollo de la presente tesis.

A mis amigos de la Carrera de Informática, con quienes estuvimos juntos por este camino de aprendizaje.

A la Universidad Mayor de San Andrés, por acogerme y brindarme muchas oportunidades. A cada uno de los docentes que intervinieron en mi formación como profesional a lo largo de la carrera.

reynelsanchez70@gmail.com

RESUMEN

El prototipo de collar para mascotas permite visualizar la última posición en la cual se encontrará nuestras mascotas, esta información es presentada a través de una aplicación móvil que proporciona como nivel visual y en tiempo real utilizando Google Maps. la información es solicitada enviando un SMS mediante la aplicación al prototipo en cuestión y de esta manera este mismo responde enviando la longitud y latitud de su ubicación actual en un mismo SMS al número solicitante, el cual es controlado por el dispositivo móvil y reflejado en un *widget*. El presente prototipo busca enfocarse de la manera más concisa con la precisión para ello cabe destacar dos puntos de suma importancia. El punto de ubicación y Seguimiento en tiempo real, los cuales nos permiten tener de una manera más precisa la ubicación deseada.

Las mascotas en estos tiempos se volvieron un integrante más de las familias y es por ese motivo que el cuidado y el poder verificar su ubicación se han vuelto tan importantes para todos, así de esta manera poder evitar extravíos, robos o pérdidas de nuestras queridas mascotas.

La tecnología del sistema de posicionamiento global está permitiendo grandes cambios en la sociedad. Las aplicaciones que usan GPS crecen constantemente y cada vez están siendo más indispensables en nuestra vida cotidiana. Es por eso que aprovecharemos sus bajos costos y su evolución constante para utilizarlo a nuestro favor en este prototipo de geolocalización.

Palabras clave: Prototipo, aplicación móvil, mascotas, geolocalización, GPS.

Metodología: Para la realización de esta tesis se utilizaron las metodologías de Mobil-D, Metodología V y Tropos.

ABSTRACT

The pet collar prototype allows us to visualize the last position in which our pets will be found, this information is presented through a mobile application that provides as a visual level and in real time using Google Maps. The information is requested by sending an SMS through the application to the prototype in question and in this way it responds by sending the longitude and latitude of its current location in the same SMS to the requesting number, which is controlled by the mobile device and reflected in a widget. The present prototype seeks to focus in the most concise way with precision. For this, it is worth highlighting two points of great importance. The location point and Real-time tracking, which allow us to have the desired location in a more precise way.

Pets in these times have become one more member of families and that is why care and being able to verify their location have become so important for everyone, so in this way we can avoid misplacement, theft or loss of our beloved pets.

Global Positioning System technology is enabling great changes in society. Applications that use GPS are constantly growing and are becoming more and more indispensable in our daily lives. That is why we will take advantage of its low costs and its constant evolution to use it to our advantage in this geolocation prototype.

Keywords: Prototype, mobile application, pets, geolocation, GPS.

Methodology: To carry out this thesis, the methodologies of Mobil-D, Methodology V and Tropos were used.

Contenido

CAPITULO I	1
1. MARCO REFERENCIAL	1
1.1. INTRODUCCIÓN.....	1
1.2. PROBLEMA.....	2
1.2.1. ANTECEDENTES	2
1.2.2. TRABAJOS SIMILARES	3
1.2.3. LISTA DE PROBLEMAS	5
1.2.4. FORMULACIÓN DEL PROBLEMA.....	5
1.3. OBJETIVOS	5
1.3.1. OBJETIVO GENERAL.....	5
1.3.2. OBJETIVOS ESPECÍFICOS.....	5
1.4. HIPÓTESIS	5
1.5. JUSTIFICACIONES	6
1.5.1. SOCIAL.....	6
1.5.2. ECONÓMICA	6
1.5.3. TECNOLÓGICA.....	6
1.5.4. CIENTÍFICA	6
1.6. ALCANCES, LÍMITES Y APORTES	7
1.6.1. ALCANCES	7
1.6.2. LIMITES	7
1.6.3. APORTES	7
1.6.3.1. PRÁCTICO.....	7
1.6.3.2. TEÓRICO	8
1.7. METODOLOGÍA	8
1.7.1. METODOLOGÍA DE DESARROLLO	8
1.7.2. METODOLOGÍA CIENTÍFICA.....	8
1.7.3. METODOLOGÍA DE DESARROLLO DEL SOFTWARE LIBRE Y HARDWARE.....	9
CAPITULO II	10
2. MARCO TEÓRICO	10
2.1. INGENIERÍA DEL SOFTWARE	10

2.2.	MASCOTAS	11
2.3.	METODOLOGIA V	11
2.4.	METODOLOGÍA MOBILE-D.....	13
2.4.1.	FASE DE EXPLORACIÓN	14
2.4.1.1.	ESTABLECIMIENTO DE INTERESADOS.....	14
2.4.1.2.	DEFINICIÓN DEL ALCANCE.....	15
2.4.1.3.	ESTABLECIMIENTO DE PROYECTOS	15
2.4.1.4.	ROLES.....	16
2.4.2.	FASE DE INICIALIZACIÓN	16
2.4.2.1.	PUESTA EN MARCHA DEL PROYECTO	16
2.4.2.2.	PLANIFICACIÓN INICIAL.....	16
2.4.2.3.	EL DÍA DE PRUEBA.....	17
2.4.2.4.	ROLES.....	17
2.4.3.	FASE DE PRODUCCIÓN.....	17
2.4.3.1.	DÍA DE LA PLANIFICACIÓN	18
2.4.3.2.	DÍA DE TRABAJO	18
2.4.3.3.	DÍA DE LANZAMIENTO	19
2.4.3.4.	ROLES.....	20
2.4.4.	FASE DE ESTABILIZACIÓN.....	20
2.4.4.1.	ROLES	20
2.4.5.	FASE DE PRUEBAS.....	21
2.4.5.1.	ROLES.....	21
2.5.	TROPOS	21
2.5.1.	DEFINICIÓN DE TÉRMINOS.....	21
2.5.2.	ANÁLISIS TEMPRANO DE REQUERIMIENTOS	25
2.5.3.	ANÁLISIS DE REQUERIMIENTOS TARDÍOS	25
2.5.4.	DISEÑO ARQUITECTURAL	26
2.5.5.	DISEÑO DETALLADO.....	26
2.6.	SISTEMA DE GEOLOCALIZACIÓN.....	26
2.6.1.	FUNDAMENTOS DE LA TECNOLOGÍA GPS	27
2.6.2.	TRILATERACIÓN DEL SATÉLITE	27
2.7.	ARDUINO.....	28

2.7.1.	HARDWARE ARDUINO.....	30
2.7.2.	COMPONENTES.....	30
2.7.2.1.	ARDUINO LILYPAD.....	30
2.7.2.2.	A9G DE AI THINKER.....	31
2.7.2.3.	SIM CARD.....	32
2.8.	APLICACIÓN MÓVIL.....	35
2.9.	GEOREFERENCIACIÓN.....	35
2.9.	SISTEMA DE POSICIONAMIENTO GLOBAL.....	36
2.10.	TECNOLOGÍA CELULAR Y GPS.....	37
2.10.1.	FUNCIONAMIENTO DEL GPS.....	38
2.10.2.	GPS DIFERENCIAL.....	40
2.10.2.1.	SEGMENTO ESPACIAL.....	41
2.10.2.2.	SEGMENTO DE CONTROL.....	41
2.10.2.3.	SEGMENTO DE USUARIO.....	43
2.10.2.4.	INSTRUMENTOS GPS TRACKINGS.....	46
2.11.	GEOLOCALIZACIÓN.....	47
CAPITULO III.....		48
3.	DISEÑO METODOLÓGICO.....	48
3.1.	METODOLOGÍA V, MOBILE D Y TROPOS.....	48
3.2.	DEFINICIÓN DE ESPECIFICACIONES.....	51
3.2.1.	ACTORES.....	51
3.2.2.	OBJETIVOS Y DEPENDENCIAS.....	52
3.2.3.	DIAGRAMA DE ACTORES.....	52
3.3.	DISEÑO GLOBAL.....	52
3.3.1.	ALCANCE.....	52
3.3.2.	ESTABLECIMIENTO DE PROYECTO.....	53
3.3.3.	PLANIFICACIÓN.....	53
3.4.	DISEÑO EN DETALLE.....	54
3.4.1.	CONFIGURACIÓN.....	54
3.4.2.	PLANIFICACIÓN.....	57
3.5.	IMPLEMENTACIÓN.....	58
3.5.1.	PLANIFICACIÓN.....	58

3.5.2.	DISEÑO DETALLADO.....	59
3.6.	TEST UNITARIO	60
3.6.1.	TAREA 1	60
3.6.2.	TAREA 2	62
3.6.3.	TAREA 3	64
3.7.	INTEGRACIÓN.....	68
3.7.1.	INTEGRACIÓN COMPLETA DEL PROTOTIPO DE COLLAR CON TECNOLOGÍA DE GEOLOCALIZACIÓN	68
3.8.	TEST OPERACIONAL DEL SISTEMA	71
3.8.1.	PRUEBAS.....	71
CAPITULO IV		73
4.	EVALUACION DE RESULTADOS.....	73
4.1.	ANÁLISIS Y RESULTADOS	73
4.1.1.	CÁLCULO DE DATOS.....	73
4.1.1.1.	DISTANCIA MÁS CORTA ENTRE DOS PUNTOS	77
4.1.1.2.	CÁLCULO ESTADÍSTICO.....	79
4.1.2.	ANÁLISIS DE RESULTADOS.....	82
CAPÍTULO V		83
5.	CONCLUSIONES Y RECOMENDACIONES.....	83
5.1.	CONCLUSIONES.....	83
5.1.1.	COSTO DEL PROTOTIPO DE COLLAR CON GEOLOCALIZACIÓN	83
5.2.	RECOMENDACIONES.....	84
BIBLIOGRAFÍA.....		85
ANEXOS		90

INDICE DE FIGURAS

Figura 2.1 Gasto promedio en las mascotas del hogar	11
Figura 2.2 Metodología V	12
Figura 2.3 Metodología mobile d.....	14
Figura 2.4 Actor en tropos	22
Figura 2.5 Agente en tropos	22
Figura 2.6 Rol en tropos.....	22
Figura 2.7 Posición en tropos.....	23
Figura 2.8 Meta dura y suave en tropos	23
Figura 2.9 Plan en tropos	23
Figura 2.10 Recurso en tropos	23
Figura 2.11 Medios - fin en tropos.....	24
Figura 2.12 Dependencia de meta dura en tropos	24
Figura 2.13 Dependencia de recurso en tropos	24
Figura 2.14 Dependencia de plan en tropos	24
Figura 2.15 Dependencia de meta suave en tropos	25
Figura 2.16 Intersección de dos esferas es un círculo	28
Figura 2.17 Tres esferas se interceptan en dos puntos.....	28
Figura 2.18 Arduino Lilypad.....	31
Figura 2.19 A9G de ai thinker.....	32
Figura 2.20 Sim Card	34
Figura 2.21 Sistema de posicionamiento global	37
Figura 2.22 Cambios de posición.....	40
Figura 2.23 Segmento espacial	41
Figura 2.24 Estaciones de control	43
Figura 2.25 Antena gps	44
Figura 2.26 Receptor gps	44
Figura 2.27 Terminal GPS	44
Figura 2.28 Segmento de usuario.....	45
Figura 3.1: Metodología combinada mobile d y tropos	48
Figura 3.2: Metodología combinada mobile d, tropos y metodología v.....	49
Figura 3.3: Diagrama de actores	52

Figura 3.4 Diagrama de metas de actores	57
Figura 3.5 Interfaz de Aplicación.....	60
Figura 3.6 Diagrama de Actividades de Actores	60
Figura 3.7 Diagrama de Componentes.....	61
Figura 3.8 Arduino Lilypad conectado con A9G de ai	62
Figura 3.9 Primera Pantalla “Inicio”	63
Figura 3.10 Segunda pantalla “Proceso”.....	64
Figura 3.11 Verificación de Aplicación	66
Figura 3.12 Consumo de las coordenadas por la aplicación	65
Figura 3.13 Configuración de Widget.....	66
Figura 3.14 Integración Completa del Dispositivo	69
Figura 3.15 Diagrama lógico de Arduino Lilypad con A9G de ai y su batería 5V	69
Figura 4.1 Región critica.....	80
Figura 4.2 Tabla distribución normal.....	80
Figura 4.3 Distribución z para la toma de decisión.....	81

INDICE DE TABLAS

Tabla 2.1 Lanzamientos de bloques de satélites.....	39
Tabla 3.1 Cronograma de desarrollo	54
Tabla 3.2 Clasificación de requisitos	58
Tabla 3.3 Tareas de desarrollo	58
Tabla 3.4 Prueba de conexión Arduino Lilypad y A9G de ai	59
Tabla 3.5 Desarrollo de la aplicación móvil.....	61
Tabla 3.6 Prueba de conexión aplicación con prototipo collar	64
Tabla 3.7 Prueba de Integración Completa del Prototipo de collar.....	68
Tabla 3.8 Documento de Resultados Finales	71
Tabla 4.1 Coordenadas google eart pro y conversión a unidades utm.....	75
Tabla 4.2 Coordenadas del Prototipo de collar y Conversión a unidades UTM.....	77
Tabla 4.3 Distancia más corta entre dos puntos	79
Tabla 5.1 Costo del prototipo de collar con geolocalización.	84

CAPITULO I

1. MARCO REFERENCIAL

1.1. INTRODUCCIÓN

Actualmente una gran parte de la población fue tomando a las mascotas como uno más de la familia. Su salud y bienestar es nuestra preocupación, así que procedemos a integrarlos en la sociedad identificándolos al igual que nosotros. Para ello, los llevamos al veterinario para que le hagan la pertinente revisión, le pongan sus respectivas vacunas y demás.

Por otro lado, en la actualidad existen diversos sistemas de monitoreo y rastreo satelital aplicados en diferentes ámbitos como usos militares, transportes terrestres, aéreos, marítimos, uno de los más utilizados es el sistema de posicionamiento global “GPS”¹ que por décadas ha sido útil a la sociedad.

El GPS es un sistema que indica la posición exacta de algún objeto en cualquier lugar del planeta, funciona captando la señal de satélites a través de un receptor el cual fija la posición de un objeto, lo realiza por medio de triangulación, la ubicación es entregada en coordenadas terrestres representadas en grados de longitud y latitud, se necesitan mapas para establecer gráficamente la ubicación y visualización correcta del punto geográfico, a medida que el objeto se mueve a las coordenadas cambian.

GPS aumenta la productividad a través de una amplia franja de la economía, para incluir agricultura, construcción, minería, topografía, entrega de paquetes, y gestión de cadenas de suministros logísticos. Las principales redes de comunicaciones, sistemas bancarios, los mercados financieros, y las redes de energía dependen en gran medida de GPS para la sincronización de tiempo preciso. Algunos servicios inalámbricos no pueden funcionar sin ella. La calidad y el alcance a diferencia de otras tecnologías es universal a más de que es un sistema en uso en casi la mayoría de países y la base a los satélites que prestan dicho servicio es libre para fines no militares.

1 GPS: El Sistema de Posicionamiento Global (GPS) es un sistema de radionavegación de los Estados Unidos de América, basado en el espacio, que proporciona servicios fiables de posicionamiento, navegación, y cronometría gratuita e ininterrumpidamente a usuarios civiles en todo el mundo.

El presente trabajo tiene como objetivo demostrar tanto el funcionamiento como la importancia de aplicar esta tecnología en mascotas, dando a conocer la ubicación del animal en caso de extraviarse. Debido al incremento de animales domésticos en la ciudad y a nivel nacional, se presenta la necesidad de aplicar esta tecnología en esta área, de tal manera beneficiando a los propietarios de mascotas.

1.2. PROBLEMA

1.2.1. Antecedentes

Una pequeña cápsula de cristal especial del tamaño de un grano de arroz que contiene un código único hace de un microchip una herramienta para evitar accidentes y extravíos de mascotas.

La implantación del microchip² será subcutánea en la región dorsal del cuello. Dependerá del tamaño de la mascota que sea atendida en la unidad para ver si se utiliza anestesia local.

Se estima que, en la ciudad de La Paz, sede de gobierno, existen al menos 200 mil animales callejeros y que muchos de ellos cuentan con dueños que los descuidan y los dejan fuera de casa.

Esa situación provocó que la Unidad de Atención Integral de Animales, dependiente del Gobierno Municipal de La Paz, capacite a su personal para ofrecer la incorporación de estos chips en gatos y perros.

La lectura de la información que tiene integrado el microchip se realiza mediante un lector de RFID (*Radio Frequency IDentification*), que es un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tagsRFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas Auto ID (*Automatic IDentification* o identificación automática).

2 MICROCHIP: Pastilla pequeña de material semiconductor, de algunos milímetros cuadrados de área, sobre la que se fabrican circuitos electrónicos.

“Este microchip servirá para tener un registro de todas las mascotas que están en nuestro municipio. Vamos a comenzar con una cantidad específica para ver el funcionamiento del software”, según la veterinaria de la Unidad de Atención Integral de Animales, (Claros,2020).

El microchip permitirá a la unidad municipal, conocida también como la Casa de la Mascota, contar con una base de datos informática para identificar a los gatos y canes. El control ayudará a saber desde las características principales hasta las vacunas que los animales reciben.

1.2.2. Trabajos similares

En este punto se dará a conocer estudios e investigaciones que servirán de ayuda a la elaboración de este trabajo investigativo, también se mencionara tecnologías ofertadas en el mercado mundial a cerca de rastreo de mascotas, las cuales ayudarán a tener una visión para desarrollar el presente trabajo investigativo.

- “GPS mascotas una alternativa tecnológica para perros y gatos” (Robledo, 2015) En dicha investigación se concluye que hay un alto interés por parte de las personas en adquirir un dispositivo de rastreo, también se pudo conocer la falta de conocimiento y manejo de esta tecnología, lo cual nos da una pauta muy importante ya que es un punto para tomar en cuenta y mejorarlo en la investigación de esta tesis.
- “Diseño de un equipo de rastreo satelital de elementos usando tecnologías GPS y GSM describe el funcionamiento y las diferentes etapas de construcción del prototipo” (Henao,2014) .Cumpliendo con el objetivo principal de la fabricación de un equipo que permita la ubicación, seguimiento y monitoreo, por medio de tecnología satelital (GPS) y tecnología móvil (GSM), los resultados de este trabajo concluyen con la posibilidad de aprovechar la tecnología de punta para crear equipos más pequeños y que puedan ser usados en áreas del cuerpo sin que sean pesados o muy grandes, así aprovechándose para otros fines según la necesidad requerida.
- “Sistema GPS utilizado en Kenia para evitar la caza de leones” (Griggs, 2011) Describe el funcionamiento de unos collares GPS que calculan la ubicación exacta de leones, enviando un mensaje de texto a un servidor el cual automáticamente genera un correo

electrónico con las coordenadas, posteriormente se agregan los datos un mapa satelital de código abierto donde se muestra el desplazamiento del león.

Finalmente, mediante una revisión de las tesis realizadas en la Carrera de Informática de la Universidad Mayor de San Andrés no se encontraron investigaciones que tengan relación con el tema en estudio, sin embargo, se pueden encontrar diversos proyectos y tesis de grado que se basan en ayudar a localizar a alguien que esté en peligro de perderse. Entre éstas, mencionamos las siguientes:

- “Sistematización del modelo de concientización sobre tenencia responsable de mascotas dirigido a voluntarios de asociaciones en la ciudad de la paz” (León, 2016) Esta tesis fue realizada con el fin de sistematizar el modelo de concientización de Paulo Freire, sobre tenencia responsable de mascotas, que permita a los voluntarios de las asociaciones en defensa de los animales capacitarse. El Sistema de Capacitación facilita a los voluntarios que adquieran conciencia sobre el tema.
- “Geolocalización³ para niños con autismo” (Aruni, 2015) Una de las principales preocupaciones para los padres de un niño con autismo es que el hijo se escape o se pierda. En una reciente encuesta en internet conducida por la NAA, un impresionante 92% de los padres respondió que su hijo estaba en riesgo de perderse. Este es un problema que debe ser enfrentado en cada ciudad y pueblo del país.
- “Sistema web para el registro y control de mascotas” (Alvarez, 2014) Con el sistema se debe poder registrar a los propietarios y sus mascotas, tanto como las denuncias de la población en general sobre algún incidente con algún animal, además de la colaboración a la población en general con los servicios de listas de mascotas desaparecidas y encontradas y que los propietarios puedan tener a la mano la información de su mascota.
- “Geolocalización basada en tecnologías GPS a personas con enfermedad de Alzheimer” (Monroy, 2018) El prototipo de esta aplicación web permite visualizar la última posición y el recorrido, también es posible ver los datos guardados con anterioridad del portador del GPS familiar con la enfermedad de Alzheimer, esta información es presentada a través

3 GEOLOCALIZACIÓN: La **geolocalización** consiste en obtener la ubicación geográfica de un objeto como puede ser un teléfono móvil, un coche o una calle.

de una página web que proporciona como nivel visual y en tiempo real Google Maps, la información de las posiciones está resguardada en base de datos que contiene nombre día, fecha y hora. El presente prototipo busca enfocarse de la manera más concisa con la precisión para ello cabe destacar tres puntos de suma importancia. El punto de ubicación, Seguimiento en tiempo real, Ubicaciones y/o anteriores registros.

1.2.3. Lista de Problemas

- Robos de mascotas debido a la raza y por ende al lucro de ellos.
- Abandono de las mascotas por propietarios irresponsables.
- Pérdida de las mascotas por descuido de los propietarios o por que la mascota es muy inquieta y tiende a desaparecer por largos lapsos de tiempo.
- Al momento de ser encontrada la mascota por una persona ajena, es casi imposible encontrar al dueño sin la información adecuada.

1.2.4. Formulación del problema

¿De qué manera se puede determinar la posición actual de la mascota?

1.3. OBJETIVOS

1.3.1. Objetivo General

Desarrollar un prototipo de collar para el monitoreo de mascotas usando una placa A9G de AI thinker con geolocalización basada en tecnología GPS.

1.3.2. Objetivos específicos

- ✓ Analizar el funcionamiento de dispositivos GPS para el rastreo de mascotas.
- ✓ Establecer una conexión de la aplicación con el prototipo de collar con tecnología GPS.
- ✓ Determinar los beneficios de la utilización de GPS y SIM para el cuidado de mascotas sin necesidad de una conexión a internet.
- ✓ Demostrar la efectividad del sistema de posicionamiento global utilizando la placa “A9G de ai thinker” aplicado a mascotas.

1.4. HIPÓTESIS

El prototipo de collar con geolocalización basada en tecnologías GPS permite rastrear la ubicación de una MASCOTA, con un margen de error menor igual a 3 metros.

1.5. JUSTIFICACIONES

1.5.1. Social

En el ámbito social con el desarrollo de este prototipo de collar inteligente para monitoreo de mascotas se evitará la pérdida de estos mismo, así las familias podrán cuidar más a estos pequeños seres queridos.

1.5.2. Económica

Actualmente existen un sin número de actividades que en base a servicios y aplicaciones nos permiten interactuar con los beneficios de la tecnología del internet y de las comunicaciones mundiales y satelitales, dicha tecnología está siendo usada por el público en general, dirigiremos el estudio de factibilidad del dispositivo el cual será enfocará hacia el público dueño de mascotas. Se usará con una aplicación gratuita que es intuitiva y fácil de uso, es por eso que, el presente desarrollo del prototipo del collar es económicamente justificable debido a que la aplicación será realizada en su totalidad con herramientas de software libre, esto hace que el producto solo dependa de los costos materiales necesarios.

1.5.3. Tecnológica

Hoy en día el acceso a un dispositivo móvil (*Smartphone*⁴) es ya considerada una necesidad, porque a diferencia de años atrás cuando era considerado como un objeto de lujo, es por eso que esta investigación se decidió hacer en base a una aplicación móvil para que esta pueda ser distribuida a la mayor cantidad de personas y con un collar el cual es indispensable para una mascota, pero con la diferencia que esta contara con un dispositivo GPS controlado por Arduino.

1.5.4. Científica

El impacto que puede tener esta investigación será debido a que forma intereses de las ciencias de la informática en ayuda a la sociedad, precisamente porque irán de la mano la metodología móvil junto con la disposición de una placa de ARDUINO acompañado de un dispositivo GPS, de tal manera que pueden ser estudiados y aplicados para la creación de más herramientas tecnológicas ayudando a las personas.

4 SMARTPHONE: Teléfono celular con pantalla táctil, que permite al usuario conectarse a internet, gestionar cuentas de correo electrónico e instalar otras aplicaciones y recursos a modo de pequeño computador.

1.6. ALCANCES, LÍMITES Y APORTES

1.6.1. Alcances

Se elaborará un prototipo de collar inteligente para monitoreo de mascotas usando geolocalización, además de una aplicación móvil de geolocalización personal para la seguridad que merecen las mascotas. Se explicará de la mejor manera la aplicabilidad del mismo, el cual permitirá al tutor o cuidador contar con un instrumento específico, que facilite emitir resultados sobre ellos.

Contará con los siguientes procedimientos:

- Identificar el lugar donde está situada la mascota.
- Rastrear el o los lugares por donde se encontraba.
- Mostrar las rutas más cortas y sin congestión para llegar donde los mencionados.
- El alcance geográfico del presente trabajo de investigación abarca el País de Bolivia y más específicamente la ciudad de La Paz.
- La aplicación no podrá tener uso si no se cuenta con un dispositivo móvil que tenga como sistema operativo: Android.
- Los usuarios deben aprender a manejar el dispositivo móvil y el colocado del collar.

1.6.2. Límites

- Los usuarios y las personas que deseen ubicar a sus mascotas tendrán que tener acceso a Internet.
- Debido a que la información a ser manejada es privada, no tendrá conexión a redes sociales.
- La plataforma ofrece servicios de geolocalización para dispositivos móviles.

1.6.3. Aportes

1.6.3.1. Práctico

El prototipo de collar inteligente para monitoreo de mascotas, será un gran aporte a la sociedad, debido a que es una herramienta para la seguridad de nuestras mascotas y a reducir los peligros de pérdida o hurto. También mejorar de una manera y/o concientizar a la población en la tenencia responsable de mascotas.

1.6.3.2. Teórico

Se busca que el resultado permita la incorporación del modelo de concientización en una herramienta tecnológica mediante la sistematización, una vez sistematizado el modelo, debe contribuir a dar solución al problema de la investigación de implementar el prototipo de collar inteligente para monitoreo de mascotas usando geolocalización con GPS.

1.7. METODOLOGÍA

1.7.1. Metodología de desarrollo

La investigación a realizar será de la exploratoria, debido a que se desarrollara una representación del fenómeno estudiado que es la pérdida o hurto de mascotas, a la descriptiva esta será debido trabaja sobre realidades de hecho y sus características y se dará una correcta ubicación de las mascotas en caso de pérdida o hurto.

La metodología de investigación a implementar es el método científico porque es la única forma de conocer la verdad objetiva del prototipo de collar inteligente tomando en cuenta la observación, hipótesis, experimentación, teoría, prueba.

El tipo de metodología a utilizar será la Mobile D y Metodología V debido a la comunicación, simplicidad, productividad, calidad y flexibilidad del desarrollo y la elaboración del prototipo de collar para monitoreo de mascotas usando geolocalización basada en tecnologías GPS.

1.7.2. Metodología científica

Para el desarrollo de la investigación se implementará la metodología de investigación científica y deductiva planteada por Mario Bunge identificamos las siguientes etapas:

- **Observación:** Se realizará un análisis de los equipos de localización en el cual analizaremos el funcionamiento mecánico actual para poder implementar el nuevo sistema de control.
- **Planteamiento de la hipótesis:** Se propone implementar Arduino para lograr la automatización del prototipo de collar para mascotas.
- **Diseño de la aplicación:** En esta etapa se hará la estructuración de la metodología para guiar la construcción del sistema basado en Arduino.

- **Casos de prueba:** Se basará en los postulados teóricos de la ingeniería del software, para tal verificación se aplicará las pruebas de caja negra.
- **Conclusiones:** En esta etapa final se mencionarán los resultados obtenidos y recomendaciones para nuevos temas de investigación, en base al conocimiento obtenido en el trabajo actual.

1.7.3. Metodología de desarrollo del software libre y Hardware

Para el desarrollo del prototipo se implementó el modelo en v. El cual se desarrolló para terminar con algunos de los problemas que se vieron utilizando el enfoque de cascada tradicional.

Los defectos estaban siendo encontrados demasiado tarde en el ciclo de vida, ya que las pruebas no se introducían hasta el final del proyecto.

El modelo en v dice que las pruebas necesitan empezarse lo más pronto posible en el ciclo de vida. También muestra que las pruebas no son sólo una actividad basada en la ejecución.

Hay una variedad de actividades que se han de realizar antes del fin de la fase de codificación. Estas actividades deberían ser llevadas a cabo en paralelo con las actividades de desarrollo, y los técnicos de pruebas necesitan trabajar con los desarrolladores y analistas de negocio de tal forma que puedan realizar estas actividades y tareas y producir una serie de entregables de pruebas.

Los productos de trabajo generados por los desarrolladores y analistas de negocio durante el desarrollo son las bases de las pruebas en uno o más niveles. El modelo en v es un modelo que ilustra cómo las actividades de prueba (verificación y validación) se pueden integrar en cada fase del ciclo de vida. Dentro del modelo en v, las pruebas de validación tienen lugar especialmente durante las etapas tempranas, por ejemplo, revisando los requisitos de usuario y después, por ejemplo, durante las pruebas de aceptación de usuario.

El modelo en v es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Describe las actividades y resultados que han de ser producidos durante el desarrollo del producto. La parte izquierda de la v representa la descomposición de los requisitos y la creación de las especificaciones del sistema. El lado derecho de la v representa la integración de partes y su verificación. V significa “Validación y Verificación”.

CAPITULO II

2. MARCO TEÓRICO

2.1. INGENIERÍA DEL SOFTWARE

La ingeniería del software es una disciplina de ingeniería cuya meta es el desarrollo costeable de sistemas del software. Este es abstracto e intangible. No está restringido por materiales, o gobernado por leyes físicas o procesos de manufactura, de alguna forma esto simplifica la ingeniería del software ya que no existen limitaciones físicas del potencial del software. Sin embargo, la falta de restricciones naturales significa que el software puede llegar a ser extremadamente complejo, y por lo tanto muy difícil de entender.

La experiencia previa en la construcción de estos sistemas mostro que un enfoque informal para el desarrollo del software no era muy bueno. Los grandes proyectos a menudo tenían años de retraso. Costaban mucho más de lo presupuestado, eran irrealizables, difíciles de mantener y con un desempeño pobre. El desarrollo de software estaba en crisis. Los costos del hardware tambaleaban mientras que los del software se incrementaban con rapidez. Se necesitaban nuevas técnicas y métodos para controlar la complejidad inherente a los sistemas grandes.

Estas nuevas técnicas han llegado a ser parte de la ingeniería del software y son ampliamente utilizadas. Sin embargo, cuanto más crezca nuestra capacidad de producir software, también la hará la complejidad de los sistemas de software solicitados. Las nuevas tecnologías resultantes de la convergencia de las computadoras y de los sistemas de comunicación y complejas interfaces graficas de usuario impusieron nuevas demandas de ingenieros de software. Debido a que muchas compañías no aplican de forma efectiva las técnicas de ingeniería del software, demasiados proyectos todavía producen software que es irrealizable, entregando tarde y sobre presupuestado.

La amplia diversidad de diferentes tipos de sistemas y organizaciones que usan sistemas significa que necesitamos una diversidad de enfoques al desarrollo de software. Sin embargo, las nociones fundamentales de proceso y organización de sistema son base de todas estas técnicas y estas son la esencia de la ingeniería del software. (Sommerville, 2005)

2.2. MASCOTAS

Un animal de compañía, mascota o animal doméstico, es un animal domesticado que se conserva con el propósito de brindar compañía o para el disfrute del cuidador. A diferencia de los animales de laboratorio, animales para la crianza o ganado, animales para el transporte o animales para el deporte, los animales de compañía no son conservados para traer beneficios económicos o alimenticios, aunque sí un beneficio personal.

Los animales de compañía son seleccionados por su comportamiento, adaptabilidad y por su interacción con los humanos, en la que posiblemente se utilicen como herramientas de caza o seguridad. Poseer un animal de compañía puede traer beneficios a la salud, ya que el cuidado diario hace olvidar a muchas personas otro tipo de preocupaciones, evita la depresión y nos hace sentir útiles. Los animales de compañía más populares son el perro y el gato.

El animal de compañía también se caracteriza por ser amigable. Se puede decir que la mayoría de estos animales se convierten en domésticos si la persona con la que conviven les enseña buenos modales en la convivencia diaria, no olvidando su naturaleza animal.

Un animal de compañía se adapta al ambiente de un ser humano sin importar su naturaleza: se adhiere a un grupo, familia o sociedad, la cual le dota de todo lo necesario para su desarrollo. (Diario mascota, 2002)



Figura 2.1 Gasto promedio en las mascotas del hogar

Fuente:

2.3. METODOLOGIA V

Consiste en V en los niveles lógicos del 1 al 4, para cada fase del desarrollo, existe una fase

correspondiente o paralela de verificación o validación. Esta estructura obedece al principio de que para cada fase del desarrollo debe existir un resultado verificable.

En la misma estructura se advierte también que la proximidad entre una fase del desarrollo y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V. La longitud de esta separación intenta ser proporcional a la distancia en el tiempo entre una fase y su homóloga de verificación.

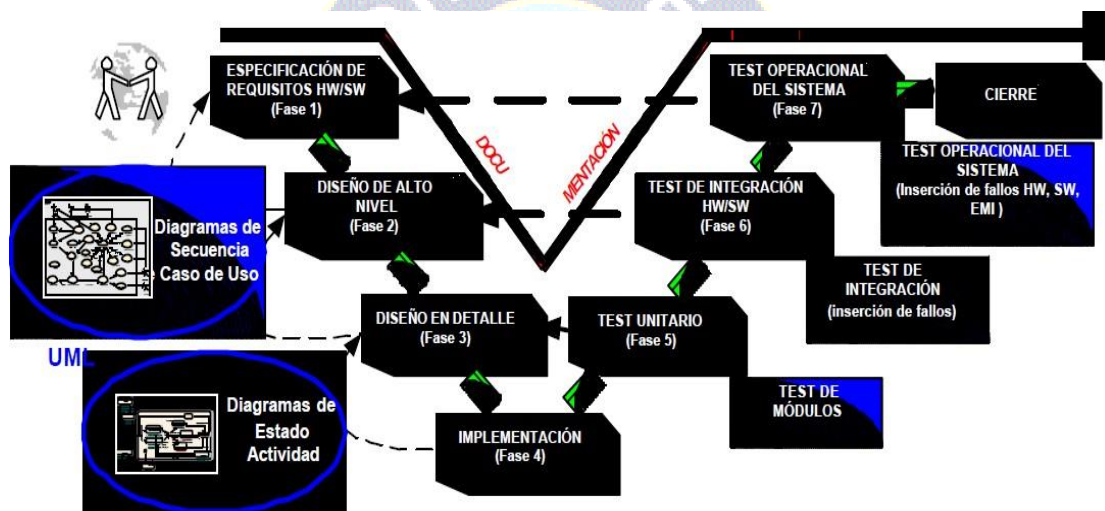


Figura 2.2 Metodología V

Fuente: (Pérez et. al., 2006)

El modelo en V define las siguientes etapas de desarrollo:

- **Definición De Especificaciones (Fase 1):** Está orientado al “cliente”. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones.
- **Diseño Global (Fase 2):** También llamado diseño de alto nivel. Su objetivo es obtener un diseño y visión general del sistema.
- **Diseño En Detalle (Fase 3):** Consiste en detallar cada bloque de la fase anterior. Define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.
- **Implementación (Fase 4):** Es la fase en la que se materializa el diseño en detalle.
- **Test Unitario (Fase 5):** En esta fase se verifica cada módulo HW y SW de forma unitaria,

comprobando su funcionamiento adecuado.

- **Integración (Fase 6):** En esta fase se integran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas. Por una parte, se debe comprobar en todo el sistema el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, debe verificarse que ante la presencia de un fallo persiste el funcionamiento correcto. Se comprueba el cumplimiento de los requisitos establecidos.
- **Test Operacional Del Sistema (Fase 7):** Se realizan las últimas pruebas, pero sobre un escenario real, en su ubicación final, anotando una vez más las pruebas realizadas los resultados obtenidos. (Pérez et. al., 2006)

2.4. METODOLOGÍA MOBILE-D

El método Mobile-D se desarrolló junto con un proyecto finlandés en el 2004. Fue realizado, principalmente, por investigadores de la VTT (Instituto de Investigación Finlandés) y, a pesar de que es un método antiguo, sigue en vigor (se está utilizando en proyectos de éxito y está basado en técnicas que funcionan).

Se trata de método basado en soluciones conocidas y consolidadas: *Extreme Programming* (XP), *Crystal Methodologies* y *Rational Unified Process* (RUP), XP para las prácticas de desarrollo, Crystal para escalar los métodos y RUP como base en el diseño del ciclo de vida. (Ramírez., 2010)

Las fases son las siguientes:

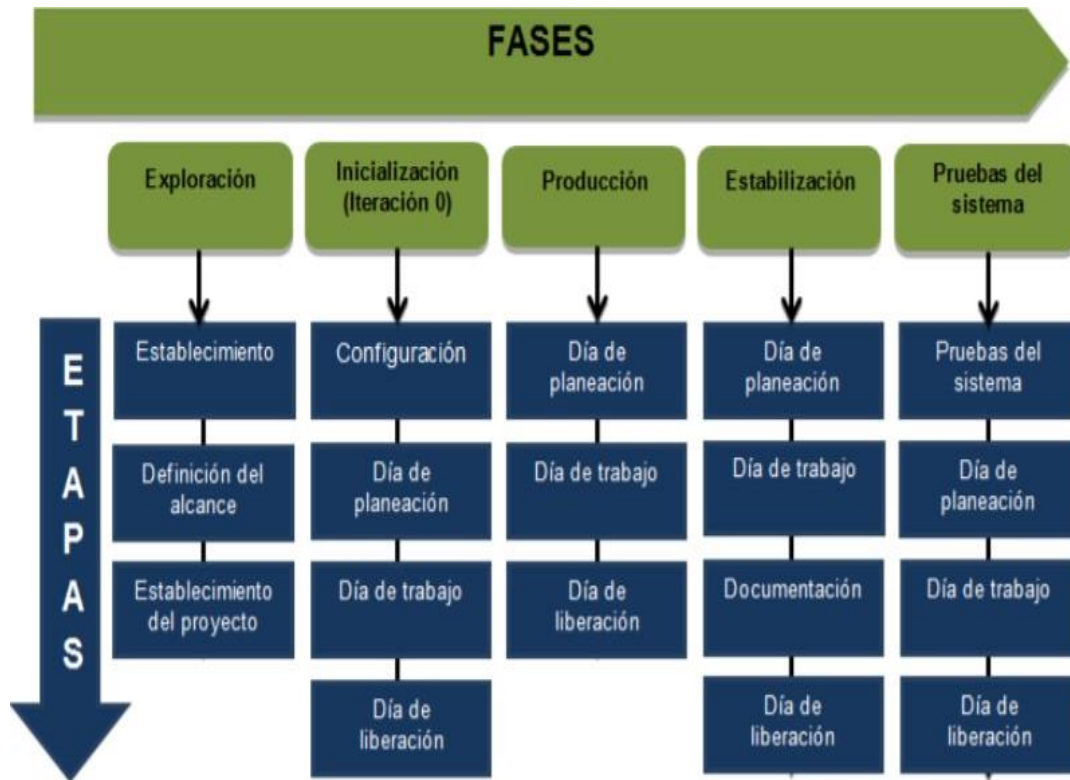


Figura 2.3 Metodología Mobile D

Fuente: (Ramírez., 2010)

2.4.1. Fase de exploración

El equipo de desarrollo debe generar un plan y establecer las características y los conceptos básicos que están alrededor de todo el proyecto. Este proceso se realiza en tres etapas: Establecimiento actores, definición del alcance y el establecimiento de proyectos. En esta fase inicial incluyen a los clientes que toman parte activa en el proceso de desarrollo, la planificación inicial del proyecto, los requisitos y el establecimiento de procesos. (Gómez & Hernández, 2016).

2.4.1.1. Establecimiento de interesados

El propósito de esta etapa es identificar y establecer los grupos de interés que serán necesarios en diversas tareas. Amplia variedad de conocimientos y la cooperación que se necesita para planificar de forma controlada y efectiva aplicación del producto de software. (Gómez &

Hernández, 2016).

a) Establecimiento de actores

El propósito de esta tarea es reconocer que el grupo de clientes tiene la experiencia, conocimiento del dominio y de los requisitos para el desarrollo del producto software.

2.4.1.2. Definición del alcance

El propósito de esta etapa es definir los objetivos para el proyecto incipiente con respecto tanto los contenidos, así como la línea de tiempo del proyecto.

a) Planeación inicial del proyecto: El propósito de esta tarea es establecer el plan inicial de los procesos de desarrollo de software con respecto a una línea de tiempo, el ritmo y las inversiones del proyecto.

b) Conjunto de requisitos iniciales: El propósito de esta tarea es la de producir una definición general inicial del alcance del producto, el propósito y la funcionalidad. Esto es necesario la etapa de establecimiento del proyecto (tamaño, cuestiones técnicas, arquitectura, entre otros.).

2.4.1.3. Establecimiento de proyectos

El propósito de esta etapa es definir y asignar los recursos (tanto técnicos como humanos) que se necesita para que el proyecto de desarrollo de software inicie. También se establece la línea de base para el proyecto, esta es una tarea importante de esta etapa. (Gómez & Hernández, 2016).

a) Definición de la arquitectura

El propósito de esta tarea es definir suficientemente los problemas de arquitectura para que el proyecto se pueda realizar con éxito.

b) Proceso de establecimiento

Esta se construye desarrollando una serie de pasos. (Gómez & Hernández, 2016):

- Establecer la línea de proceso base
- Planificar la documentación necesaria
- Planificar el seguimiento y la medición
- Identificar las necesidades de formación

2.4.1.4. Roles

Las siguientes funciones están relacionadas con la Fase de Exploración. (Arias, 2014):

- Equipo del proyecto
- Grupo de apoyo
- Grupo de Cliente / Clientes
- Grupo de Dirección
- Equipo de Exploración.

2.4.2. Fase de inicialización

Se preparan los planes para las siguientes fases y se establece el entorno técnico como los recursos físicos, tecnológicos y de comunicaciones, incluyendo el entrenamiento del equipo de desarrollo. Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida (opcional). (Gómez & Hernández, 2016).

2.4.2.1. Puesta en marcha del proyecto

El propósito de esta etapa es la configuración de los recursos físicos y técnicos para el proyecto, así como el medio para el seguimiento del proyecto, capacitar al equipo de proyecto como es necesario, y establecer las formas específicas de proyectos para comunicarse con el grupo de clientes. (Gómez & Hernández, 2016).

a) Configuración del entorno

El procedimiento muy general y depende en gran medida del proyecto y del ambiente donde se va a desarrollar el proyecto, por lo tanto, en Mobile D este procedimiento no es especificado.

b) Formación

El procedimiento muy general y depende en gran medida del proyecto y del ambiente donde se va a desarrollar el proyecto, por lo tanto, en Mobile D este procedimiento no es especificado.

c) Establecer comunicación con el cliente

El propósito de esta tarea es llegar a un acuerdo sobre cómo el jefe de proyecto y el equipo se comunicarán con el cliente durante el desarrollo de software.

2.4.2.2. Planificación inicial

El propósito del modelo de planificación inicial es obtener una buena comprensión general del producto a desarrollar, para preparar y perfeccionar los planes para las próximas fases del proyecto y preparar planes de comprobación y resolución de todas las cuestiones fundamentales del desarrollo final de la fase actual. (Gómez & Hernández, 2016).

a) Planeamiento de la arquitectura

El propósito de esta tarea es preparar los problemas críticos de la arquitectura de manera que estén a disposición cuando se haga la implementación de los requisitos del cliente.

b) Análisis de los requisitos iniciales

El propósito de esta tarea es priorizar y analizar los requisitos. La búsqueda de un conjunto de requisitos obliga a crear componentes más importantes e interfaces del sistema. Un esqueleto arquitectónico de trabajo no debe ser encontrado después que el final de la primera iteración.

2.4.2.3. El día de prueba

El propósito de esta etapa es de prueba y para asegurarse de que todo está listo para la implementación del software. Además, el propósito es implementar algunas funciones al núcleo del sistema (por ejemplo, la comunicación cliente - servidor) o resolver algún problema crítico de desarrollo sin producir ningún código de trabajo. También se pueden producir nuevas investigaciones tecnológicas en esta etapa. (Gómez & Hernández, 2016).

2.4.2.4. Roles

Las siguientes funciones están relacionadas con la fase de inicialización. (Arias, 2014):

- Equipo del proyecto
- Equipo del proyecto / jefe de proyecto
- Equipo del proyecto / arquitecto
- Grupo de apoyo
- Grupo de clientes.

2.4.3. Fase de producción

Se repite la programación iterativamente (planificación, trabajo, liberación) hasta implementar todas las funcionalidades usando el desarrollo dirigido por pruebas para llevar a

cabo toda la implementación. (Gómez & Hernández, 2016).

2.4.3.1. Día de la planificación

El propósito es seleccionar y planificar el contenido de trabajo. Al participar activamente en las actividades de planificación, el cliente se asegura que el producto proporciona más valor para el negocio y se identifica con esos requisitos que son correctamente entendidos.

a) Análisis de requisitos.

El propósito de esta tarea avanzar cuidadosamente en el análisis de los requisitos seleccionados para cada iteración (fase).

b) Revisión de Pruebas de Aceptación

El propósito de esta tarea es asegurarse de que el equipo entiende los requisitos del sistema correctamente. Esta tarea también permite a los miembros del equipo comentar sobre las pruebas de aceptación para mejorar su calidad.

c) Generación de Pruebas de Aceptación

El propósito de esta tarea es verificar los requisitos que cliente ha fijado para el software. Esta tarea también actúa como una herramienta de comunicación entre el cliente y el equipo de desarrollo. Esta tarea incluye una plantilla de aceptación de las pruebas.

d) Planificación de las iteraciones

El propósito de esta tarea es generar el programa y el contenido para la iteración (fase) a ejecutar. El contenido se define en términos de tareas que son órdenes de trabajo para el equipo. Esta tarea incluye dos plantillas: historia de trabajo y guía de tareas.

e) Tareas de post Iteración

El propósito de esta tarea es mejorar iterativamente el proceso de desarrollo de software para ajustarse mejor a las necesidades del equipo actual proyecto de software (retroalimentación). Esta tarea incluye una plantilla llamada Punto de acción. (Gómez & Hernández, 2016).

2.4.3.2. Día de trabajo

El propósito de esta etapa es la implementación de la funcionalidad del sistema previsto durante el día de planificación. El equipo de desarrollo se centra en la funcionalidad de

prioridad más alta definida por el cliente.

a) Programación en parejas

El propósito de la programación en parejas es mejorar la comunicación, mejorar el proceso de la fidelidad y la difusión del conocimiento dentro del equipo, y garantizar la calidad del código.

b) Integración continua

El propósito de la integración continua es integrar de forma continua con el nuevo código.

c) Informes al cliente

El propósito de esta tarea es proporcionar una visión honesta de los avances al cliente, para dar la posibilidad de dar su opinión sobre las características implementadas y para guiar el desarrollo.

2.4.3.3. Día de lanzamiento

El propósito de esta etapa es hacer una versión totalmente funcional del sistema bajo desarrollo.

a) Integración del sistema

Los productos complejos pueden requerir que el sistema se divida en subsistemas más pequeños. En el caso del proyecto multi - equipo, el propósito de esta tarea es integrar subsistemas, que se generan en equipos separados, en un solo producto.

b) Pruebas pre - lanzamiento

El propósito de esta tarea es asegurarse de que el software que está siendo producido está listo para las pruebas de aceptación y liberación.

c) Prueba de aceptación

El propósito de esta tarea es verificar que los requisitos que el cliente ha fijado para el software se han implementa correctamente.

d) Ceremonia de lanzamiento

El propósito de esta tarea es confirmar que todo se ha hecho justo en la iteración actual y la base para un mayor desarrollo está garantizada. Las ceremonias de lanzamiento son los pasos

finales antes de hacer una versión del software. En la práctica, las ceremonias de lanzamiento consisten en dos actividades esenciales; auditoría de lanzamiento y la creación una línea de base. (Gómez & Hernández, 2016).

2.4.3.4. Roles

La fase de Producción utiliza el mismo equipo de proyecto y los actores presentes en las fases anteriores. Sin embargo, el papel de la comunicación con el cliente debe enfatizar como una rápida retroalimentación durante la aplicación, es beneficioso para la consecución de resultados satisfactorios, es decir, mejor correspondencia con los requisitos. (Gómez & Hernández, 2016).

Las siguientes funciones están relacionadas con la fase de Producción. (Arias, 2014):

- Equipo del proyecto
- Grupo de apoyo
- Grupo de clientes
- Grupo de Dirección.

2.4.4. Fase de estabilización

Se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyectos multi - equipo con diferentes subsistemas desarrollados por equipos distintos. En esta fase, los desarrolladores realizarán tareas similares a las que debían desplegar en la fase de “producción”, aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación. (Gómez & Hernández, 2016).

2.4.4.1. Roles

Las siguientes funciones están relacionadas con la fase Estabilización. (Arias, 2014):

- Equipo del proyecto
- Equipo del proyecto / arquitecto
- Grupo de apoyo
- Cliente
- Grupo de Dirección.

2.4.5. Fase de pruebas

En esta fase se prueba y repara el sistema, se pasa una fase de test hasta tener una versión estable según lo establecido anteriormente por el cliente, esto como meta para así lograr la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados. (Gómez & Hernández, 2016).

2.4.5.1. Roles

Las siguientes funciones están relacionadas con la prueba del sistema y la fase de reparación. (Arias, 2014):

- Equipo del proyecto
- Grupo de apoyo
- Cliente
- Grupo de Dirección
- Sistema de grupo de prueba.

Una vez finalizadas todas las fases se debería tener una buena producción de la aplicación la cual ya puede ser publicable y entregable al usuario final.

2.5. TROPOS

Tropos, es una metodología de desarrollo de software orientada a los actores que cubre todo el proceso del desarrollo de software, cuya base es la búsqueda de la respuesta a dos preguntas que se consideran básicas: ¿Qué? y ¿Cómo? (También se trata de responder el ¿Por Qué? En ocasiones en las que se considera necesario).

Esta metodología de desarrollo de software tiene 2 fundamentos:

- Los actores y todo lo relacionado con ellos se usan en todas las fases del desarrollo.
- Analiza las fases más tempranas del análisis de requerimientos lo cual permite un más apropiado desarrollo ya que son conocidos el entorno y las interacciones entre el sistema y los agentes. (Rivera, 2013).

2.5.1. Definición de términos

Actor: Representa a una persona o sistema. Las metas deben estar asociadas a actores que

puedan satisfacerlas. La representación gráfica de un actor es un círculo y sus límites están representados por otro círculo de líneas punteadas.

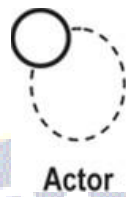


Figura 2.4 Actor en Tropos

Fuente: (Rivera, 2013).

Agente: Es un actor con manifestaciones concretas y físicas, tales como una persona individual. Un agente tiene dependencias que implican independientemente de que rol espera desempeñar.



Figura 2.5 Agente en Tropos

Fuente: (Rivera, 2013).

Rol (“Plays”): Es una característica de un comportamiento de un acto.



Figura 2.6 Rol en Tropos

Fuente: (Rivera, 2013).

Posición (“occupies”): Es una abstracción intermedia entre un rol y un agente, esto es un conjunto de roles típicamente asignados a un agente.



Figura 2.7 Posición en Tropos

Fuente: (Rivera, 2013).

Asociación: se refiere a una colección de roles posiciones y agentes, los cuales están interconectados mediante relaciones: Juega (plays), ocupa (occupies) y cubre (covers).

Meta dura y meta suave (hardgoal/softgoal): Las metas representan los intereses estratégicos de un actor. Las metas duras se distinguen de las metas suaves porque las segundas no tienen un claro criterio de definición para decidir si ellas son satisfechas o no.

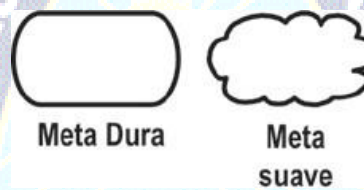


Figura 2.8 Meta dura y suave en Tropos

Fuente: (Rivera, 2013).

Plan: Este elemento representa una manera de hacer algo. La ejecución del plan puede ser una manera de lograr una meta dura o satisfacer una meta suave.

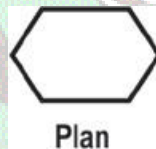


Figura 2.9 Plan en Tropos

Fuente: (Rivera, 2013).

Recurso: Este elemento representa una entidad física o un documento de información.



Figura 2.10 Recurso en Tropos

Fuente: (Rivera, 2013).

Medios-fin (*means-end*): Esta liga se utiliza para definir el conjunto de medios que son necesarios para alcanzar un fin, que usualmente es una meta a satisfacer.

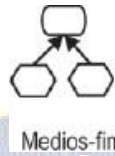


Figura 2.11 Medios - fin en Tropos

Fuente: (Rivera, 2013).

Dependencia de meta dura: Este tipo de dependencia es una relación en el cual un actor depende de otro para satisfacer una meta.



Figura 2.12 Dependencia de meta dura en Tropos

Fuente: (Rivera, 2013).

Dependencia de recurso: Este tipo de dependencia es una relación en la cual un actor depende de otro actor para entregar un recurso que puede ser material o un documento de información.

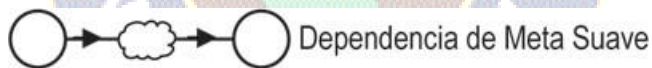


Figura 2.13 Dependencia de recurso en Tropos

Fuente: (Rivera, 2013).

Dependencia de plan: Es una relación en la cual existe una dependencia para llevar a cabo una tarea. En este tipo de dependencia se recomienda la manera en la cual debe ser ejecutado el plan.



Figura 2.14 Dependencia de plan en Tropos

Fuente: (Rivera, 2013).

Dependencia de meta suave: Este tipo de dependencia es muy similar a la dependencia de

meta dura, con la deferencia que no es posible definir en forma precisa la forma en la cual se satisface la meta suave. (Rivera, 2013).



Figura 2.15 Dependencia de meta suave en Tropos

Fuente: (Rivera, 2013).

Se tienen cinco fases a seguir en la metodología:

2.5.2. Análisis temprano de requerimientos

Entendimiento del problema. Se obtiene un modelo organizacional en el cual se incluyen los actores relevantes y sus dependencias. Cada actor tiene sus objetivos que serán logrados en conjunto en virtud de conocimientos y dependencias recíprocas, por medio de las cuales se distribuirán tareas y se hará una utilización correcta de los recursos.

En esta etapa el análisis de los objetivos de los usuarios del futuro sistema conduce a obtener los requerimientos funcionales y no funcionales. Se obtienen dos clases de diagramas; los diagramas de actores (agente, rol, posición) que muestran una red de dependencias sociales de manera general, y los diagramas de racionalidad, en los cuales se hace un análisis de los objetivos (descomposición en sub - objetivos) que se establecieron para cada uno de los actores del diagrama anterior. Por medio de este análisis se pueden añadir más actores y dependencias. (Rodríguez, 2013)

2.5.3. Análisis de requerimientos tardíos

El sistema que se quiere obtener es descrito en su ambiente operacional con sus funciones y cualidades. Sistema es un conjunto pequeño de actores con dependencias sociales. En el análisis realizado se puede hacer una descomposición en sub - actores y sub - objetivos. Se logra mostrar un análisis de casos de uso, ya que en el diagrama de racionalidad se muestra la manera en la que el objetivo de un actor puede ser cumplido por medio del sistema. (Rodríguez, 2013)

2.5.4. Diseño arquitectural

Arquitectura definida en términos de subsistemas (actores), interconectados a través de datos y flujos de control (dependencias). Se definen las capacidades de los agentes y los tipos de agentes (donde los agentes son una clase especial de actores). Se termina con la especificación de los agentes del sistema. Lo anterior se logra por medio del análisis detallado del diagrama extendido de actores. (Rodríguez, 2013)

2.5.5. Diseño detallado

Cada agente de la arquitectura es definido en términos de eventos internos y externos, planes, creencias y protocolos de comunicación. La especificación de las capacidades es importante para modelar los eventos internos y externos que motivan los planes y creencias envueltas en el razonamiento de los agentes. Para este diseño, se pueden adaptar algunos diagramas propuestos para diseñar agentes los cuales se tienen algunos de los propuestos por UML: Diagramas de Capacidad, Diagramas de Actividades, Diagramas de planes, Diagramas de Interacción de Agentes. (En esta etapa podemos sugerir una arquitectura para los agentes, basándonos en diferentes ideas, como Wooldridge, g-nets, Ferber). (Rodríguez, 2013)

2.6. SISTEMA DE GEOLOCALIZACIÓN

La geolocalización, se define como una identificación de un sitio señalado por un dispositivo, como por ejemplo podría ser un smartphone, una antena o bien cualquier tecnología conectada a una red. Esta señal se halla siempre y en todo momento enlazada con diferentes sistemas detectores de posicionamiento. Y a su vez incorpora diversa información relativa al ambiente como locales, distritos, calles. (Perfil., 2020).

La geolocalización ha proveído de servicios tanto en aplicaciones militares como comerciales que va más allá de los servicios de emergencia. Dentro de las aplicaciones comerciales se tienen la necesidad de localización de los pacientes de un hospital

Este tipo de tecnología de sistema de posicionamiento está teniendo gran importancia en la evolución de los sistemas 4G o de cuarta generación, ya que actualmente el sistema más utilizado para posicionamiento es el **GPS** (*Global Position System*), aunque suele ser muy costoso. (Osorio, et.al., s.f.).

2.6.1. Fundamentos de la tecnología GPS

La tecnología GPS (*Global Positioning System*) consiste en una constelación de 24 satélites en órbita que transmiten información precisa de tiempo y posición (órbita del satélite) a estaciones terrestres. El sistema GPS provee información de navegación: Latitud, Longitud, Velocidad, Rumbo, Tiempo, entre otros. Las estaciones terrestres no transmiten información de vuelta al satélite GPS, solamente utilizan la información recibida para realizar cálculos matemáticos.

Cuando el receptor GPS recibe la señal y rastrea al menos 3 satélites, tiene la capacidad de calcular su posición y se dice que “ha obtenido un punto GPS” (GPS Fix). La estación terrestre (receptor GPS) es la encargada de calcular su posición utilizando la información transmitida por los satélites. El receptor GPS utiliza un método matemático conocido como trilateración.(Geek, 2017).

2.6.2. Trilateración del satélite

Si se mide la distancia desde un grupo de satélite a una posición cualquiera de la Tierra pueden calcularse las coordenadas exactas de dicha posición. Los satélites actúan como puntos de referencia precisos. Asumiendo que se conoce la distancia desde un satélite, puede reducirse la posición a la superficie de una esfera que circunde a dicho satélite.

Si también se conoce la distancia desde un segundo satélite, esto reduce la posición a la intersección de dos esferas que forma un círculo.

Añadir un tercer satélite, entonces podremos ubicarnos con exactitud, hay solo dos puntos en el espacio que pueden ser verdaderos. Esto es correcto, haciendo mediciones desde tres satélites podremos reducir el lugar donde estamos a solo dos puntos en el espacio. Tres esferas se interceptan en dos puntos.

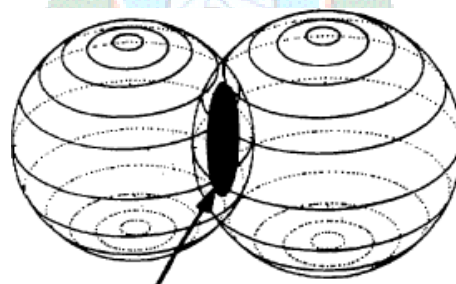


Figura 2.16 Intersección de dos esferas es un círculo

Fuente: (Chihuan, 2002)

De estas dos posiciones una no se considera ya que esta fuera del espacio o moviéndose a gran velocidad y no es una respuesta probable. Si se sabe la distancia a la que se encuentra la posición de tres satélites cualesquiera, pueden calcularse sus coordenadas.

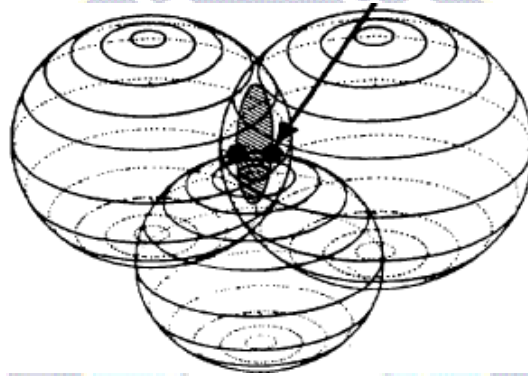


Figura 2.17 Tres esferas se interceptan en dos puntos

Fuente: (Chihuan, 2002)

En la práctica, se necesita un cuarto satélite para resolver las cuatro incógnitas, X, Y, Z, y el tiempo. (Chihuan, 2002)

2.7. ARDUINO

Arduino es una plataforma de prototipos electrónica de código abierto (*open-source*) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos.

El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo, con Flash, *Processing*, *MaxMSP*, entre otros.).

Las placas se pueden ensamblar a mano o encargarse preensambladas; el software se puede

descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia *open-source*, por lo que eres libre de adaptarlas a tus necesidades. Arduino recibió una mención honorífica en la sección *Digital Communities* del Ars Electrónica Prix en 2006. (Herrador, 2009)

Arduino también simplifica el proceso de trabajar con microcontroladores, pero ofrece algunas ventajas para profesores, estudiantes y aficionados interesados sobre otros sistemas:

- **Barato:** los tableros Arduino son relativamente económicos en comparación con otras plataformas de microcontroladores. La versión más económica del módulo Arduino se puede ensamblar a mano.
- **Multiplataforma:** El software de Arduino (IDE) se ejecuta en Windows, Macintosh OS X, y Linux. La mayoría de los sistemas de microcontroladores están limitados a Windows.
- **Entorno de programación simple y claro:** el software Arduino (IDE) es fácil de usar para los principiantes, pero es lo suficientemente flexible como para que los usuarios avanzados también puedan aprovechar. Para los maestros, está convenientemente basado en el entorno de programación de Procesamiento, por lo que los estudiantes que aprenden a programar en ese entorno estarán familiarizados con el funcionamiento del IDE de Arduino.
- **Software de código abierto y extensible:** el software Arduino se publica como herramientas de código abierto, disponibles para la extensión por programadores experimentados. El lenguaje se puede expandir a través de las bibliotecas de C ++, y las personas que deseen comprender los detalles técnicos pueden dar el salto de Arduino al lenguaje de programación AVR C en el que se basa. De manera similar, puede agregar código AVR-C directamente a sus programas Arduino si lo desea.
- **Fuente abierta y hardware extensible:** los planes de las placas Arduino se publican bajo una licencia de *Creative Commons*, por lo que los diseñadores de circuitos con experiencia pueden crear su propia versión del módulo, ampliarlo y mejorarlo. Incluso los usuarios relativamente inexpertos pueden construir la versión de tablero del módulo para entender cómo funciona y ahorrar dinero. (Arduino, 2019).

2.7.1. Hardware Arduino

Los Arduino y en general los microcontroladores tienen puertos de entrada y salida y puertos de comunicación. En Arduino podemos acceder a esos puertos a través de los pines.

- Pines digitales: pueden configurarse como entrada (para leer, sensores) o como salida (para escribir, actuadores)
- Pines analógicos de entrada: usan un conversor analógico/digital y sirven para leer sensores analógicos como sondas de temperatura.
- Pines analógicos de salida (PWM): la mayoría de Arduino no tienen conversor digital/analógico y para tener salidas analógicas se usa la técnica PWM. No todos los pines digitales soportan PWM.
- Puertos de comunicación: USB, serie, I2C y SPI

Otro aspecto importante es la memoria, Arduino tiene tres tipos de memoria:

- **SRAM:** donde Arduino crea y manipula las variables cuando se ejecuta. Es un recurso limitado y debemos supervisar su uso para evitar agotarlo.
- **EEPROM:** memoria no volátil para mantener datos después de un reset o apagado. Las EEPROMs tienen un número limitado de lecturas/escrituras, tener en cuenta a la hora de usarla.
- **Flash:** Memoria de programa. Usualmente desde 1 Kb a 4 Mb (controladores de familias grandes). Donde se guarda el sketch. (Crespo, s.f.).

2.7.2. Componentes

2.7.2.1. Arduino LilyPad

La placa LilyPad Arduino 328 es un micro controlador programado por Arduino diseñado para integrarse fácilmente en e-textiles y proyectos portátiles. Ofrece la misma funcionalidad que se encuentra en otras placas Arduino, en un paquete liviano y redondo, diseñado para minimizar el enganche y el perfil, con pestañas anchas que se pueden coser y conectar con hilo conductor.

El LilyPad Arduino consiste en un ATmega328 con el gestor de arranque Arduino y un número mínimo de componentes externos para mantenerlo lo más pequeño (y simple) como sea posible. Esta placa funciona con voltaje de 2 V a 5 V y ofrece orificios de salida grandes que facilitan la costura y la conexión. Cada uno de estos pines, con la excepción de (+) y (-), puede controlar un dispositivo de entrada o salida conectado (como una luz, un motor o un interruptor). (Arduino, 2020)

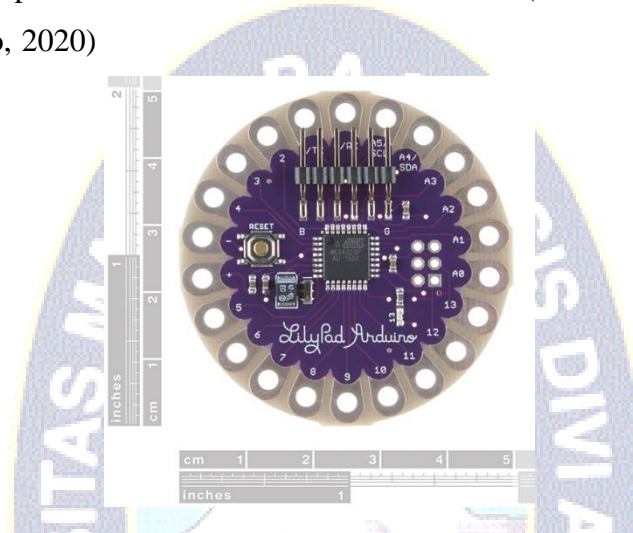


Figura 2.18 Arduino LilyPad
Fuente: (Arduino, 2020).

2.7.2.2. A9G de AI Thinker

Desarrollar proyectos de Internet de las cosas (IoT) utilizando la red celular 2G es ahora más sencillo con la ayuda del módulo A9G, que además permite conocer la ubicación mediante GPS. El módulo A9G integra en un solo dispositivo un módem GSM/GPRS y un receptor GPS/GNSS. El A9G es la evolución del módulo A7, fabricado por Ai-Thinker: los creadores del magnífico módulo ESP-12E(ESP8266) y ESP-WROOM-32(ESP32). El A9G es un Sistema En Módulo o SoM (*System On Module*) pues integra en un módulo los siguientes chips: Chip CPU(SoC): RDA8955L (RISC 32 bit), Chip GSM/GPRS: RDA6625E y Chip GPS/GNSS: GK9501. Ideal para aplicaciones móviles de: rastreo y monitoreo de vehículos (automóviles, buses, bicicletas), supervisión de flotas de buses o camiones, monitoreo de ganado o mascotas.

La placa de desarrollo A9G permite realizar y recibir llamadas de voz, recepción y envío de mensajes SMS y también conectarse a Internet mediante GPRS y entrar así al mundo del Internet de las cosas (IoT). Soporta GSM en las bandas de

850/900/1800/1900MHz. Incluye socket para tarjeta micro-SIM y socket para memoria Micro-SD. La placa puede ser alimentado por el puerto USB (micro-USB) o por los pines VBAT a una fuente externa o una batería de Litio de 1 celda, incluye también la electrónica de gestión de carga de batería. La placa A9G incluye además micrófono, interfaz de altavoz, interfaz de comunicación USB, interfaz SPI, interfaz I2C, interfaz ADC.

El módulo A9G posee 2 modos de trabajo: A. Modo Módem AT funciona como un clásico módem similar al SIM800/SIM900/A6/A7 con comunicación por comandos AT por Serial UART con un microcontrolador. B. Modo Tarjeta de desarrollo SDK en el cual funciona como un dispositivo todo en uno: microcontrolador + modem, con la gran ventaja de no necesitar un MCU adicional reduciendo costos, energía y tamaño, podemos programar el dispositivo directamente en C (utilizando las herramientas de Ai-Thinker SDK) y cargar el firmware utilizando un conversor USB-Serial TTL como el CP2102/PL2303. Ai-Thinker promete que dentro de unos meses podremos programar nuestro A9G directamente desde el IDE Arduino de forma similar al ESP8266 o ESP32. (Naylamp,2021).



Figura 2.19 A9G DE AI THINKER
Fuente: (Naylamp, 2021).

2.7.2.3. SIM Card

La tarjeta SIM o *Subscriber Identity Module* es una pequeña tarjeta de plástico que tiene un chip pegado a ella, y que tienes que insertar en tu teléfono móvil o smartphone. En

este chip, almacena de manera segura tu número de teléfono, así como las claves de acceso de un usuario concreto en una operadora de telefonía.

Concebidas en los años 70, la tecnología de las tarjetas inteligentes o *Smart Card* tuvo distintos usos a lo largo de los años. A principios de los 90, se incorporaron como elemento de identificación obligatorio de los usuarios en las redes de telefonía GSM. Tendrían que ser obligatorias, y todos los móviles tendrían que tener una para ser identificados a la hora de hacer llamadas.

El chip de una SIM tiene ocho contactos eléctricos que le permiten interactuar con el móvil en el que está insertada. Con el tiempo, además de los datos de autenticación del usuario en la red, las tarjetas SIM han ido ampliando su capacidad de almacenamiento. Pasaron de los 16 KB originales a poder almacenar más información, como por ejemplo los datos de conexión a Internet o una agenda simple en la que guardar algunos números de teléfono.

El sistema criptográfico y de seguridad de este tipo de tarjetas hace tiempo que fue superado y dejó de ser tan útil. Sin embargo, el uso de las tarjetas SIM sigue siendo generalizado en todo el mundo, tanto en las redes GSM como en las CDMA, que son las dos más utilizadas (sobre todo la primera) para enviar las señales de telefonía móvil en todo el mundo.

Por lo tanto, la razón por la que a día de hoy se siguen utilizando los diferentes tipos de tarjeta SIM no es tanto por su seguridad como por su gran comodidad. Y es que es muy cómodo poder almacenar los datos del usuario en un medio físico que puedes pasar fácilmente de un teléfono a otro sin tener que hacer ninguna configuración extra. Además, en el plástico de la SIM suele venir impreso tu número de teléfono, algo que también puede ayudarte cuando le cambies la tarjeta al móvil. (Xataka,2020).

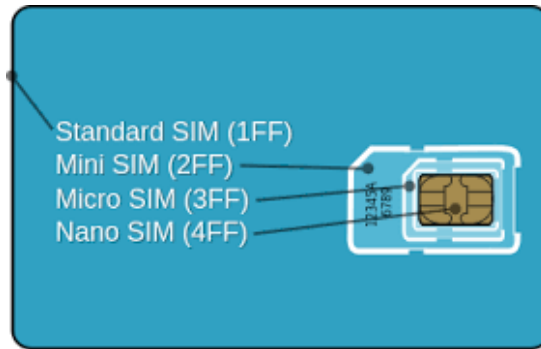


Figura 2.20 SIM CARD

Fuente: (Xataka,2020).

a) SIM

Las tarjetas SIM originales ya no se utilizan, y eran de un tamaño parecido al de una tarjeta de crédito. Medía apenas 33×66 milímetros, y tenía tan poco espacio que cabían apenas 20 contactos y 5 o 6 mensajes. (Xataka,2020).

b) MINISIM

Es la que actualmente conocemos como SIM a secas, pero que no fue el primer modelo existente. Fue el estándar desarrollado para que la tarjeta SIM cupiera en los teléfonos móviles, y tiene un tamaño de 15×25 milímetros. (Xataka,2020).

c) MICROSIM

En 2003 se desarrolló este nuevo tipo de tarjeta, aunque no fue hasta la llegada del iPad que Apple hizo que toda la industria empezase a adoptarla. Con un tamaño de 12×15 milímetros, se creó para ampliar la memoria y seguridad de las SIM, pero acabó aprovechándose para hacerla más pequeña. (Xataka,2020).

d) NANOSIM

Una evolución sobre las Micro SIM, con la que se consiguió crear una tarjeta todavía más pequeña de sólo 12×9 milímetros. Empezó a llegar en 2012, y su misión fue la de optimizar el espacio del hardware de un teléfono, manteniendo su tamaño externo, pero con mayor volumen para ciertos componentes. (Xataka,2020).

2.8. APLICACIÓN MÓVIL

Una aplicación móvil, una aplicación, una apli o una *app* (acortamiento del inglés *application*), es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Este tipo de aplicaciones permiten al usuario efectuar un variado conjunto de tareas profesionales, de ocio, educativas, de acceso a servicios, etcétera, facilitando las gestiones o actividades a desarrollar.

Por lo general, se encuentran disponibles a través de ciertas plataformas de distribución, o por intermedio de las compañías propietarias de los sistemas operativos móviles tales como Android, iOS, BlackBerry OS, Windows *Phone*, entre otros. Al ser aplicaciones residentes en los dispositivos están escritas en algún lenguaje de programación compilado, y su funcionamiento y recursos se encaminan a aportar una serie de ventajas tales como:

- Un acceso más rápido y sencillo a la información necesaria sin necesidad de los datos de autenticación en cada acceso.
- Un almacenamiento de datos personales que, a prioridad, es de una manera segura.
- Una gran versatilidad en cuanto a su utilización o aplicación práctica.
- La atribución de funcionalidades específicas.
- Mejorar la capacidad de conectividad y disponibilidad de servicios y productos (usuario-usuario, usuario-proveedor de servicios, etc.).

Un sistema operativo es un programa o conjunto de programas informáticos que gestiona el hardware de un dispositivo y administra el servicio de aplicaciones informáticas (Windows, iOS, Android, etc.).

2.9. GEOREFERENCIACIÓN

Se entiende por Georreferenciación o Geolocalización, a la identificación de la posición geográfica real de un objeto o persona, ya sea un dispositivo conectado a Internet, un teléfono móvil o cualquier otro aparato que sea posible rastrear. Dicha localización puede ser en un plano de dos dimensiones (por ejemplo, Google Maps), como en un plano de tres dimensiones (GPS). En los últimos años, diferentes tipos de tecnologías han apostado por la Georreferenciación, siendo extraordinario el auge de esta en las tecnologías móviles de última generación. Actualmente es posible el uso de la Geolocalización en la mayoría de plataformas:

- En el caso de la Geolocalización de un ordenador, esta se hace a través de una serie de bases de datos que “aproximan” la zona en la que el usuario se encuentra.
- En el caso de un dispositivo Móvil, existen diferentes tecnologías actualmente, como Sistema de Posicionamiento Global (GPS). Sobre la Geo localización en dispositivos Móviles, los usos se disparan. El conocer la posición actual en un momento determinado nos ofrece el poder disponer de un sinfín de información de nuestro alrededor. La primera y más conocida tecnología es el sistema GPS, que informa de nuestra posición actual, que, a través de mapas, indican diferentes rutas a fin de llegar al destino. Actualmente, esta tecnología ofrece información como bares, restaurantes, puntos de encuentro, localización de móviles para niños.

La utilización de los satélites de la constelación NAVSTAR con técnicas GPS ha abierto en las Ciencias Geográficas un inmenso abanico de posibilidades, al permitir situar puntos, con grandes precisiones, en aplicaciones geodésicas y topográficas, y precisiones ampliamente satisfactorias para navegación en tiempo real por tierra, mar y aire.

2.9. SISTEMA DE POSICIONAMIENTO GLOBAL

El sistema de Posicionamiento Global es un sistema de localización, diseñado por el Departamento de Defensa de los Estados Unidos con fines militares para proporcionar estimaciones precisas de posición, velocidad y tiempo, operativo desde 1995 utiliza conjuntamente una red de ordenadores y una constelación de satélites para determinar por triangulación, la altitud, longitud y latitud de cualquier objeto en la superficie terrestre.

La navegación por satélite es un sistema de radio localización desde el espacio con una o más constelaciones de satélites que aumenta según las necesidades de apoyo a la operación prevista y con los que se suministra información tridimensional las 24 horas del día acerca de la posición, velocidad y hora a usuarios debidamente equipados en cualquier parte de la superficie, o cerca de la superficie, de la tierra. El GPS es el primer elemento básico de los sistemas de navegación por satélite que está a disposición de muchos usuarios civiles

Mediante el empleo de receptores portátiles que reciben la señal de los 24 satélites que orbital la Tierra, es posible identificar cada punto sobre ésta y asociar el acontecimiento en tiempo real, Así poder localizar mediante unas coordenadas únicas cualquier equipo radiorreceptor terrestre sin importar su posición en cualquier parte del planeta e incluso

fuera de ella.

El Sistema de Posicionamiento Global o GPS, aunque su nombre correcto es NAVSTAR-GPS1, es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave. Podemos alcanzar una precisión hasta de centímetros, usando el GPS diferencial, pero lo habitual son unos pocos metros.

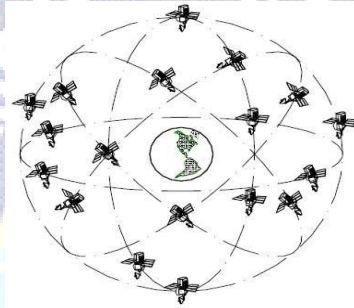


Figura 2.21 Sistema de Posicionamiento Global
Fuente: (Leonardo Casanova M, ULA, 2002).

Llegado a este punto la pregunta sería, ¿Qué hace un GPS?

- Identificación y seguimiento de los códigos asociados a cada satélite.
- Determinación de la distancia.
- Decodificación de las señales de datos de navegación para obtener las efemérides, el almanaque.
- Aplicar las correcciones (del reloj).
- Determinación de la posición y velocidad.
- Validación de los resultados obtenidos.
- Presentación de la información.

2.10. TECNOLOGÍA CELULAR Y GPS

El desarrollo de la tecnología de las comunicaciones se ha vuelto multi alcance por lo que la interacción de varios aspectos dinámicos como la geo localización GPS es algo en la actualidad común en los dispositivos de telefonía celular. Hoy en día, los dispositivos de

comunicación móviles se están convirtiéndose en algo mucho más avanzado y ofrecen más que la capacidad de llevar sólo en una conversación. Todos los teléfonos celulares emiten constantemente una señal de radio, incluso cuando no está en una llamada.

Las empresas de telefonía celular han sido capaces de estimar la localización de un teléfono celular durante muchos años usando información de triangulación de las torres que reciben la señal. Sin embargo, la introducción de la tecnología GPS en los teléfonos celulares ha significado que el seguimiento del teléfono celular con GPS sea mucho más preciso. Aplicaciones GPS de seguimiento proporciona un campo de aplicaciones para iPhone, Android, Windows Mobile entre otros para realizar un seguimiento de unos a otros. La tecnología de localización se basa en la medición de los niveles de potencia y los diagramas de antena y utiliza el concepto de que un teléfono móvil siempre se comunica de forma inalámbrica con una de las estaciones base más cercanas, así que si se sabe qué estación base de la telefonía se comunica con el receptor se sabe que el teléfono está cerca de la estación base respectiva.

Además, dicha aproximación se puede lograr mediante la interpolación de señales entre torres de antena adyacentes. Servicios calificados pueden alcanzar una precisión de hasta 50 metros en zonas urbanas en las que el tráfico móvil y la densidad de torres de antenas (estaciones base) es suficientemente alta. Las zonas rurales y desoladas pueden tener estaciones bases con distancias considerables entre sí y, por tanto, determinar los lugares con menos precisión.

2.10.1. Funcionamiento del GPS

El funcionamiento del sistema GPS se basa también, al igual que los sistemas electrónicos antiguos de navegación, en el principio matemático de la triangulación. Por tanto, para calcular la posición de un punto será necesario que el receptor GPS determine con exactitud la distancia que lo separa de los satélites. Los sistemas de posicionamiento global permiten situar a una persona mediante un aparato la posición dentro del mundo con una precisión de casi centímetros. Para su buen funcionamiento es necesario tener en cuenta la diferencia de frecuencias a las cuales son emitidas y recibidas las señales en dos localizaciones con distinto potencial gravitatorio.

Cada satélite dispone de un pequeño reflector, que es usado para el seguimiento de los satélites por láser desde las estaciones de control. Existen 4 prototipos o modelos de satélite.

El primer prototipo lo componen un total de 10 satélites que forman el Bloque I, lanzados entre Octubre82 y May-85. Otros 6 satélites del segundo prototipo forman el Bloque Iia, lanzados entre Mayo85 y Septiembre-86. Un total de 12 satélites forman el Bloque Iib del tercer prototipo, lanzados entre Abril-87 y Mayo-88, de los cuales seis se perdieron en los dos fallos anteriormente reseñados por fallo del vehículo de lanzamiento. El cuarto prototipo forma el Bloque Iiv, constituido por 43 satélites, de los cuales se han lanzado la totalidad de ellos hasta la fecha de 14 de diciembre de 1995. Cada subsiguiente generación de satélites contiene equipamientos más modernos y tienen un mayor periodo de vida.

Prototipos Lanzados	Bloques	Satélites	Fecha
P1	I	10	Oct-82 y May-85
P2	Iia	6	Mayo-85 y Sep-86
P3	Iribú	12	Abril-87yMay-88
P4	Inv.	43	14 Dic de 1995

Tabla 2.1 Lanzamientos de Bloques de Satélites

Fuente:

Forman parte del equipamiento de cada satélite dos paneles solares y un sistema de propulsión. Los paneles solares se utilizan para recargar los acumuladores que permiten el funcionamiento mientras el satélite pasa por la sombra de la Tierra. Desde el sistema de control terrestre es posible activar los sistemas de propulsión con el objetivo de corregir las órbitas de cada satélite o incluso cambiar de posición dentro de la misma órbita.

- a) Altitud: 20.000 km
- b) Periodo: 11h 59 min
- c) Inclinación: 55 grados (respecto al ecuador).
- d) Vida útil: 7,5 años
- e) Utilizan paneles solares y baterías

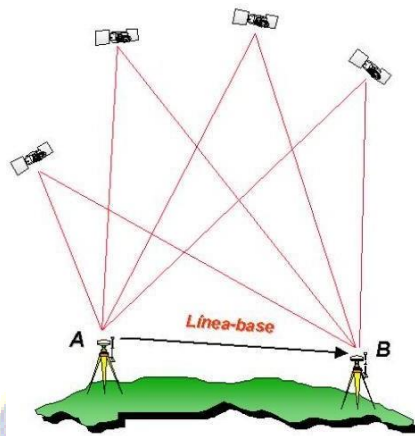


Figura 2.22 Cambios de Posición

Fuente: (Leonardo Casanova M, ULA, 2001)

La identificación de los satélites puede hacerse de varias formas, por su orden de lanzamiento, por la órbita y posición que ocupa en ella, o por su PRN o Ruido Pseudo Aleatorio (Pseudo Random Noise) característico y exclusivo de cada satélite.

2.10.2. GPS diferencial

El DGPS o GPS diferencial, es un sistema que proporciona a los receptores de GPS correcciones de los datos recibidos de los satélites GPS, con el fin de proporcionar una mayor precisión en la posición calculada. Se concibió fundamentalmente debido a la introducción de la disponibilidad selectiva.

El fundamento radica en el hecho de que los errores producidos por el sistema GPS afectan por igual a los receptores situados próximos entre sí. Los errores están fuertemente correlacionados en los receptores próximos. Un receptor GPS fijo en tierra que conoce exactamente su posición basándose en otras técnicas, recibe la posición dada por el sistema GPS, y puede calcular los errores producidos por el sistema GPS, comparándola con la suya, conocida de antemano. Este receptor transmite la corrección de errores a los receptores próximos a él, y así estos pueden, a su vez, corregir también los errores producidos por el sistema dentro del área de cobertura de transmisión de señales del equipo GPS de referencia.

2.10.2.1. Segmento Espacial

El Segmento Espacial está constituido por los satélites que soportan el sistema y las señales de radio que emiten. Estos satélites conforman la llamada constelación NAVSTAR (Navegación Satélite Timing and Ranging).

La señal en sí es muy compleja. Está formada por varios componentes que se estructuran sobre una señal principal con frecuencia en MHz.

Los satélites del GPS transmiten dos señales de radio de baja potencia, llamadas "L1" y "L2". Cada señal GPS contiene tres componentes de información: un código pseudoaleatorio, los datos de efemérides de satélite y datos de almanaque.

- El código pseudoaleatorio: identifica al satélite que transmite su señal.
- Los datos de efemérides de satélite: proporcionan información sobre la ubicación del satélite en cualquier momento.
- El almanaque: contiene información sobre el estado del satélite y la fecha y hora actuales.

Para cada satélite, el tiempo es controlado por los relojes atómicos a bordo que son cruciales para conocer su posición exacta.

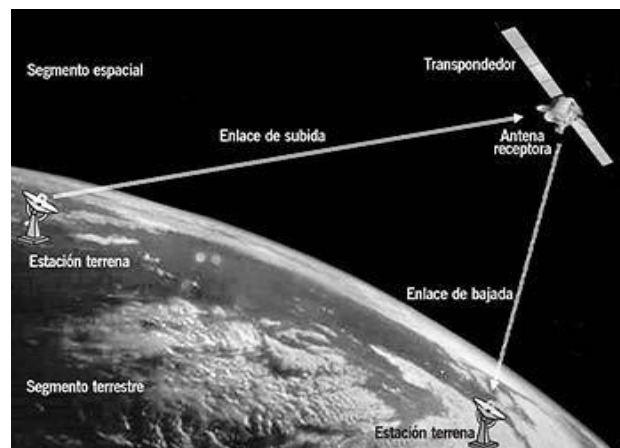


Figura 2.23 Segmento Espacial
Fuente: (Ruben Gabriel, 2012)

2.10.2.2. Segmento de control

El segmento de control está constituido por todas las infraestructuras en tierra necesarias para el control de la constelación de satélites, mantenidas por la fuerza aérea estadounidense.

Dichas infraestructuras tienen coordenadas terrestres de muy alta precisión y consisten en cinco grupos de instalaciones repartidas por todo el planeta, para tener un control homogéneo de toda la constelación de satélites.

Estas infraestructuras realizan un seguimiento continuo de los satélites que pasan por su región del cielo, acumulando los datos necesarios para el cálculo preciso de sus órbitas.

Dichas órbitas son muy predecibles, dado que no existe fricción atmosférica en el entorno donde se mueven los satélites, a las predicciones de las órbitas de los satélites, cuyo cálculo depende también del segmento de control. Sin embargo, aunque muy predecibles, las órbitas también tienen una degradación debido a una serie de factores: desigual densidad de la gravedad terrestre, mareas gravitatorias provocadas por el alineamiento de la luna y los planetas, viento solar, etc.

Todos estos factores conllevan a pequeñas degradaciones sobre las órbitas que hay que tener en cuenta para que el sistema GPS sea preciso. Por ello, aquellas estaciones del segmento de control que están dotadas de antenas de referencia tienen también la función de enviar a los satélites las correcciones de órbita para sus sistemas de navegación. A estas órbitas recalculadas con los datos de corrección (suministrados por las estaciones de tierra) y su información de tiempo se les denomina efemérides.

El usuario no experimentado no ve por ninguna parte rastro de las efemérides, pero hasta el navegador más sencillo las está utilizando en el momento en que estamos midiendo.

Las funciones principales del segmento de control, denominado internacionalmente con las siglas OCS (Operational Control Segment) son:

- Monitoreo y control permanente de los satélites con el objeto de determinar y predecir las órbitas y los relojes de a bordo.
- Sincronización de los relojes de los satélites con el tiempo GPS Transmisión, a cada satélite, de la información procesada.

Está integrado por una Estación de Control Maestra (MCS), varias Estaciones de Monitoreo (MS) y Antenas Terrestres (GA).

El segmento de control está integrado por 10 estaciones.

- Colorado Springs (EUA)

- Isla Ascensión (Atlántico Sur)
- Diego García (Índico)
- Kwajalein (Pacífico Occidental)
- Hawaii (Pacífico Oriental)
- Quito (Ecuador)
- Buenos Aires (Argentina)
- Hermitage (Inglaterra)
- Bahrein (Golfo Pérsico)
- Smithfield (Austral)

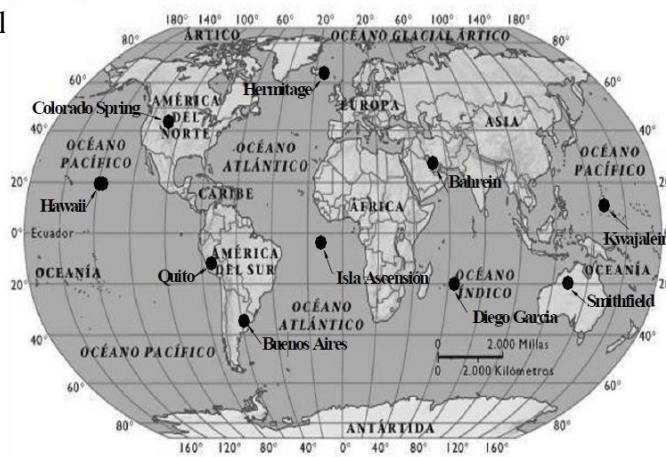


Figura 2.24 Estaciones de Control
Fuente: (Carlos Punch,2002)

2.10.2.3. Segmento de usuario

El segmento de Usuarios comprende a cualquiera que reciba las señales GPS con un receptor, determinando su posición y/o la hora. Algunas aplicaciones típicas dentro del segmento Usuario tenemos la navegación en tierra para excursionistas, ubicación de vehículos, topografía, navegación marítima y aérea, control de maquinaria, etc.

Hablando de la utilización del GPS como instrumento topográfico este segmento comprende los siguientes elementos:

- **Antena:** Componente que se encarga de recibir y amplificar la señal recibida por los satélites.



Antena GPS Tipo Choke Ring



Antena GPS Tipo MicroStrip

Figura 2.25 Antena GPS
Fuente: (Eduardo Graña, 2009)

- **Receptor:** Recibe la señal recogida por la antena y decodifica esta para convertirla en información legible.

Antena GPS Tipo MicroStrip



Figura 2.26 Receptor GPS
Fuente: (Eduardo Graña, 2009)

- **Terminal GPS o Unidad de Control:** Ordenador de campo que muestra la información transmitida por los satélites y recoge todos datos útiles para su posterior cálculo, de aplicaciones Topográficas.



Figura 2.27 Terminal GPS
Fuente: (Eduardo Graña, 2009)

Un equipo de recepción de señales GLONASS, al igual que uno de GPS, está formado por una antena y un receptor, así como terminal o colector de datos. La antena suele llevar un plano de tierra para evitar el efecto multipath. Los receptores disponen de un reloj para sincronizar las señales recibidas. Lo normal y aplicado para Topografía es encontrar un receptor combinada GPS/GLONASS

El GPS es el conjunto de elementos (Software y Hardware) que permiten determinar la posición, velocidad y tiempo de un usuario, además de los parámetros necesarios adicionales que requiera. A continuación, enumeraremos las partes de las que consta un GPS:

- Antena con preamplificador.
- Sección de radio frecuencia o canal.
- Micro procesador para almacenamiento y procesamiento de datos.
- Oscilador de precisión para la generación de los códigos Pseudo aleatorios utilizados en la medición del tiempo de viaje de la señal.
- Fuente de energía eléctrica.
- Interfases del usuario (pantalla, teclado de comandos).
- Memoria de almacenamiento.



Figura 2.28 Segmento de Usuario
Fuente: (Eduardo Graña, 2009)

Información en el receptor

Una vez en funcionamiento, el receptor puede ofrecer al operador una muy amplia y diversa información sobre el proceso de observación, mientras recibe las señales de los satélites. Aunque varía entre diferentes modelos, se suele disponer de la información siguiente:

- Satélites localizados.

- Satélites en seguimiento.
- Intensidad de cada señal recibida.
- Condición de cada satélite en seguimiento.
- Posición: longitud, latitud, altitud.

2.10.2.4. Instrumentos GPS trackings

Son instrumentos de vigilancia de la ubicación a través de la utilización del Sistema de Posicionamiento Global (GPS) lo hacen desde ubicaciones remotas y triangulan su ubicación con un servidor GPS. La tecnología puede determinar con precisión la longitud, latitud, velocidad de avance y dirección del objetivo a monitorearse. Las posiciones pueden incluso ser computado en vistas tridimensionales con la ayuda de cuatro señales de satélite GPS.

El segmento espacial del Sistema de Posicionamiento Global consta de 27 satélites GPS que orbitan la Tierra. Hay 24 satélites operativos y 3 satélites adicionales (en caso de que uno falle) se mueven alrededor de la Tierra cada 12 horas y se envían señales de radio del espacio que son recibidas por el receptor GPS. Mediante estaciones de monitoreo se dan seguimiento de los satélites GPS que orbitan alrededor de la tierra de forma continua.

Los vehículos espaciales transmiten señales portadoras de microondas. El funcionamiento del sistema se basa en un sencillo principio matemático llamado trilateración. La Trilateración se divide en dos categorías: Trilateración 2D y Trilateración 3D. Con el fin de hacer el cálculo matemático sencillo el receptor GPS debe saber dos cosas. En primer lugar, se debe conocer la ubicación del lugar es para ser trazada por al menos tres satélites sobre el lugar. En segundo lugar, se debe conocer la distancia entre el lugar y cada uno de los vehículos espaciales.

Estas ondas de radio transmisión son energía electromagnética que viaja a la velocidad de la luz. Un sistema de seguimiento GPS puede trabajar de varias maneras. Desde un punto de vista comercial, los dispositivos GPS se utilizan generalmente para registrar la posición de los vehículos que hacen sus viajes. Algunos sistemas almacenan los datos dentro del sistema de seguimiento de GPS en sí (conocido como el seguimiento pasivo) y algunos envían la información a una base de datos o sistema centralizado (servidor GPS) a través de un módem dentro de la unidad del sistema GPS sobre una base regular (conocido como el seguimiento activo).

2.11. GEOLOCALIZACIÓN

Geolocalización es un concepto relativamente nuevo, que ha proliferado de unos dos años a esta parte y que hace referencia al conocimiento de la propia ubicación geográfica de modo automático. También denominada georreferenciación, la geolocalización implica el posicionamiento que define la localización de un objeto en un sistema de coordenadas determinado.

Este proceso es generalmente empleado por los sistemas de información geográfica, un conjunto organizado de hardware y software, más datos geográficos, que se encuentra diseñado especialmente para capturar, almacenar, manipular y analizar en todas sus posibles formas la información geográfica referenciada, con la clara misión de resolver problemas de gestión y planificación. La geolocalización permite localizar un teléfono móvil o un ordenador en un lugar preciso. Esta tecnología ha sido aprovechada por redes sociales como Facebook o Foursquare. Sin embargo, la localización geográfica de un *Smartphone* o un ordenador portátil no se limita a las redes sociales. Esta también puede ser utilizada a diario, como para planear un recorrido en un mapa desde un *Smartphone*, o facilitar las compras.

La geolocalización determina las coordenadas geográficas de un teléfono móvil o un ordenador. Con un Smartphone, la localización es posible incluso sin activar la función GPS del teléfono. En este caso, la localización es efectuada con la ayuda de estaciones bases cercanas al Smartphone. Dependiendo de la distancia de las estaciones base, la precisión varía de 3 metros a 30 kilómetros.

El Sistema de Posicionamiento Global (GPS) es una red compuesta por cerca de 30 satélites que orbitan la Tierra a una altitud de 20.000 kilómetros. Según explican desde *Institute of Physics*, es un sistema diseñado originalmente por el gobierno de los Estados Unidos para la navegación militar. Hoy en día, es una tecnología accesible para muchos, bien sea desde un móvil, o un dispositivo portátil de GPS, como los que se instalan en los coches. Cualquiera de esos terminales puede recibir las señales que los satélites envían.



CAPITULO III

3. DISEÑO METODOLÓGICO

3.1. METODOLOGÍA V, MOBILE D Y TROPOS

Las metodologías V, Mobile D y Tropos se combinaron haciendo una refactorización de metodologías para el mejor desarrollo del prototipo de collar con geolocalización GPS.

Primero se hará una combinación de las metodologías Mobile D y Tropos.



Figura 3.1: Metodología combinada Mobile D y Tropos

Fuente: (Elaboración Propia).

Al combinar las tres metodologías se tiene como resultado una metodología robusta que cumple todos los requerimientos para el desarrollo del prototipo de collar con geolocalización para monitorear mascotas.

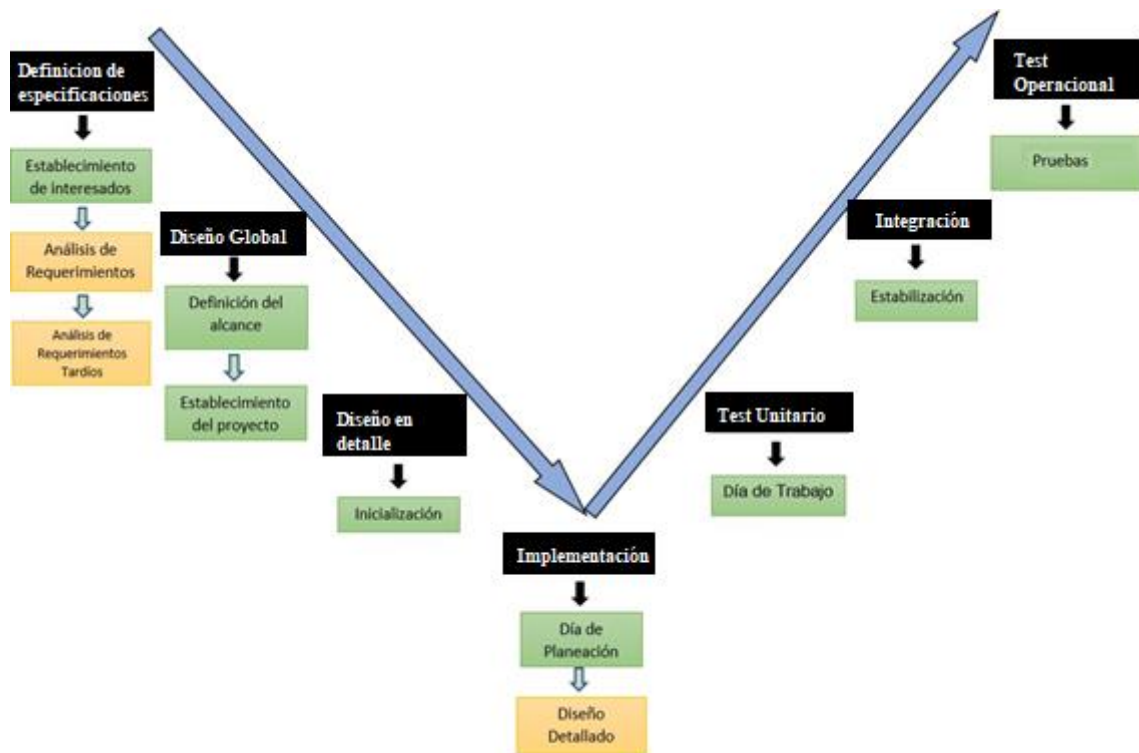


Figura 3.2: Metodología combinada Mobile D, Tropos y Metodología V

Fuente: (Elaboración Propia, 2021).

Usa fases de la Metodología V como guía base se tiene:

Fase 1 - Definición de Especificaciones: Esta fase está orientado especialmente a los clientes o usuarios tanto como en la primera etapa de la metodología Mobile D y en las primeras fases la metodología Tropos.

En esta fase se llevará a cabo los siguientes procesos:

- Definición de Actores, donde se especificará los actores involucrados y sus funciones principales para el desarrollo del prototipo de collar.
- Objetivos y Dependencias, donde se especificará los objetivos del usuario en el futuro prototipo de collar, para obtener los requerimientos necesarios.

- Diagrama de actores, usando la descripción que se encontró en la fase anterior se hará un diagrama con objetivos y sus dependencias del prototipo de collar.

Fase 2 – Diseño Global: En esta fase se hará un diseño y visión general del prototipo de collar, usando las dos últimas etapas de la fase de exploración de la metodología Mobile D.

En esta fase se llevará a cabo los siguientes procesos:

- Alcance, se establecerá los objetivos principales, el plan inicial para el comienzo del prototipo de collar y el conjunto de requisitos iniciales.
- Establecimiento del prototipo de collar, se definirá y asignará los recursos necesarios para el desarrollo y se hará la planificación de la línea del proceso base.

Fase 3 – Diseño en Detalle: Se detallará los componentes de hardware y software a usar de la fase anterior del prototipo final y se definirá la arquitectura del prototipo de collar.

En esta fase se llevará a cabo los siguientes procesos:

- Configuración, se verificará las características cada componente de hardware y lo necesario de software para empezar el desarrollo del prototipo de collar.
- Planificación, se hará una comprensión general del prototipo de collar, donde también se hará un análisis detallado de los actores y requisitos principales.

Fase 4 – Implementación: En esta fase se planificará la materialización del diseño en detalle del prototipo de collar.

En esta fase se llevará a cabo los siguientes procesos:

- Diseño Detallado, se modelará los eventos internos y externos, que digan los usuarios que podrían pasar.
- Día de Planificación, se realizará análisis profundo de requisitos y se hará una historia de trabajo y guía de tarea.

Fase 5 – Test Unitario: Se comprobará que todos los módulos hardware y software estén con un buen funcionamiento, para que estén listos para la siguiente etapa y así no tener ningún retraso en la planificación, generando un documento de pruebas verificando su funcionamiento.

En esta fase se llevará a cabo los siguientes procesos:

- Dia de Trabajo, se implementará el desarrollo funcional de la fase de Diseño Global utilizando la parte de planificación.

Fase 6 – Integración: Se integrará todos los módulos del prototipo de collar, comprobando que se encuentre en un correcto funcionamiento y garantizando que no tiene fallos, también se comprobara el cumplimiento de los requisitos.

En esta fase se llevará a cabo los siguientes procesos:

- Dia de lanzamiento, se realizará una versión totalmente funcional del prototipo, y estar listo para las pruebas de aceptación.
- Estabilización, se verificará si hay fallos y se resolverá los inconvenientes y adicionalmente se hará documentación de la verificación realizada.

Fase 7 – Test Operacional: Se realizará pruebas en un escenario real anotando los resultados obtenidos.

En esta fase se llevará a cabo los siguientes procesos:

- Fase de Prueba, se tendrá una versión estable según todo establecido por el usuario, además que el prototipo de collar se probará con los requisitos del usuario.

Se comenzará el desarrollo del prototipo de collar para monitoreo de mascotas usando geolocalización basada en tecnologías GPS

3.2. DEFINICIÓN DE ESPECIFICACIONES

En esta fase primeramente se definen los actores necesarios para el desarrollo del prototipo de collar para monitoreo de mascotas, con lo que se desarrollará el diagrama de actores.

3.2.1. Actores

- Usuario Principal: Serán las mascotas las cuales llevarán puesto el prototipo de collar el cual mandara su localización al Internet.
- Usuario Secundario: Dueños o propietarios, serán las personas que podrán visualizar los datos que mande el dispositivo de geolocalización.

- Los datos de la ubicación serán enviados mediante SMS al número del usuario secundario.
- La aplicación consumirá la información del SMS para ser visualizados en la aplicación.
- El prototipo de collar podrá ser manipulado mediante la aplicación que tiene el usuario secundario instalado en su celular.
- En caso de que la mascota fuera encontrada por otra persona podrá pedir ayuda mediante un led que tendrá el collar con un botón, este mandara un SMS a su dueño o propietario para alertarlo de la situación.

3.3.2. Establecimiento de proyecto

Se realizará el prototipo de collar que será de uso para mascotas, donde se utilizará la placa Arduino Lylipad y la placa A9G de ai thinker, entre los más principales y también estará basado tecnología Android donde se utilizará una aplicación echa para el prototipo de collar.

El dispositivo será desarrollado por una persona que realizará el rol de los participantes en esta etapa: desarrollador, diseñador, jefe del proyecto y pruebas.

Para la recolección de información de la población a la cual va dirigida el prototipo de collar se debe identificar cada parte del problema que se pretende solucionar:

- ✓ Los robos de mascotas debido a la raza y por ende al lucro de ellos.
- ✓ Al abandono de las mascotas por propietarios irresponsables.
- ✓ Pérdida de las mascotas por descuido de los propietarios o por que la mascota es muy inquieta y tiende a desaparecer por largos lapsos de tiempo.
- ✓ Al momento de ser encontrada la mascota por una persona ajena, es casi imposible encontrar al dueño sin la información adecuada.

3.3.3. Planificación

La planificación que se llevara a cabo se muestra en la siguiente tabla.

Iteración	Desarrollo	duración	Fecha de inicio
Primera	Implementación entre placa Arduino Lylipad y A9G de ai thinker	15	13/04/2021

Segunda	Implementación de circuitos	5	28/04/2021
Tercera	Integración de los módulos anteriores	20	18/05/2021
Cuarta	Diseño de la aplicación para la funcionalidad del prototipo de collar	15	27/05/2021
Quinta	Conexión del collar con la aplicación	15	05/06/2021

Tabla 3.1 Cronograma de Desarrollo

Fuente: (Elaboración Propia, 2021).

3.4. DISEÑO EN DETALLE

En esta fase detallaremos los componentes de software y hardware que mencionamos anteriormente para el desarrollo del prototipo de collar.

3.4.1. Configuración

Componentes de hardware que se usara para el desarrollo del prototipo de collar.

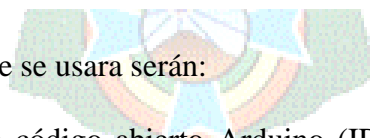
- ARDUINO LILYPAD.
 - a) Microcontrolador: ATmega 328P
 - b) Voltaje de funcionamiento: 2.7-5.5 V
 - c) Voltaje de entrada: 2.7-5.5 V
 - d) Pines digitales de I/O : 14
 - e) Canales PWM: 6
 - f) Canales de entrada analógica: 6
 - g) Corriente CC por Pin I/O: 40 mA
 - h) Memoria Flash: 16 KB (de los cuales 2 KB son utilizados por el gestor de arranque)
 - i) SRAM: 1 KB
 - j) EEPROM: 512 bytes
 - k) Velocidad de reloj: 16

- MODULO A9G GSM/GPRS/GPS

- Voltaje de alimentación USB: 5V DC
 - Voltaje de alimentación VBAT: 3.8 ~ 4.2V DC
 - Voltaje de I/O: 2.8 VDC
 - Corriente stand-by menor a 2mA
 - Modelo placa: PUDDING A9G-DEV
 - Modulo principal (SoM): Ai-Thinker A9G
 - Chip CPU(SoC): RDA8955L (RISC 32 bit)
 - Chip GSM/GPRS: RDA6625E
 - Chip GPS/GNSS: GK9501
 - Conector antena GSM/GPRS IPEX-1
 - Conector antena GPS IPEX-1
 - Incluye antena GSM y antena GPS
 - Micrófono en placa
 - Unidad de administración de energía: administración de carga de la batería de litio, DC-DC y LDO integrados, voltaje de IO variable
 - Socket para tarjeta micro-SIM
 - Socket para MicroSD Card
 - Pulsador de encendido
 - Pulsador de reinicio
 - 2 LEDs indicadores en placa
 - Temperatura de trabajo: -30° a +80°C
 - Dimensiones: 41*22*5mm
 - Peso: 23 gramos
- **CONECTIVIDAD:**
 - Cuatro bandas GSM/GPRS: 850,900,1800,1900 MHz
 - Compatible con sistemas GNSS: GPS/BDS/GLONASS/GALILEO/QZSS/SBAS
 - Soporta llamadas de voz
 - Servicio de mensajería SMS
 - GPRS clase 10
 - Sensibilidad <-105dBm

- Datos por GPRS, velocidad máx: 85.6 Kbps (Descarga), 42.8 Kbps (subida)
- Utiliza el estándar de comandos AT y TCP/IP
- Estándar GSM 07.07 y 07.05, Comandos AT y comandos adicionales de Ai-Thinker
- Soporte para audio digital y analógico (HR, FR, EFR, AMR)
- Certificaciones: FCC, CE, CCC
- **MCU (SDK mode):**
- SoC RDA8955 32 bit RISC core, frecuencia hasta 312 MHz, con caché de instrucciones 4kB, caché de datos 4kB
- LASH: 32Mb (4MB) SPI NOR FLASH
- RAM: 32Mb (4MB) DDR PSRAM
- Hasta 29 GPIO con espaciado de 2.45 mm
- RTC (reloj de tiempo real) con alarma
- 1x interfaz de dispositivo USB1.1
- 2x interfaz UART con control de flujo (+1 puerto serie de descarga/depuración HST_TX, HST_RX)
- 2x Interfaz SPI
- 3x Interfaz I2C
- 1x Controlador SDMMC (interfaz)
- 2x Interfaces ADC 10 bits
- ADC micrófono: 8kHz, 13bits/muestra
- DAC audio: 48kHz, 16bits/muestra

- Interruptor switch de 3 patitas, batería lipo de 5v, cables, protoboard, Sim card, que son elementos simples pero necesarios para el desarrollo del proyecto, además que estos componentes se usaran para las primeras pruebas preliminares a realizar a lo largo del desarrollo del prototipo de collar.



Los componentes de software que se usara serán:

- Ide Arduino. El software de código abierto Arduino (IDE) facilita escribir código. El entorno está escrito en Java y se basa en Procesamiento y otro software de código abierto. Este software se puede utilizar con cualquier placa Arduino, se puede adquirir descargando de su página oficial de internet, además de las diferentes librerías que se

usara para los módulos necesarios.

- Ide Android Studio. Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores. Este IDE nos ayudara a desarrollar la aplicación para la recepción y visualización de nuestro prototipo de collar.

3.4.2. Planificación

Las metas a detalle que cumplirá cada actor se muestran en la siguiente imagen, donde se tomó en cuenta el modelo de actores definido en la fase anterior, donde se definía a los actores involucrados en el desarrollo del prototipo de collar.

También se especifica lo necesario que se debe tener para el buen funcionamiento y buen uso del prototipo de collar para tener la idea clara de lo que se quiere desarrollar.

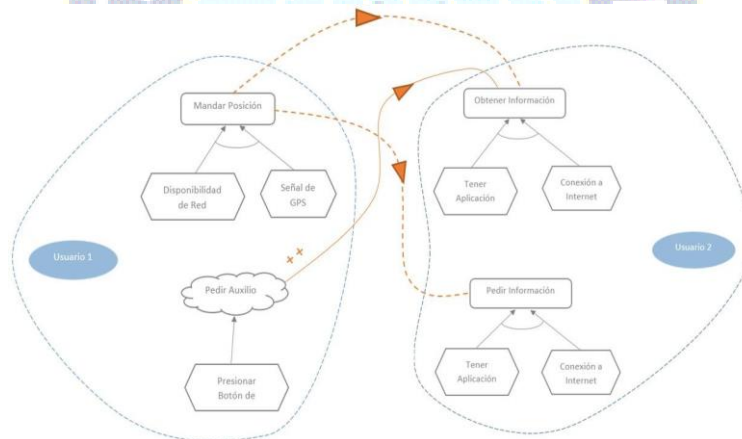


Figura 3.4 Diagrama de metas de actores

Fuente: (Elaboración Propia, 2021).

Para la clasificación de requisitos basado en el anterior grafico será:

Requisitos	Prioridad
Mandar SMS para recibir Posición	1
Visualizar Información	2

Mandar ubicacion	3
-------------------------	---

Tabla 3.2 Clasificación de Requisitos
Fuente: (Elaboración Propia, 2021).

3.5. IMPLEMENTACIÓN

Se hará una planificación de implementación en detalle para el análisis de requerimientos profundos y se modelará los diferentes eventos que pudiera tener el prototipo de collar.

3.5.1. Planificación

Se hará una guía de tarea de lo planificado en la fase de Diseño Global, para verificar los componentes de hardware y software que se usará para el prototipo de collar.

Tarea	DESSAROLLO
Tarea 1	<ul style="list-style-type: none"> ✓ Conexión de la placa A9G de ai thinker con el Arduino Lilypad ✓ Verificar el buen funcionamiento y código con el IDE Arduino
Tarea 2	<ul style="list-style-type: none"> ✓ Desarrollo de la aplicación en el IDE Android Studio para la funcionalidad del prototipo de collar.
Tarea 3	<ul style="list-style-type: none"> ✓ Conexión de la aplicación con el prototipo. ✓ Verificar si hay los consumos necesarios para la aplicación.

Tabla 3.3 Tareas de Desarrollo
Fuente: (Elaboración Propia, 2021).

Para el usuario secundario que visualizará y pedirá la posición del prototipo de collar tendrá la siguiente interfaz en la aplicación.

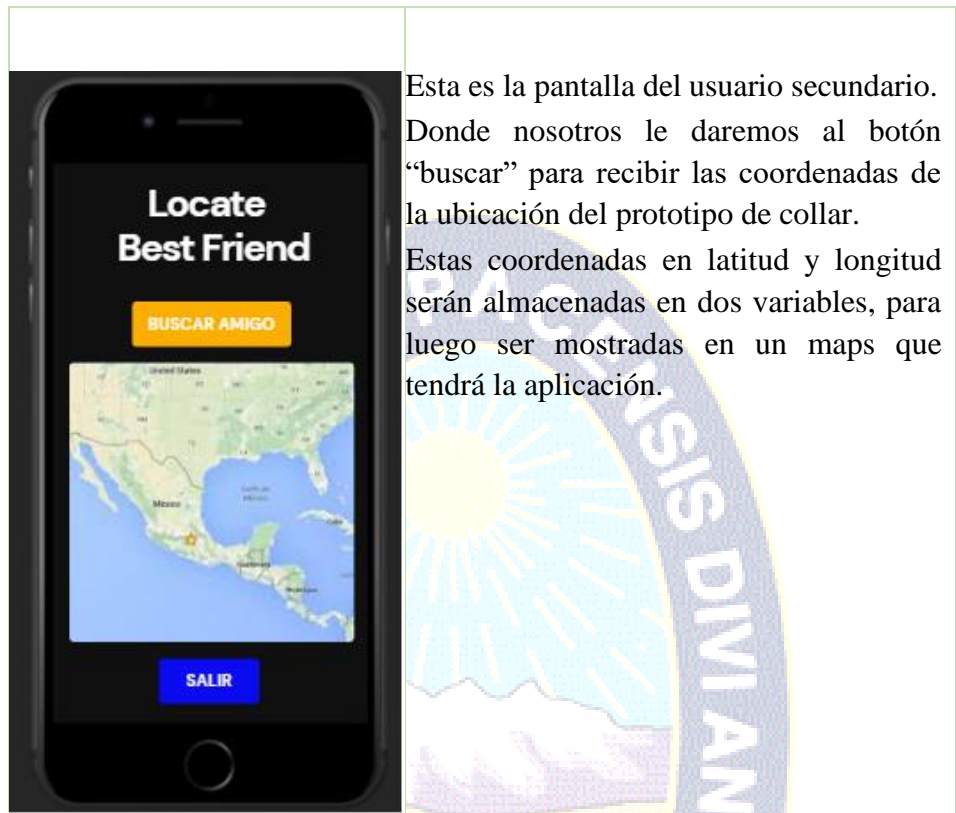


Tabla 3.5 Interfaz de Aplicación
Fuente: (Elaboración Propia, 2021)

3.5.2. Diseño detallado

En el siguiente grafico se detalla los eventos que podría pasar de acuerdo a los usuarios tomados, con un diagrama de actividades.



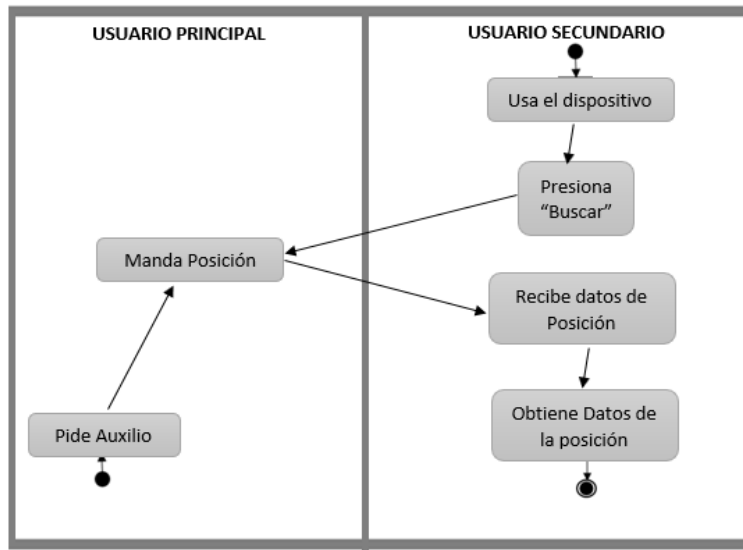


Figura 3.6 Diagrama de Actividades de Actores
Fuente: (Elaboración Propia,2021)

El siguiente gráfico proporciona una vista de alto nivel de los componentes dentro del prototipo de collar con geolocalización, los componentes que se muestran son de hardware como de software.

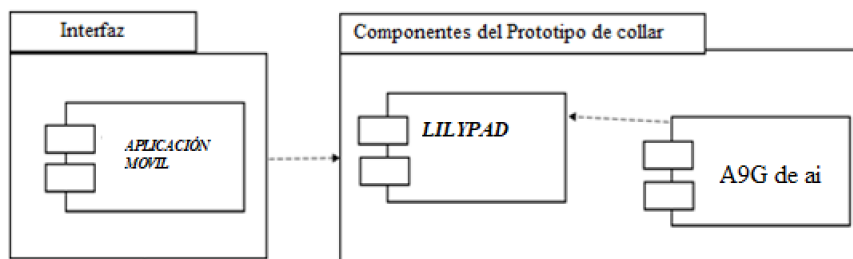


Figura 3.7 Diagrama de Componentes
Fuente: (Elaboración Propia,2021)

3.6. TEST UNITARIO

Se hará una verificación de los componentes de hardware mencionados en la fase anterior que es la fase de Implementación, generando el documento correspondiente.

3.6.1. Tarea 1

Conexión del Módulo A9G de ai con el Arduino Lilypad.

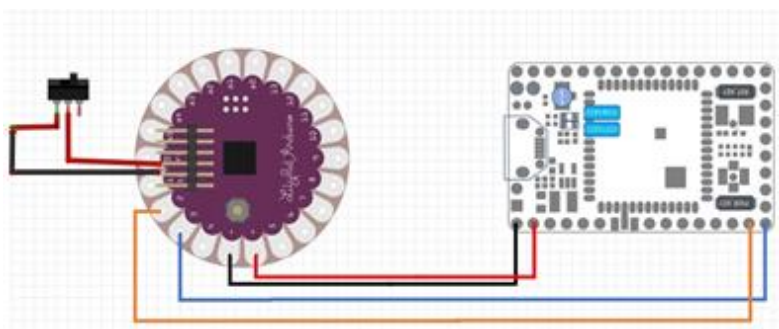


Figura 3.8 Arduino Lilypad conectado con A9G de ai
Fuente: (Elaboración propia, 2021).

El resultado detallado de la prueba conexión física, se describe a continuación en la tabla.

Documento de Prueba 1	
Modulo	A9G de ai
Actividad	<ul style="list-style-type: none"> ➤ Se obtendrá la posición de la mascota que tenga puesto el prototipo de collar, obteniendo su latitud y longitud ➤ Se pudo verificar el correcto funcionamiento del módulo se hizo la prueba con comandos, así verificamos conectividad a la red de telefonía, poder hacer llamadas y enviado de SMS. ➤ Se hizo la conexión para verificar el uso de la función GPRS. ➤ Se verifico que este módulo puede trabajar con cualquier chip de telefonía, en este caso se trabajara con la empresa de Entel.
Observación	La conexión respondió correctamente
Detalle u observación	Generalmente este módulo es compatible con cualquier Arduino.

Tabla 3.4 Prueba de conexión Arduino Lilypad y A9G de ai
Fuente: (Elaboración Propia).

3.6.2. Tarea 2

Iniciaremos desarrollando dos pantallas una la cual llamaremos “Inicio” y la segunda que llamaremos “Proceso”.

Desarrollo de interfaz de la primera pantalla “INICIO”.



Figura 3.9 Primera Pantalla “Inicio”

Fuente: (Elaboración propia, 2021).

En esta figura se muestra la primera pantalla que se tendrá al ingresar a la aplicación se desarrollará de la siguiente manera:

- **Text**, el cual tendrá el título dado por nuestra propia creación en este caso se le dará el nombre de “*Locate Best Friend*”.
- **Imagen**, es una imagen decorativa al tema de la aplicación para poder distinguir mediante algo visual de que trata la aplicación.
- **Button(Iniciar)**, nos servirá para ingresar a la siguiente pantalla(*layout*) en la cual nosotros tendremos más funciones.

En el siguiente gráfico se muestra el interfaz de la segunda pantalla “Proceso”.



Figura 3.10 Segunda pantalla “Proceso”

Fuente: (Elaboración Propia,2021).

En esta figura se muestra la segunda pantalla que se obtendrá después de oprimir el *Button* “INICIAR” y se desarrollará de la siguiente manera:

- **Text**, el cual tendrá el título dado por nuestra propia creación en este caso se le dará el nombre de “*Locate Best Friend*”.
- **Button(Buscar Amigo)**, el cual nos servirá para obtener las coordenadas de nuestra mascota y así poder reflejarlas en el mapa siguiente.
- **Map**, es un mapa en el cual se mostrará la ubicación de nuestra mascota con la ubicación ya obtenida por el anterior *Button* utilizado.
- **Button(Salir)**, este nos servirá para salir de nuestra aplicación cuando dejemos de utilizarla.

El resultado detallado de la prueba conexión física, se describen a continuación en la tabla.

Documento de Prueba 2

Actividad	Se hizo el desarrollo de la aplicación móvil para la funcionalidad del prototipo de collar, con el respectivo IDE Android Studio, pudiendo consumir y reflejar los mensajes como coordenadas y así poder mostrarlas en el Maps.
Observación	El Desarrollo de la aplicación fue exitosa.
Detalle u observación	Esta aplicación solo es para el uso del prototipo de collar

Tabla 3.5 Desarrollo de la aplicación móvil

Fuente: (Elaboración Propia,2021).

3.6.3. Tarea 3

Conexión de la aplicación con el prototipo de collar, se verifica que la aplicación mande un mensaje al oprimir el *Button* “Buscar Amigo”.

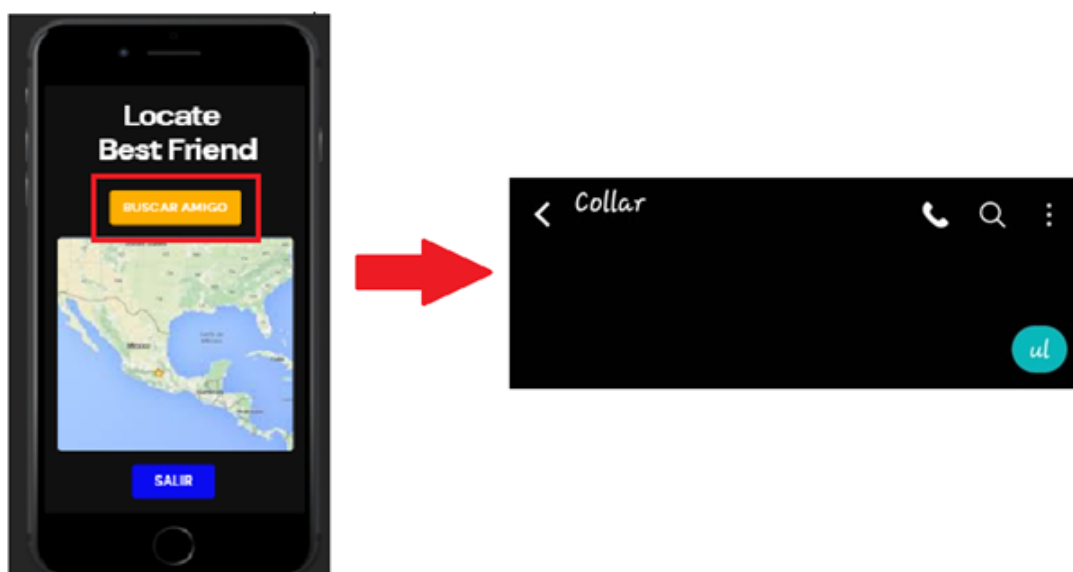


Figura 3.11 Verificación de Aplicación

Fuente: (Elaboración Propia, 2021)

- Para la elaboración de la aplicación verificamos que tengan el código necesario para ser reconocido por el prototipo de collar.
- El código que la aplicación deberá mandar es de “u1” para que el prototipo entienda el

mensaje y mande las coordenadas.

- Una vez mandadas las coordenadas la aplicación deberá mostrar esas coordenadas en el maps indicando la ubicación de nuestra mascota.

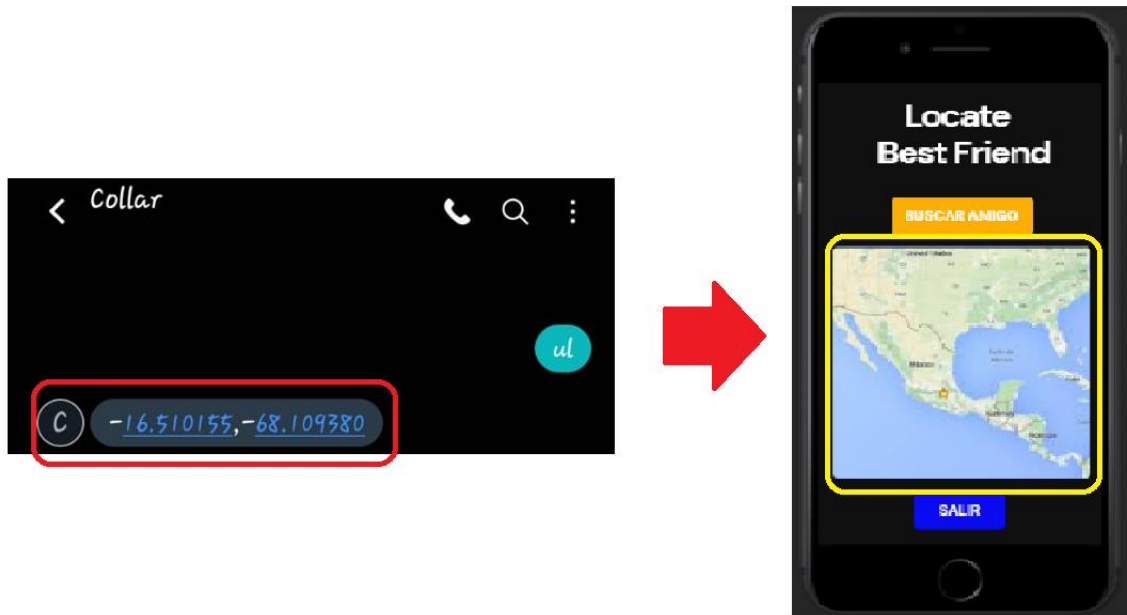


Figura 3.12 Consumo de las coordenadas por la aplicación.

Fuente: (Elaboración Propia, 2021).

Se selecciona el widget y se pondrá el nombre y se utilizará un pin virtual para el uso en la aplicación.

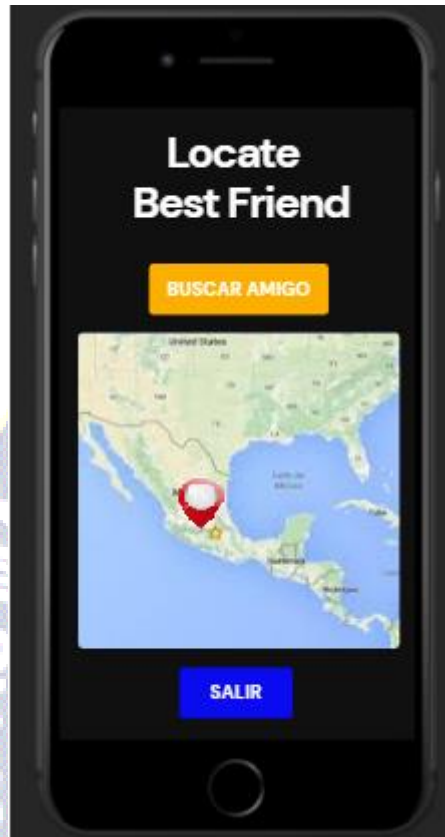


Figura 3.13 Configuración de Widget

Fuente: (Elaboración Propia, 2021).

Se selecciona el widget y se hará la configuración donde se pondrá el nombre y se usará un pin virtual para su uso en la aplicación.

El código utilizado para la implementación del consumo y envío del SMS es el siguiente:

```
public void onReceive(Context context, Intent intent) {  
    if (intent.getAction().equals(Telephony.Sms.Intents.SMS_RECEIVED_ACTION)) {  
        String smsSender = "";  
        String smsBody = "";  
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {  
            SmsMessage smsMessage = Telephony.Sms.Intents.getMessagesFromIntent(intent)[0];  
            smsSender = smsMessage.getDisplayOriginatingAddress();  
            smsBody += smsMessage.getMessageBody();  
        }  
    }  
}
```

```

} else {
    Bundle smsBundle = intent.getExtras();
    if (smsBundle != null) {
        Object[] pdus = (Object[]) smsBundle.get("pdus");
        if (pdus == null) {
            return;
        }
        SmsMessage[] messages = new SmsMessage[pdus.length];
        for (int i = 0; i < messages.length; i++) {
            messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
            smsBody += messages[i].getMessageBody();
        }
        smsSender = messages[0].getOriginatingAddress();
    }
}

if (smsSender.contains(serviceProviderSmsCondition)) {
    if (listener != null) {
        String[] smsWords = smsBody.split(" ");
        listener.onTextReceived(smsWords[smsWords.length-1], smsSender);
    }
}
}
}
}

```

En este código podemos observar la lógica que seguimos para el funcionamiento de esta aplicación, cabe recalcar que una de las principales condiciones para que esta aplicación funcione es la de tener una versión de Android mayor a (4.4 – 4.4.4) *KITKAT*.

El resultado detallado de la prueba de aplicación se describe a continuación en la tabla.

Documento de Prueba 3	
Modulo	Conexión de aplicación con prototipo de collar
Actividad	Se hizo la conexión correcta de la aplicación con el prototipo para que esta pueda enviar un mensaje oprimiendo el <i>button</i> y posteriormente se pueda consumir el mensaje de llegada a la aplicación y poder mostrarla mediante un widget.
Observación	La conexión respondió correctamente.
Detalle u observación	Para usar la aplicación el dispositivo de celular debe contar con una versión de Arduino 5.0 o superior.

Tabla 3.6 Prueba de conexión aplicación con prototipo collar

Fuente: (Elaboración Propia, 2021).

3.7. INTEGRACIÓN

Para la integración completa del prototipo de collar de geolocalización se llevará a cabo mediante la planeación que se hizo en la fase del Diseño Global y se verificara que esté totalmente funcional la integración que se realizará y que no tenga fallos ni inconvenientes, adicionalmente se documentará como se hizo en la anterior fase.

3.7.1. Integración completa del prototipo de collar con tecnología de geolocalización

Se hizo la integración completa del prototipo, terminando así el módulo, en la siguiente imagen se mostrará la conexión de todos los componentes y el diagrama lógico de los mismos, también se verificará su correcto funcionamiento y se hará el documento correspondiente, así evitar fallos e inconvenientes para la fase final del desarrollo del prototipo de collar.

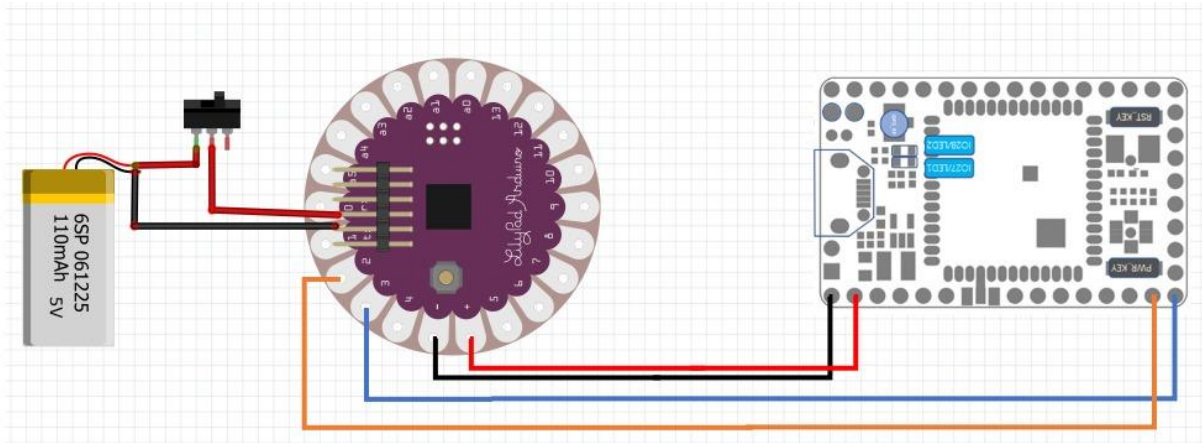


Figura 3.14 Integración Completa del Dispositivo
Fuente: (Elaboración Propia, 2021).

En el Siguiete grafico se muestra el Diagrama lógico de Arduino Lilypad con A9G de ai y su batería 5V.

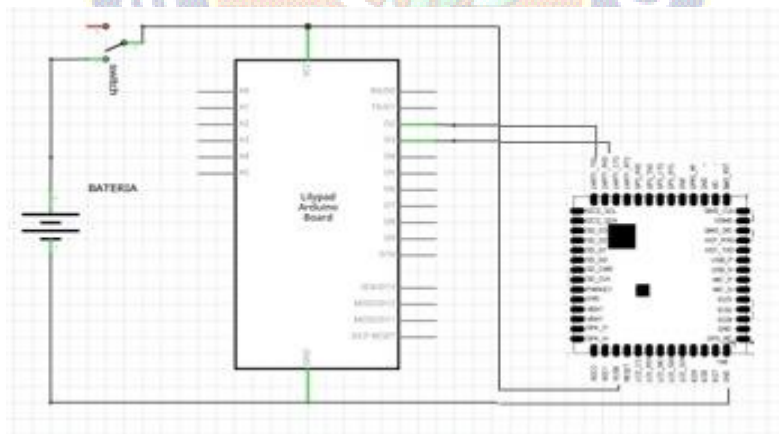


Figura 3.15 Diagrama lógico de Arduino Lilypad con A9G de ai y su batería 5V
Fuente: (Elaboración Propia, 2021).

El código utilizado para la integración entre el Arduino Lilypad con el módulo A9G de ai y su batería 5V fue el siguiente:

```

SoftwareSerial localizador(3, 2); //definimos los pines de conexion tx,rx

String Ndueno = "78816330";
String Nmascota = "KAYLA";
String ubi = ""; // variable para habilitar la ubicación gps
int ubigps = 0;
String SMSubicacion = ""; // variable que almacena la respuesta total de la ubicacion
String enviarUbicacion = ""; //variable que almacena la ubicacion en coordenadas latitud y longitud
String sms = ""; // variable que almacena el mensaje recibido
String destinatario = "AT+CMGS="; // comando para configurar el numero al cual se enviara el sms
String rEnvio=""; // variable que almacena la respuesta del envio de sms

void setup() {
  pinMode(13, OUTPUT); // definimos el pin 13 como salida
  Serial.begin(115200); // inicia la comunicacion serial del arduino
  localizador.begin(115200); //inicia la comunicacion serial con el módulo A9G
  iniciarLocalizador(); //método que inicia el módulo A9G
}

```

Como se puede observar en el recuadro tenemos como en el (*void setup*) definimos las variables que se utilizaran, y los métodos que iniciaran las diferentes comunicaciones que se tendrá entre placa y modulo.

El resultado detallado de la prueba conexión física de los módulos, se describen a continuación en la tabla.

Documento de Prueba 5	
Modulo	Integración Completa del prototipo
Actividad	Se hizo la conexión para verificar el correcto funcionamiento del módulo A9G de ai, además de la batería externa.
Observación	La integración respondió correctamente, comprobando que no existe fallos.

Detalle u observación	El módulo A9G de ai ya cuenta con el SIM y El GPS incluidos.
------------------------------	--

Tabla 3.7 Prueba de Integración Completa del Prototipo de collar

Fuente: (Elaboración Propia, 2021).

3.8. TEST OPERACIONAL DEL SISTEMA

Se llega a finalizar el desarrollo del prototipo de collar basado en tecnología GPS, realizando las pruebas respectivas del funcionamiento de la aplicación y del hardware en un ambiente real.

3.8.1. Pruebas

Como se pudo observar en esta fase se contempla la ejecución de pruebas y de aceptación. Dando paso a la aprobación final y una tabla de resultados.

Documento Resultados Finales	
Datos de GPS	Para los datos del GPS se tomaron en cuenta: Datos Validos. Valores dentro el límite aceptado.
Conexión del SIM	Para las pruebas de tiempo se tomó en cuenta los siguientes criterios: La posición del prototipo de collar enviara su ubicación en latitud y longitud mediante sms.
Prototipo de Collar	Para esta prueba se tomó en cuenta: La usabilidad del prototipo. El poder mandar un SMS para la respuesta de la ubicación.

<p>Aplicación para Visualizar los datos</p>	<p>Para las pruebas de visualización de la posición del prototipo se tomó en cuenta:</p> <ul style="list-style-type: none"> ➤ Una interfaz amigable en la aplicación de celular desarrollada para la visualización de la posición en un Widget Map.
--	--

Tabla 3.8 Documento de Resultados Finales.
Fuente: (Elaboración Propia, 2021).



CAPITULO IV

4. EVALUACION DE RESULTADOS

4.1. ANÁLISIS Y RESULTADOS

A continuación, se presenta el desarrollo de las pruebas y el análisis de los datos obtenidos a través del prototipo de collar de geolocalización dando lugar a la prueba de la hipótesis planteada.

La hipótesis planteada es:

“El prototipo de collar con geolocalización basada en tecnologías GPS permite rastrear la ubicación de una MASCOTA, con un margen de error menor igual a 3 metros.”

Para demostrar la hipótesis se utilizará la distribución Normal o Gaussiana, la cual frecuentemente es utilizada en las aplicaciones estadísticas, se utiliza con frecuencia o normalidad ya que muchos fenómenos tienden a parecerse en su comportamiento a esta distribución.

Para que la hipótesis sea aceptada se utilizará un nivel de confianza de 95% y un nivel de significancia del 5%.

Se plantea la H_0 : hipótesis nula y la H_1 : que es la hipótesis alternativa.

H_0 : El prototipo de collar con geolocalización basada en tecnologías GPS permite rastrear la ubicación de una MASCOTA, con un margen de error mayor a 3 metros.

H_1 : El prototipo de collar con geolocalización basada en tecnologías GPS permite rastrear la ubicación de una MASCOTA, con un margen de error menor igual a 3 metros.

4.1.1. Cálculo de datos

Para el cálculo de datos se hará una conversión de unidades puesto que los datos obtenidos latitud y longitud del prototipo de collar trabajan en grados decimales y la hipótesis está planteada en metros, entonces estos datos serán transformados en datos UTM que son medidas de localización trabajadas en metros, de la misma manera para verificar la medida tomada de Google Maps será transformado en datos UTM y hacer los cálculos respectivos

para la prueba de hipótesis.

N°	Coordenadas de Google Eart Pro		Observación	Coordenadas de Google Eart Pro	
	Grados Decimales			UTM (metros)	
	Latitud	Longitud		Este	Norte
1	-3,90522469	-79,57483494	Domicilio	594308,00	8177105,00
2	-3,90522469	-79,57483494	UMSA	592844,61	8175015,81
3	-3,90522469	-79,57483494	Carrera de Informática	592888,00	8175135,00
4	-3,90522469	-79,57483494	Plaza Del Estudiante	592713,00	8175085,00
5	-3,90522469	-79,57483494	Librería Don Bosco Prado	592704,00	8175166,00
6	-3,90522469	-79,57483494	Hotel Plaza	592700,00	8175213,00
7	-3,90522469	-79,57483494	Dumbo	592561,10	8175396,36
8	-3,90522469	-79,57483494	Cine Moje Campero	592564,01	8175468,96
9	-3,90522469	-79,57483494	Plaza Camacho	592668,65	8175522,41
10	-3,90522469	-79,57483494	Mercado Camacho	592730,24	8175489,15
11	-3,90522469	-79,57483494	Cruz Roja Boliviana	592665,38	8175562,52
12	-3,90522469	-79,57483494	Banco Nacional de Bolivia	592568,88	8175589,74
13	-3,90522469	-79,57483494	Banco Unión	592506,72	8175621,64
14	-3,90522469	-79,57483494	Banco Ganadero	592439,08	8175655,23
15	-3,90522469	-79,57483494	Banco Nacional de Bolivia	592378,70	8175683,08
16	-3,90522469	-79,57483494	Plaza Obelisco	592288,28	8175727,99

17	-3,90522469	-79,57483494	Facultad de Ingeniería UMSA	592313,78	8175752,24
18	-3,90522469	-79,57483494	Iglesia de San Francisco	592092,00	8175962,00
19	-3,90522469	-79,57483494	Mercado Lanza	592113,00	8176095,00
20	-3,90522469	-79,57483494	Plaza Alonzo de Mendoza	591956,00	8176309,00
21	-3,90522469	-79,57483494	Museo Tambo Quirquincho	591991,00	8176273,00
22	-3,90522469	-79,57483494	Plaza Eguino	591691,00	8176245,00
23	-3,90522469	-79,57483494	Teleférico Naranja Estación Central	591297,00	8176437,00
24	-3,90522469	-79,57483494	Puente Vita	591262,97	8176293,68
25	-3,90522469	-79,57483494	Teleférico Rojo Est. Central	591311,00	8176504,00
26	-3,90522469	-79,57483494	Estación Central	591350,00	8176525,00
27	-3,90522469	-79,57483494	Hospital Luis Uría	594276,00	8177365,00
28	-3,90522469	-79,57483494	Hospital Arcoiris	593905,00	8177315,00
29	-3,90522469	-79,57483494	Plaza Villaroel	593762,00	8177242,00
30	-3,90522469	-79,57483494	Teleférico Naranja Estación Villaroel	593692,00	8177245,00
31	-3,90522469	-79,57483494	Instituto Técnico Ayacucho	593751,00	8177160,00
32	-3,90522469	-79,57483494	Club Tenis Sucre	593774,00	8176621,00
33	-3,90522469	-79,57483494	Universidad Loyola	593788,00	8176366,00
34	-3,90522469	-79,57483494	Monumento Busch	593781,00	8176174,00
35	-3,90522469	-79,57483494	Hospital Obrero	594105,00	8175614,00
36	-3,90522469	-79,57483494	Colegio Hugo Dávila	593977,00	8175621,00
37	-3,90522469	-79,57483494	Clínica Europa	593830,00	8175626,00
38	-3,90522469	-79,57483494	Estadio Hernando Siles	593705,00	8175617,00
39	-3,90522469	-79,57483494	Pollos Copacabana	593525,00	8175549,00
40	-3,90522469	-79,57483494	Mercado Ketal	593808,00	8175281,00

Tabla 4.1 Coordenadas Google Eart Pro y Conversión a unidades UTM.

Fuente: (Elaboración Propia, 2021).

En esta tabla se visualizará los datos tomados por el prototipo de collar con geolocalización y las coordenadas transformadas en UTM.

N°	Coordenadas del Prototipo de collar		Observación	Coordenadas del Prototipo de collar	
	Grados Decimales			UTM (metros)	
	Latitud	Longitud		Este	Norte
1	-16,48408127	-68,12017765	Domicilio	594307,00	8177104,00
2	-16,48472461	-68,12151894	UMSA	592844,91	8175014,81
3	-16,48469366	-68,12217722	Carrera de Informática	592890,00	8175138,00
4	-16,48544634	-68,121615	Plaza Del Estudiante	592713,65	8175084,90
5	-16,49032773	-68,1213804	Librería Don Bosco Prado	592705,00	8175168,00
6	-16,49264142	-68,12126147	Hotel Plaza	592700,00	8175215,00
7	-16,49438339	-68,12129492	Dumbo	592561,90	8175397,56
8	-16,49942982	-68,11826678	Cine Moje Campero	592565,01	8175467,30
9	-16,49935756	-68,11947836	Plaza Camacho	592667,74	8175522,41
10	-16,49933875	-68,12080868	Mercado Camacho	592731,54	8175487,35
11	-16,49939708	-68,122	Cruz Roja Boliviana	592664,54	8175563,77
12	-16,50002596	-68,12366423	Banco Nacional de Bolivia	592567,95	8175587,93
13	-16,50244267	-68,12099985	Banco Unión	592507,31	8175623,34
14	-16,49912038	-68,13385089	Banco Ganadero	592440,08	8175655,20
15	-16,49885959	-68,134438	Banco Nacional de Bolivia	592377,54	8175684,10
16	-16,49843915	-68,13526024	Plaza Obelisco	592289,98	8175730,99
17	-16,49824293	-68,13502557	Facultad de Ingeniería UMSA	592315,12	8175752,59
18	-16,49634882	-68,13713285	Iglesia de San Francisco	592093,10	8175965,10
19	-16,49515659	-68,13692171	Mercado Lanza	592114,20	8176094,90
20	-16,49321746	-68,13839833	Plaza Alonzo de Mendoza	591957,51	8176310,10
21	-16,49353536	-68,1380691	Museo Tambo Quirquincho	591992,50	8176274,78
22	-16,4938079	-68,14088422	Plaza Eguino	591691,90	8176245,91
23	-16,49208309	-68,14459817	Teleférico Naranja Estación Central	591296,30	8176438,41
24	-16,49340107	-68,14488029	Puente Vita	591265,57	8176292,73

25	-16,49148506	-68,14444303	Teleférico Rojo Est, Central	591313,14	8176504,50
26	-16,49130267	-68,14407263	Estación Central	591352,76	8176524,51
27	-16,48359228	-68,11672843	Hospital Luis Uría	594275,12	8177364,94
28	-16,48408127	-68,12017765	Hospital Arcoíris	593906,71	8177312,45
29	-16,48472461	-68,12151894	Plaza Villaroel	593763,23	8177241,90
30	-16,48469366	-68,12217722	Teleférico Naranja Estación Villaroel	593692,98	8177245,63
31	-16,48544634	-68,121615	Instituto Técnico Ayacucho	593752,63	8177162,10
32	-16,49032773	-68,1213804	Club Tenis Sucre	593775,32	8176621,96
33	-16,49264142	-68,12126147	Universidad Loyola	593786,90	8176365,94
34	-16,49438339	-68,12129492	Monumento Busch	593783,49	8176177,24
35	-16,49942982	-68,11826678	Hospital Obrero	594103,26	8175613,54
36	-16,49935756	-68,11947836	Colegio Hugo Dávila	593973,98	8175622,10
37	-16,49933875	-68,12080868	Clínica Europa	593832,00	8175624,80
38	-16,49939708	-68,122	Estadio Hernando Siles	593704,82	8175618,90
39	-16,50002596	-68,12366423	Pollos Copacabana	593526,89	8175550,10
40	-16,50244267	-68,12099985	Mercado Ketal	593810,10	8175281,50

Tabla 4.2 Coordenadas del Prototipo de collar y Conversión a unidades UTM.

Fuente: (Elaboración Propia, 2021).

4.1.1.1. Distancia más corta entre dos puntos

Para hallar la distancia que hay entre el punto real dado por las coordenadas de Google Eart Pro y el punto tomado por el prototipo de collar con geolocalización se hace un cálculo buscando la distancia más corta entre dos puntos en un plano, dada por la siguiente formula, este es el dato que se usara para los cálculos posteriores.

$$d = \sqrt{(X_{CG} - X_{CD})^2 + (Y_{CG} - Y_{CD})^2}$$

N°	Observación	Coordenadas de Google Eart Pro		Coordenadas de Prototipo de collar		d
		UTM (metros)		UTM (metros)		
		Este	Norte	Este	Norte	
1	Domicilio	594308,00	8177105,00	594307,00	8177104,00	1,41
2	UMSA	592844,61	8175015,81	592844,91	8175014,81	1,04
3	Carrera de Informática	592888,00	8175135,00	592890,00	8175138,00	3,61
4	Plaza Del Estudiante	592713,00	8175085,00	592713,65	8175084,90	0,66
5	Librería Don Bosco Prado	592704,00	8175166,00	592705,00	8175168,00	2,24
6	Hotel Plaza	592700,00	8175213,00	592700,00	8175215,00	2,00
7	Dumbo	592561,10	8175396,36	592561,90	8175397,56	1,44
8	Cine Moje Campero	592564,01	8175468,96	592565,01	8175467,30	1,94
9	Plaza Camacho	592668,65	8175522,41	592667,74	8175522,41	0,91
10	Mercado Camacho	592730,24	8175489,15	592731,54	8175487,35	2,22
11	Cruz Roja Boliviana	592665,38	8175562,52	592664,54	8175563,77	1,51
12	Banco Nacional de Bolivia	592568,88	8175589,74	592567,95	8175587,93	2,03
13	Banco Unión	592506,72	8175621,64	592507,31	8175623,34	1,80
14	Banco Ganadero	592439,08	8175655,23	592440,08	8175655,20	1,00
15	Banco Nacional de Bolivia	592378,70	8175683,08	592377,54	8175684,10	1,54
16	Plaza Obelisco	592288,28	8175727,99	592289,98	8175730,99	3,45
17	Facultad de Ingeniería UMSA	592313,78	8175752,24	592315,12	8175752,59	1,38
18	Iglesia de San Francisco	592092,00	8175962,00	592093,10	8175965,10	3,29
19	Mercado Lanza	592113,00	8176095,00	592114,20	8176094,90	1,20
20	Plaza Alonzo de Mendoza	591956,00	8176309,00	591957,51	8176310,10	1,87
21	Museo Tambo Quirquincho	591991,00	8176273,00	591992,50	8176274,78	2,33
22	Plaza Eguino	591691,00	8176245,00	591691,90	8176245,91	1,28
23	Teleférico Naranja Estación Central	591297,00	8176437,00	591296,30	8176438,41	1,57
24	Puente Vita	591262,97	8176293,68	591265,57	8176292,73	2,77
25	Teleférico Rojo Estación Central	591311,00	8176504,00	591313,14	8176504,50	2,20

26	Estación Central	591350,00	8176525,00	591352,76	8176524,51	2,80
27	Hospital Luis Uría	594276,00	8177365,00	594275,12	8177364,94	0,88
28	Hospital Arcoíris	593905,00	8177315,00	593906,71	8177312,45	3,07
29	Plaza Villaroel	593762,00	8177242,00	593763,23	8177241,90	1,23
30	Teleférico Naranja Estación Villaroel	593692,00	8177245,00	593692,98	8177245,63	1,17
31	Instituto Técnico Ayacucho	593751,00	8177160,00	593752,63	8177162,10	2,66
32	Club Tenis Sucre	593774,00	8176621,00	593775,32	8176621,96	1,63
33	Universidad Loyola	593788,00	8176366,00	593786,90	8176365,94	1,10
34	Monumento Busch	593781,00	8176174,00	593783,49	8176177,24	4,09
35	Hospital Obrero	594105,00	8175614,00	594103,26	8175613,54	1,80
36	Colegio Hugo Dávila	593977,00	8175621,00	593973,98	8175622,10	3,21
37	Clínica Europa	593830,00	8175626,00	593832,00	8175624,80	2,33
38	Estadio Hernando Siles	593705,00	8175617,00	593704,82	8175618,90	1,91
39	Pollos Copacabana	593525,00	8175549,00	593526,89	8175550,10	2,19
40	Mercado Ketal	593808,00	8175281,00	593810,10	8175281,50	2,16

Tabla 4.3 Distancia más corta entre dos puntos

Fuente: (Elaboración Propia).

4.1.1.2. Cálculo estadístico

Para el cálculo estadístico se determinará la región crítica a utilizar de la Función de la Curva de la Normal y se encontrar los valores “ \bar{x} ” y “ S ”, que son el promedio y la desviación estándar de la muestra

a) Determinación de región crítica

Para la hipótesis planteada se utilizará la prueba unilateral, debido a que la hipótesis tiene la siguiente forma.

$$H_0: \mu > 3$$

Entonces la región de aceptación será:

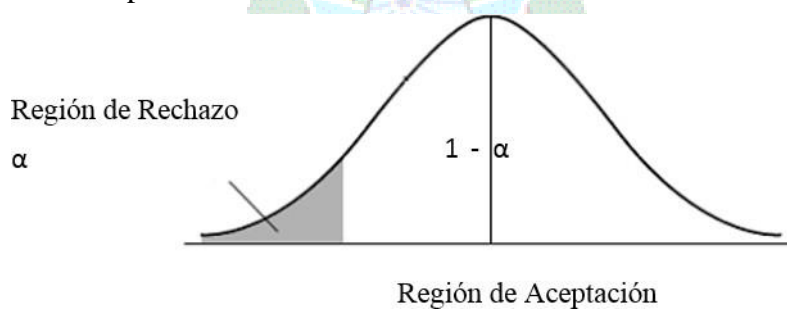


Figura 4.1 Región Crítica
Fuente:(Elaboración Propia, 2021)

Donde el nivel de confianza es del 95% y determinaremos Z.

$$95\% = 1 - \alpha$$

$$1 - \alpha = 0.95$$

Para hallar el valor se recurrirá a la tabla de la normal.

t	0.00	.01	.02	.03	.04	.05
0.0	.5000	.5040	.5080	.5120	.5160	.5199
0.1	.5398	.5438	.5478	.5517	.5557	.5596
0.2	.5793	.5832	.5871	.5910	.5948	.5987
0.3	.6179	.6217	.6255	.6293	.6331	.6368
0.4	.6554	.6591	.6628	.6664	.6700	.6736
0.5	.6915	.6950	.6985	.7019	.7054	.7088
0.6	.7257	.7291	.7324	.7357	.7389	.7422
0.7	.7580	.7611	.7642	.7673	.7704	.7734
0.8	.7881	.7910	.7939	.7967	.7995	.8023
0.9	.8159	.8186	.8212	.8238	.8264	.8289
1.0	.8413	.8438	.8461	.8485	.8508	.8531
1.1	.8643	.8665	.8686	.8708	.8729	.8749
1.2	.8849	.8869	.8888	.8907	.8925	.8944
1.3	.9032	.9049	.9066	.9082	.9099	.9115
1.4	.9192	.9207	.9222	.9236	.9251	.9265
1.5	.9332	.9345	.9357	.9370	.9382	.9394
1.6	.9452	.9463	.9474	.9484	.9495	.9505
1.7	.9554	.9564	.9573	.9582	.9591	.9599
1.8	.9641	.9649	.9656	.9664	.9671	.9678
1.9	.9713	.9719	.9726	.9732	.9738	.9744
2.0	.9772	.9778	.9783	.9788	.9793	.9798
2.1	.9821	.9826	.9830	.9834	.9838	.9842
2.2	.9861	.9864	.9868	.9871	.9875	.9878
2.3	.9893	.9896	.9898	.9901	.9904	.9906
2.4	.9918	.9920	.9922	.9925	.9927	.9929

Figura 4.2 Tabla Distribución Normal
Fuente: (Perez,2019).

Entonces:

$$Z_{(1-\alpha)} = -1.63$$

b) Cálculo estadístico de la hipótesis

Para el cálculo estadístico de la hipótesis se hallará, que son el promedio y desviación estándar de la muestra, usando las siguientes formulas.

Promedio:

Desviación Estándar:

$$S = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}}$$

Donde:

x: es la distancia en metros que hay entre los puntos encontrados

Los resultados obtenidos son:

$$\bar{x} = 1.97 \quad s = 0.83$$

Usando la distribución de Z dada la siguiente transformación se calculará si la hipótesis es aceptada o no.

$$Z = \frac{X - \mu}{\delta / \sqrt{n}}$$

Remplazando los datos.

$$Z = \frac{1.97 - 3}{0.83 / \sqrt{40}} = -7.84$$

Conclusión:

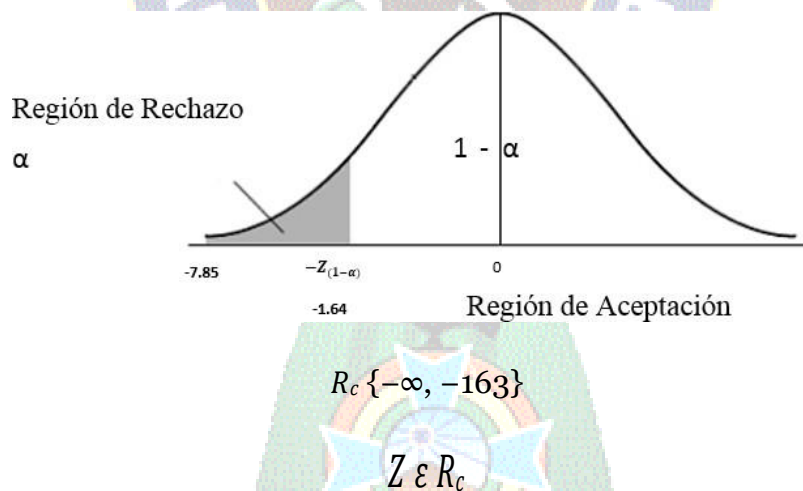


Figura 4.3 Distribución Z para la toma de decisión

Fuente: (Elaboración Propia, 2021).

Por lo tanto, se rechaza H_0 .

H_0 : El prototipo de collar con geolocalización basada en tecnologías GPS permite rastrear la ubicación de una MASCOTA, con un margen de error mayor a 3 metros.

4.1.2. ANÁLISIS DE RESULTADOS

La mayoría de los datos obtenidos con el prototipo de collar con geolocalización basada en tecnologías GPS fueron exitosos teniendo un nivel de error más bajo a lo propuesto donde se requería tener un margen de error menor o igual a 3 metros.

Por lo tanto, podemos decir que el prototipo de geolocalización basada en tecnologías GPS tiene una función óptima al momento de tomar los datos.



CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

El objetivo planteado en el capítulo 1 “Desarrollar un prototipo de collar para el monitoreo de mascotas usando una placa A9G de ai thinker con geolocalización basada en tecnología GPS”, se cumplió con el diseño, implementación y pruebas del prototipo desarrollado, que con la ayuda del módulo A9G de ai y tecnología Arduino y Android, permite obtener la ubicación de la mascota que use el prototipo de collar con geolocalización, para disminuir extravíos, hurtos u otros que pueden sufrir las mascotas por diferentes motivos.

Los objetivos específicos fueron cumplidos:

- ✓ Se analizó el funcionamiento de dispositivos GPS para el rastreo de mascotas.
- ✓ Se estableció una conexión de la aplicación con el prototipo de collar con tecnología GPS.
- ✓ Se pudo determinar los beneficios de la utilización de GPS y SIM para el cuidado de mascotas sin necesidad de una conexión a internet.
- ✓ Se pudo demostrar la efectividad del sistema de posicionamiento global utilizando la placa “A9G de ai thinker” aplicado a mascotas.

La conectividad del prototipo de collar para el monitoreo de mascotas usando una placa A9G de ai thinker con geolocalización basada en tecnología GPS se realizó con éxito

El uso de hardware y software libre permite la visualización del módulo utilizado en el prototipo y las diferentes librerías que se usaron para su desarrollo.

Armar el prototipo es una parte muy dinámica, pero a la vez compleja ya que se debe tener cuidado al momento de la elección del módulo a usar, revisar si son los adecuados y más apropiados para la optimización tanto en lo económico como en lo físico, trabajar con un módulo nuevo del monto puede tener muchas ventajas al optimizar.

5.1.1. Costo del prototipo de collar con geolocalización

El costo del prototipo de collar se basó en el hardware que se utilizó.

Numero	Componente	Cantidad	Costo en bs
1	Placa Arduino Lilypad	1	30
2	Módulo A9G de ai	1	145
3	Bateria 5V	1	60
4	Cables, soldaduras entre otros	8	70
5	Chip (Entel)	1	10
Total			315

Tabla 5.1 Costo del prototipo de collar con geolocalización.

Fuente: (Elaboración Propia, 2021).

5.2. RECOMENDACIONES

Las recomendaciones de la presente tesis son:

- Usar un chip de telefonía Entel, por ser la que tiene una mejor cobertura de señal.
- Usar el módulo A9G de ai como placa.
- Verificar el saldo del crédito del chip, ya que se manda la ubicación mediante SMS.
- Investigar más sobre la placa A9G de ai ya que tiene muchos otros usos.
- Adaptar una pantalla LED al prototipo ya que es compatible con A9G de ai.
- Introducir una base de datos para poder almacenar rutas anteriores en las cuales se encontraba nuestra mascota, para un mejor seguimiento.

BIBLIOGRAFÍA

- Abc Tecnología / Madrid (2017).” La geolocalización: una tecnología para desplazarse fácilmente”. Recuperado de: <http://www.abc.es/tecnologia/moviles-telefonía/20140320/abci-localizacion-movil201403192024.html>
- Alcaraz M. (s.f). Internet De Las Cosas. Universidad Católica Nuestra Señora De La Asunción. Recuperado en línea[<http://jeuazarru.com/wp-content/uploads/2014/10/Internet-of-Things.pdf>]. Consulta [1 de marzo de 2019].
- Animaleando (2020). “¿Qué es una mascota?”. Recuperado de: <https://animalear.com/b/que-es-una-mascota-2>
- Arduino. Recuperado de <https://www.arduino.cc/en/Guide/Introduction> [consulta 05 de noviembre de 2018]
- Arias X. (2014). Mapa y Recorrido Virtual Interactivo con Georreferenciación en Dispositivos Móviles. Tesis de Grado. Universidad Mayor de San Andrés. La Paz - Bolivia.
- Aruini M. (2015) Geolocalización para Niños con Autismo. Tesis de Grado. Universidad Mayor de San Andrés. La Paz Bolivia
- Baltazar S. (2016). Manilla de Control de Frecuencia Cardíaca Para La Prevención de Arritmias. Tesis de Grado. Universidad Mayor de San Andrés, La Paz – Bolivia.
- Beltrán G. (2015). Geolocalización Social. España. Recuperado en línea: [<http://revpubli.unileon.es/ojs/index.php/poligonos/article/view/3290>]. Consulta: [3 de noviembre 2018]
- Beltrán, G (2012). “Geolocalización y redes Sociales”. Recuperado de: www.redactalia.com
- Blanco P. et al., (2009). Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone. Tesis de Grado. Universidad Politécnica de Madrid
- Bonifaz D. & Vásquez A. (2017). Diseño Y Construcción De Un Dispositivo Electrónico, Para La Ignición De Vehículos Mediante Reconocimiento De Huella Dactilar, Con Opción De Monitoreo GPS E Inmovilización Vehicular, Programado En Arduino, Con Visualización En Dispositivos Móviles. Tesis de Grado. Universidad Politécnica Salesiana Sede Cuenca

- Canal Bolivisión (2020). “Familias buscan a sus mascotas perdidas”. Recuperado de: <https://www.redbolivision.tv.bo/noticia/familias-buscan-a-sus-mascotas-perdidas-12/>
- Castro, L (2014). “Que es geolocalización”. Recuperado de: <http://aprenderinternet.about.com/od/Glosario/g/Que-es-geolocalizacion.htm>
- Charca R., 2018, Robo de Niños Express, El Alteño. Recuperado en línea [<http://www.troposproject.eu/node/93>] Consulta: [15 de noviembre de 2018]
- Chihuan R. (2002). Propuesta De Una Metodología Para El Levantamiento Catastral De Predios Rurales Mediante El Uso Del GPS En La Selva. Tesis de Grado. Universidad Nacional de Ingeniería. Lima-Perú.
- Ciberseguridad (2016). “Geolocalización: virtudes y riesgos”. Recuperado de: <https://www.osi.es/es/actualidad/blog/2016/09/20/geolocalizacion-virtudes-y-riesgos>
- Consulta [9 de abril de 2019].
- Crespo E. (s.f.). Aprendiendo Arduino. Recuperado en línea [<https://aprendiendoarduino.wordpress.com/2016/12/11/hardware-arduino/>] Consulta: [19 de marzo de 2019].
- Cupi J. (2014). Georreferenciación Y Su Uso En Smartphones Para La Localización De Personas Tesis de Grado. Universidad Mayor de San Andrés, La Paz – Bolivia.
- Dave E. (2011). Internet De Las Cosas Como La Próxima Evolución De Internet Lo Cambia Todo. Cisco IBSG. Recuperado en línea: [https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf]. Consulta: [3 de marzo de 2019].
- Diario Urgente (2020). “Las mascotas en La Paz ya pueden llevar chips para evitar extravíos”. Recuperado de: <https://urgente.bo/noticia/las-mascotas-en-la-paz-ya-pueden-llevar-chips-para-evitar-extrav%C3%ADos>
- Fernández, Y (2020). “Qué es Arduino, cómo funciona y qué puedes hacer con uno”.
- Garmin y Tomtom, S (2017). “Qué es un GPS (sistema de posicionamiento global)”. Recuperado de: <https://www.carvalza.es/que-es-un-gps>
- Geek Factory. (2017). Tutorial Arduino. Recuperado de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/shield-o-modulo-gps-con-arduino/>. [consulta 20 de marzo de 2019].

- Gómez J. & Hernández D. (2016). Desarrollo De Aplicaciones Para Dispositivos Móviles - Mobile D. Tesis de Grado. Universidad Del Quindío
- Gómez M. & Venegas A. (2018). Wearable Computing. Tesis de Grado. Universidad Peruana De Ciencias Aplicadas
- Gómez, J (2016). “Mobile D (programación dispositivos móviles)”. Recuperado de: <https://es.slideshare.net/pipehernandez1020/mobile-d-programacion-dispositivos-moviles#:~:text=Se%20compone%20de%20distintas%20fases,la%20metodolog%C3%A9ica%20de%20Gil%20sea%20eficiente.>
- Herrador R. (2009). Guía de Usuario de Arduino. Universidad de Córdoba. Recuperado en línea: [http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf]. Consulta: [3 de marzo de 2019].
- Ingenia, B (2020). “Qué es la geolocalización y cómo funciona”. Recuperado de: <https://www.evaluandosoftware.com/la-geolocalizacion-funciona/>
- Kiosquera (junio 2014).” La geolocalización: una tecnología para desplazarse fácilmente”. Recuperado de: [file:///C:/Users/NEAR/Downloads/lageolocalizacion-una-tecnologia-para-desplazarse-facilmente-10861-mnus1h%20\(1\).pdf](file:///C:/Users/NEAR/Downloads/lageolocalizacion-una-tecnologia-para-desplazarse-facilmente-10861-mnus1h%20(1).pdf)
- Madakam S., Ramaswamy R. & Tripathi S. (2015). Internet of Things (IoT): A Literature Review. Revistta de Informática y Comunicaciones. Recuperado en línea: [<http://www.scirp.org/journal/jcc>].
- Mamani J. (2014) Localización Y Monitoreo De Personas Dentro De Los Limites Del Perímetro Urbano Con Cobertura, Basada En Tecnología GPS/GSM. Universidad Mayor de San Andrés, La Paz – Bolivia.
- Mann, Steve (1998), Wearable Computing as means for personal empowerment. Conferencia pronunciada el 12 de mayo de 1998, Toronto. Recuperado en línea: [<http://www.eyetap.org/wearcam/icwc98/keynote.html>]. Consulta: [3 de marzo de 2019].
- Mega Practical (2018). “¿Qué es una Metodología de Desarrollo de Software?”. Recuperado de: <https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/metodologias-de-desarrollo-de-software>
- Mendoza K & Sandoval L. (2015). Diseño de un Sistema Móvil con tecnología GPS Para La Localización de Niños Extraviados. Tesis de Grado. Universidad Nacional de Trujillo. Lima– Perú.

- MIT, Massachusetts Institute of Technology (2000). Augmented Reality Through Wearable Computing. Recuperado en línea [https://www.media.mit.edu/wearables/papers/thesis_MarkBlum.pdf]. Consulta [9 de abril de 2019].
- Monrroy S. (2018) Geolocalización Basada en Tecnologías GPS a Personas con Enfermedad de Alzheimer. Universidad Mayor de San Andrés. La Paz – Bolivia.
- OBS (2020). “¿Qué son las metodologías de desarrollo de software?”. Recuperado de: https://obsbusiness.school/es/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software
- Osorio R., Duran S., González P., Peña M. & Lefranc G. (s.f.). Geolocalización Usando Tecnología Wifi. Recuperado en línea [https://www.researchgate.net/publication/280353577_System_Administration_and_Monitoring_Moving_Objects_Using_Mobile_Devices]. Consulta [9 de abril de 2019].
- Pérez A. et. al. (2006). Una Metodología Para El Desarrollo De Hardware Y Software Embebidos En Sistemas Críticos De Seguridad. Recuperado: [http://www.iiisci.org/journal/CV%24/risci/pdfs/C863GM.pdf]. Consulta [1 de abril de 2019].
- Pérez Corimayta, Alex (2014). Software Móvil de Geolocalización para la Banca en la Ciudad de La Paz, La Paz – UMSA
- Perfil Topografía. (2017) [en línea]. Recuperado https://perfiltopografia.es/geolocalizacion-georreferenciacion/ [consulta 11 de marzo de 2019].
- Quispe I. (2016) Internet de Las Cosas, Control y Seguimiento de un Automóvil. Universidad Mayor de San Andrés. La Paz - Bolivia.
- Ramírez R. (2010). Métodos Para El Desarrollo De Aplicaciones Móviles. Recuperado en línea [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf]. Recuperado de: https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno.

- Rivera C. (2013). Metodología Tropos. Recuperado <https://sites.google.com/site/analisisydisenoudea/metodologias-requisitos-no-funcionales/tropos> [consulta 6 de marzo de 2019]
- Rodríguez J. (2013). Metodología Tropos Descripción. Recuperado <https://sites.google.com/site/analisisydisenoudea/metodologias-requisitos-no-funcionales/tropos> [consulta 7 de marzo de 2019]
- Serpas G., Ramírez M. & Obaldía F. (2004) Sistema de Posicionamiento Global. Universidad Nacional Heredia. Costa Rica.
- Sommerville I. (2005). Ingeniería del Software, Madrid España: Pearson Educación, S.A.
- Toro E., 2013, Instalan GPS en prendas de niños para localizarlos, La Razón, La Paz - Bolivia
- UNICEF & ICMEC, (2011), Niños Desaparecidos en Centroamérica. Recuperado en línea [[https://www.icmec.org/wp-content/uploads/2015/10/Spanish-Central_America_Missing_Children_Report ES_.pdf](https://www.icmec.org/wp-content/uploads/2015/10/Spanish-Central_America_Missing_Children_Report_ES_.pdf)]. Consulta [9 de abril de 2019].
- Vilca R. (2015). Influencia de un sistema de geolocalización en el control y monitoreo de vehículos con dispositivos GPS en una empresa logística. Universidad Cesar Vallejo. Lima- Perú
- Zambetti, H (2019). “¿Sabes qué es un Arduino y para qué sirve?”. Recuperado de: <https://www.fundacionaquae.org/sabes-arduino-sirve/>

ANEXOS

ANEXO 1.- Prototipo de collar con geolocalización basada en tecnologías GPS. Terminado



ANEXO 2.- Código del XML de la aplicación móvil

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/black"
    tools:context=".MainActivity">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:paddingVertical="25dp"
            android:textColor="@color/white"
            android:fontFamily="@font/worksans_bold"
            android:textSize="45sp"
            android:text="Locate\nBest Friend" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:paddingBottom="20dp"
            android:textColor="@color/white"
            android:textSize="17sp"
            android:text="Introduce el número del collar" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center">
            <TextView
                android:fontFamily="@font/worksans_bold"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/white"
                android:layout_marginRight="5dp"
                android:textSize="18sp"
                android:text="+591 " />

            <com.google.android.material.textfield.TextInputLayout
                android:layout_width="200dp"
                android:layout_height="wrap_content"
                app:hintAnimationEnabled="false"
                app:hintEnabled="false"
            >
            <com.google.android.material.textfield.TextInputEditText
```



```

        android:id="@+id/cellphone"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:background="#19FFFFFF"
        android:textSize="18sp"
        android:hint="Celular"
        android:maxLength="8"
        android:inputType="number"
        android:paddingTop="10dp"
        android:textColor="@color/white"
        android:textColorHint="#66FFFFFF"
    />

</com.google.android.material.textfield.TextInputLayout>

</LinearLayout>

<com.google.android.material.button.MaterialButton
    android:id="@+id/findButton"
    android:layout_marginVertical="15dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Buscar Amigo"
    style="@style/MaterialButtonStyle"
    android:layout_gravity="center"
    />

<RelativeLayout
    android:id="@+id/waiting"
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <LinearLayout
        android:id="@+id/loading"
        android:gravity="center"
        android:visibility="gone"
        tools:visibility="visible"
        android:layout_marginTop="45dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="15sp"
            android:textColor="@color/white"
            android:text="Esperando la ubicación"
            android:layout_marginRight="10dp"/>

        <ProgressBar
            android:id="@+id/progressBar"
            style="?android:attr/progressBarStyle"
            android:layout_width="30dp"
            android:layout_height="30dp" />

    </LinearLayout>

```

```

<ImageView
    android:id="@+id/image"
    android:layout_width="wrap_content"
    android:layout_height="230dp"
    android:alpha="0.2"
    android:src="@drawable/ic_guide_dog"/>

</RelativeLayout>
<androidx.cardview.widget.CardView
    android:id="@+id/mapContainer"
    android:visibility="gone"
    android:layout_marginTop="15dp"
    android:layout_width="match_parent"
    android:layout_marginHorizontal="20dp"
    app:cardCornerRadius="20dp"
    app:cardElevation="12dp"
    android:layout_height="270dp">

    <fragment
xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/map"

        tools:context=".MapsActivity"

        android:name="com.google.android.gms.maps.SupportMapFragment" />

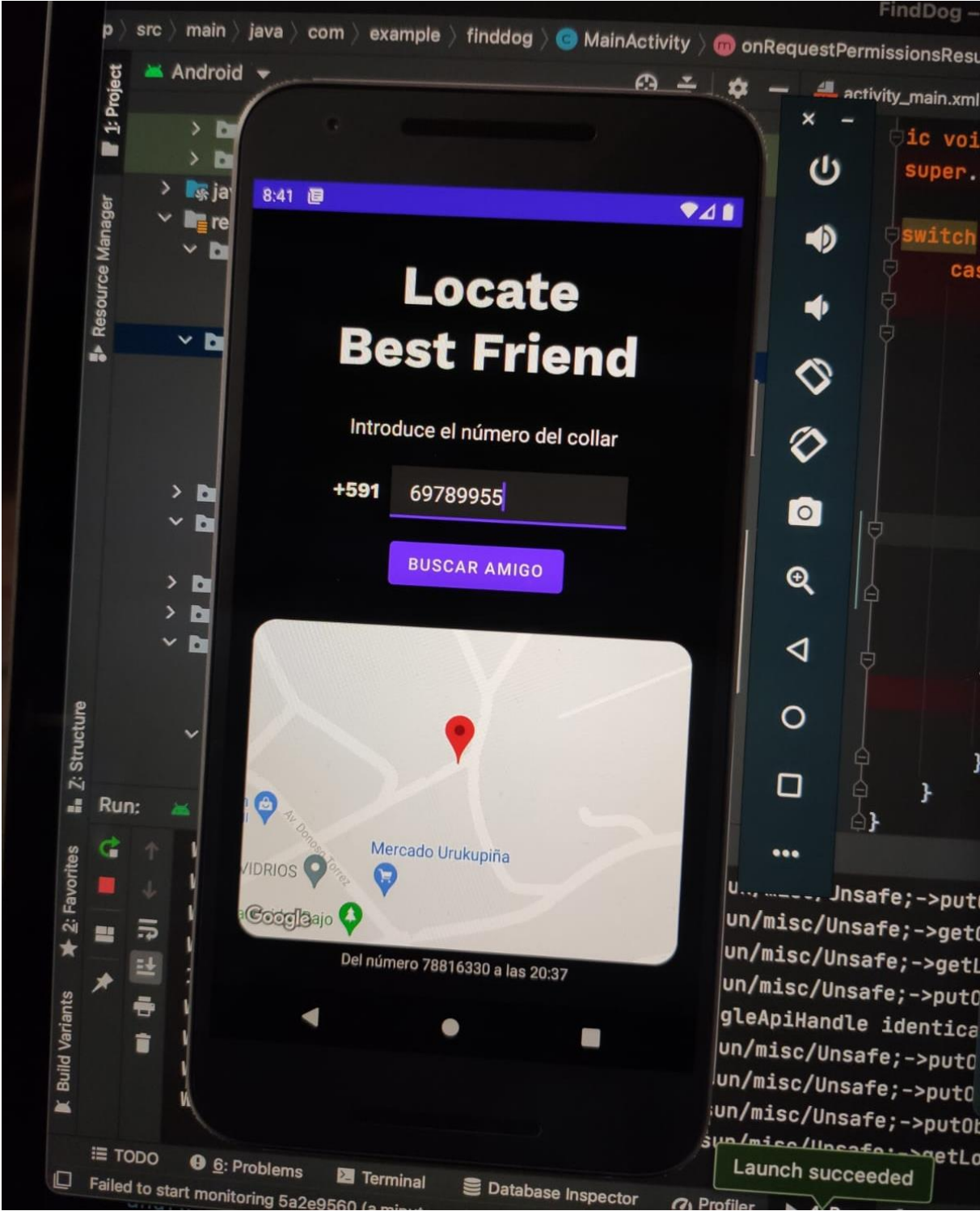
    </androidx.cardview.widget.CardView>
    <TextView
        android:id="@+id/timeMessage"
        android:visibility="gone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/white"
        android:alpha="0.7"
        android:gravity="center"
        android:layout_marginTop="5dp"
        android:textSize="13sp"
        android:layout_marginBottom="15dp"
        android:text="Del número 69789955 a las: 12:00"/>
</LinearLayout>

</ScrollView>

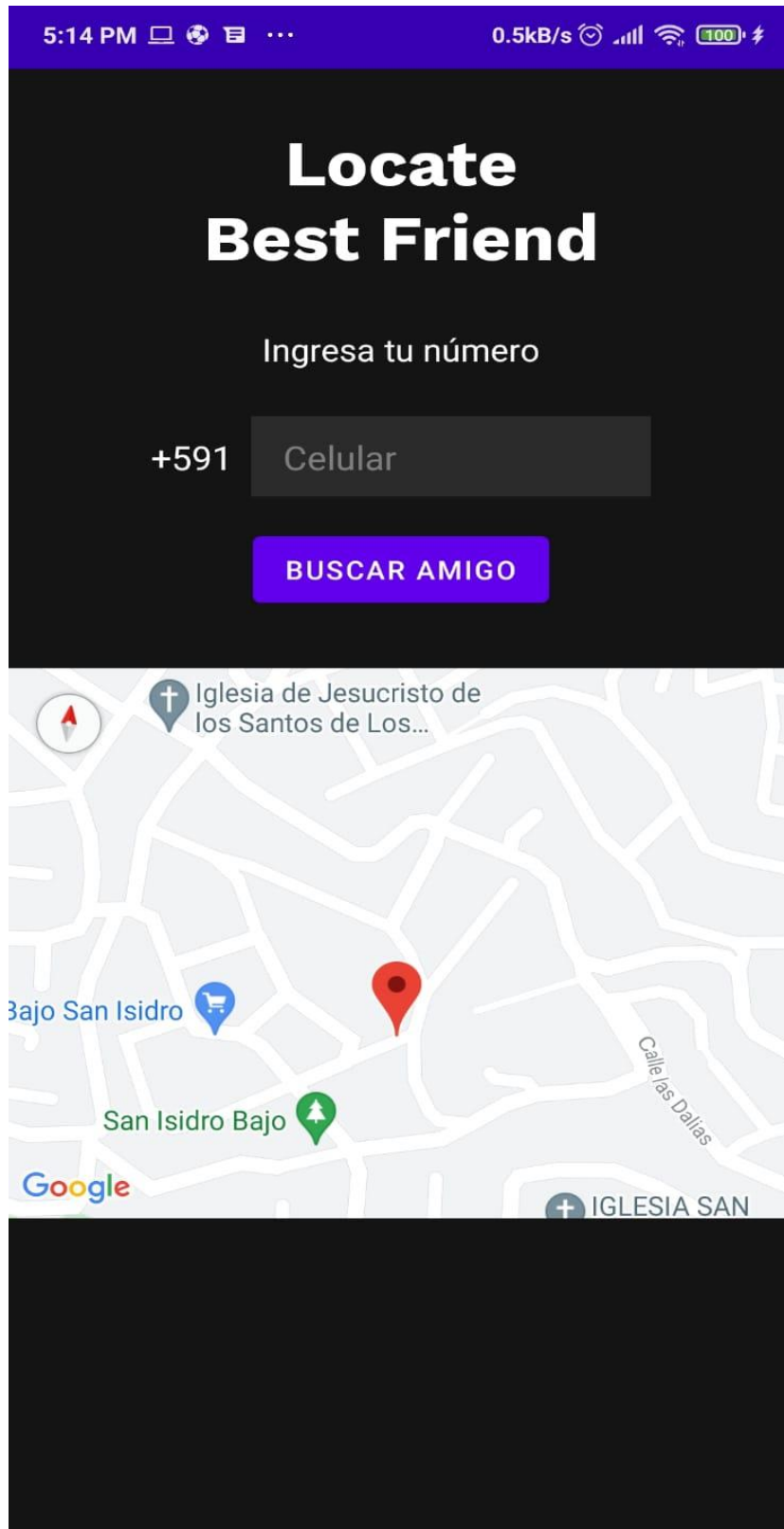
```



ANEXO 3.- Imagen de la aplicación móvil en emulador de Android Studio.



ANEXO 4.- Captura de pantalla de la aplicación móvil funcionando en un dispositivo.



ANEXO 5.- Código Completo del Arduino Lilypad conectado con A9G de ai y su batería 5V.

```
#include<SoftwareSerial.h>

SoftwareSerial localizador(3, 2); //definimos los pines de conexion tx,rx

String Ndueno = "78816330";
String Nmascota = "KAYLA";
String ubi = ""; // variable para habilitar la ubicación gps
int ubigps = 0;
String SMSubicacion = ""; // variable que almacena la respuesta total de la ubicacion
String enviarUbicacion = ""; //variable que almacena la ubicacion en coordenadas latitud y longitud
String sms = ""; // variable que almacena el mensaje recibido
String destinatario = "AT+CMGS="; // comando para configurar el numero al cual se enviara el sms
String rEnvio="";// variable que almacena la respuesta del envio de sms

void setup() {
  pinMode(13, OUTPUT); // definimos el pin 13 como salida
  Serial.begin(115200); // inicia la comunicacion serial del arduino
  localizador.begin(115200); //inicia la comunicacion serial con el módulo A9G
  iniciarLocalizador(); //método que inicia el módulo A9G
}

void iniciarLocalizador() {
  Serial.println("iniciando localizador...");
  localizador.println("AT"); // inicia la comunicación por comandos AT
  respuestaA9G(); // método que muestra la respuesta del módulo A9G por el puerto serial
  localizador.println("AT+CSQ"); // verifica la intensidad de la señal
  respuestaA9G();
  localizador.println("AT+CCID"); // verifica la informacion de la sim
  respuestaA9G();
  localizador.println("AT+CGSN"); // muestra el IMEI del módulo A9G
  respuestaA9G();
  localizador.println("AT+EGMR=2,7"); //verifica si hay red
  respuestaA9G();
  localizador.println("AT+CMGF=1"); //configura el módulo en modo texto para la lectura y envio de mensajes
  respuestaA9G();
  localizador.println("AT+CMGD=1,4"); //borra los mensajes del sim
  respuestaA9G();
  localizador.println("AT+CPMS=SM,SM,SM"); //verifica la capacidad de mensajes
  respuestaA9G();
  iniciaGPS(); //iniciamos el gps del módulo A9G
  localizador.println("AT+CMGF=1");
  respuestaA9G();
  localizador.println("AT+CNMI=2,2,0,0,0"); //configura ver los sms por puerto serial
  respuestaA9G();
}

void respuestaA9G() {
  delay(600);
  while (localizador.available() > 0) { // verifica si existe entradas en el puerto serial
    Serial.write(localizador.read()); // imprime en el puerto serial la respuesta del puerto serial del módulo A9G
  }
}

void iniciaGPS() {
  localizador.println("AT+GPS=1"); //activa el GPS
  respuestaA9G();
}
```

```

delay(100);
localizador.println("AT+GPSRD=5"); //muestra por pantalla la lectura de GPS
lecturaGPS(); // método que habilita la localización del módulo A9G
}
void lecturaGPS() {
  ubigps = 0;
  while (ubigps == 0) {
    while (localizador.available() > 0) {
      char ubicacion = localizador.read();
      ubi.concat(ubicacion);
      delay(10);
    }
    if (ubi != "") {
      Serial.println(ubi);
      int i = ubi.indexOf("W");
      if ((ubi.substring(i + 2, i + 3).equals("1") || (ubi.substring(i + 2, i + 3)).equals("2"))) { //realiza la
comparación para ver si capto las coordenadas,
//en caso de ser 0 no logro captar las coordenadas
guardarUbicacion(); // método que guarda la localización obtenida por el GPS
      ubigps = 1;
    }
    ubi = "";
  }
}
void guardarUbicacion() {
  digitalWrite(13,1);
  delay(200);
  digitalWrite(13,0);
  delay(200);
  digitalWrite(13,1);
  delay(200);
  digitalWrite(13,0);
  delay(200);
  digitalWrite(13,1);
  Serial.println("obteniendo coordenadas...");
  localizador.println("AT+LOCATION=2");//comando para obtener solo la longitud y latitud detectado por el
GPS
  int coord = 0;
  digitalWrite(13,1);
  while (coord == 0) {
    while (localizador.available() > 0) {
      char coordenada = localizador.read();
      SMSubicacion.concat(coordenada);
      delay(100);
    }
    if (SMSubicacion != "") {
      Serial.println(SMSubicacion);
      int i = SMSubicacion.indexOf("-");
      enviarUbicacion = SMSubicacion.substring(i, i + 21); // almacena solo las coordenadas en la variable
      coord = 1;
    }
  }
  digitalWrite(13,0);
}
}

```

```

void leerSMS() {
  while (localizador.available() > 0) {
    char mensaje = localizador.read();
    sms.concat(mensaje);
    delay(10);
  }

  if (sms != "") {
    Serial.println(sms); // muestra por pantalla el mensaje recibido
    int i = sms.indexOf("u");
    if ((sms.substring(i , i + 2)).equalsIgnoreCase("u1")) {
      enviarSMS(); // método para enviar el mensaje de ubicación
    }
    sms = "";
  }
}

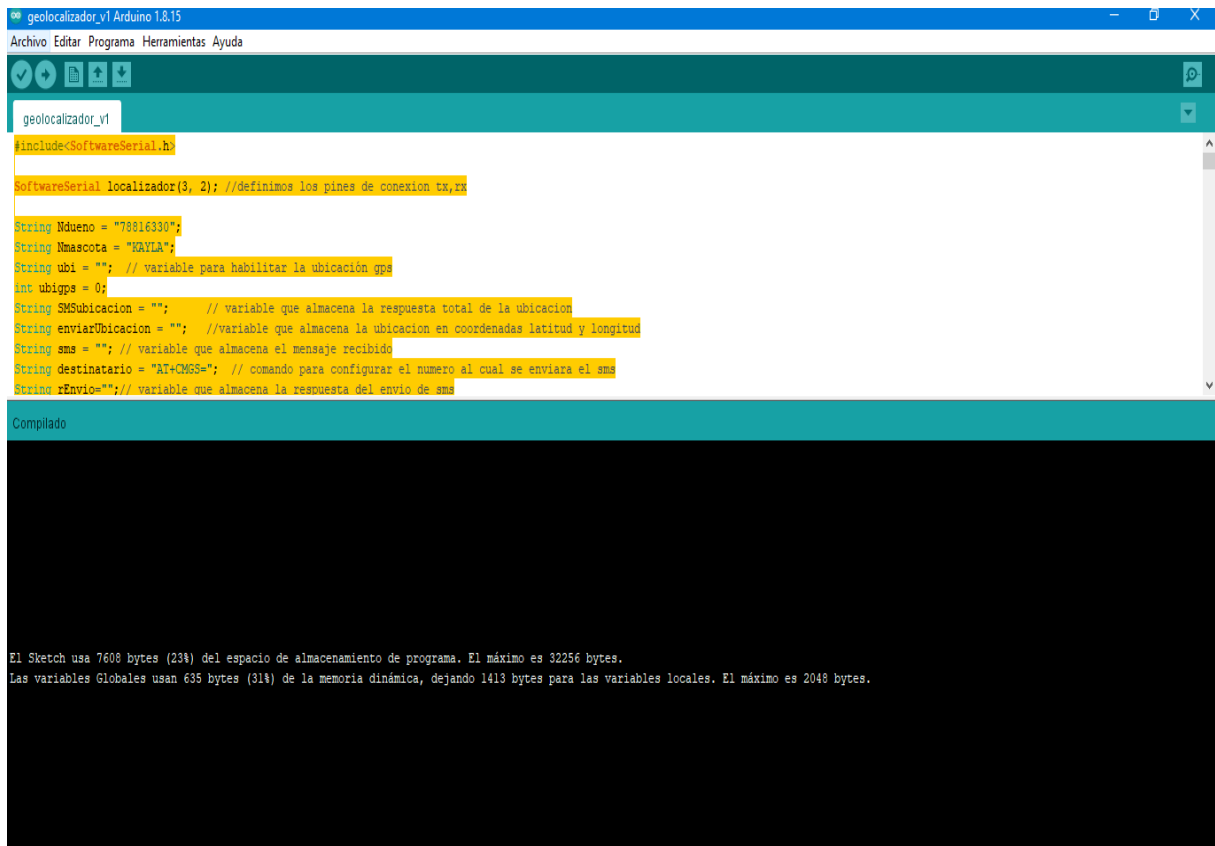
void enviarSMS() {
  Serial.println("enviando mensaje...");
  digitalWrite(13, 1); // enciende el led integrado del arduino para indicarnos que se esta enviando un mensaje
  delay(20);
  Serial.print("ubicacion mascota");
  lecturaGPS();
  localizador.println("AT+GPSRD=10"); //detenemos la transmision del gps por pantalla
  respuestaA9G();
  localizador.println("AT+CMGF=1"); //configuramos en modo texto
  respuestaA9G();
  destinatario = destinatario + Ndueno;
  localizador.println(destinatario);
  respuestaA9G();
  localizador.print(enviarUbicacion);
  localizador.write(26); // envia el comando para enviar el mensaje <= equivale a un ctrl+z
  // todo: comprobar envio sms
  while (localizador.available() > 0) {
    char respuesta = localizador.read();
    rEnvio.concat(respuesta);
    delay(10);
  }

  if (rEnvio != "") {
    Serial.println(rEnvio); // muestra por pantalla la respuesta recibida
    int i = rEnvio.indexOf("E");
    Serial.println(sms.substring(i , i + 5));
    if ((sms.substring(i , i + 5)).equalsIgnoreCase("ERROR")) { //si existe error reenvia el mensaje de ubicación
      enviarSMS();
    }
    rEnvio = "";
  }
  digitalWrite(13, 0); // apaga el led, confirmando que se envio el sms
}

void loop() {
  leerSMS();
}

```

ANEXO 6.- Captura de pantalla del código siendo verificado



```
geolocalizador_v1 Arduino 1.8.15
Archivo Editar Programa Herramientas Ayuda
geolocalizador_v1
#include<SoftwareSerial.h>
SoftwareSerial localizador(3, 2); //definimos los pines de conexion tx,rx
String Ndueno = "78816330";
String Nmascota = "KAYLA";
String ubi = ""; // variable para habilitar la ubicacion gps
int ubigps = 0;
String SMSubicacion = ""; // variable que almacena la respuesta total de la ubicacion
String enviarUbicacion = ""; //variable que almacena la ubicacion en coordenadas latitud y longitud
String sms = ""; // variable que almacena el mensaje recibido
String destinatario = "AT+CMGS="; // comando para configurar el numero al cual se enviara el sms
String rEnvio=""; // variable que almacena la respuesta del envio de sms
Compilado
El Sketch usa 7608 bytes (23%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 635 bytes (31%) de la memoria dinámica, dejando 1413 bytes para las variables locales. El máximo es 2048 bytes.
```



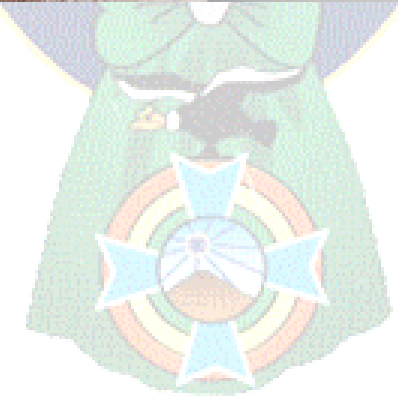
ANEXO 7.- Probando el prototipo de collar en una mascota llamada “Perlita”



ANEXO 8.- Probando el prototipo de collar en una mascota llamada “Lunita”



ANEXO 9.- Probando el prototipo de collar en una mascota llamada “Morita”



ANEXO 9.- Probando el prototipo de collar en una mascota llamada “Mayita”

