

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniera e Ingeniero de Sistemas**

**TEMA:
DESARROLLO DE UN PROTOTIPO PARA LA GEOLOCALIZACIÓN Y
MONITORIZACIÓN DE FRECUENCIA CARDIACA DE MASCOTAS EN LA CIUDAD
DE QUITO**

**AUTORES:
NATHALY ANABEL PÉREZ VARGAS
BRYAN DAVID TORRES PACHAR**

**TUTOR:
MANUEL RAFAEL JAYA DUCHE**

Quito, agosto 2020

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Nathaly Anabel Pérez Vargas con documento de identificación N.º 1723112189 y Bryan David Torres Pachar con documento de identificación N.º 1719839423, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “DESARROLLO DE UN PROTOTIPO PARA LA GEOLOCALIZACIÓN Y MONITORIZACIÓN DE FRECUENCIA CARDIACA DE MASCOTAS EN LA CIUDAD DE QUITO”, mismo que ha sido desarrollado para optar por el título de: INGENIERA E INGENIERO DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
PÉREZ VARGAS
NATHALY ANABEL
CI: 1723112189



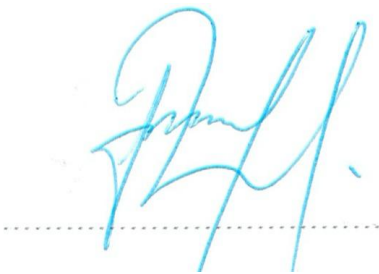
.....
TORRES PACHAR
BRYAN DAVID
CI: 1719839423

Quito, agosto del 2020

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “DESARROLLO DE UN PROTOTIPO PARA LA GEOLOCALIZACIÓN Y MONITORIZACIÓN DE FRECUENCIA CARDIACA DE MASCOTAS EN LA CIUDAD DE QUITO”, realizado por Nathaly Anabel Pérez Vargas y Bryan David Torres Pachar, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, agosto del 2020

A handwritten signature in blue ink, appearing to read 'Manuel Rafael Jaya Duché', is written over a horizontal dashed line.

MANUEL RAFAEL JAYA DUCHE
CI: 1710631035

DEDICATORIA

Dedico el presente trabajo de titulación en primer lugar a Dios, por darme la vida y la salud quien con su sabiduría, paciencia y misericordia me brindo las fuerzas necesarias para luchar y sobrepasar todas las adversidades, manteniéndome firme para cumplir uno de mis mayores sueños en mi vida profesional, no hay mejor recompensa por la que se lucha día a día a pesar de todas las tormentas.

Dedico a mis queridos Padres, Patricia Vargas, Jesús Pérez porque gracias a su ejemplo me ayudaron hacer una persona fuerte, luchadora, y así cumplir mis objetivos en mi vida profesional. Gracias por ser como son, por su apoyo incondicional en cada momento que necesite una mano, por llorar juntos y por levantarnos en cada dificultad son mi orgullo mi gran fortaleza de lucha.

Dedico a mi querida hermana Joselin Pérez ya que se ha convertido en mi gran amiga, confidente mi compañera de lucha, mi mano derecha, y la que cada día me regalaba palabras de aliento para seguir adelante.

Dedico a Bryan Torres que se ha convertido en mi compañero en todo este camino de la universidad por ser mi apoyo, mi fuerza para salir adelante, gracias por la paciencia y por convertirte en el gran amor de mi vida.

Nathaly Pérez

Dedico este trabajo en primer lugar a Dios que me ha dado la salud y la vida para poder llegar hasta esta etapa de mi vida y por todas las bendiciones que me ha ofrecido a lo largo de este arduo trayecto de la vida universitaria y no dejarme solo y permitirme compartir este logro con mi familia.

A mis padres que han sido el pilar fundamental en mi vida y que gracias a ellos puedo culminar satisfactoriamente mi carrera y empezar mi vida profesional. Gracias a mi madre Mercedes Pachar que me ha inculcado buenos valores y su apoyo incondicional, en los buenos, pero sobre todo en los malos momentos de mi vida, me ha enseñado a luchar y no rendirme pese a las adversidades que se presenten en el camino y por ser el mejor ejemplo de lo que es ser una buena persona. A mi padre Gonzalo Torres por todos sus consejos, su lucha constante, por su esfuerzo y por enseñarme que la vida por más dura que sea siempre habrá una razón por la cual seguir avanzando y a los dos por su gran sacrificio a lo largo de todos estos años.

A mi hermana Karol por siempre estar ahí insistiendo en que siga adelante y no me rinda aconsejándome para seguir adelante. Por su cariño y su espontaneidad para conmigo y su apoyo incondicional durante todos estos años. Así como a mi primo Diego, mi tía Mónica, que fueron parte de este trayecto y a mi abuelita que desde el cielo me bendice y hace que esto sea posible.

Dedico por su puesto este trabajo a mi amiga, mi compañera incondicional Nathaly Pérez que durante estos últimos años me ha brindado su apoyo, su cariño, su comprensión, siempre me alentado a superarme y seguir adelante cuando ya estaba por desfallecer, pero sobre todo gracias por convertirse en mi gran amor.

Bryan Torres

AGRADECIMIENTOS

Agradecemos principalmente a Dios por brindarnos la sabiduría necesaria para culminar nuestros estudios y ser nuestra fuerza para seguir luchando día a día.

Agradecemos a nuestro querido tutor Rafael Jaya, ya que con los conocimientos brindados nos han servido de gran ayuda para la realización del proyecto, gracias por su paciencia, su amistad y por la gran persona que ha sido con nosotros.

A todos y cada uno de los docentes que siempre han sido un pilar fundamental para saber guiarnos correctamente y ayudarnos a corregir nuestros errores e impulsarnos a sobresalir y ser mejores en el ámbito personal y profesional.

A toda nuestra familia quienes siempre han estado ahí para apoyarnos, ayudarnos y aconsejarnos que debemos seguir formándonos y alcanzar nuestros objetivos.

A todos nuestros amigos y amigas que a lo largo de la carrera universitaria nos han colaborado de alguna u otra manera para poder lograr este propósito tan anhelado.

ÍNDICE

INTRODUCCIÓN	1
Antecedentes	1
Problema.....	2
Justificación.....	3
Objetivos	4
Objetivo General	4
Objetivos Específicos	4
Metodología	4
CAPÍTULO I.....	8
MARCO TEÓRICO	8
1.1. Geolocalización.....	8
1.1.1. Usos de la geolocalización	8
1.1.2. GPS.....	8
1.1.2.1. Historia GPS.....	9
1.1.2.2. Técnicas del GPS	10
1.1.2.3. Funcionamiento del Sistema GPS	11
1.2. Sistemas Embebidos y dispositivos IoT.....	12
1.2.1. Definición de sistema embebido	12
1.2.2. Sistema eléctrico	13
1.2.3. Sistema de comunicaciones electrónicas.....	14
1.2.4. Microcontrolador.....	14
1.2.5. Arduino.....	15
1.2.6. Frecuencia Cardiaca	15
1.2.7. Dispositivo IoT.....	17
1.2.8. Elementos de IoT	17
1.3. Redes de Comunicación	18
1.3.1. Comunicación GSM.....	18
1.3.2. Arquitectura de GSM	18
1.4. Evolución de plataformas en la nube	20
1.4.1. Servicios en la nube.....	20
1.4.2. Arquitectura de la nube	20
1.4.2.1. Software como Servicio (SaaS).....	20

1.4.2.2. Plataforma como Servicio (PaaS).....	20
1.4.2.3. Infraestructura como Servicio (IaaS)	21
1.4.3. Plataformas de servicios en la nube	23
1.4.4. Firebase	23
1.5. Aplicaciones de monitoreo y gestión	25
1.5.1. Aplicaciones Móviles.....	25
1.5.2. Android.....	26
1.5.3. Arquitectura de Android.....	26
CAPÍTULO II	28
ANÁLISIS DE COMPONENTES.....	28
2.1. Análisis de materiales.....	28
2.1.1. Microcontroladores	28
2.1.1.1. Arduino Nano.....	28
2.1.1.2. Raspberry Pi	29
2.1.1.3. BeagleBone Black	29
2.1.1.4. Módulo ESP3266 NODE MCU	30
2.1.1.5. Análisis de los componentes	31
2.2. Tipos de tecnologías aplicadas para geolocalización	32
2.2.1. Módulo GPS	32
2.2.2. Redes LPWAN.....	33
2.2.3. Bluetooth.....	34
2.3. Medios de comunicación.....	36
2.3.1. Wi-Fi	36
2.3.2. Redes móviles GSM.....	37
2.3.3. ZigBee	38
2.3.4. Tipo de conexiones a las redes móviles	39
2.4. Buzzer.....	40
2.4.1. Tipos de Buzzer.....	41
2.5. Frecuencia cardiaca	42
2.6. Entornos de desarrollo.....	43
2.6.1. Android Studio	43
2.6.2. App Inventor	44
2.6.3. Xamarin.....	45
2.6.4. Comparación de los entornos de desarrollo	46

2.6.5. Arduino IDE.....	46
2.7. Bases de datos	46
2.7.1. MySQL.....	46
2.7.2. AWS (Amazon Web Services).....	47
2.7.3. MongoDB.....	48
2.7.4. Firebase	48
2.7.5. Comparación de Bases de Datos	49
2.8. Análisis de factibilidad.....	50
2.8.1. Factibilidad técnica.....	50
2.8.2. Factibilidad operacional	51
2.8.3. Factibilidad Económica.....	51
2.8.3.1. Cálculo de VAN y TIR.....	52
2.8.4. Factibilidad Legal.....	54
CAPÍTULO III	55
DISEÑO Y DESARROLLO	55
3.1. Arquitectura del Prototipo IoT	55
3.2. Construcción del prototipo IoT (hardware).....	56
3.2.1. Elementos del prototipo	57
3.2.1.1. Arduino Nano Versión 3	57
3.2.1.2. Micromódulo Sim800L GSM/GPRS	57
3.2.1.3. Módulo GPS Ublox Neo 6m	57
3.2.1.4. Placa PCB.....	57
3.2.1.5. Batería Turnigy Nano-tech.....	57
3.2.1.6. Módulo de carga Tp4056	58
3.2.1.7. Sensor de pulso de frecuencia cardiaca.....	58
3.2.1.8. Diagrama de bloques de la conectividad de componentes	58
3.2.2. Diseño e implementación del prototipo	59
3.2.2.1. Diagrama de flujo general	61
3.2.3. Fases del desarrollo del prototipo.....	62
3.2.3.1. Fase del medio de comunicación.....	62
3.2.3.2. Fase de geolocalización.....	64
3.2.3.3. Fase de almacenamiento.....	66
3.2.3.4. Fase de Monitoreo	67
3.2.4. Diagramas UML.....	68

3.2.5. Diagramas de casos de usos	68
3.2.6. Diagramas de secuencia	71
3.2.7. Diagramas de clases	73
3.2.8. Diagrama de base de datos no relacional	73
3.2.9. Diseño de la App	74
3.2.9.1. Desarrollo de la App	74
3.2.9.2. Esquema del software.....	74
3.2.10. Permisos de conexión.....	75
CAPÍTULO IV	77
PRUEBAS Y RESULTADOS DEL PROTOTIPO	77
4.1. Pruebas de funcionamiento de hardware.....	77
4.1.1. Pruebas de encendido de prototipo.....	77
4.2. Pruebas de funcionamiento de software.....	79
4.2.1. Pruebas de toma de datos	80
4.2.2. Pruebas de almacenamiento	81
4.2.2.1. Monitoreo en el Realtime Database de Firebase	82
4.2.3. Pruebas de la App.....	83
4.2.3.1. Prueba de registro.....	83
4.2.3.2. Prueba de entrada al sistema	83
4.2.3.3. Pruebas de registro de mascota	84
4.2.3.4. Pruebas de localización y visualización de la frecuencia cardiaca	85
4.2.3.5. Pruebas de validación de campos de la App	86
4.2.4. Pruebas y resultados del prototipo	86
4.2.4.1. Resultados del GPS	86
4.2.4.2. Resultados del sensor de frecuencia cardiaca.....	88
4.2.4.3. Resultados de elección del prototipo (hardware)	89
4.2.4.4. Resultados de perímetro del GPS.....	90
CONCLUSIONES	93
RECOMENDACIONES	95
LISTA DE REFERENCIAS	96
ANEXOS.....	101

ÍNDICE DE FIGURAS

Figura 1. Ciclo para el diseño de SE.	5
Figura 2. Disposición del prototipo.	6
Figura 3. Desarrollo del prototipo.	7
Figura 4. Satélites NAVSTAR representación gráfica.	9
Figura 5. Evolución del GPS.	10
Figura 6. Conformación del sistema GPS.	11
Figura 7. Funcionalidad del sistema GPS.	12
Figura 8. Representación de bloques de un sistema eléctrico.	13
Figura 9. Representación de bloques del enlace de las comunicaciones electrónicas.	14
Figura 10. Elementos básicos de un microcontrolador.	15
Figura 11. Variación del ritmo cardiaco.	16
Figura 12. Bloques de la arquitectura GSM.	19
Figura 13. Capas de la nube.	22
Figura 14. Fortalezas y debilidades de una infraestructura en la nube.	22
Figura 15. Características de Android.	26
Figura 16. Arquitectura de Android.	27
Figura 17. Arduino Nano.	28
Figura 18. Placa de Raspberry Pi.	29
Figura 19. BeagleBone Black.	30
Figura 20. Módulo Node MCU ESP8266.	30
Figura 21. Descripción de los componentes.	31
Figura 22. Módulo GPS.	32
Figura 23. Placa ESP8266 con Sigfox.	34
Figura 24. Punto de acceso (Router).	36
Figura 25. Módulo GSM SIM800L.	37
Figura 26. Módulo XBeePRO de ZigBee.	38
Figura 27. Matriz de comparación de los estándares de comunicación.	39
Figura 28. Matriz de comparación de redes móviles.	40
Figura 29. Estructura interna buzzer electromagnético.	41
Figura 30. Estructura interna de un piezoeléctrico.	42
Figura 31. Sensor pulso cardiaco.	43
Figura 32. Arquitectura del prototipo IoT.	55
Figura 33. Estructura interna del prototipo.	56
Figura 34. Diseño de conectividad.	58
Figura 35. Conectividad de los componentes del prototipo.	59
Figura 36. Piezas adaptadas en el prototipo.	60
Figura 37. Prototipo montando en placa PCB.	60
Figura 38. Fases del desarrollo del prototipo (hardware).	61
Figura 39. Tasa de transferencia de 9600 baudios.	63
Figura 40. Comandos AT.	63
Figura 41. Comandos de inicialización para conectarse a la red GPRS del APN.	64
Figura 42. Solicitud HTTP utilizando los comandos AT.	64
Figura 43. Librería para configurar pines para transmisión UART.	65
Figura 44. Variables tipo String para almacenar valores de latitud y longitud.	65

Figura 45. Conversión de valores de latitud y longitud.	65
Figura 46. Enlace con la solicitud HTTP.	66
Figura 47. Recepción de solicitudes en HTTP.	66
Figura 48. Inicialización de variables obtención del pulso cardiaco.	68
Figura 49. Función getBeatsPerMinute().	68
Figura 50. Esquema de inicio de sesión.	69
Figura 51. Diagrama de especificaciones de actividades de aplicación móvil.	70
Figura 52. Diagrama de secuencia para el inicio de sesión de la aplicación móvil.	71
Figura 53. Esquema de secuencia del registro de mascotas.	72
Figura 54. Diagrama de clases de la aplicación móvil.	73
Figura 55. Diagrama NoSQL de aplicación móvil.	73
Figura 56. Pantallas de interacción del software.	75
Figura 57. Conceder permiso de conexión a internet y ubicación.	75
Figura 58. Uso de la llave de Google API.	76
Figura 59. Plugin para el GSM.	76
Figura 60. Dependencias de Gradle para Firebase.	76
Figura 61. Switch ON/OFF del prototipo.	77
Figura 62. Comprobación de encendido de luces led.	78
Figura 63. Prototipo final adaptado a un arnés.	78
Figura 64. Prototipo final adaptado a un arnés.	79
Figura 65. Búsqueda de señal del GSM.	80
Figura 66. Búsqueda del APN de Claro.	80
Figura 67. Valores de las coordenadas obtenidas por el GPS.	81
Figura 68. Envío y verificación de datos a la base.	82
Figura 69. Gráficas del consumo y gestión de los datos.	82
Figura 70: Autenticación y almacenamiento de usuario.	83
Figura 71. Inicio de sesión y almacenamiento en Firebase.	84
Figura 72. Registro de datos de mascota.	85
Figura 73. Localización y frecuencia cardiaca.	85
Figura 74. Frecuencia cardiaca de la mascota.	86
Figura 75. Gráfica de frecuencia cardiaca.	89
Figura 76. Medidas del prototipo para la elección.	90
Figura 77. Alerta de perímetro de la mascota.	91
Figura 78. Validación de campos en diferentes escenarios.	92

ÍNDICE DE TABLAS

Tabla 1. Matriz de comparación de microcontroladores.....	31
Tabla 2. Matriz de comparación de geolocalización.....	36
Tabla 3. Matriz de comparación de los entornos de desarrollo.....	46
Tabla 4. Matriz de comparación de Bases de Datos	50
Tabla 5. Análisis de factibilidad técnica para el desarrollo del prototipo	50
Tabla 6. Factibilidad económica de los recursos apropiados para el prototipo.....	51
Tabla 7. Flujo de caja.	54
Tabla 8. Proceso de conexión de sensores y módulos.....	62
Tabla 9. Proceso para el envío de datos Firebase.....	67
Tabla 10. Especificación del inicio de sesión a la aplicación móvil	69
Tabla 11. Especificaciones de las actividades de la aplicación móvil	70
Tabla 12. Actividades del usuario en la aplicación móvil.....	71
Tabla 13. Actividades en la aplicación móvil	72
Tabla 14. Resultado de las muestras.	87
Tabla 15. Resultados de pulsaciones según el estado energético.....	88

RESUMEN

El presente proyecto de titulación tiene como objetivo la geolocalización y visualización de la frecuencia cardiaca de la mascota en la ciudad de Quito. Partiendo del problema que genera para la sociedad la pérdida de la mascota.

Para ello se propone un prototipo que pueda mitigar dicho problema, por tal motivo, se realizó un análisis previo en base a matrices de comparación que permitieron la elección de los componentes idóneos para el prototipo, para brindar las mejores características, de bajo costo y con un buen rendimiento cumpliendo así los requerimientos previos.

Para el desarrollo del prototipo se llevaron a cabo varios procedimientos para obtener un producto funcional en los procesos de geolocalización, comunicación y conectividad. Adicional a ello se implementó una aplicación móvil interactiva con el usuario, para la localización de las mascotas.

Finalmente se realizaron las pruebas de integración, donde se obtuvo la precisión del módulo GPS del prototipo con un margen de error de 10,81 metros con respecto al dato que ofrece el GPS de la aplicación móvil, adicionalmente para los resultados del sensor de la frecuencia cardiaca se realizó un promedio en base a cinco estados diferentes de la mascota que dieron como resultado un margen de error del 10.54% con respecto al dato real tomado manualmente.

ABSTRACT

The present titling project aims to geolocate and visualize the heart rate of the mascot in the city of Quito. Starting from the problem that the loss of the mascot generates for society.

For this purpose, a prototype is proposed that can mitigate this problem. For this reason, a previous analysis was made based on comparison matrices that allowed the selection of the ideal components for the prototype, to provide the best characteristics, low cost and good performance, thus fulfilling the previous requirements.

For the development of the prototype, several procedures were carried out to obtain a functional product in the processes of geolocation, communication and connectivity. In addition to this, a mobile application was implemented, interactive with the user, for the location of the pets.

Finally, integration tests were carried out, where the accuracy of the prototype's GPS module was obtained with a margin of error of 10.81 meters with respect to the data provided by the mobile application's GPS. Additionally, for the results of the heart rate sensor, an average was made based on five different states of the pet, resulting in a margin of error of 10.54% with respect to the real data taken manually.

INTRODUCCIÓN

Antecedentes

El cariño que se tiene por las mascotas dentro del núcleo familiar ha aumentado en la sociedad, puesto que son consideradas un miembro más de la familia y por ende ocupan un lugar especial dentro de la misma, convirtiéndose en un pilar fundamental sobre todo para los más pequeños del hogar.

Con respecto a las leyes de protección animal que rigen en el país, no son severas pues no permitan castigar rigurosamente a las personas que se dediquen a maltratar, abandonar o terminar con la vida de las mascotas. Por este motivo se tendrá en cuenta el artículo 249 de la Ley Orgánica de Bienestar Animal (L.O.B.A) (Ver Anexo 1).

El presente trabajo está encaminado a colaborar a los propietarios de mascotas para que tengan un control más directo con las mismas y tratar de mitigar en un porcentaje significativo las pérdidas o extravíos gracias a un sistema de geolocalización implementado en un prototipo que estará situado en el cuerpo de la mascota, además empleando una aplicación móvil mostrar la ubicación y visualizar su frecuencia cardiaca.

Este tipo de técnicas fueron probadas exitosamente en laboratorios de Agro electrónica del Instituto de Ingeniería Rural (Buenos Aires) para aplicarlo a un collar con tecnología GPS para lograr el monitoreo a tiempo real de los animales. Gracias a los datos de georreferenciación se sabe exactamente la ubicación del animal y se tiene la posibilidad de trazar una ruta a través de una aplicación web (Garandi, Clemares, & Moltoni, 2015, pág. 2).

El aumento de familias que consideran adoptar una mascota genera la preocupación por parte de las entidades que se dedican al cuidado y protección de los animales desprotegidos, en mayor medida de los perros y gatos, ya que no siempre terminan proporcionándoles un buen cuidado y en muchas ocasiones incluso vuelven a ser víctimas de abandono por parte de sus dueños. Esto ha motivado a las organizaciones protectoras de animales optar por adquirir herramientas adicionales basadas en las tecnologías actuales tales como un dispositivo GPS para obtener información en sobre el posicionamiento, recorridos o movimientos que realiza la mascota (Romarís Caamaño, 2019, págs. 1,2).

Problema

En diversos países del mundo tales como la India, alrededor de 20.000 personas mueren a causa de mordeduras de perros callejeros que poseen rabia (M.K. Sudarshan a, 2005, pág. 51). Este tipo de situaciones obliga a la población de estos países llevar a cabo prácticas como el sacrificio, sin embargo, también se toman otro tipo de medidas como la esterilización y vacunación contra la rabia (M.K. Sudarshan a, 2005, pág. 52). Países como Holanda en cambio, gracias a sus buenas prácticas y el apoyo del gobierno han aminorado el exceso de perros callejeros por habitante. Todo esto gracias al nivel de cultura que tienen las personas y las estrictas leyes que se aplican en ese país contra el maltrato animal. (Sternheim, 2012, pág. 6) .

A nivel de Latinoamérica con varios países aún en vías de desarrollo, los gobernantes de distintos territorios no ofrecen realizar cambios significativos para mejorar la frágil situación de aquellos animales que viven en las calles y ponen en peligro la salud de los ciudadanos al tener que convivir con toda la insalubridad que los perros de la calle producen, ya que no cuentan con los cuidados apropiados. Un claro ejemplo, es la cantidad de excrementos que se encuentran en la vía pública, así como los cuerpos de los perros atropellados que con el tiempo se descomponen y todo se mezcla

con el aire que respiran las personas o a su vez las enfermedades como la rabia o la sarna de las cuales los perros son portadores. (López, 2016, pág. 8)

Ecuador no se queda atrás ya que cuenta con aproximadamente 300.000 perros callejeros a nivel nacional (Aguaguña, 2017, pág. 5). Un estudio realizado para censar a los perros callejeros en el Distrito Metropolitano de Quito en el año 2018 arrojó el resultado final que existe 1 perro callejero por cada 22 habitantes (Stella de la Torre1, 2018).

Justificación

La población de perros callejeros en Quito se ha acrecentado en los últimos años (Castellanos, 2018). Es común encontrarse con nuevas tecnologías que aportan soluciones innovadoras y vanguardistas a problemas cotidianos que le dan un nuevo enfoque. Es así como varias compañías dedicadas a la manipulación de los datos evolucionan y ofrecen nuevos servicios para controlar toda esta información para almacenarla en plataformas donde se visualicen esos datos en cualquier parte del mundo, ya sean servicios como Google Cloud, Azure, Amazon Web Services, Firebase entre otros.

Estos datos necesitan ser generados por un conjunto de componentes electrónicos para ser procesados en una aplicación para visualizar dicha información y posteriormente almacenarla en la nube para que esté disponible en todo momento, comportándose como un sistema embebido combinando características de hardware como de software.

Se necesita una solución que otorgue una perspectiva diferente frente al entorno que se vive a diario con animales de la calle y los problemas que traen a la sociedad puesto que no existen políticas estrictas de control en la ciudad, ni tampoco una concientización por parte de las autoridades locales que permita un cambio de mentalidad en la sociedad.

El motivo del presente proyecto será diseñar un prototipo para obtener las coordenadas geográficas de la mascota que permita al usuario visualizar la frecuencia cardiaca y saber sobre el estado de salud de la mascota.

Objetivos

Objetivo General

Desarrollar un prototipo para la geolocalización y monitorización de la frecuencia cardiaca de mascotas en la ciudad de Quito.

Objetivos Específicos

Analizar las características de los módulos que se ajusten al prototipo tanto de hardware (sensores, partes electrónicas) como de software (lenguajes de programación, plataformas, aplicaciones) que se adapten mejor a los requerimientos para obtener una solución viable.

Diseñar el prototipo en hardware y software aplicando metodologías de diseño adecuadas.

Desarrollar el prototipo en base al diseño para localizar y monitorizar la frecuencia cardiaca de las mascotas.

Realizar las pruebas correspondientes, tanto de hardware como software y verificar el correcto funcionamiento de ambas partes integradas para así obtener un resultado adecuado del prototipo.

Metodología

Para el presente proyecto se utilizará la metodología de diseño de IoT con Sistema Embebido ya que un dispositivo IoT está asociado a un sistema embebido, gracias a un microcontrolador que almacena en su memoria varias instrucciones que el usuario ejecutará. En la Figura 1 se detalla el proceso que ayudará a la ejecución de la geolocalización y la visualización de la frecuencia cardiaca.

Proceso del diseño de SE



Figura 1. Ciclo para el diseño de SE.
Elaborado por: Los autores.

Definición del Sistema

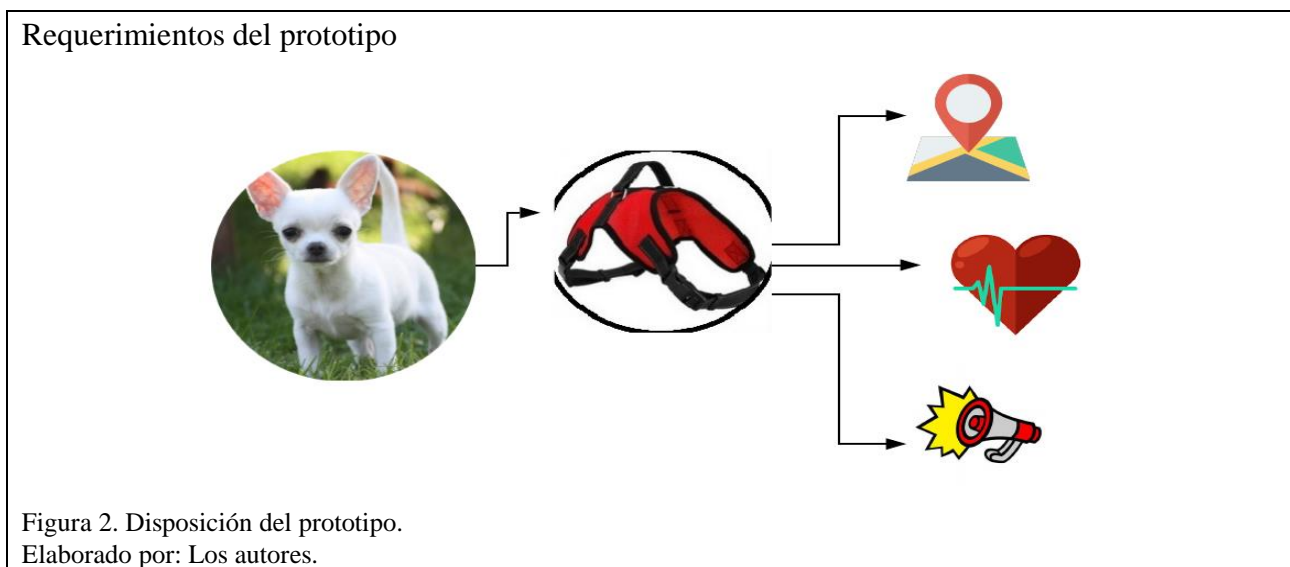
El presente sistema embebido consta de un módulo que brindará la información del geoposicionamiento de la mascota y de un sensor para visualizar la frecuencia cardiaca del animal.

El prototipo estará ubicado en el cuerpo del animal, será adecuado sobre todo para entornos abiertos, como parques o calles aledañas de donde la mascota pueda extraviarse, eventualmente se necesitará de una buena señal de telecomunicación ya que se usará un módulo GSM.

Toda esta información se visualizará gracias a una aplicación móvil desarrollada para el entorno de Android ya que es uno de las más elegidos por los usuarios. Los datos se gestionarán en la nube usando la plataforma de Firebase accediendo a ellos desde cualquier lugar en que se encuentre el dueño de la mascota.

Definición de Requisitos

Como se considera en la Figura 2, las características fundamentales que requiere el prototipo se basan en la geolocalización, es decir, latitud y longitud de la ubicación de la mascota, además del control de los latidos del corazón, la emisión de una notificación si sale de un perímetro establecido por el usuario que esté utilizando la aplicación, los datos de la mascota, del posicionamiento, así como la frecuencia cardiaca.



Operaciones para desarrollar

Las dos variables que se ocuparán tanto en la geolocalización como en la visualización de la frecuencia cardiaca se enviarán a través de una comunicación GSM a la nube, donde esos datos se guardarán y se visualizarán en el smartphone gracias a una aplicación móvil que estará interconectada al dispositivo IoT. En la Figura 3 se tienen las características del prototipo.

Características del prototipo

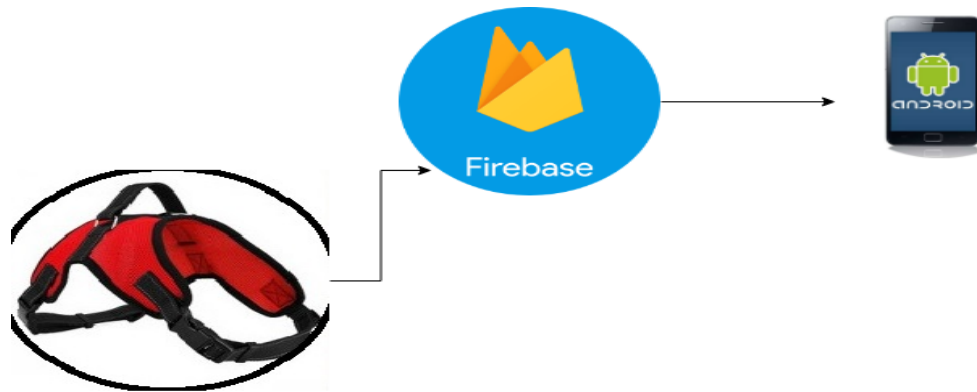


Figura 3. Desarrollo del prototipo.
Elaborado por: Los autores.

Elaboración del Modelo

Para la elaboración del modelo se tendrá en cuenta la parte de alimentación, comunicación, microcontrolador, entradas y salidas de los sensores tanto de la geolocalización como de la frecuencia cardiaca.

Pruebas

Se ejecutarán pruebas de funcionamiento como de integración de los elementos analizados respectivamente para el prototipo, tanto en la parte de hardware como en la parte de software para que no exista ningún problema y se evaluarán los datos recibidos por los sensores.

CAPÍTULO I

MARCO TEÓRICO

1.1. Geolocalización

El propósito del sistema de geolocalización es obtener un punto específico en cualquier lugar de la tierra y así adquirir coordenadas de latitud y longitud, ya sea de un objeto que permanece inmóvil en un plano fijo o bien para situarlo cuando se encuentre en movimiento (Huerta, Mangiaterra, & Noguera, 2005, págs. I-2).

Para obtener la ubicación se necesita la ayuda de satélites que se encuentren en órbita en el espacio para que emitan señales entre el emisor y el receptor logrando una triangulación entre los satélites (Huerta, Mangiaterra, & Noguera, 2005, págs. I-2).

1.1.1. Usos de la geolocalización

Debido a los requerimientos y las necesidades que se tienen en la actualidad para obtener la ubicación exacta de un elemento físico, este tipo de prácticas se han vuelto comunes debido a que los usuarios utilizan más servicios de pedido a domicilio, búsquedas de restaurantes, hoteles, gasolineras, así como para rastrear vehículos robados o como es el caso, para obtener la ubicación de las mascotas con la ayuda de una aplicación móvil.

1.1.2. GPS

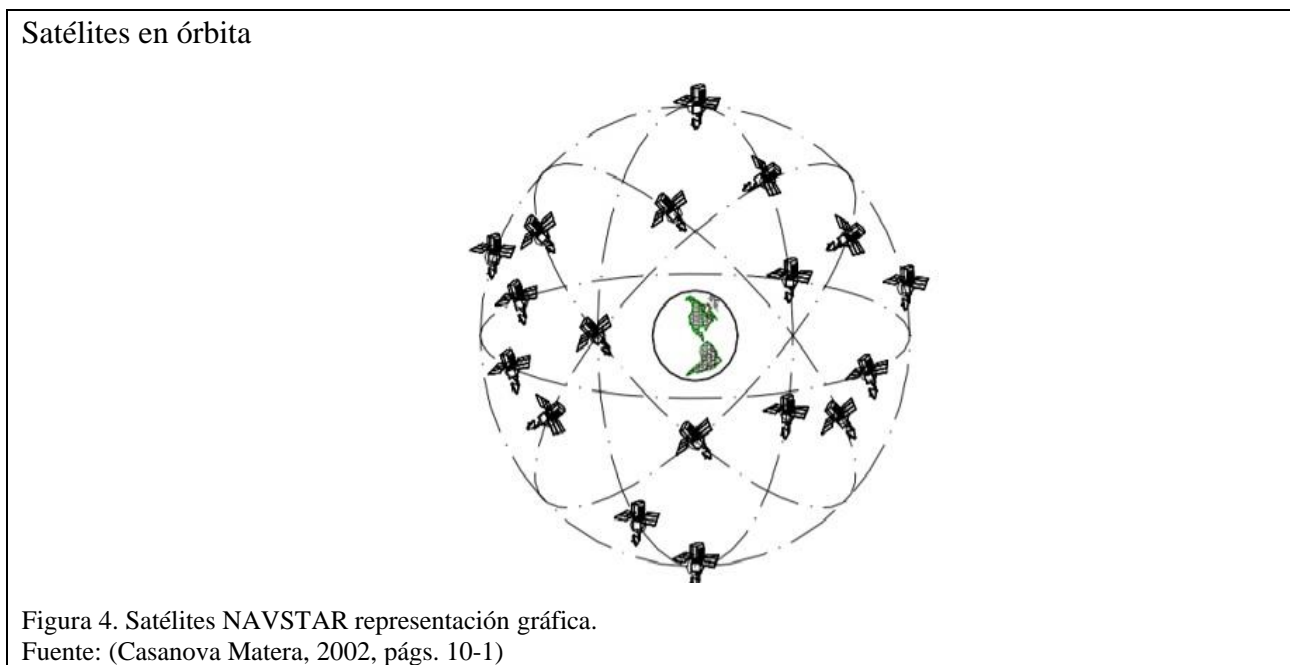
El sistema de localización GPS¹ está basado en la geolocalización pues necesita de satélites que triangulen la posición y de un receptor para obtener los datos concebidos, actualmente consta de 24 satélites artificiales (Casanova Matera, 2002, págs. 10-1), que determinan la posición de un objeto en la superficie de la Tierra (Hernández Blanco).

¹ GPS: Global Positioning System

Para diferenciar el tipo de funciones se separa los servicios por niveles:

- **De posicionamiento normal:** Este servicio de posicionamiento está disponible para todos los usuarios de GPS (Tomasi, 2003, pág. 853).
- **De posicionamiento preciso:** La función que realiza es la presentación de su posicionamiento con exactitud, velocidad y tiempo que se encuentran disponibles solamente a usuarios con autorización previa por el Departamento de Defensa de Estados Unidos (Tomasi, 2003, pág. 853).

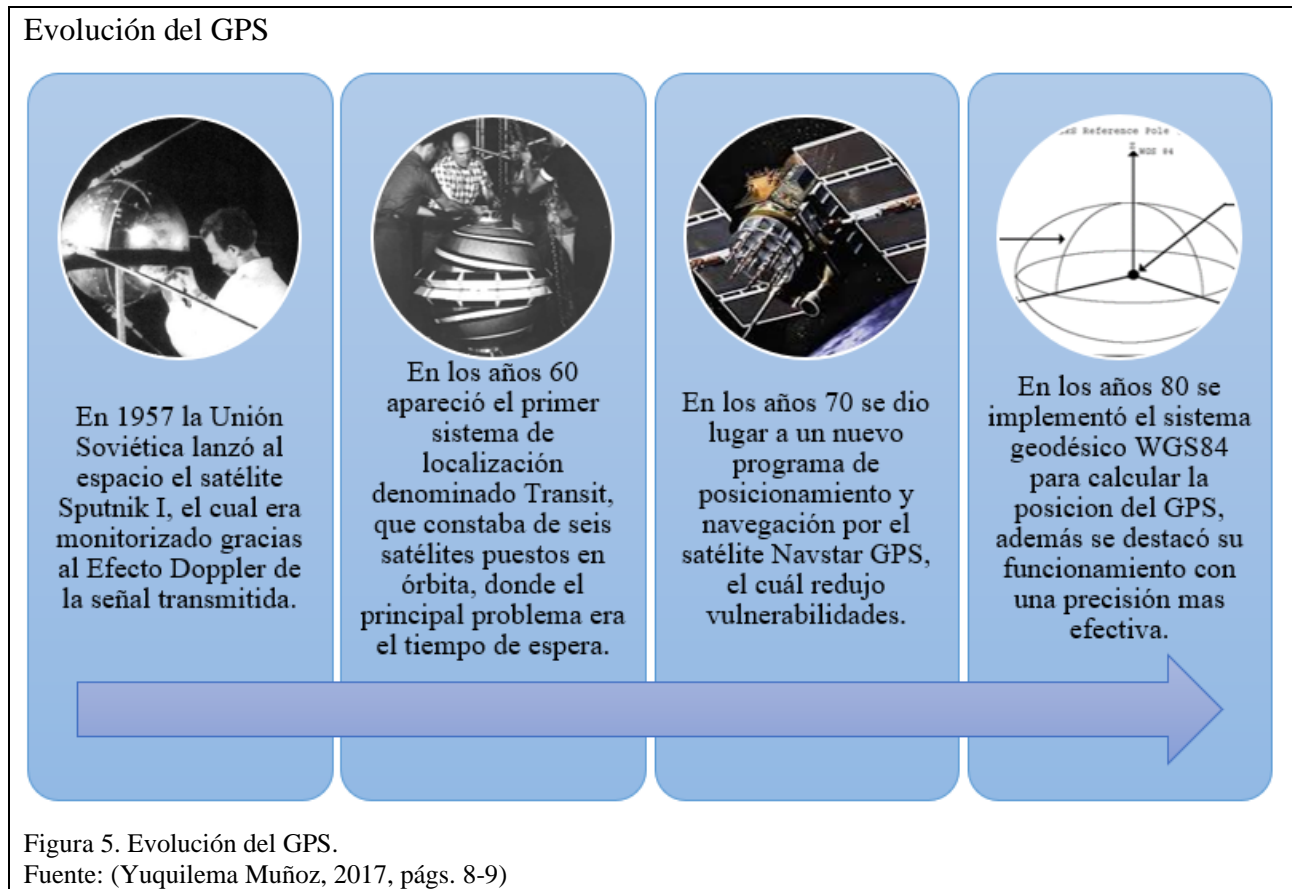
En la Figura 4, está representado gráficamente la trayectoria de los satélites alrededor de la tierra.



1.1.2.1. Historia GPS

Es importante percatarse sobre la evolución que han sufrido las comunicaciones y como los humanos indagan en nuevas ideas para obtener mejores métodos de comunicación que impliquen nuevas tecnologías, como el seguimiento de objetos y su ubicación precisa sin importar el lugar que estén ubicados dichos elementos.

En la Figura 5, se detalla la historia del GPS.



Debido a este crecimiento, en la actualidad si se adquiere un smartphone se tiene acceso a la geolocalización, junto con otras tecnologías que realizan tareas específicas dependiendo de los requerimientos del usuario.

1.1.2.2. Técnicas del GPS

Está formado por una serie de segmentos que se observan en la Figura 6, se dividen en tres aspectos apropiados.

Conformación del sistema GPS

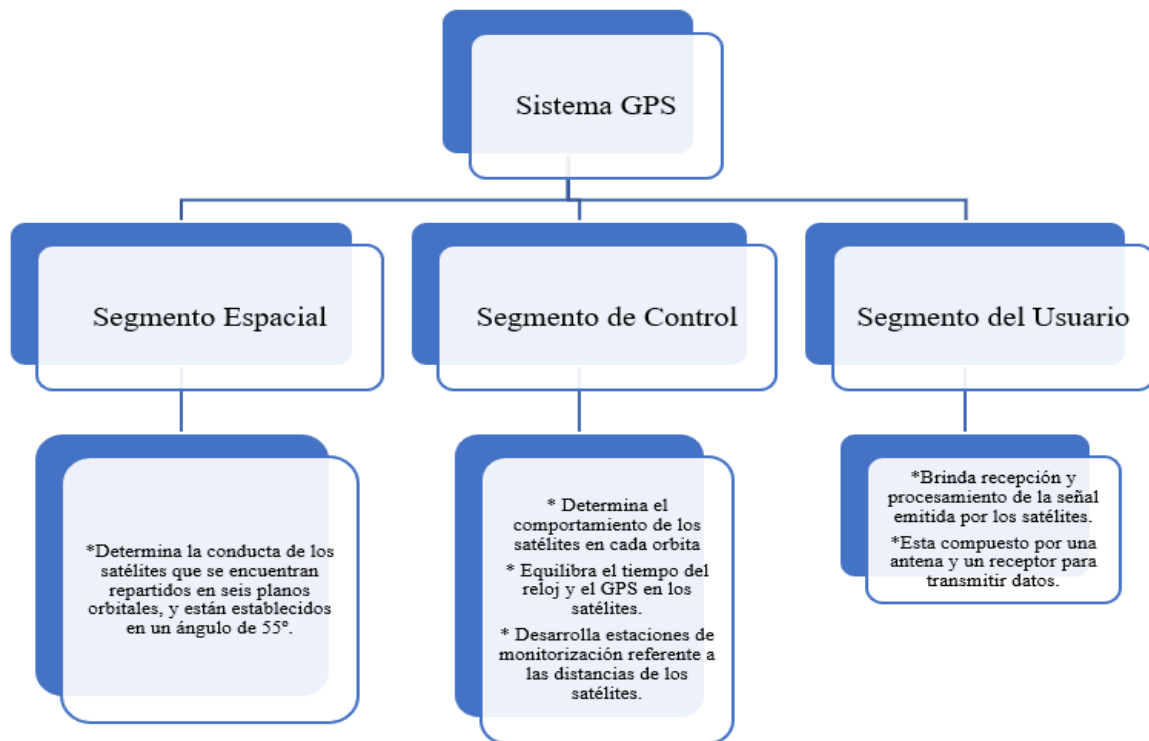
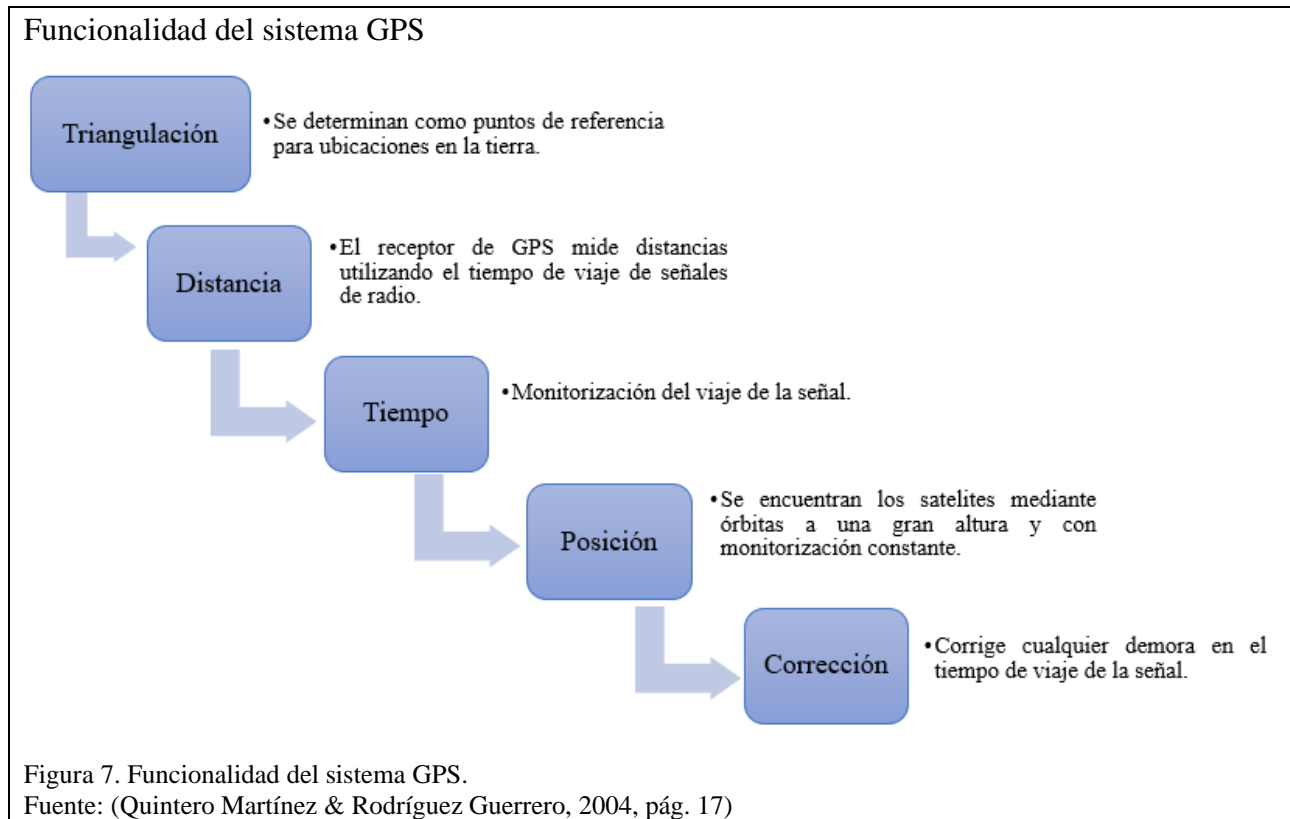


Figura 6. Conformación del sistema GPS.
Fuente: (Huerta, Mangiaterra, & Noguera, 2005)

1.1.2.3. Funcionamiento del Sistema GPS

La meta del GPS es calcular el recorrido que se produce en base a una señal que posteriormente es difundida por medio de una agrupación de satélites artificiales hacia los receptores ubicados en cualquier punto de la tierra, interpretando las señales captadas para obtener una mejor precisión (León Quiñones, 2005, págs. 10-2). En la Figura 7, se determina la funcionalidad del sistema GPS.



1.2. Sistemas Embebidos y dispositivos IoT

1.2.1. Definición de sistema embebido

Es un sistema eléctrico que está conformado por hardware y software con la capacidad para controlar diferentes componentes ejecutando una labor específica, con la particularidad que se realizará en un periodo inmediato. Este tipo de sistemas se programan directamente en el lenguaje ensamblador del microcontrolador (UNED, 2011), sus características son las siguientes:

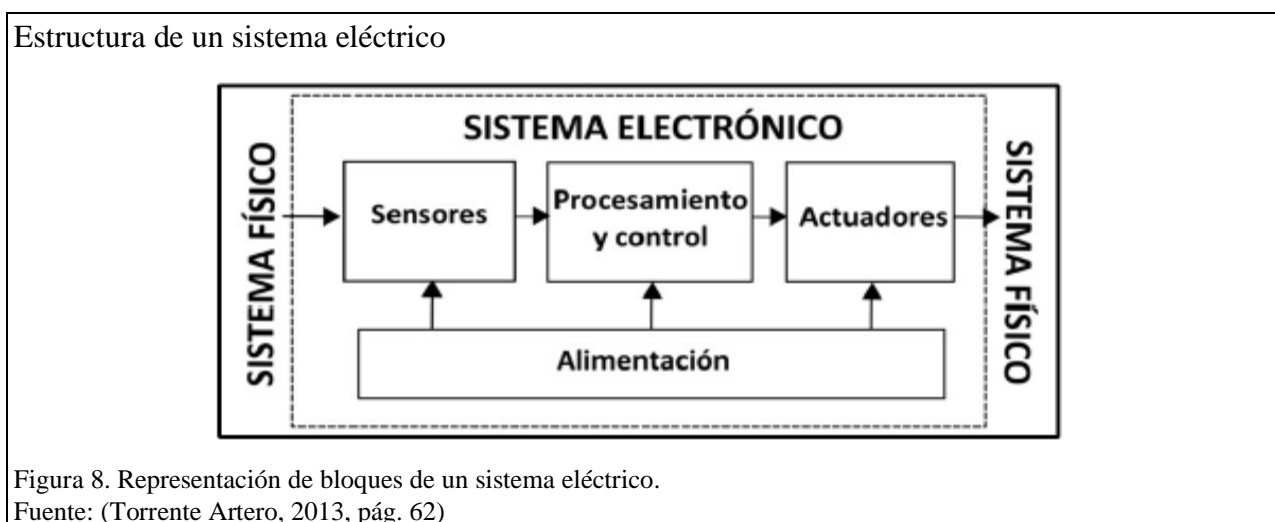
- Deben ser confiables para el usuario.
- Deben ser eficientes en cuanto a energía, tamaño de código, peso y costo.
- Facilidad para elaborar nuevas versiones modificadas (UNED, 2011).

1.2.2. Sistema eléctrico

La integración de sensores, el esquema de procesamiento y control, así como la fuente de alimentación forman un sistema eléctrico donde se cumplen diferentes tareas:

- El circuito intrínseco de dominio convierte la información obtenida del exterior en una marca eléctrica gracias a los sensores (Torrente Artero, 2013, pág. 61).
- Los circuitos que se ubican en el interior de un sistema eléctrico utilizan un método propio para procesar la muestra eléctrica que será manipulada y dependerá del diseño de las distintas partes de hardware como la agrupación lógica de instrucciones (Torrente Artero, 2013, pág. 61).
- La señal eléctrica previamente tratada es transformada por medio de los actuadores en energía que actuara inmediatamente en el exterior (Torrente Artero, 2013, pág. 61).
- Para dotar de fuerza al suministro eléctrico se necesita que el hardware sea capaz de cumplir dicha tarea (Torrente Artero, 2013, pág. 61).

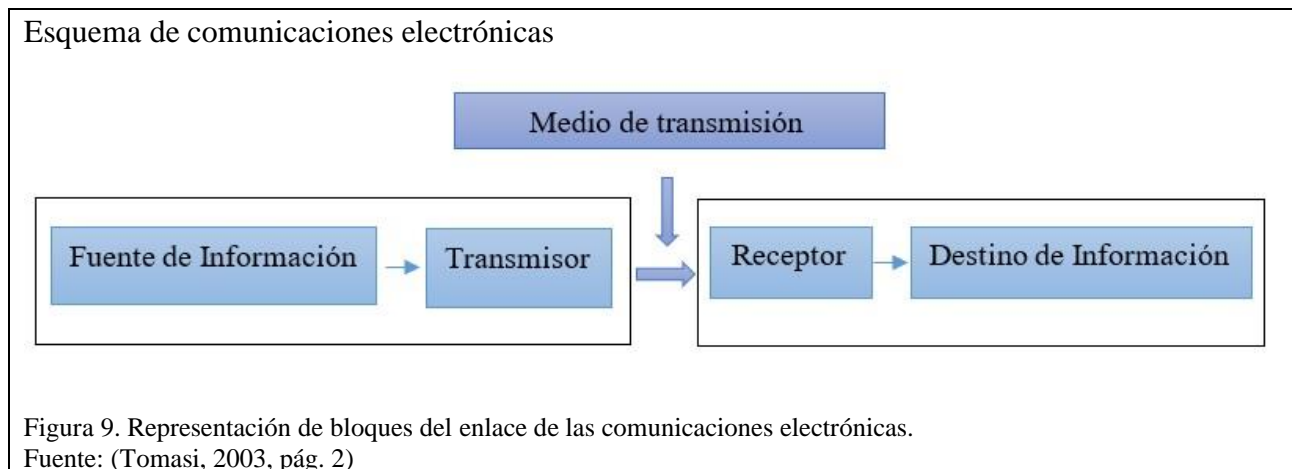
En la Figura 8. se detalla el esquema de un sistema eléctrico.



1.2.3. Sistema de comunicaciones electrónicas

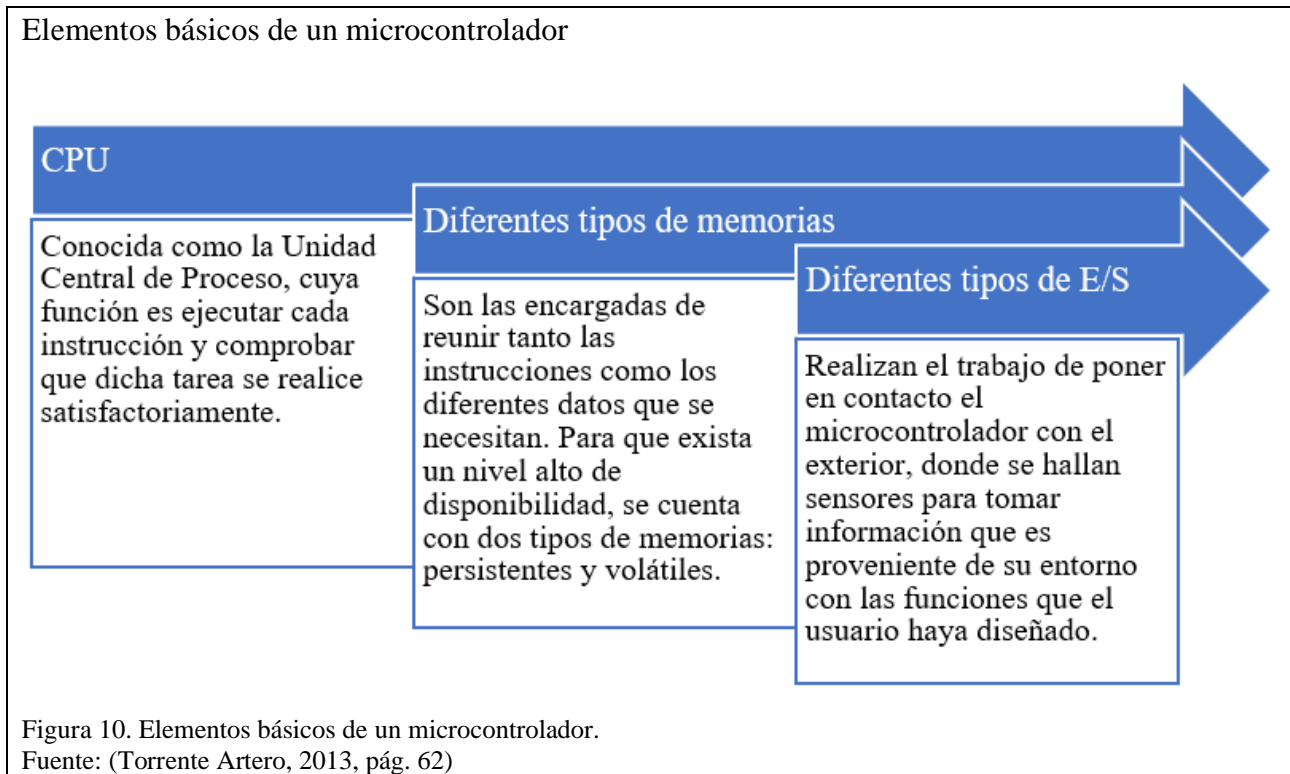
El propósito de la técnica de comunicación electrónica es la transferencia, ingreso y el tratamiento de información por medio de dos o más puntos, a través de circuitos eléctricos, ya sea que la información esté dispuesta en formato analógico o digital, la intención de la fuente principal es transformar la energía electromagnética previa a ser distribuida con la ayuda de un método electrónico de conexiones (Tomasi, 2003, pág. 1).

En la Figura 9, se detalla la representación de bloques del enlace de las comunicaciones electrónicas y la manera como opera.



1.2.4. Microcontrolador

Es un mecanismo electrónico que está compuesto por un único encapsulado con un enorme número de piezas, con la primordial característica de ser programable, siendo aptos para verificar de manera independiente una lista de ordenes anteriormente fijadas por el usuario (Torrente Artero, 2013, pág. 62). En la Figura 10, se exponen los elementos básicos de un microcontrolador.



1.2.5. Arduino

Representa a una minicomputadora con funciones esenciales para el campo de la electrónica y de la informática, su estructura se entiende fácilmente, al tratarse de una placa de código abierto que cuenta con pines de entrada, así como de salida para conexiones del tipo analógico y digital con un lenguaje de programación basado en Java (Proserquisa, 2016).

La plataforma es bastante didáctica y amigable con el usuario, permitiéndole que desarrolle proyectos interesantes sin necesidad de pagar por una licencia de uso (Proserquisa, 2016).

1.2.6. Frecuencia Cardíaca

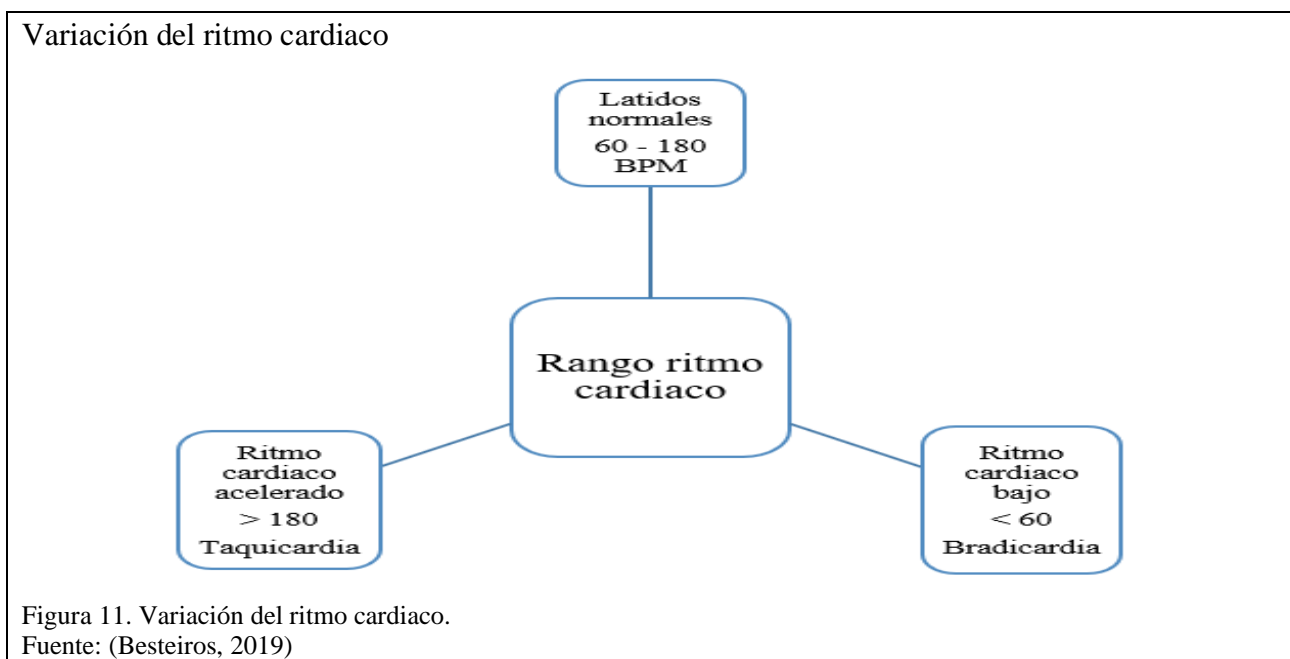
En los animales este tipo de mediciones es distinta, debido a que los valores normales son diferentes y los números de latidos por minuto del corazón son mayores que el de los humanos, y también están expuestos a sufrir contratiempos cardíacos durante su vida, además en razas con angostos

orificios nasales el problema es persistente pues debido a esta patología de nacimiento necesitan de cuidados especiales y evitar someterlos a ejercicios extremos.

La frecuencia cardiaca hace referencia al número de latidos por minuto del corazón y es la encargada de bombear sangre a todos los órganos del cuerpo para un buen funcionamiento del organismo (Besteiros, 2019).

El ritmo cardiaco de un perro oscila entre 60 a 160 BPM², sin embargo, en este punto va a depender bastante del tipo de raza, así como el tamaño del perro, pues el ritmo cardiaco varía entre 60 a 180 BPM en perros de razas pequeñas (Besteiros, 2019).

Para detallar los rangos del ritmo cardiaco dependiendo el BPM está la Figura 11.



² BPM: Beats per minute

1.2.7. Dispositivo IoT

Para enlazar diferentes dispositivos se habla del internet de las cosas como un medio en donde todos están conectados a internet junto a un grupo de sensores, actuadores y un canal de transmisión formando una comunicación entre dichos dispositivos.

Esta tecnología está diseñada para varios ámbitos como la industria, la salud, la educación, etc., otorgando bondadosas prestaciones a los usuarios interesados en mudarse a estas nuevas formas de comunicare.

Los dispositivos IoT³ son elementos que se componen de un microcontrolador el cual realiza una serie de instrucciones programables para diferentes actividades o una en específico.

1.2.8. Elementos de IoT

Consta de tres componentes primordiales que se relacionan entre sí:

Hardware: Dentro del hardware entran los componentes eléctricos y electrónicos como por ejemplo sensores y actuadores que van a controlar el sistema.

Soporte Middleware: Esta parte contiene el software encargado de aprobar el intercambio de la información entre aplicaciones para ejecutar el estudio de los datos.

Aplicaciones: Donde proporciona la comprensión de los datos analizados anteriormente y que son accesibles para el usuario (Bonilla Fabela, Tavizon Salazar, Morales Escobar, Guajardo Muñoz, & Laines Alamnia, 2016).

³ IoT: Internet of Things

1.3. Redes de Comunicación

1.3.1. Comunicación GSM

En un principio las comunicaciones telefónicas no eran lo suficientemente estables ni compatibles unas con otras, es por esa razón que GSM fue implementado como una solución para obtener un alcance global con un mismo sistema de comunicación telefónica, y a diferencia de sus predecesores este fue totalmente digital para facilitar múltiples utilidades a la red (Tomasi, 2003, pág. 898).

Debido a la gran transformación que ofrece la tecnología en tiempos modernos, la comunicación GSM cuenta con la ventaja de obtener cobertura en otros países con los que han establecido convenios gracias a la implementación de roaming⁴. (Bedoya, Salazar, & Muñoz, 2013, pág. 34).

1.3.2. Arquitectura de GSM

La arquitectura de GSM se distribuye en tres partes:

- Sistema de conmutación.
- Sistema de estación base.
- Sistema de operación y mantenimiento.

Cada uno de estos procedimientos cuentan con la disposición para desarrollar varias funciones que el GSM es idóneo para realizar (Bedoya, Salazar, & Muñoz, 2013, pág. 36).

En la Figura 12, se detalla la arquitectura GSM.

⁴ El Roaming: En las telecomunicaciones hace referencia a la utilización de una línea de telefonía móvil para llamar o recibir mensajes de texto en una ubicación geográfica distinta (otros países) a través de una red visitada. (Fernández G. M., 2008)

Arquitectura GSM

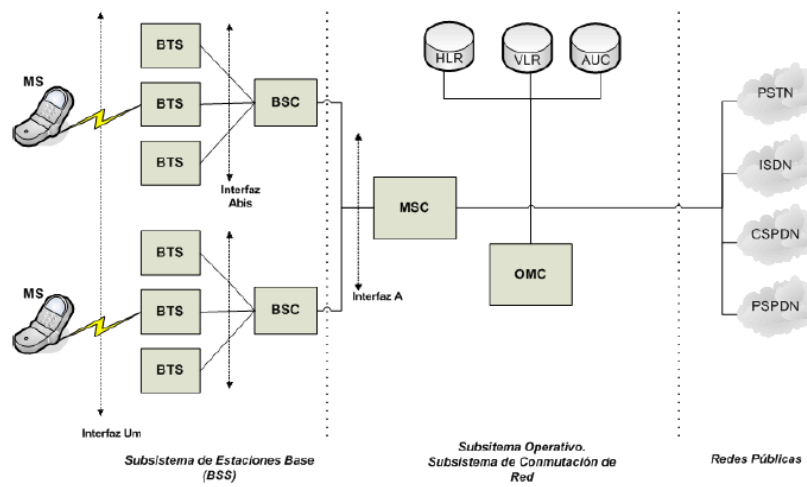


Figura 12. Bloques de la arquitectura GSM.

Fuente: (Bedoya, Salazar, & Muñoz, 2013, pág. 36)

La estructura de la arquitectura GSM donde funcionan las estaciones móviles MS (Mobile Station) son dispositivos que el usuario o abonado utiliza para comunicarse con la red de telefonía móvil al subsistema de estaciones base BSS (Base Station Subsystem). El BSS se compone por dos tipos de elementos:

Estación base transceptora BTS (Base Transceiver Station): se encarga de controlar las interfaces hacia la estación móvil MS (Bedoya, Salazar, & Muñoz, 2013, pág. 36).

Estación base controlada BSC (Base Station Controller): permite la codificación y decodificación de los distintos canales, así como el cifrado y descifrado (Bedoya, Salazar, & Muñoz, 2013, pág. 36).

Centro de conmutación móvil MSC (Mobile Switching Center): tiene la capacidad para establecer y mantener las llamadas entrantes por medio de la red y gestionar servicios complementarios para la comprobación y recuperación de datos (Bedoya, Salazar, & Muñoz, 2013, pág. 36).

1.4. Evolución de plataformas en la nube

1.4.1. Servicios en la nube

Los servicios en la nube se utilizan a través de internet, brindando prestaciones de visualización de datos en cualquier parte del mundo. Las diferentes ventajas que brindan estos servicios son seguridad, capacidad de almacenamiento entre otros, incluyendo que los recursos que ofrece la nube son mayores que la de un computador.

La base de esta arquitectura hace referencia a la granja de servidores que es parecida a una arquitectura de procesamiento de red (Ávila Mejía, Computación en la nube, 2011, pág. 46).

1.4.2. Arquitectura de la nube

Existen diferentes etapas en la arquitectura de la nube:

1.4.2.1. Software como Servicio (SaaS)

El software se encuentra hospedado en servidores pertenecientes a los encargados de brindar el servicio y el consumidor se adhiere a ellos mediante el navegador web donde los mismos son responsables tanto del funcionamiento como la disponibilidad que requiera el usuario de cada uno de sus servicios.

La mayoría de las actividades desarrolladas se desempeñan desde una ubicación central, en lugar de realizarlas en la sede o ubicación de cada cliente, lo que permitirá el acceso remoto a cada uno de sus servicios dependiendo lo que el usuario necesita.

1.4.2.2. Plataforma como Servicio (PaaS)

Es un tipo de servicio en la nube donde el proveedor ofrece acceso a un entorno basado en cloud donde los usuarios crean y distribuyen sus propias aplicaciones brindando un servicio de plataforma con todo lo necesario para dar soporte al ciclo del desarrollo de las aplicaciones, sin

preocuparse en la infraestructura que se ejecutarán las aplicaciones mediante el uso de servicios y herramientas que apoyan para agilizar todos los procesos de desarrollo e implementación, en esta parte el usuario administra el despliegue de las aplicaciones sin preocuparse de armar la infraestructura ya que no tiene acceso a ese perfil.

Gracias al uso del PaaS se reduce el hardware físico del usuario, y es lo que llama la atención a varios desarrolladores web ya que no se necesita un control del sistema (Ávila Mejía, Computación en la nube, 2011, pág. 47).

1.4.2.3. Infraestructura como Servicio (IaaS)

El encargado de facilitar prestaciones provee el software y las aplicaciones a través del internet, para que los usuarios se registren e ingresen a través de la web o API⁵ que les brinda el administrador.

IaaS ofrece una propuesta con mucha flexibilidad para el uso que necesite el usuario, gracias a este servicio se tiene una solución en la que se paga solamente por los recursos utilizados, como por ejemplo tiempo CPU, espacio para base de datos, transferencia de datos.

Es importante mencionar que IaaS permite la escalabilidad automática donde se controla más recursos según se requiera.

En la Figura 13, se especifica las etapas de la nube.

⁵ API: Application Programming Interface

Etapas de la nube

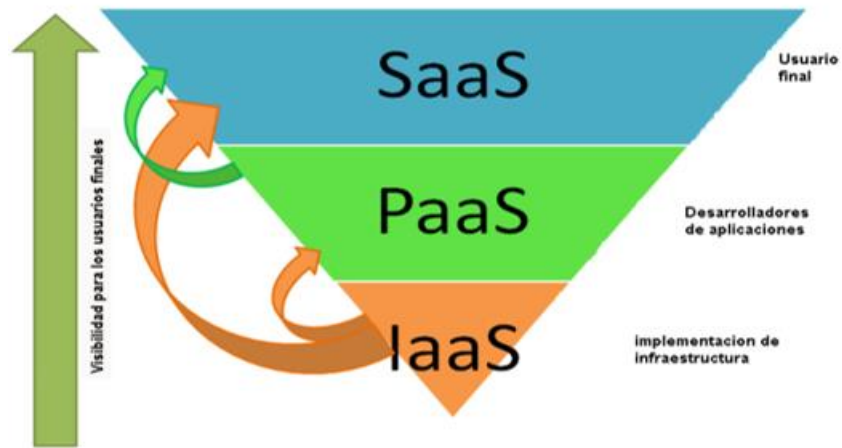


Figura 13. Capas de la nube.

Fuente: (Ávila Mejía, Computación en la nube, 2011)

En la Figura 14, se analiza los riesgos y beneficios de la nube.

Riesgos y beneficios de la nube



Figura 14. Fortalezas y debilidades de una infraestructura en la nube.

Elaborado por: Los autores.

1.4.3. Plataformas de servicios en la nube

Entre ciertas plataformas funcionales en la nube se tiene:

Google Apps

Se realizan aplicaciones en los mismos sistemas que funcionan las aplicaciones de Google (ManagementSolutions, 2012) las que proporcionan un sin número de herramientas para el manejo de alias de internet, así como para la administración de funciones adicionales.

Amazon EC2

Es un servicio web que proporciona una gran capacidad computacional con el objetivo de brindar tecnologías de virtualización, lo cual permite la utilización de varios sistemas operativos a través de servicios web. Brinda varios servicios como almacenamiento, base de datos, migración, redes, mensajería, desarrollo de juegos, etc.

1.4.4. Firebase

Es una aplicación web desarrollada por Firebase Inc. en el año 2011 y tres años más tarde fue adquirida por Google. Se considera como plataforma de aplicación web la cual ayuda a los desarrolladores a la construcción de aplicaciones de buena calidad.

La información es almacenada en formato de notación de objetos JavaScript JSON⁶ la cual no utiliza consultas para insertar, actualizar, eliminar o agregar datos.

El backend de un sistema se utiliza como base de datos para almacenar información (Khawas & Shah, 2018).

Entre los servicios que ofrece Firebase se encuentran los siguientes:

⁶ JSON: JavaScript Object Notation

- **Firestore Analytics:** Brinda información correspondiente a la aplicación, permite la participación activa de usuarios, y permite al desarrollador comprender como los usuarios utilizan el servicio (Khawas & Shah, 2018).
- **Firestore Cloud Messaging:** Se basa en un servicio de pago multiplataforma para mensajes y notificaciones para Android, aplicaciones web e IOS (Khawas & Shah, 2018).
- **Firestore Auth:** Los desarrolladores administran mediante correo electrónico y contraseña el inicio de sesión almacenada en Firestore para así autenticar a los usuarios (Khawas & Shah, 2018).
- **Firestore Real-Time Database:** Brinda acceso a la DataBase para proporcionar un API que será de utilidad para enlazar los datos de los usuarios y almacenarla en la sección correspondiente a Firestore (Khawas & Shah, 2018).
- **Firestore Storage:** Este servicio permite la transferencia de archivos de manera fácil y segura, independientemente de la calidad de las aplicaciones de Firestore, el servicio de almacenamiento de datos está respaldado por Google Cloud (Khawas & Shah, 2018).
- **Firestore Test Lab for Android:** Servicio de pruebas Firestore como capturas de pantalla. Los videos y registros están disponibles en la consola de Firestore ya que proporciona una infraestructura basada en la nube para probar las aplicaciones (Khawas & Shah, 2018).
- **Firestore Crash Reportinf:** Servicio donde se detallan los errores que se crean dentro de la aplicación y corregirlos a tiempo para su correcto funcionamiento en la plataforma (Khawas & Shah, 2018).
- **Firestore Notifications:** Servicio que envía notificaciones de usuario para las aplicaciones móviles (Khawas & Shah, 2018).

1.5. Aplicaciones de monitoreo y gestión

1.5.1. Aplicaciones Móviles

En los últimos años los dispositivos móviles se han convertido en unas herramientas indispensables para el día a día, ya que se cuenta con más demandas de líneas telefónicas contratadas que el número de usuarios activos.

La evolución no solo se da por el aumento de usuarios, pues los terminales también evolucionan y en la actualidad la mayoría cuentan con acceso a internet, esto hace posible que aumente el uso de aplicaciones móviles, ya sea para compras en línea o para pedir comida a domicilio (Fernández R. F., 2020).

Un dispositivo móvil se adapta a estas tres funcionalidades:

- **Terminal:** Se implementa el hardware por el fabricante, un ejemplo es cómo captan la señal las antenas.
- **Sistema Operativo:** Entran los servicios básicos que son utilizadas por el usuario, un ejemplo es como convertir la intensidad de la señal en un dato numérico.
- **Aplicaciones:** Funcionalidad adicional en donde se encuentra todo lo que el dispositivo móvil ofrece como tal (Fernández R. F., 2020).

Cada capa construye sobre los servicios de las anteriores y proporciona una visión uniforme. La funcionalidad básica es limitada, donde se incluye el terminal más el sistema operativo, las aplicaciones son descargadas en los terminales por los usuarios. Se tiene dos tipos de sistemas operativos, Android con casi el 80 % y el sistema de IOS con el 20 % en el mercado (Casas, 2019).

1.5.2. Android

Los dispositivos móviles al igual que el computador personal necesitan de un sistema operativo para ofrecer funcionalidades a nivel de software. Es así como surge la necesidad de desarrollar por parte de la corporación Android Inc el sistema operativo de Android en el año 2005 con la característica de ser promocionado como software libre y tres años más tarde de sus inicios ya posicionarse ampliamente en el mercado (Báez , y otros, 2019, pág. 1).

Android es la plataforma más popular en todo el mundo para el uso de aplicaciones móviles, pues la mayoría de los usuarios cuentan con este sistema, el cual se basa en el kernel de Linux (UA MASTER MOVILES, 2012).

En la Figura 15, se detallan las funcionalidades que brinda el sistema operativo Android.

Características de Android	
Características	Android
Código	Abierto.
Núcleo	Kernel de Linux.
Adaptabilidad	Varias pantallas y resoluciones.
Soporte	HTML, HTML5, Adobe Flash Player.
Funcionalidad	Multitarea real de aplicaciones.
Soporta	Emulador de dispositivo.

Figura 15. Características de Android.
Elaborado por: Los autores.

1.5.3. Arquitectura de Android

En la Figura 16, se muestra el funcionamiento de la arquitectura de Android y sus partes más características.

Esquema de arquitectura de Android

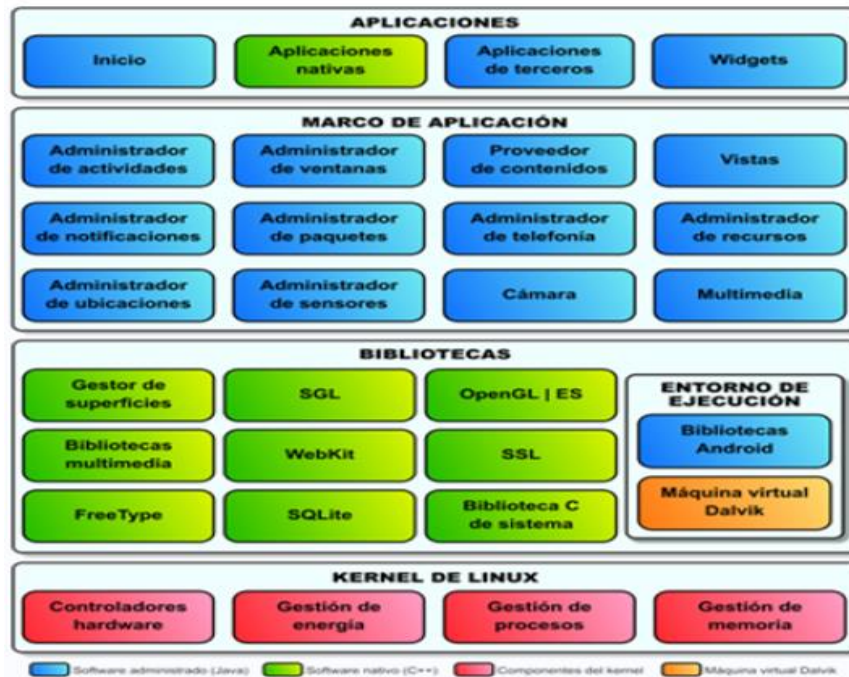


Figura 16. Arquitectura de Android.

Fuente: (Molina Riviera, Sandoval Cardona, & Toledo Franco, 2012, pág. 39)

Núcleo Linux: Toma como referencia a Linux y así desempeña distintas tareas como proporcionar memoria, procedimientos y seguridad. El núcleo también realiza la conceptualización entre el hardware, así como del software, alcanzando la mayoría de los elementos sin la necesidad de conocer el modelo del dispositivo (Ávila Mejía, Android, 2012, pág. 45).

Bibliotecas: Dentro de esta capa se cuenta con un grupo de bibliotecas de C/C++ utilizadas por los desarrolladores, ya que ofrecen utilidades atractivas y una serie de escenarios por descubrir (Ávila Mejía, Android, 2012, pág. 45).

CAPÍTULO II

ANÁLISIS DE COMPONENTES

En el presente capítulo, se realizará la investigación de los componentes para el prototipo, además del análisis técnico para la elección de los materiales y las herramientas necesarias para el diseño, además su funcionamiento tanto en hardware como en software, en base a la matriz de comparación para la mejor elección de componentes, asimismo se implementará la factibilidad técnica, económica y legal.

2.1. Análisis de materiales

2.1.1. Microcontroladores

2.1.1.1. *Arduino Nano*

Hace referencia a una plataforma open-source basada en un IDE de programación denominado Arduino IDE y que funciona bajo código C++, en el mercado se encuentran diferentes opciones con similares características a las que ofrece Arduino, sin embargo, no todas cuentan con una comunidad que se encuentra activa compartiendo nuevas ideas y proyectos para ser mejorados. Cuenta con la ventaja de ser un entorno de programación multiplataforma que funciona en sistemas como Windows, Mac OS y Linux, esto no ocurre con el software de otras placas de similares características (Abrajan Arias, 2020). En Figura 17, se visualiza la placa de Arduino Nano.

Arduino Nano



Figura 17. Arduino Nano.
Fuente: (Abrajan Arias, 2020, pág. 21)

2.1.1.2. Raspberry Pi

Con esta plataforma se revolucionó el mercado de los microprocesadores debido a que Raspberry Pi evolucionó a otro nivel. Diseñada para convertirse en lo que se denominaría un minicomputador. Esta plataforma utiliza lenguajes de alto nivel como C++, Python y Java. Tiene la capacidad de funcionar como un computador convencional, ya que es capaz de arrancar un sistema operativo sin ningún problema. Posee un puerto Ethernet y puertos dedicados para un monitor externo y periféricos como teclado, mouse, etc (Casco, 2014, pág. 3). En la Figura 18, se contempla una placa de Raspberry Pi.

Placa de Raspberry Pi



Figura 18. Placa de Raspberry Pi.
Fuente: (Chávez Castrillón & Yuquilema León, 2013, pág. 21)

2.1.1.3. BeagleBone Black

Se trata de una plataforma dedicada a desarrolladores poco expertos que requieren de este tipo de placas para sus proyectos, ya que posee la ventaja de ser tan potente como la Raspberry Pi, pero al mismo tiempo ofrece una opción de interfaz externa similar al Arduino. Cuenta con una memoria flash interna y no posee la opción de añadir una pantalla, ni tampoco añade tantos puertos USB como la Raspberry Pi (Casco, 2014, pág. 4). Esta placa es una opción para perfeccionar proyectos avanzados que posteriormente puedan ser comercializados. En la Figura 19, se muestra la placa de BeagleBone.

Placa de BeagleBone

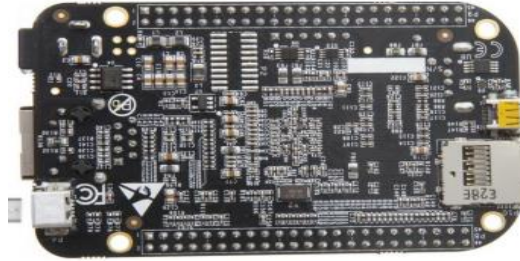


Figura 19. BeagleBone Black.
Fuente: (Coley, 2014, pág. 19)

2.1.1.4. Módulo ESP3266 NODE MCU

Esta placa tiene integrado un módulo Wi-Fi ESP8266 que utiliza el protocolo de comunicación TCP/IP y es capaz de permitir el acceso a cualquier microcontrolador (Ayala Delgado & Lemos Londoño, 2019, pág. 45).

Un inconveniente a la hora de optar por este módulo es que debido a que funciona vía Wi-Fi su utilización se reduce a espacios en los que exista un determinado número de puntos de acceso, esto en espacios abiertos no siempre es favorable debido a que las redes públicas de acceso inalámbrico no siempre cuentan con una buena intensidad de señal y no sería posible obtener la localización de la mascota. En la Figura 20, se considera el módulo ESP8266.

Módulo ESP8266



Figura 20. Módulo Node MCU ESP8266.
Fuente: (Ayala Delgado & Lemos Londoño, 2019)

2.1.1.5. Análisis de los componentes

En la Figura 21, se examinan las principales características de los componentes analizados.

Descripción de los componentes

	Arduino Nano	Raspberry Pi	BeagleBone Black	Node MCU ESP8266
Microcontrolador	ATMega 328	ARM11	ARM Cortex-A8	ESP8266
Voltaje de operación	5V	5V	5V	3.3V
Voltaje de alimentación recomendado	7-12V	5V	5V	3.3V
I/O digitales	14 (6 son PWM)	8	66	13
Entradas analógicas	8	N/A	7-12 bit	1
Memoria flash	32KB	(SD Card)	4GB (microSD)	4MB
Frecuencia de trabajo	16MHz	700MHz	700MHz	80MHz
Dimensiones (pulgadas)	0,73'' x 1,70''	3.37'' x 2.125''	3.4'' x 2.1''	0,82'' x 1,85
Ethernet	N/A	10/100	10/100	N/A
USB Master	N/A	2 USB 2.0	1 USB 2.0	N/A
Salida de Video	N/A	HDMI, Compuesto	N/A	N/A
Salida de Audio	N/A	HDMI, Analógico	Analógico	N/A
Sistema Operativo	N/A	Linux	Android, Linux, Windows, etc	N/A

Figura 21. Descripción de los componentes.
Elaborado por: Los autores.

En la Tabla 1, se evalúa la matriz de comparación de los microcontroladores.

Tabla 1. Matriz de comparación de microcontroladores

	Arduino Nano	Raspberry Pi	BeagleBone Black	Node MCU ESP8266
Adaptación a espacios exteriores	✓	X	X	✓
Ausencia de un sistema operativo	✓	X	X	✓
Dimensiones muy reducidas	✓	X	X	X
Contar con puertos de conexión	✓	✓	✓	X
Ethernet	X	✓	✓	X

Acoplamiento a un protoboard pequeño	✓	✓	✓	✓
Bajo costo	✓	X	X	✓

Nota: En la presente tabla se puede observar un análisis de los componentes descritos anteriormente.
Elaborado por: Los autores.

2.2. Tipos de tecnologías aplicadas para geolocalización

Existe una gran variedad de tecnologías aplicables para trabajos de geolocalización en la actualidad, sin embargo, el presente trabajo estará orientado sobre todo a zonas exteriores como parques, plazas, calles, y otros lugares donde exista mayor incidencia de pérdida o extravió de mascotas. Al tratarse de un prototipo para animales de compañía se opta por escoger hardware de dimensiones reducidas, que se adapten a la mascota tratando de ocasionarle la menor incomodidad a la misma.

2.2.1. Módulo GPS

Este tipo de módulos cuentan con la capacidad de rastrear cualquier tipo de objetos en la tierra que cuenten con un sistema GPS instalado, ya que para encontrar un punto determinado utiliza la triangulación de tres satélites que emiten señales que son captadas por el receptor en un punto en la tierra hasta encontrar la ubicación del objeto, tiene un margen de error que oscila entre los 3 a 15 metros pero este dato puede variar según el tipo y modelo de GPS (Alambiaga Pascual, 2017, pág. 5). En la Figura 22, se observa el módulo GPS.

Módulo GPS



Figura 22. Módulo GPS.
Fuente: (Benitez Silva, 2018, pág. 10)

Beneficios

- Cuenta con una amplia cobertura debido al gran número de antenas móviles distribuidas en el país.
- Es capaz de soportar el envío de gran cantidad de datos, lo que beneficia para la recepción de múltiples coordenadas simultáneamente (MAPIT, 2017).

Inconvenientes

- Coste mensual de planes considerables.
- La cobertura internacional dependerá de la operadora móvil.
- Consumo de batería alto (MAPIT, 2017).

2.2.2. Redes LPWAN

La implementación de IoT (Internet Of Things) es el siguiente salto de la tecnología para que las personas tengan un mayor control de sus actividades cotidianas y con una aplicación móvil puedan encender o apagar las luces de su casa, preparar la lista de la compra, preparar un café o incluso localizar un objeto en movimiento (Quintero Martínez & Rodríguez Guerrero, 2004).

Características importantes de LPWAN⁷

Repercusión geográfica: Está pensado para el transporte inalámbrico de datos entre dispositivos y el rango de alcance está comprendido en kilómetros.

La afluencia de los datos transmitidos: Este tipo de redes se encargan de medir y calibrar un flujo constante pero pequeño de datos, lo que se convierte en una gran debilidad.

⁷ LPWAN: Low Power Wide Area Network

Bajo consumo eléctrico: LPWAN ofrece una larga duración de sus baterías (PANDORAFMS, s.f.).

En la Figura 23, se considera el módulo ESP8266 PYCOM.

Módulo ESP8266 PYCOM



Figura 23. Placa ESP8266 con Sigfox.
Fuente: (Programarfacil.com, 2020)

Beneficios

- Posee una amplia cobertura.
- Reducido coste anual.
- Consumo de batería reducido (MAPIT, 2017).

Inconvenientes

- Cuenta con un menor ancho de banda, pues envía menos cantidad de datos.
- La monitorización en tiempo real de objetos en movimiento resulta complicada.
- La localización en interiores es menos precisa (MAPIT, 2017).

2.2.3. Bluetooth

Este tipo de tecnología surge por la necesidad de obtener una comunicación totalmente inalámbrica.

Utiliza un pequeño microchip que tiene un radio transmisor encargado de vincular las conexiones de corto alcance.

Como cualquier otro tipo de tecnología de comunicación inalámbrica, Bluetooth tiene un rango de alcance que se encuentra entre los 10-100 metros y una capacidad de transmisión de voz y datos de hasta 720 kbps por canal (Lorefice Sparacino, 2003, pág. 39).

- **Transmisión de voz:** Dispone de tres canales síncronos de voz a la vez, que soportan hasta 64 Kbps.
- **Transmisión de datos:** Utiliza un canal asíncrono para transmitir los datos y es capaz de soportar hasta 723.2 Kbps (Lorefice Sparacino, 2003, pág. 39).

Beneficios

- Bajos costes de infraestructura.
- Tamaño de infraestructura pequeño.
- Transmisiones de voz y datos seguras.
- Permite conexiones entre dispositivos sin usar cables (MAPIT, 2017).

Inconvenientes

- Localización en interiores menos precisa.
- Cuenta únicamente con un alcance de 10 metros para la localización de personas (MAPIT, 2017).

Como se observa en la Tabla 2, el módulo GPS a pesar de tener un menor alcance a comparación con LPWAN, para el prototipo se precisa de mayor ancho de banda para el envío de información a la nube, obteniendo un módulo de dimensiones reducidas a un bajo costo.

Tabla 2. Matriz de comparación de geolocalización

	GPS	LPWAN	BLUETOOTH
Alcance	5-10 m	> 1000 m	10-100 m
Cobertura	Amplia	Amplia	Reducida
Consumo de batería	Alto	Bajo	Bajo
Envío de datos	Mayor ancho de banda	Menor ancho de banda	N/A
Localización en interiores	Media	Baja	Baja
Costo	\$14.99	\$99	\$17.99

Nota: Comparación de los tipos de tecnología aplicadas para geolocalización.
Elaborado por: Los autores.

2.3. Medios de comunicación

2.3.1. Wi-Fi

Una de las principales características destacables de Wi-Fi, es su fácil implementación y la no dependencia de antenas móviles o satélites, solo es necesario el reparto de diversos puntos de accesos inalámbricos para disponer de conexión.

Dentro de los usos comunes de Wi-Fi esta la localización de objetos o dispositivos, sobre todo en interiores (Gómez Ruiz, 2015, pág. 3). En espacios cerrados no cuenta con una precisión totalmente acertada, pues existen varios obstáculos que impiden una comunicación directa, debido a impedimentos físicos como paredes no permite tener una línea de vista directa y esto perjudica la calidad de la señal. En la Figura 24, se contempla un Router que hace las veces de punto de acceso.

Router



Figura 24. Punto de acceso (Router).
Fuente: (TP-LINK, 2020)

Beneficios

- Permite una movilidad amplia dentro del rango establecido.
- El tiempo de obtención de respuesta es menor a 1 minuto (PANDORAFMS, s.f.).

Inconvenientes

- En espacios abiertos no tiene una buena precisión debido a que es necesario la presencia de varios puntos de acceso a nivel externo.
- Cuenta con un menor ancho de banda puesto que no utiliza cables para mayor ancho de banda (PANDORAFMS, s.f.).

2.3.2. Redes móviles GSM

La tecnología GSM brinda una serie de beneficios que son favorables para utilizarlo en el prototipo de la mascota, ya que el sistema GSM utiliza las antenas de telefonía móvil para realizar una triangulación del dispositivo que resulta ser más efectiva que la del GPS, solo necesita cobertura y antenas a los alrededores con las que se pueda triangular la posición del objeto (Quintero Martínez & Rodríguez Guerrero, 2004, pág. 25). En la Figura 25, se observa el módulo GSM.

Módulo GSM

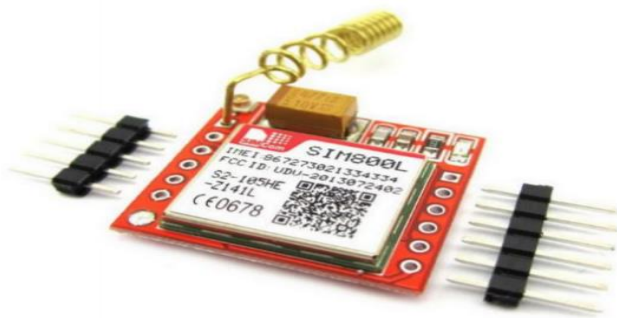


Figura 25. Módulo GSM SIM800L.
Fuente: (Leiva, 2017, pág. 8)

Beneficios

- Uso de datos.
- Mayor velocidad de transmisión.
- Cobertura extensa (Gavilanes Bayas, 2018).

Inconvenientes

- Bajas bandas de transmisión.
- Interferencia con dispositivos eléctricos.
- Dependiendo la operadora, cargos por roaming (Gavilanes Bayas, 2018).

2.3.3. ZigBee

Se analiza una tecnología inalámbrica que posee un descenso en la tasa de transferencia y disipado de potencia. La función principal de ZigBee es la de crear una red jerárquica a través de una topología que permita la conectividad de múltiples dispositivos siguiendo el protocolo de la IEEE 802.15.4 y ZigBee Alliance. Está pensado para las comunicaciones en redes de área personal (Vera Romero, Barbosa Jaimes, & Pabón González, 2017, pág. 238). En la Figura 26, se visualiza el módulo XBee ZigBee.

Módulo XBee ZigBee



Figura 26. Módulo XBeePRO de ZigBee.
Fuente: (Barneda Faudot, 2008)

Beneficios

- Se puede aplicar en usos para la domótica.
- Es adaptable para el uso de sensores médicos (Barneda Faudot, 2008, pág. 9).

Inconvenientes

- Baja transmisión y recepción de datos.
- Velocidad de transmisión de datos de hasta 250 Kbps (Barneda Faudot, 2008, pág. 14).

Como se observa en la Figura 27, el medio de comunicación más acorde para este proyecto es el GSM, ya que tiene un alcance amplio entre celdas y esto permite un alcance aceptable en caso de que la mascota recorra una distancia considerable, a diferencia de Wi-Fi y ZigBee que cuentan con un alcance corto, así como una velocidad de transmisión de datos menor que GSM.

Matriz de comparación de medios de comunicación

	GSM	Wi-Fi	ZigBee
Estándar	GSM-GPRS	IEEE 802.11	IEEE 802.15.4
Alcance	30 km por celda	10-300 m	100-300 m
Banda de transmisión	800 MHz 1800 MHz 1900 MHz	2.4 GHz 5 GHz	2.4 GHz
Velocidad de transmisión de datos	100 Mbps	54 Mbps	250 kbps
Costo	\$19.99	\$49.99	\$41.99

Figura 27. Matriz de comparación de los estándares de comunicación.
Elaborado por: Los autores.

2.3.4. Tipo de conexiones a las redes móviles

Una vez escogidos los componentes y teniendo claro cómo se realizará la transmisión de la información, es necesario disponer de una tarjeta SIM que requiere el módulo GSM/GPRS para obtener conectividad gracias a los datos móviles.

En Ecuador se cuenta con la ARCOTEL⁸, ente que regula las empresas que ofrecen un servicio de red de telefonía en el país tales como: Movistar legalmente registrada como Otecel S.A, Claro legalmente registrada como Conecel y CNT EP empresa estatal ecuatoriana de telecomunicaciones (Gavilanes Bayas, 2018, pág. 38).

Mediante la información facilitada en la Figura 28, se observa que la operadora móvil de CONECEL S.A perteneciente a Claro cuenta con mayor cobertura a nivel nacional debido a que tiene una multitud de estaciones base propagadas alrededor de todo el país. Al contar con una buena cobertura, la mascota estará mejor controlada evitando perder la señal constantemente.

Matriz de comparación de redes móviles

Operadoras	Tecnología	Frecuencia	Red	Servicio
Claro (CONECEL S.A)	GSM	850 / 1900 MHz	2G-3G	Voz y datos
Movistar (OTECCEL S.A)	GSM	850 / 1900 MHz	2G-3G	Voz y datos
CNT EP	GSM	1900 MHz	2G-3G	Voz y datos

Figura 28. Matriz de comparación de redes móviles.
Elaborado por: Los autores.

2.4. Buzzer

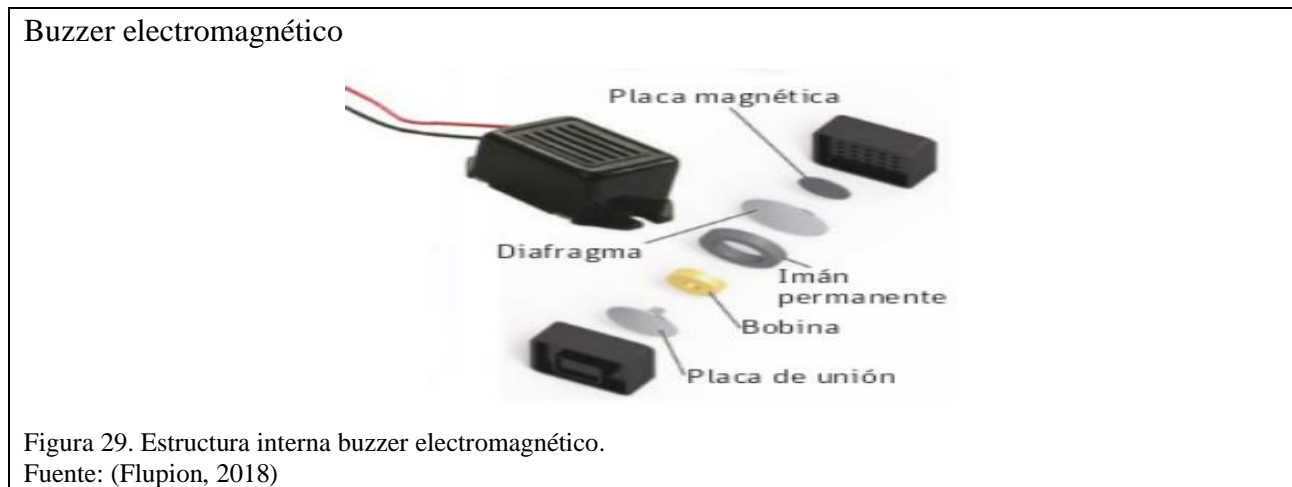
Los buzzer o mejor conocidos como zumbadores, son pequeños transductores que tienen la capacidad de transformar la energía eléctrica en sonido. Para su correcto funcionamiento se conecta el polo positivo del buzzer con el positivo del módulo y el polo negativo, con la tierra del módulo del zumbador (Ingeniería Mecafenix, 2018).

⁸ Se trata de la Agencia de Regulación y Control de las Telecomunicaciones. En Ecuador entidad encargada de la administración, regulación y control del espectro radioeléctrico y su gestión (GOBIERNO DE LA REPÚBLICA DEL ECUADOR, 2020).

2.4.1. Tipos de Buzzer

En el mercado se encuentran dos tipos de buzzer dependiendo el tipo de uso y la intensidad de sonido requeridos. Se clasifican en electromagnéticos y piezoeléctricos:

- **Electromagnéticos:** En este tipo de buzzer se cuenta con una bobina por la que circula una corriente que hace que vibre una chapa de acero que emite un sonido particular que llega a alcanzar una frecuencia de 2000 Hz (Flupion, 2018). En la Figura 29, se visualiza la estructura interna del buzzer electromagnético.



- **Piezoeléctricos:** Los buzzer piezoeléctricos al contrario que los electromagnéticos cuentan con dimensiones reducidas ya que se conforman de un componente electrónico que combina dos discos, uno de ellos metálico y otro de cerámica, contando ambos con propiedades piezoeléctricas. El funcionamiento interno de este tipo de buzzer se percibe cuando se aplica voltaje al buzzer repeliendo los discos y se genera una especie de clic audible (Flupion, 2018). En la Figura 30, se observa el buzzer piezoeléctrico.

Buzzer piezoeléctrico

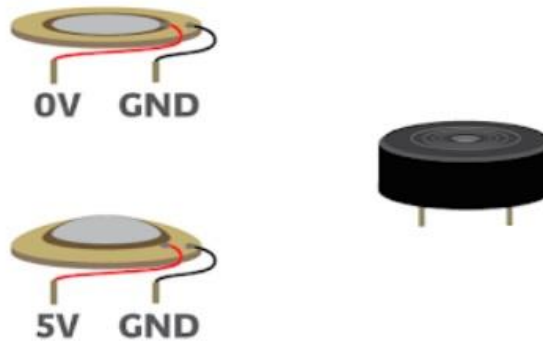


Figura 30. Estructura interna de un piezoeléctrico.

Fuente: (Flupion, 2018)

Dentro de los piezoeléctricos se identifican dos grupos:

- **Buzzer con oscilador:** Si al disco se le aplica un voltaje desde un circuito que cuente con un oscilante externo se le denomina buzzer piezoeléctrico (Flupion, 2018).
- **Buzzer sin oscilador:** Si cuenta con un oscilador interno propio incluido en la cubierta se le denomina zumbador piezoeléctrico (Flupion, 2018).

2.5. Frecuencia cardiaca

El sensor de frecuencia cardiaca servirá para obtener los valores de las pulsaciones cardiacas del animal, dependiendo del estado en que se encuentre, ya sea en reposo o después de haber realizado alguna actividad física.

Para ello se colocará el sensor en el prototipo estratégicamente para censar esos valores y cargarlos en Firebase y visualizarlos en el smartphone (AG Electrónica S.A, 2015). En la Figura 31, se contempla el sensor de frecuencia cardiaca.

Sensor frecuencia cardiaca



Figura 31. Sensor pulso cardiaco.
Fuente: (AFEL, 2020)

2.6. Entornos de desarrollo

2.6.1. Android Studio

Se define como medio multiplataforma fundamentado en IntelliJ IDE, que permite una fácil instalación para sistemas operativos como Windows, Linux y MacOS. Una importante mejora en esta plataforma es Gradle, que ayuda en la gestión y la automatización de la construcción de proyectos (Developers, 2020).

Beneficios

- Gracias a que posee su propio emulador permite codificar el programa en un periodo inmediato de tiempo.
- Ejecuta la aplicación desarrollada directamente en un teléfono móvil.
- Brinda la posibilidad de obtener una vista previa del proyecto en desarrollo.
- Android Studio permite la creación de diferentes versiones entre ellas de pago o gratuitas ofreciendo un mejor uso de los recursos (Android Studio FAQs, 2016).

Inconvenientes

- Para funcionar adecuadamente requiere de una batería en óptimas condiciones que soporte el rendimiento de la aplicación.
- Los requisitos necesarios para compilar Android Studio son altos, ya que se requiere de un buen procesador, espacio de almacenamiento y una memoria RAM de mínimo 8GB para no tener problemas de latencia en el sistema (Android Studio FAQs, 2016).

2.6.2. App Inventor

Este entorno de desarrollo surge de una idea conjunta del MIT⁹ y de un equipo de Google Education como una herramienta de desarrollo para entrar al mundo de la programación de una forma sencilla y para que estudiantes, así como personas ajenas a este mundo no tuvieran mayor inconveniente a la hora de programar su primera aplicación (fundaciónesplai, 2015, pág. 4).

Es un ambiente dedicado a la programación que sirve de gran ayuda ya que se fundamenta en bloques y resulta ser bastante interactivo (fundaciónesplai, 2015, pág. 4).

Beneficios

- Permite el desarrollo de aplicaciones incluso sin tener conocimientos previos de programación.
- Acceso constante si se cuenta con una conectividad a internet.
- Ofrece un emulador a través del cual se comprueba el avance del desarrollo del software (fundaciónesplai, 2015, pág. 8).

⁹ MIT: Massachusetts Institute of Technology

Inconvenientes

- App inventor no genera un código en Java, por lo que no es conveniente si se requiere de un desarrollo más a fondo de algún proyecto en un futuro.
- Tiene una gran limitante para programar aplicaciones que requieran de una mayor complejidad (fundaciónesplai, 2015, pág. 16).

2.6.3. Xamarin

Entorno para el desarrollo de aplicaciones en lenguaje C# compatible con sistemas Android, iOS y Windows Phone. Se obtiene el IDE y se instala en el computador o se añade en el Microsoft Visual Studio debido a la gran acogida con los programadores de .NET en la sociedad (Press, 2016).

Beneficios

- Xamarin permite que el código funcione en distintas plataformas utilizando el framework de .NET.
- No necesita mantenimiento, pues al ser multiplataforma se publican actualizaciones al código fuente constantemente (Enagora, 2017).

Inconvenientes

- Acceso limitado al uso de las bibliotecas.
- Tiene problemas de compatibilidad con las bibliotecas de entornos externos.
- No hay una comunidad grande de desarrolladores en comparación con otras comunidades nativas de Android.
- Las aplicaciones desarrolladas en Xamarin ocupan un tamaño mayor que las desarrolladas en otro tipo de plataformas (Enagora , 2017).

2.6.4. Comparación de los entornos de desarrollo

En la Tabla 3, se determinó qué tipo de entorno de desarrollo resulta más eficaz para el propósito del proyecto, optando por una plataforma que brinde más funcionalidades con un entorno propio de desarrollo y que resulte compatible con todos los sistemas que están posicionados en el mercado.

Tabla 3. Matriz de comparación de los entornos de desarrollo

	Android Studio	AppInventor	Xamarin
Compatibilidad multiplataforma	✓	✓	✓
Contiene emulador	✓	✓	X
Desarrollo de aplicaciones de gran complejidad	✓	X	X
Genera código en Java	✓	✓	X
Acceso a múltiples bibliotecas	✓	X	X

Nota: En esta tabla se encuentran las características más relevantes de los entornos de desarrollo.
Elaborado por: Los autores.

2.6.5. Arduino IDE

Para el control de diferentes módulos se utilizó líneas de código del Arduino IDE que son similares a las de C++, mismas que son usadas para modificar, compilar y enviar instrucciones a la placa de Arduino, de esta manera se logró una comunicación directa con el circuito.

2.7. Bases de datos

2.7.1. MySQL

Tiene un contexto basado en la asignación independiente y colaborativa de desarrollo de software aplicado en una base de datos relacional con múltiples funcionalidades (Sánchez, 2004).

Beneficios

- Facilita la construcción y la estructuración.

- No requiere de recursos excesivos para su ejecución.
- Soporta diversos sistemas operativos.
- Brinda un soporte de código abierto basado en licencia GPL (Pérez Hernandez, 2017).

Inconvenientes

- Gran porcentaje de las funciones de MYSQL no se encuentran bien documentadas.
- No resulta ser una herramienta tan intuitiva como otros tipos de software (Pérez Hernandez, 2017).

2.7.2. AWS (Amazon Web Services)

Es una plataforma de Amazon que ofrece más de un centenar de servicios en la nube tales como las tecnologías de infraestructura y un software para guardar datos en gran cantidad, adoptar un estudio automatizado junto al intelecto artificial. (AWS, 2020).

Beneficios

- Tiene un entorno más flexible y seguro, actualmente disponible en la nube.
- Posee un ritmo de innovación más rápido ya que se encuentra en constante evolución.
- Cuenta con una de las comunidades más grandes y activas con múltiples socios en todo el mundo (AWS, 2020).

Inconvenientes

- Es un servicio de pago por uso, pues dependerá de la cantidad de tráfico que reciba, si se maneja incorrectamente generará costos elevados no contemplados inicialmente.
- Se necesita de conocimientos previos para su correcto manejo (Demian, 2018).

2.7.3. MongoDB

Sistema de libre acceso donde se analiza la efectividad de los bancos de datos relacionales para almacenamiento de BigData, es por este motivo que MongoDB se creó como una base NoSQL basada en documentos (Fabián Alexis, 2020).

Una tabla en MongoDB se denomina colección y un registro es considerado como un documento. Los documentos se encuentran en formato JSON y MongoDB se encarga de convertirlos a formato BSON (Fabián Alexis, 2020).

Beneficios

- Base de datos de código abierto escrita en C++.
- Cuenta con una escala horizontal, fácil para agregar más servidores.
- Proporciona buen rendimiento con aplicaciones de alto tráfico (Viviana, 2018).

Inconvenientes

- No resulta ser conveniente para aplicaciones con transacciones complejas.
- Este tipo de base no cuenta con un reemplazo para soluciones de herencia (Viviana, 2018).

2.7.4. Firebase

Incluye servicios como textos, imágenes, API entre otras, Firebase incluye múltiples servicios que facilitan el escalado. No resulta tan complejo como el trabajo que realiza AWS y es más liviano, al ser una base NoSQL resulta conveniente para el almacenar una amplia gama de datos, adicionalmente una supervisión de la transferencia de datos (Fabián Alexis, 2020).

Los datos Firebase Realtime Database se almacenan como objetos JSON y se conceptualizan como un árbol que se aloja en la nube. En este tipo base de datos no hay tablas ni registros como en las SQL.

Beneficios

- Al fallar la conectividad la información se almacena en el cache local hasta volver a enlazarse a la red, una vez establece de nuevo la conexión los datos guardados en cache se sincronizan rápidamente.
- Es un repositorio que cuenta con la capacidad para almacenar y sincronizar datos al momento.
- Ofrece flexibilidad para visualizar la información en cualquier emplazamiento desde un dispositivo (Viviana, 2018).

Inconvenientes

- Al no ser un tipo de base convencional podría resultar en un problema con las empresas que manejan un modelo distinto al común.
- En su versión gratuita puede tener máximo hasta 100 conexiones simultáneas.
- No todas las funcionalidades que ofrece Firebase disponen de analíticas (Lazaro, 2017).

2.7.5. Comparación de Bases de Datos

En la Tabla 4, una vez tomadas en cuenta las fortalezas y debilidades de las posibles bases a utilizar, se observa que Firebase brinda mayores prestaciones que se acogen mejor a las condiciones del prototipo.

Tabla 4. Matriz de comparación de Bases de Datos

	MySQL	AWS	MongoDB	Firebase
Menor coste por servicios	X	X	✓	✓
NoSQL	X	✓	✓	✓
Datos en tiempo real	X	✓	X	✓
Guardar datos sin conexión	✓	✓	X	✓
Fácil interacción con el usuario	✓	X	✓	✓
API de Google	✓	X	X	✓

Nota: Se muestran características de las bases a considerar.

Elaborado por: Los autores.

2.8. Análisis de factibilidad

El siguiente análisis detallará la factibilidad del proyecto, para comprender si es viable para los autores el despliegue del prototipo, para ello se contemplarán tres factores importantes: la parte técnica, operacional y legal.

2.8.1. Factibilidad técnica

Al conocer los recursos tecnológicos que se utilizarán en el presente proyecto, en la Tabla 5, se detallarán los requerimientos necesarios para la elaboración del prototipo.

Tabla 5. Análisis de factibilidad técnica para el desarrollo del prototipo

Recurso	Nombre	Característica
IDE de desarrollo	Android Studio	Versión 4.0
	Arduino IDE	Versión 1.8.13
Base de Datos	Firebase	Realtime Database
Módulos para el Prototipo	Módulo GSM	Sim 808L
	Arduino	Nano
Sensor	Pulso Cardíaco	
	GPS	Neo-6M UBLOX
Fuente de Alimentación	Batería	7.4 V 300 mAh
Cargador	Batería	Módulo cargador
Placa	PCB	Circuito Impreso

Estación de Trabajo	Computador personal	Navegadores Web Actualizado
	Smartphones	Versión de Android 6.0

Nota: Características pertenecientes al desarrollo del hardware y software.
Elaborado por: Los autores.

2.8.2. Factibilidad operacional

Se busca una factibilidad donde se disfrute la rentabilidad concedida por el prototipo, y así ofrecer un producto funcional, para su efecto se determinaron los siguientes factores:

- La tecnología que se usará en el prototipo en la parte de hardware será de fácil implementación a la mascota, el resultado será un producto funcional donde se visualizarán la geolocalización y valores de la frecuencia cardiaca.
- La finalidad es que el usuario no tenga ningún problema al usar la aplicación móvil, por tal motivo se realizará una interfaz que sea amigable e intuitiva.

2.8.3. Factibilidad Económica

Se han considerado los costos relacionados con los sensores, módulos, recursos humanos, e insumos varios necesarios para su correcto desenvolvimiento, adicional aquello se realiza el cálculo del VAN y TIR para conocer si el prototipo es viable y factible. En la Tabla 6, se detalla la factibilidad económica de los recursos.

Tabla 6. Factibilidad económica de los recursos apropiados para el prototipo

Descripción	Cantidad	Costo Unitario	Costo Total
Materiales Tecnológicos			
Android Studio V.4.0	1	\$0	\$0
Arduino IDE V 1.8.13	1	\$0	\$0
Firebase	1	\$0	\$0
Sensores, Módulos del prototipo			
Módulo GSM 808	1	\$45	\$45
Módulo GPS	1	\$15	\$15
Sensor Frecuencia Cardiaca	1	\$5	\$5
Batería	1	\$25	\$25

Cargador de Batería	1	\$25	\$25
Placa PCB	1	\$60	\$60
Recursos Humanos			
Desarrollador App.	2	\$400	\$800
Gastos Varios			
Servicios Básicos	-	\$20 Mes	\$80
Total			\$1055

Nota: Coste aproximado de los recursos para el desarrollo del prototipo.
Elaborado por: Los autores.

2.8.3.1. Cálculo de VAN y TIR

VAN (Valor Actual Neto): El VAN hace referencia a las ganancias que obtiene un proyecto siempre y cuando se tome en cuenta el valor actual de la propuesta para conocer si es rentable o no.

$$VAN = \sum_{n=1}^n \frac{F_n}{(1+i)^n} - I_0$$

Ecuación 1. Fórmula VAN.

Parámetros:

- F_n : Flujo de efectivo de cada periodo.
- n : Periodo de tiempo.
- i : Tasa de interés o descuento.
- I_0 : Inversión Inicial.

Criterios de selección:

- $VAN > 0$: El proyecto es rentable porque generará beneficios.
- $VAN = 0$: El proyecto no generará ni beneficios ni pérdidas.
- $VAN < 0$: El proyecto no es rentable porque generará pérdidas (Mete, 2014, pág. 69).

TIR (Tasa Interna Retorno): El TIR define la aceptación de la propuesta demostrando las ganancias o las pérdidas según la estimación de los ingresos y los egresos.

$$TIR = \sum_{n=0}^n \frac{F_n}{(1+i)^n} - I_o = 0$$

Ecuación 2. Fórmula TIR.

Donde:

- F_n : Flujo de efectivo de cada periodo.
- n : Periodo de tiempo.
- TIR / i : Tasa de interés o descuento.
- I_o : Inversión Inicial.

Criterios de selección:

- $TIR > i$: El proyecto de inversión será aceptado.
- $TIR = i$: El proyecto no es ni aceptado ni rechazado.
- $TIR < i$: El proyecto debe rechazarse (Mete, 2014, pág. 73).

En la Tabla 7, una vez analizada la factibilidad económica y obtenido el valor total inicial se establece el cálculo del VAN y del TIR. Con la ayuda de las ecuaciones anteriormente descritas se realizaron los cálculos asumiendo los gastos establecidos para efectuar el desarrollo del prototipo, y es así como se obtuvo cálculo del VAN, con un interés del 12% y un valor inicial de \$1055 un resultado de \$1,815,13. A su vez, se obtuvo el cálculo del TIR una vez establecidos los valores de flujo de caja, con un valor inicial ya estipulado se obtuvo un 74%.

Tabla 7. Flujo de caja.

Flujo de ingresos		Flujo de egresos		Flujo de efectivo neto	
Año	Valor	Año	Valor	Año	Valor
1	1200	1	900	1	300
2	2100	2	550	2	1550
3	2500	3	580	3	1920
Total	5800	Total	2030	Total	3770

Nota: Se muestra el flujo de caja con los ingresos, egresos y el efectivo neto.

Elaborado por: Los autores.

Dado los resultados y de acuerdo con las características ya mencionadas se estipula que el proyecto es viable a futuro, pues se cumplió la condición de que el VAN sea mayor a cero y el TIR mayor a la tasa de interés convenida.

2.8.4. Factibilidad Legal

Para la factibilidad legal se tomó en cuenta los aspectos Jurídicos de las Aplicaciones Móviles sujeta a los derechos de la Ley de Propiedad Intelectual.

Se consideró el artículo 133 del Código Orgánico de Economía Social de Conocimientos, Creatividad e Innovación (Ver Anexo 1)

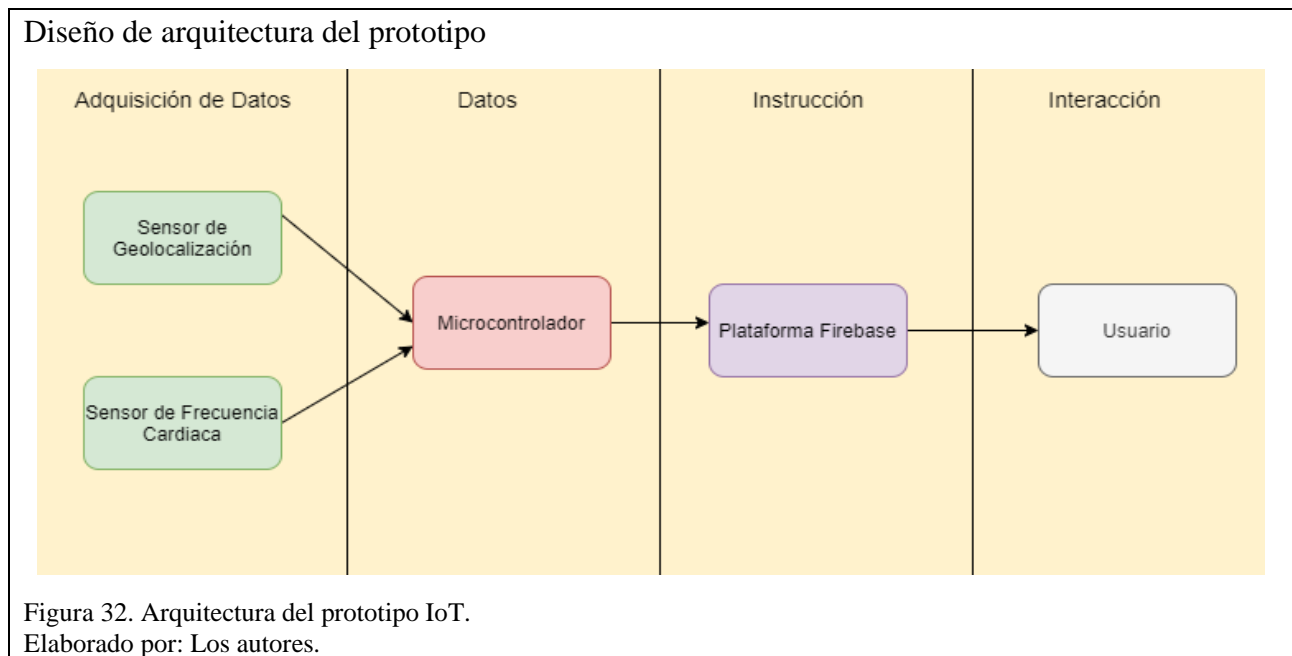
CAPÍTULO III

DISEÑO Y DESARROLLO

En el presente capítulo, se detallará el diseño y desarrollo del prototipo IoT, se visualizará mediante un diagrama de arquitectura en caso de hardware y diagramas UML en caso de software, de manera que el usuario interactúe con la aplicación acorde a los parámetros establecidos.

3.1. Arquitectura del Prototipo IoT

En la Figura 32, se detalla la arquitectura utilizada para el prototipo IoT el cual consta de cuatro bloques principales:



- **Adquisición de datos:** Función que realizan los sensores tanto de geolocalización y el sensor de la frecuencia cardíaca al enviar datos de la mascota en tiempo real.
- **Datos:** Los datos son generados y procesados por el Arduino nano que se convierte en el microcontrolador programable para la evaluación de los datos correspondientes.

- **Instrucción:** Se guardará la información recogida en Firebase y de esta forma tenerla siempre disponible para el libre acceso del usuario.
- **Interacción:** Permite la visualización de los datos por medio de la aplicación móvil, requiriendo un registro e inicio de sesión por parte del usuario, y así comenzar la localización de la mascota.

3.2. Construcción del prototipo IoT (hardware)

En la Figura 33, se tuvo en cuenta los componentes a utilizar para el armado del prototipo así como sus respectivas medidas para asignarles un lugar específico a cada uno con el fin de que todo el circuito quede en armonía y ningún componente produzca algún tipo de contacto o interfiera en el correcto funcionamiento del otro, obteniendo finalmente el modelo de 70 [mm] x 61,5 [mm], medidas ideales para realizar las pruebas con una mascota.

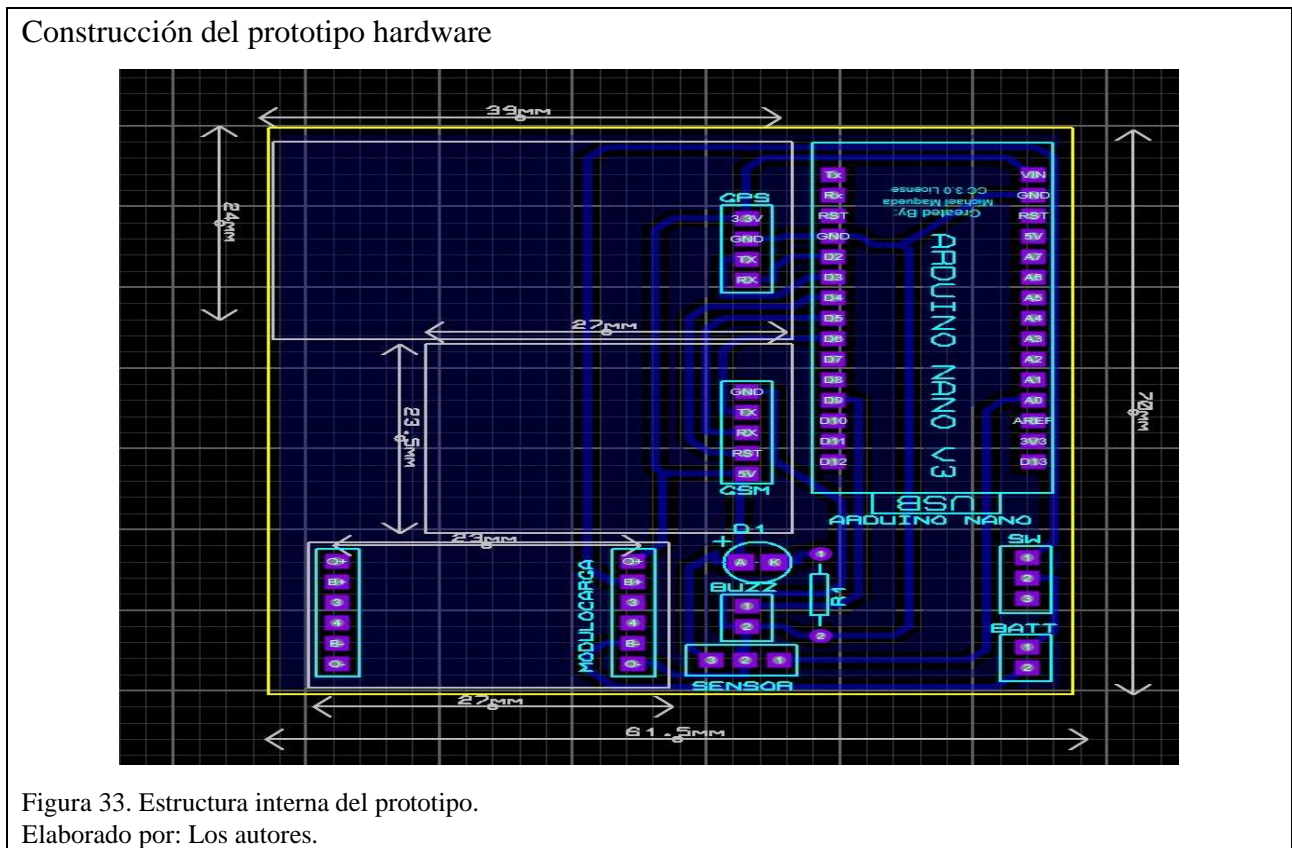


Figura 33. Estructura interna del prototipo.
Elaborado por: Los autores.

3.2.1. Elementos del prototipo

3.2.1.1. Arduino Nano Versión 3

Este es el microcontrolador del circuito, mismo que controlará las funciones de los sensores, del GPS, GSM/GPRS, el módulo de carga, la bocina y la batería.

3.2.1.2. Micromódulo Sim800L GSM/GPRS

Este módulo se utilizó como medio de comunicación para obtener la conexión a internet con la ayuda de una tarjeta SIM incorporada que se encarga de recibir la señal 2G para posteriormente enviar los datos captados por el GPRS y procesarlos por el microcontrolador.

3.2.1.3. Módulo GPS Ublox Neo 6m

Se utilizó el módulo GPS Neo 6m para realizar la triangulación entre los satélites públicos que tiene acceso el módulo, así obtener los datos de geoposicionamiento como latitud y longitud. Para que el módulo se conecte, el GSM inicia la búsqueda de red y una vez establezca la conexión el GPS procederá a obtener datos.

3.2.1.4. Placa PCB

Una vez armado el prototipo inicial y probado las conexiones y la correcta recolección de la información, además de la adaptación de un suministro de corriente para abastecer el circuito, se procedió a imprimir una placa PCB (Printed Circuit Board), es decir, placa de circuito impreso para conectar los caminos o pistas los componentes que conforman el circuito.

3.2.1.5. Batería Turnigy Nano-tech

El circuito necesita ser alimentado por una batería externa, en este caso se utilizó la Turnigy nano-tech de 350 mAh a 3.7V, recargable a través de un módulo de carga incorporado que le proveerá corriente.

3.2.1.6. Módulo de carga Tp4056

Este módulo de hace las veces de alimentador de corriente una vez la batería del prototipo llegue a su fin. Facilita la forma en que el usuario cargará el prototipo, ya que dispone de un conector tipo microUSB que se conecta con un cable de carga tipo B.

3.2.1.7. Sensor de pulso de frecuencia cardiaca

Con este sensor incorporado en el prototipo, se obtiene el valor de las pulsaciones para ser enviadas a Firebase para posteriormente mostrarlos en la aplicación móvil.

3.2.1.8. Diagrama de bloques de la conectividad de componentes

En la Figura 34, se muestra el diseño de conectividad.

Diseño de conectividad

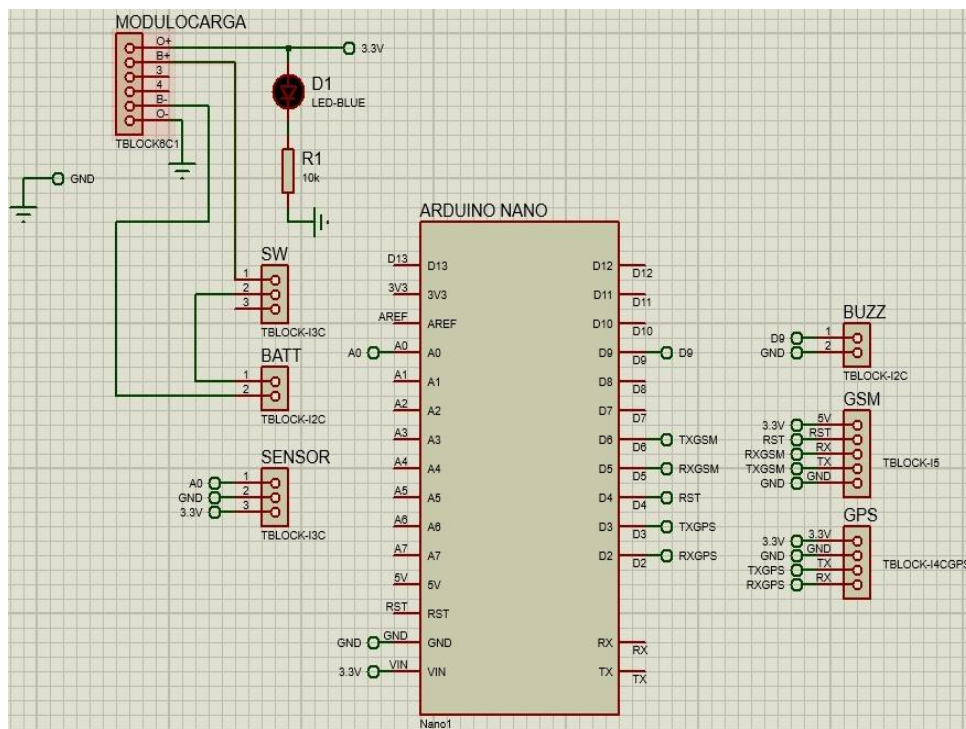


Figura 34. Diseño de conectividad.
Elaborado por: Los autores.

En la Figura 35, se detalla la conectividad de los diferentes componentes.

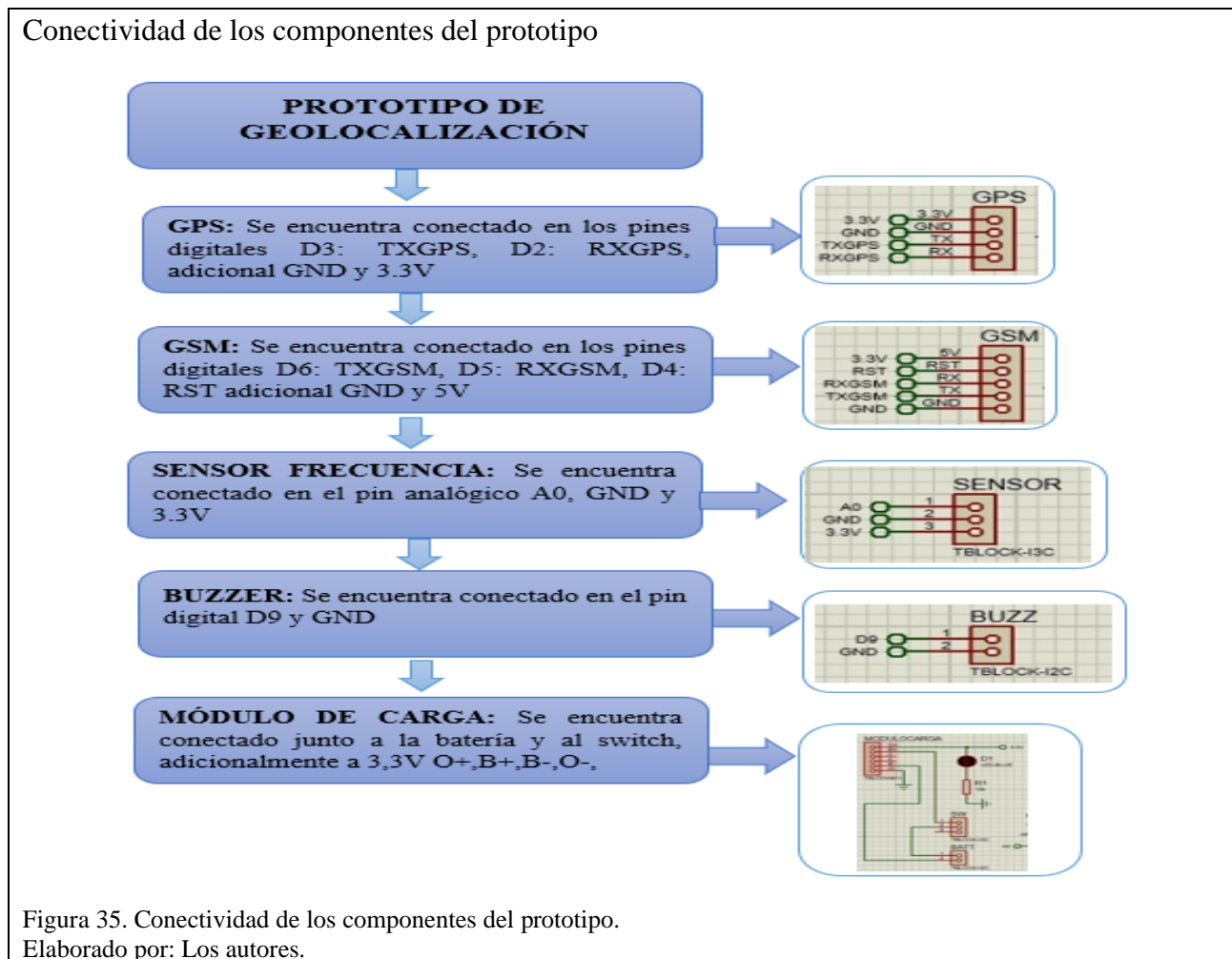


Figura 35. Conectividad de los componentes del prototipo.
Elaborado por: Los autores.

3.2.2. Diseño e implementación del prototipo

Para estructurar el prototipo se tuvo en cuenta los componentes utilizados, las medidas y su colocación dentro de un estuche para implementarlo en una pechera adaptada a la mascota.

En la Figura 36, se observan todos los componentes utilizados para la integración del prototipo final.

Componentes del prototipo

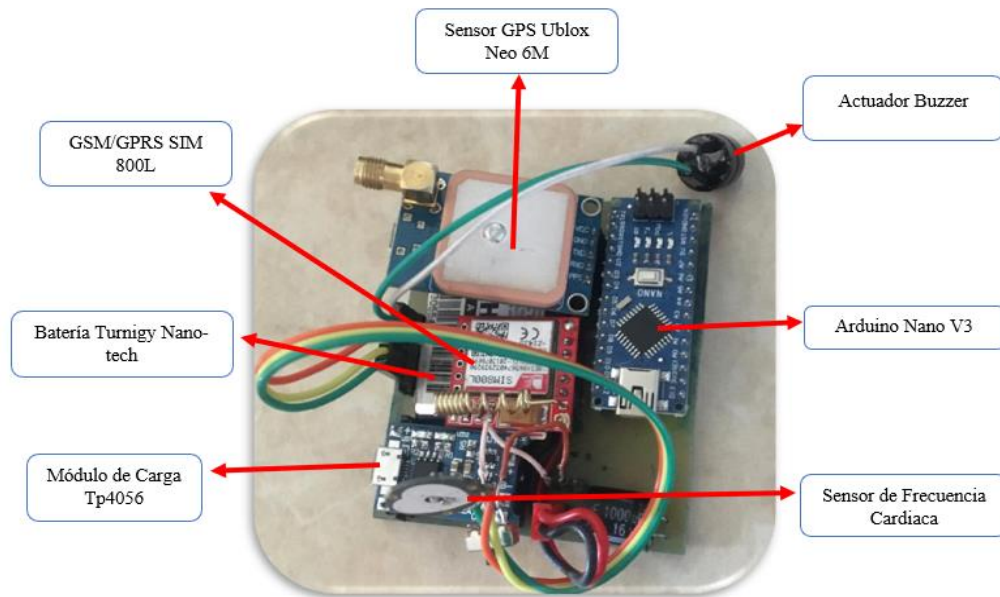


Figura 36. Piezas adaptadas en el prototipo.
Elaborado por: Los autores.

En la Figura 37, se examinan los componentes utilizados y ensamblados en una caja para tal fin.

Prototipo montado en placa PCB

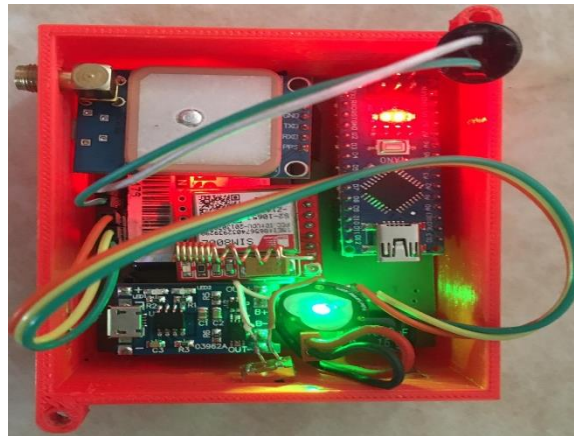
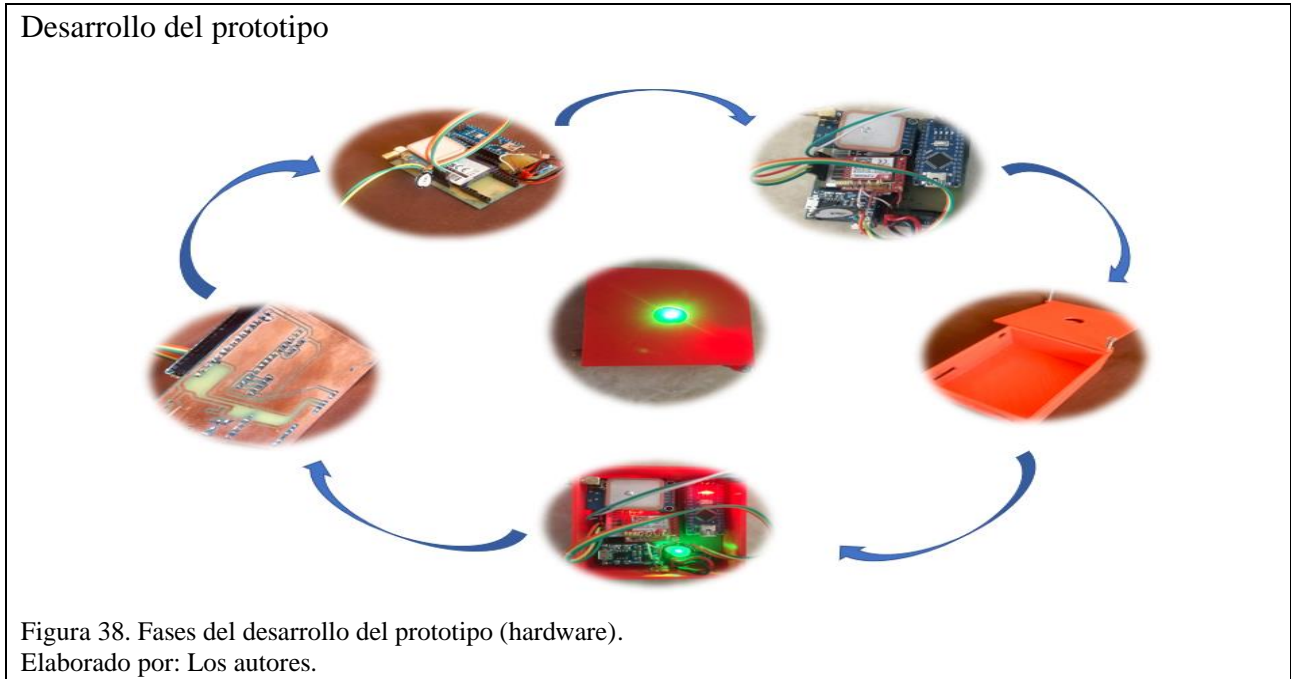


Figura 37. Prototipo montado en placa PCB.
Elaborado por: Los autores.

Las fases del proceso plasmadas en la Figura 38, corresponden al avance del prototipo, desde el inicio del circuito en placa PCB, hasta el montaje de las piezas en una caja de dimensiones reducidas para posteriormente probarla ya acoplada en la mascota.



3.2.2.1. Diagrama de flujo general

En la Tabla 8, se visualiza de manera general la estructura de la programación realizada del funcionamiento de los componentes más importantes del prototipo. Se comprueba que el GSM inicie correctamente y establezca una conexión satelital, misma que permitirá que los datos se almacenen sin generar ningún error.

Tabla 8. Proceso de conexión de sensores y módulos

<p>Diagrama de Flujo: Proceso de conexión de todos los sensores y módulos del prototipo IoT.</p>	<p>Proceso: Conexión</p>
<p>Descripción: El proceso de conexión de los sensores consiste en la adquisición de los datos del GPS en un espacio abierto para el prototipo. Si existe buena comunicación los datos serán correctos, y esa información se guardará en la plataforma de Firebase.</p>	<p>Subproceso: Adquisición de Datos</p> <p>Involucrados: Usuario, Prototipo IoT</p>
<pre> graph TD INICIO([INICIO]) --> GSM[Conexión GSM] GSM --> GPS[Toma de datos latitud y longitud GPS] GPS --> D{Latitud y Longitud distinta de 0} D -- No --> GSM D -- Sí --> S[Toma de señal GPS, verificación batería] S --> POST[Envío de datos usando método POST hacia Firebase] POST --> R[Retardo] R --> FIN([FIN]) </pre>	

Nota: Se muestran las fases para la adquisición de los valores del GPS.
 Elaborado por: Los autores.

3.2.3. Fases del desarrollo del prototipo

3.2.3.1. Fase del medio de comunicación

Para la etapa de comunicación dentro del circuito se estableció el módulo GSM/GPRS debido a que ofrece una comunicación UART (Universal Asynchronous Receiver-Transmitter) que es la

encargada de verificar que los puertos de las placas y los dispositivos seriales establezcan satisfactoriamente una conexión entre sí.

En la Figura 39, se muestra la inicialización de la tasa de transferencia para el módulo de GSM con una velocidad de 9600 baudios para la comunicación UART.

Tasa de transferencia para comunicación UART

```
GSM.begin(9600);
GSM.listen();
```

Figura 39. Tasa de transferencia de 9600 baudios.
Elaborado por: Los autores.

En la Figura 40, se visualizan los comandos AT utilizados por el módulo SIM800L que le permite vincularse a una red APN, para establecer el arranque de la base predeterminada alojada en Firebase mediante HTTP.

Comandos AT

Comandos AT	Función
AT+CSTT	Inicia una acción, Identifica un APN (proveedor de servicios) y el nombre.
AT+SAPBR	Comando portador para la configuración de aplicaciones basadas en internet.
AT+HTTINIT	Inicializa el servicio HTTP.
AT+HTTPSSL	Permite la activación o desactivación SSL (Secure Sockets Layer) para HTTP.
AT+HTTTPARA	Establece el valor de los parámetros HTTP.

Figura 40. Comandos AT.
Elaborado por: Los autores.

En la Figura 41, para establecer una conexión del módulo SIM800L con el GPS y enviar los datos, se necesita una respuesta a través de una solicitud para el envío de datos mediante HTTP.

Peticiones de envío con comandos AT

```
sendGSM("AT+CSTT=\"internet.claro.com.ec\", \"\", \"\");  
sendGSM("AT+SAPBR=3,1,\"APN\", \"claro\");  
sendGSM("AT+SAPBR=1,1", 3000);  
sendGSM("AT+HTTPIPINIT");  
sendGSM("AT+HTTTPSSL=1");  
sendGSM("AT+HTTTPARA=\"CID\",1");
```

Figura 41. Comandos de inicialización para conectarse a la red GPRS del APN.
Elaborado por: Los autores.

En la Figura 42, se utiliza una variable String para almacenar todos los valores de los sensores, para posteriormente ingresarlos en Firebase y mediante el comando AT+HTTTPARA=\"URL\", se realiza una solicitud HTTP cargando los valores.

Solicitud de HTTP cargando los valores

```
GSM.listen();  
String w = "AT+HTTTPARA=\"URL\", \"\"";  
w += FirebaseCloudFunctionURL;  
String total = w + longitud + lo + latitud + finalDeTrama;  
Serial.println(total);  
sendGSM("AT+HTTTPARA=\"URL\", \"https://us-central1-dogtrackerrufo.cloudfunctions.net/write?device=mascotas_activas/OMMwvGtEZ3N48NtgglPyjhAn9rt1.  
sendGSM(total);  
sendGSM("AT+HTTTPACTION=0");  
lastActionTime = now;  
actionState = AS WAITING FOR RESPONSE;
```

Figura 42. Solicitud HTTP utilizando los comandos AT.
Elaborado por: Los autores.

3.2.3.2. Fase de geolocalización

Para esta etapa al igual que en la anterior se estableció una comunicación UART entre el módulo GPS con el GSM, para que al momento de encontrar señal el GPS obtenga puntos de latitud y longitud junto al envío a la base de datos.

En la Figura 43, se utiliza las librerías TinyGPS y la SoftwareSerial para configurar los pines del dispositivo GPS para establecer la transmisión UART.

Librerías para la transmisión UART

```
#include <TinyGPS.h>
TinyGPS gps;
#include <SoftwareSerial.h>
SoftwareSerial GSM(10, 11); // RX, TX
SoftwareSerial ss(3, 2); //rx /tx
```

Figura 43. Librería para configurar pines para transmisión UART.
Elaborado por: Los autores.

En la Figura 44, se muestra los valores de latitud y longitud además los datos del sensor son almacenados en variables de tipo String.

Declaración de variables tipo String

```
String longitud = "-0.509492";
String latitud = "-78.571904";
String sensor = "-78.571904";
String lo = "&l=";
String finalDeTrama = "\"";
```

Figura 44. Variables tipo String para almacenar valores de latitud y longitud.
Elaborado por: Los autores.

En la Figura 45, se muestra la transformación de valores a formato String para posibilitar el envío de una solicitud HTTP a Firebase.

Conversión de valores

```
char buffer[5];
longitud = dtostrf(flat, 1, 6, buffer);
latitud = dtostrf(flon, 1, 6, buffer);
```

Figura 45. Conversión de valores de latitud y longitud.
Elaborado por: Los autores.

3.2.3.3. Fase de almacenamiento

Para almacenar los datos obtenidos del prototipo como información del usuario, valores de frecuencia cardiaca, puntos de latitud y longitud se utilizó la plataforma Firebase.

En la Figura 46, se muestra el enlace que redirige los valores obtenidos por el módulo de localización al servidor acompañado del nombre del proyecto creado en Firebase junto a la extensión de este para redirigirse al servidor de Cloud Function.

Enlace con solicitud en HTTP

```
String FirebaseCloudFunctionURL = "https://us-centrall-dogtrackerrufo.cloudfunctions.net/
```

Figura 46. Enlace con la solicitud HTTP.
Elaborado por: Los autores.

En la Figura 47, al enlace se incluye el nombre de la función que receptorá solicitudes en HTTP para ingresar datos en la extensión de Realtime Database.

Recepción de solicitudes en HTTP

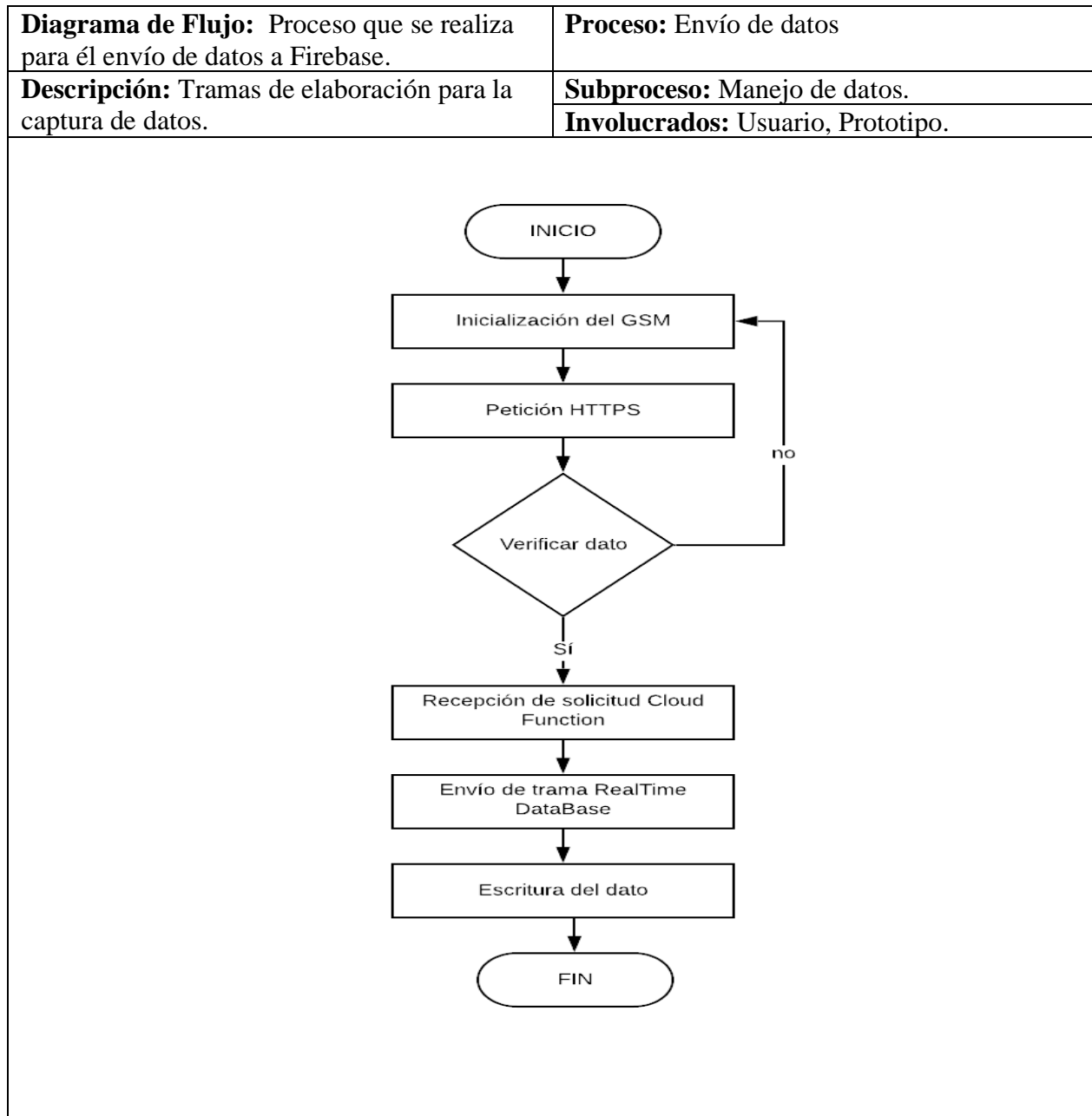
```
String FirebaseCloudFunctionURL = "https://us-centrall-dogtrackerrufo.cloudfunctions.net/write
```

Figura 47. Recepción de solicitudes en HTTP.
Elaborado por: Los autores.

3.2.3.3.1. Diagrama de flujo de Firebase

En la Tabla 9, se detallan los protocolos de conexión que se establecen a partir de las peticiones que envían los sensores para acceder a las funciones de Cloud Function.

Tabla 9. Proceso para el envío de datos Firebase



Nota: Diagrama de flujo que muestra el proceso del envío de datos a Firebase.
Elaborado por: Los autores.

3.2.3.4. Fase de Monitoreo

Para esta etapa se establece la conexión de los componentes del prototipo para obtener las coordenadas geográficas, así como la visualización de la frecuencia cardiaca de la mascota cuando se requieran dichos datos.

En la Figura 48, se inicializan las variables acordes a los pines asignados en el microcontrolador para obtener los valores del pulso cardíaco.

Inicialización de variables obtención del pulso

```
const int PULSE_INPUT = A0;
const int PULSE_BLINK = 13;
const int PULSE_FADE = 5;
const int THRESHOLD = 550;
```

Figura 48. Inicialización de variables obtención del pulso cardíaco.
Elaborado por: Los autores.

En la Figura 49, la función `getBeatsPerMinute()` refleja la cantidad de pulsaciones por minuto mediante la lectura del puerto analógico A0 del Arduino nano, a través de la comparación de un umbral definido por la variable `THRESHOLD` y el valor obtenido del sensor.

Función `getBeatsPerMinute()`

```
int myBPM = pulseSensor.getBeatsPerMinute();
```

Figura 49. Función `getBeatsPerMinute()`.
Elaborado por: Los autores.

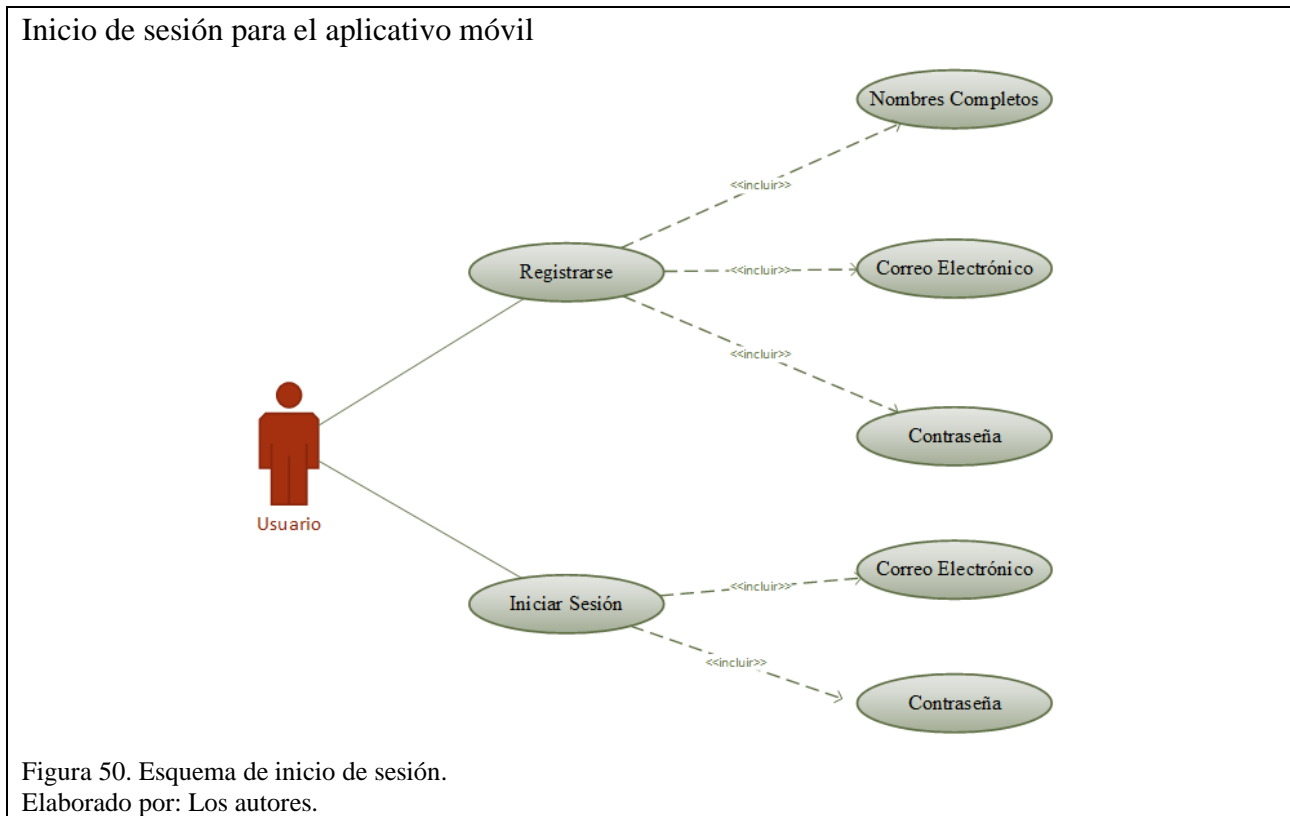
3.2.4. Diagramas UML

Lenguaje de Modelado Unificado, es un lenguaje de modelado que permite la realización de varias actividades como especificar, construir y documentar partes principales del funcionamiento del sistema de software, teniendo diferentes tipos de vistas del usuario, se implementan en cualquier momento del ciclo de vida.

3.2.5. Diagramas de casos de usos

El objetivo de un diagrama de caso de uso es la representación de la unidad funcional coherente de un sistema, subsistema o clase en la cual interactúan uno o más actores que se relacionan con las diferentes acciones que realiza el sistema.

En la Figura 50, se evalúa el diagrama de caso de uso.



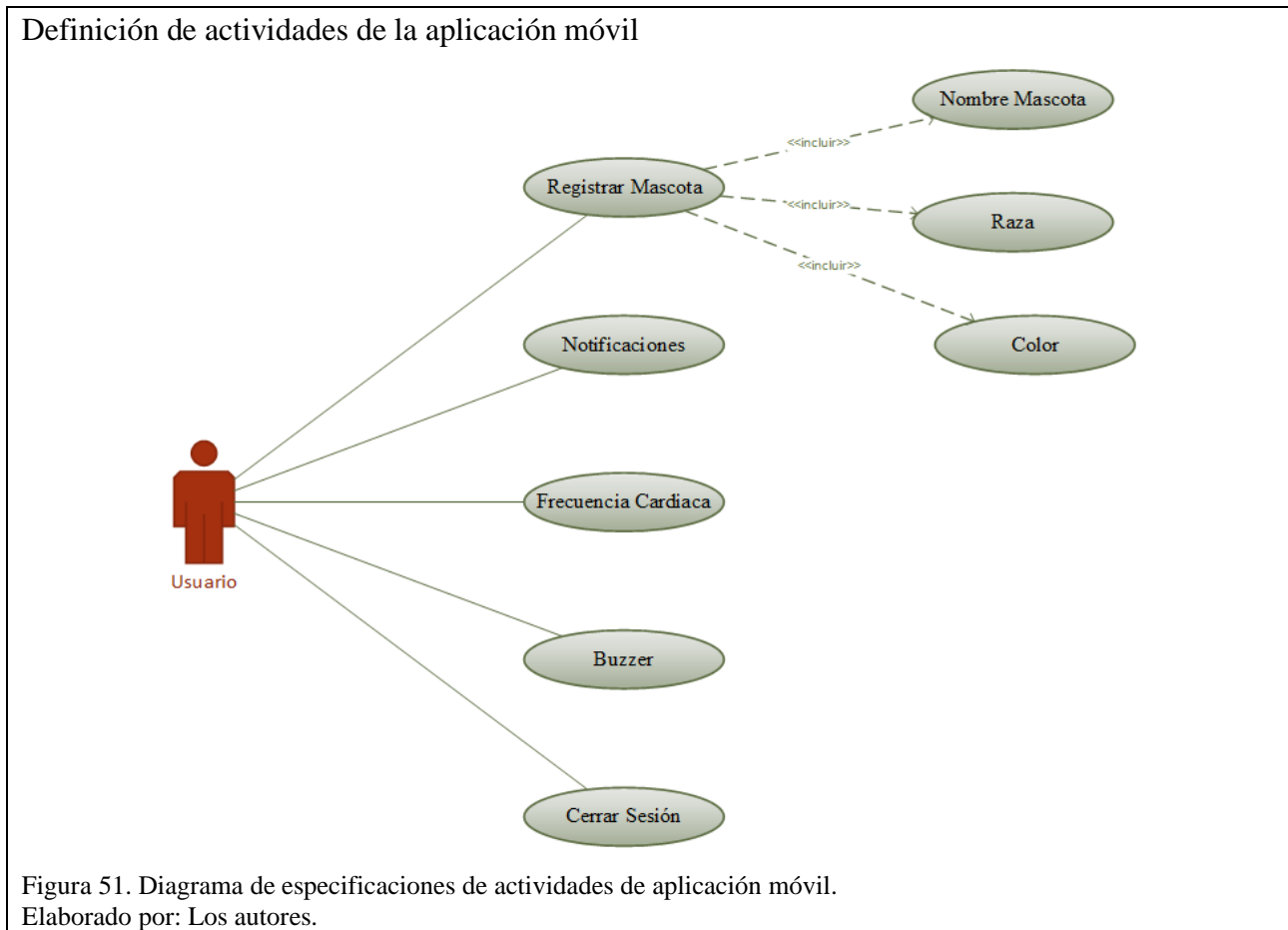
En la Tabla 10, se examina el caso de uso de inicio de sesión.

Tabla 10. Especificación del inicio de sesión a la aplicación móvil

Caso de uso	Inicio de Sesión de la Aplicación Móvil
Actores	Usuarios registrados.
Funciones Básicas	Registrarse en la Aplicación. Inicio de Sesión. Acceso a la Aplicación.
Tipo	Primario.

Nota: Determinación del caso de uso para el inicio de sesión APP.
Elaborado por: Los autores.

En la Figura 51, se analiza el caso de uso de las diferentes actividades de la aplicación.



En la Tabla 11, se detallan las actividades que cumple la aplicación móvil.

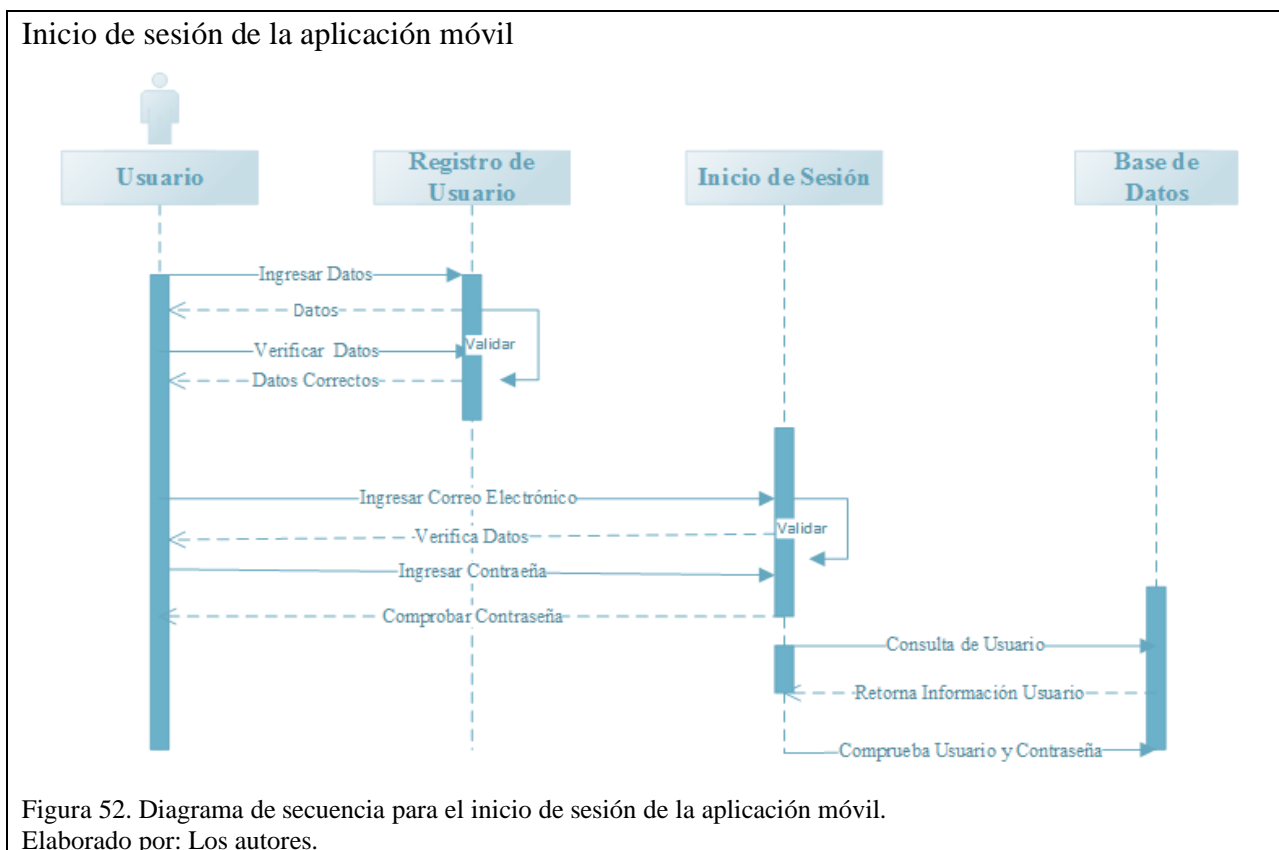
Tabla 11. Especificaciones de las actividades de la aplicación móvil

Caso de Uso	Actividades de la Aplicación Móvil
Actores	Nuevos usuarios, usuarios existentes.
Funciones Básicas	Registro de mascota. Localización de la mascota. Visualización del ritmo cardiaco de la mascota. Notificaciones, alertas.
Tipo	Primario.

Nota: Detalle de las tareas que realiza la aplicación.
Elaborado por: Los autores.

3.2.6. Diagramas de secuencia

En la Figura 52, se evalúa el esquema de secuencia para el inicio de sesión.



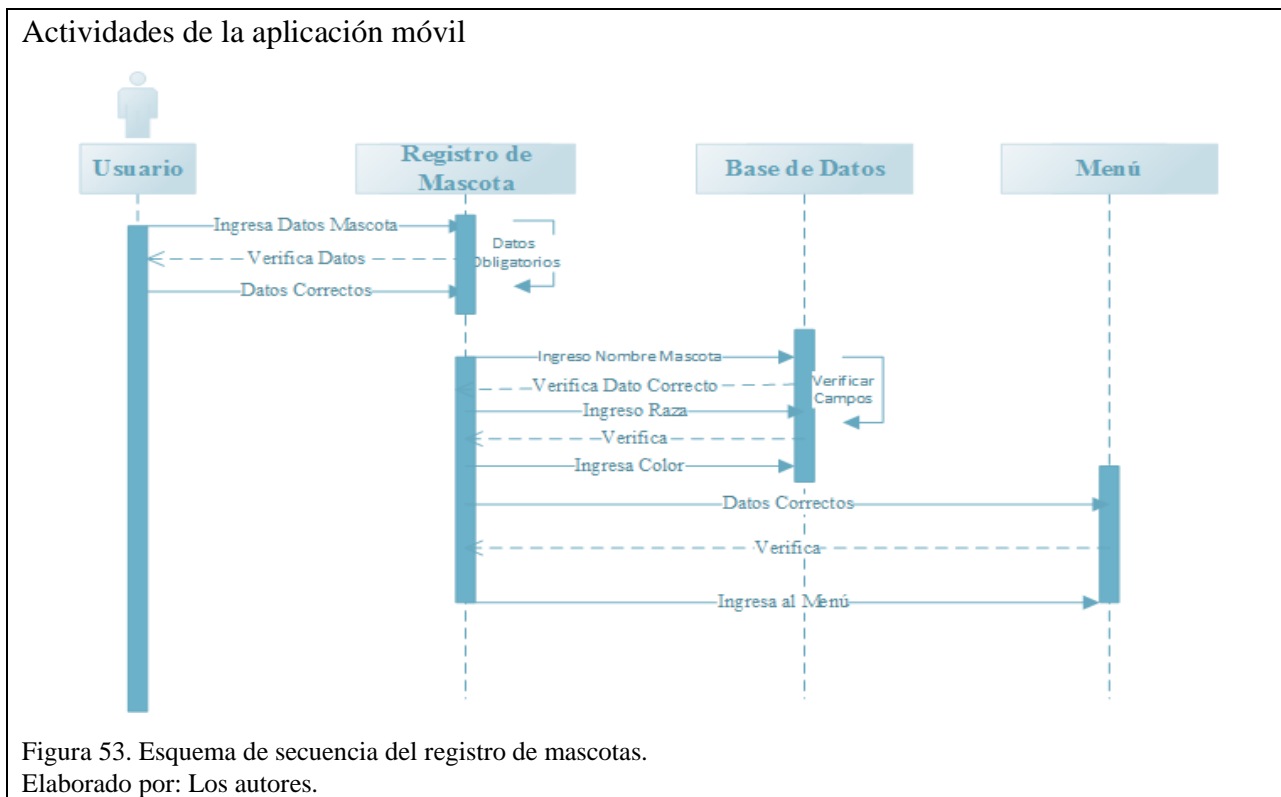
En la Tabla 12, se examinan las especificaciones de actividades del caso de secuencia.

Tabla 12. Actividades del usuario en la aplicación móvil.

Actividades del usuario	Respuesta de la aplicación móvil
Las actividades comienzan con el registro del usuario.	La aplicación verifica cada uno de los campos ingresados a la base de datos.
El usuario ingresa los datos necesarios.	La aplicación valida datos.
En caso de tener cuenta el usuario inicia la sesión.	La aplicación móvil registra el inicio de sesión.

Nota: Especificaciones de actividades del caso de uso.
Elaborado por: Los autores.

En la Figura 53, se examina el esquema de secuencia del registro de mascotas.



En la Tabla 13, se especifica la ocupación del registro de mascotas.

Tabla 13. Actividades en la aplicación móvil

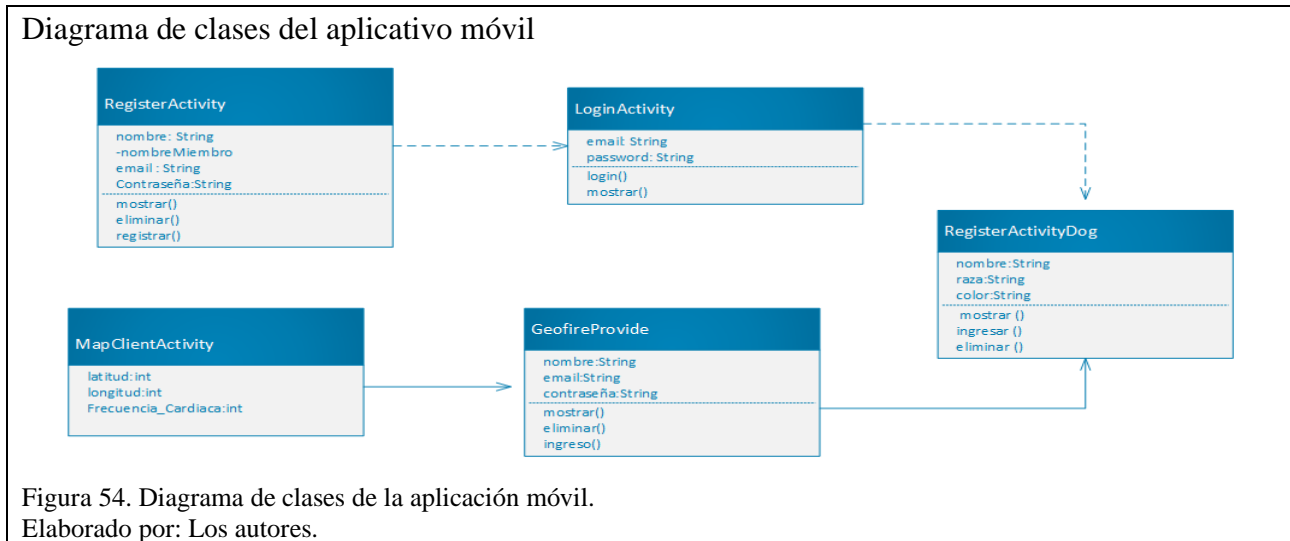
Actividades de los Actores	Respuesta de la Aplicación Móvil
La actividad comienza cuando se registra una nueva mascota.	Validar datos correctos: nombre, raza, color.
Localización de la mascota Visualización de frecuencia cardiaca	Verificación de todos los campos.
Notificaciones	Alerta al usuario.

Nota: Especificaciones de cada una de las actividades de la aplicación móvil.

Elaborado por: Los autores.

En la Figura 54, se muestran las clases principales que incluye la aplicación móvil junto con sus métodos y variables.

3.2.7. Diagramas de clases



3.2.8. Diagrama de base de datos no relacional

En la Figura 55, se aprecia que la información se almacena en Firebase Realtime Database como objetos JSON, por ende, la base se expresa como un árbol de JSON alojado en la nube.



3.2.9. Diseño de la App

Se optó por utilizar el entorno de desarrollo de Android Studio, puesto que contiene herramientas que proporcionaron un funcionamiento adecuado para lograr una interfaz que resulte interactiva y amigable al usuario.

3.2.9.1. Desarrollo de la App

Se inició con una idea sencilla y fácil de entender por el usuario evitando que se presenten dificultades al momento de manipular el aplicativo móvil. Es importante el tipo de API que se elija, pues de esto dependerá el nivel de compatibilidad que poseerá la aplicación, pues las versiones de Android tienen actualizaciones constantes y la última versión publicada es la 9.0 Pie.

3.2.9.2. Esquema del software

En la Figura 56, se muestra todo el proceso de las distintas pantallas con las que el usuario interactúa en la aplicación móvil, una vez registrado el usuario accede al mapa, así como a las funciones para registrar a su mascota y localizarla.

Pantallas del software

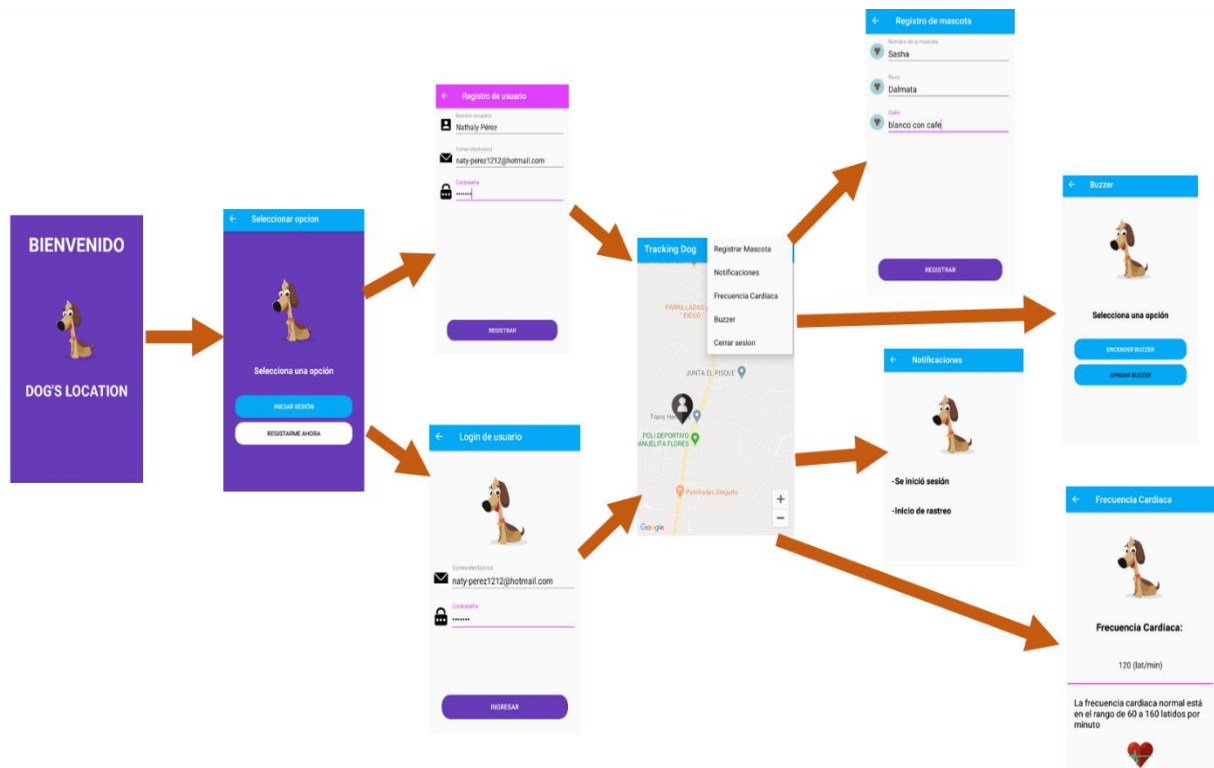


Figura 56. Pantallas de interacción del software.
Elaborado por: Los autores.

3.2.10. Permisos de conexión

En la Figura 57, se establecen los permisos necesarios para que la aplicación acceda a internet y utilice la ubicación del GPS.

Permiso de acceso a internet

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  package="com.optic.gps">

  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Figura 57. Conceder permiso de conexión a internet y ubicación.
Elaborado por: Los autores.

En la Figura 58, se otorgan las credenciales de la llave de Google API para acceder a los mapas de Google para realizar la localización.

Obtención de las llaves de Google API

```
<meta-data
  android:name="com.google.android.geo.API_KEY"
  android:value="@string/google_maps_key" />
</application>
```

Figura 58. Uso de la llave de Google API.
Elaborado por: Los autores.

En la Figura 59, se emplea un plugin de los servicios de Google para obtener las credenciales y las configuraciones de los desarrolladores.

Plugin para el GSM

```
apply plugin: 'com.google.gms.google-services'
```

Figura 59. Plugin para el GSM.
Elaborado por: Los autores.

En la Figura 60, se encuentran las dependencias que necesita Android Studio para establecer una comunicación exitosa con la plataforma de Firebase, tales como la autenticación de usuarios, el estudio analítico, las bases de datos en tiempo real entre otras.

Dependencias del Gradle para Firebase

```
dependencies {
  implementation fileTree(dir: 'libs', include: ['*.jar'])
  implementation 'androidx.appcompat:appcompat:1.1.0'
  implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
  implementation 'com.google.android.material:material:1.0.0'
  implementation 'com.google.firebase:firebase-analytics:17.2.2'
  implementation 'com.google.firebase:firebase-auth:19.1.0'
  implementation 'com.google.firebase:firebase-database:19.1.0'
  implementation 'com.google.android.gms:play-services-maps:16.1.0'
  implementation 'com.google.android.gms:play-services-location:16.0.0'
  implementation 'com.firebase:geofire-android:2.2.0'
```

Figura 60. Dependencias de Gradle para Firebase.
Elaborado por: Los autores.

CAPÍTULO IV

PRUEBAS Y RESULTADOS DEL PROTOTIPO

En el presente capítulo, se detallará un plan de pruebas correspondientes para el prototipo IoT, tanto en hardware como en software para la verificación de su correcto funcionamiento.

4.1. Pruebas de funcionamiento de hardware

4.1.1. Pruebas de encendido de prototipo

Para verificar que el prototipo final encienda se colocó para tal fin un switch de ON/OFF, mismo que se visualiza en la Figura 61, a través del cual se pone en marcha gracias a la batería incorporada destinada a proveer alimentación a los módulos situados internamente en la caja.

Switch de ON/OFF



Figura 61. Switch ON/OFF del prototipo.
Elaborado por: Los autores.

Una vez encendido el prototipo y alimentado el circuito interno, se encenderá un led de cada dispositivo mostrando que todos están en correcto estado y son funcionales para empezar la localización de la mascota equipada con el prototipo.

En la Figura 62, se observa cómo se encienden las luces de los componentes del prototipo y estas parpadean de acuerdo con las acciones que se realicen. Cuando el GSM parpadea rápido, está en busca de una red móvil. Cuando parpadea más lento y el buzzer emite dos pitidos, significa que ya

encontró una señal. Una vez encuentra la señal el GPRS, nuevamente parpadean rápido y la información se envía a la base de datos.

Encendido de luces led de componentes

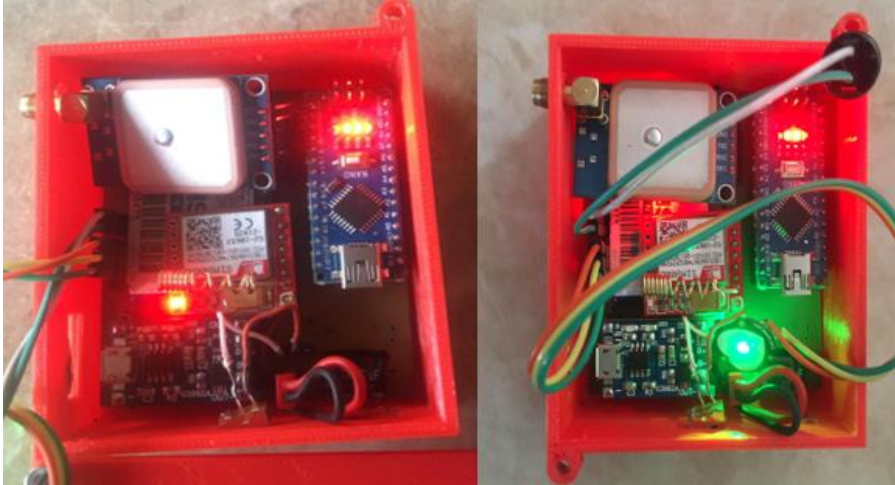


Figura 62. Comprobación de encendido de luces led.
Elaborado por: Los autores.

En la Figura 63, se muestra el prototipo final acoplado a un arnés para posteriormente probarlo en la mascota con la aplicación móvil.

Prototipo final



Figura 63. Prototipo final adaptado a un arnés.
Elaborado por: Los autores.

En la Figura 64, se observa el prototipo ya probado en la mascota. Para asegurarse que el sensor de la frecuencia cardiaca funciona correctamente, es conveniente colocar el arnés en un sitio cercano al corazón para que los latidos sean captados con más precisión.

Prueba del prototipo en la mascota



Figura 64. Prototipo final adaptado a un arnés.
Elaborado por: Los autores.

4.2. Pruebas de funcionamiento de software

Para verificar el funcionamiento del software se procedieron a realizar las pruebas de integración para saber si los módulos del prototipo se conectaban correctamente con la aplicación móvil y si estos enviaban o no los datos capturados a la base de datos.

4.2.1. Pruebas de toma de datos

Una vez encendido el prototipo para obtener los datos y cargarlos a la base, es necesario conectarlo con el cable microUSB y ejecutar el script del Arduino IDE, de esta manera se comprobará los parámetros obtenidos de los módulos a través del monitor serie. En la Figura 65, se muestra el inicio del proceso de búsqueda de señal para comunicarse con el módulo GPS.

Búsqueda de señal del GSM

```
COM14
Testing TinyGPS library v. 1.3
by Mikal Hart
-----
Sats HDOP Latitude Longitude Fix Date Time Date Alt Course Speed Card
Age (deg) Age (m) --- from GPS ---
-----
***** 0
***** 144
***** 288
***** 432
***** 576
***** 720
***** 864
***** 1008
***** 1152
***** 1296
***** 1440
***** 1584
***** 1728
***** 1872
***** 2016
***** 2160
***** 2304
***** 2448
***** 2592
***** 2736
***** 2880
***** 3024
***** 3168
***** 3312
***** 3456
***** 3600
***** 3744
***** 3888
***** 4032
***** 4176
***** 4320
```

Figura 65. Búsqueda de señal del GSM.
Elaborado por: Los autores.

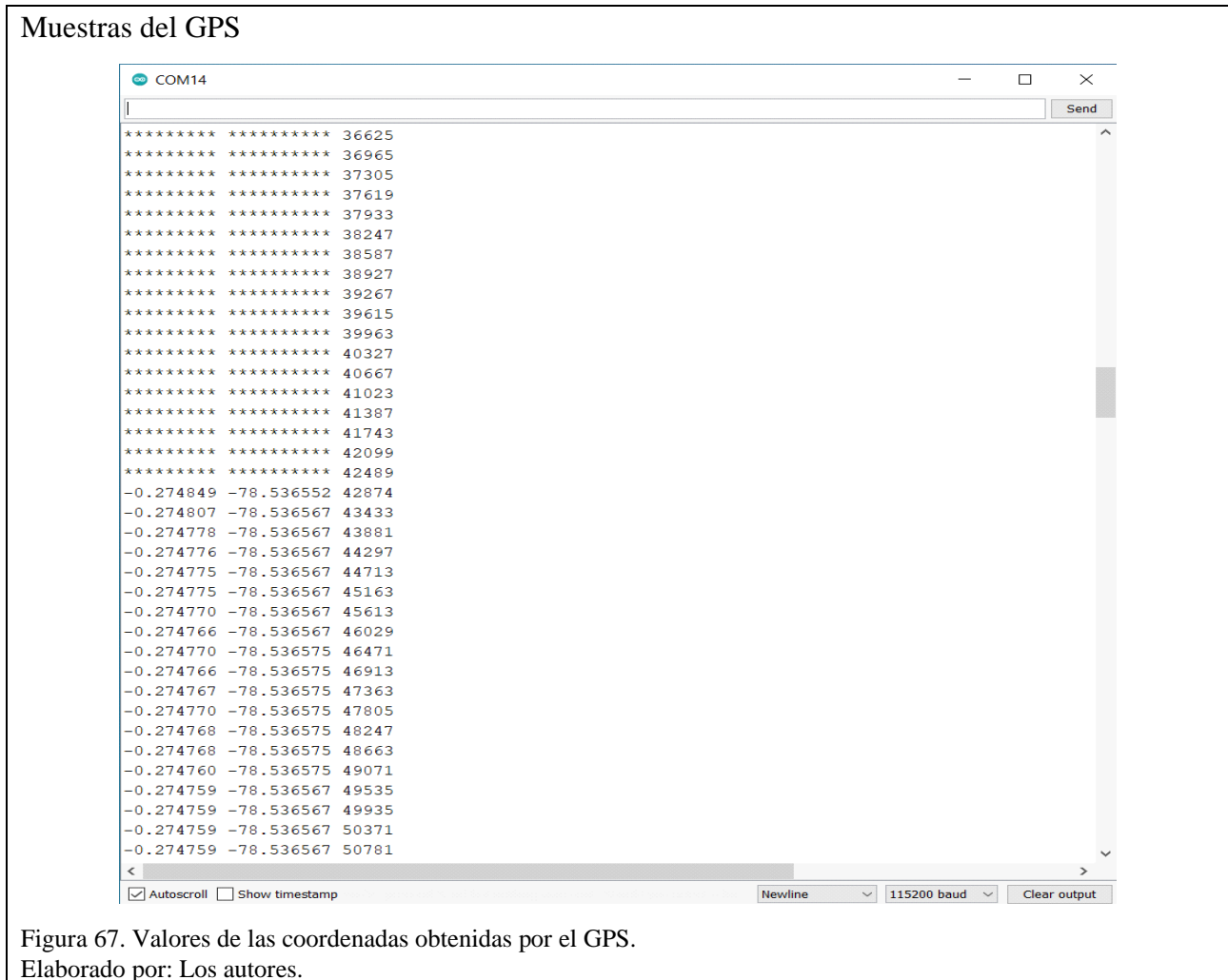
En la Figura 66, se establece la conexión del módulo con la APN de Claro a través del comando AT+SAPBR.

Búsqueda del APN

```
COM11
***** 13042
***** 13042
***** 13042
***** 13042
***** 13042
***** 13042
***** 13042
***** 13042
***** 13042
***** *Encendiendo...
leolIgnoring echo: AT+CSTT="internet.claro.com.ec","",""
Ignoring echo: AT+SAPBR=3,1,"APN","claro"
Autoscroll [x] Mostrar marca temporal [ ] Sin ajuste de línea [v] 9600 baudio [v] Limpiar salida [v]
```

Figura 66. Búsqueda del APN de Claro.
Elaborado por: Los autores.

En la Figura 67, se visualizan las coordenadas obtenidas por el GPS a través del monitor serie.



4.2.2. Pruebas de almacenamiento

Una vez realizadas las pruebas de obtención de datos, se verifica que estos sean almacenados correctamente en la plataforma de Firebase.

En la Figura 68, el módulo GSM se enlaza hacia la plataforma de Firebase donde se recopila la información obtenida por el GPS y enviada al GSM utilizando una comunicación UART.

4.2.3. Pruebas de la App

Se verificarán las pantallas que contiene la aplicación móvil y se mostrará el ingreso de los datos en la plataforma de Firebase, para obtener una sencilla interacción con el usuario final.

4.2.3.1. Prueba de registro

En la Figura 70, se detalla el registro de usuario junto a la verificación del almacén de datos.

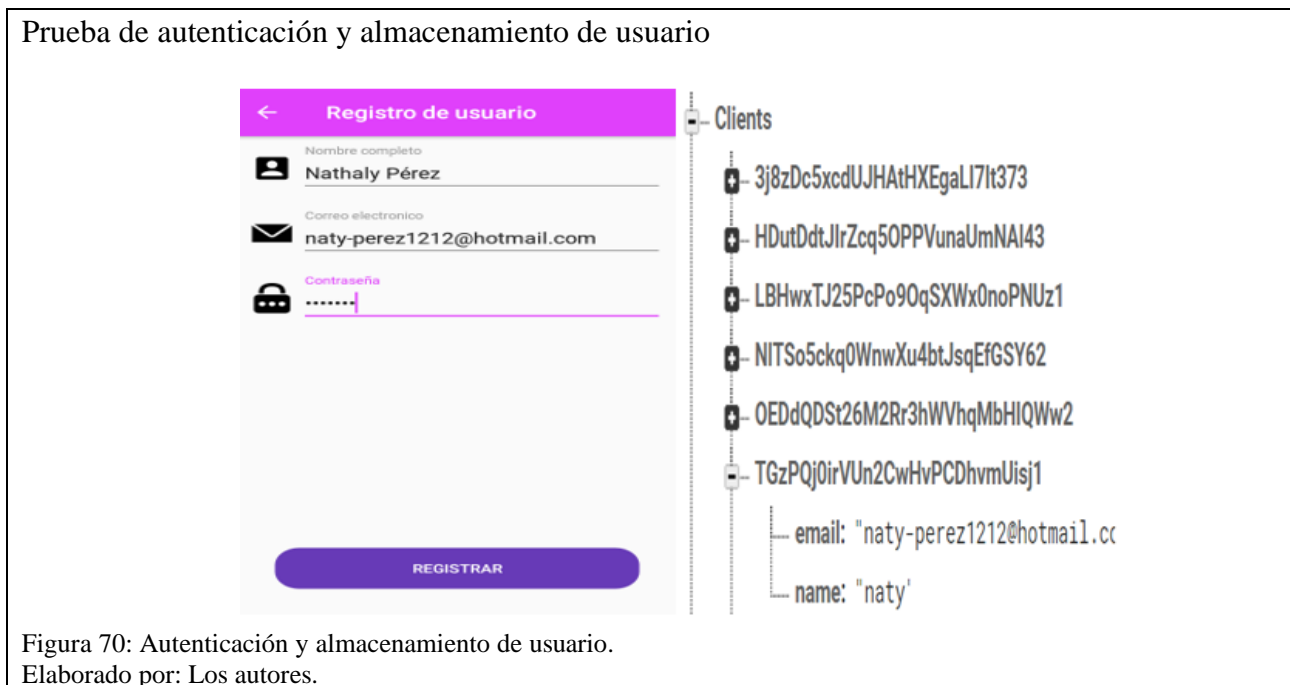


Figura 70: Autenticación y almacenamiento de usuario.
Elaborado por: Los autores.

4.2.3.2. Prueba de entrada al sistema

Se verifica el correcto inicio de sesión donde se debe completar los campos obligatorios para que sean almacenados en Firebase. En la Figura 71, el usuario inicia por medio del aplicativo móvil y esos datos se quedan guardados en la base de autenticación.

Pantalla de inicio de sesión

← Login de usuario

Correio electrónico
naty-perez1212@hotmail.com

Contraseña
.....

INGRESAR

naty-perez1212@hotmail.com

18 jul. 2020

20 jul. 2020

TGzPQj0irVUn2CwHvPCDhvmUisj1

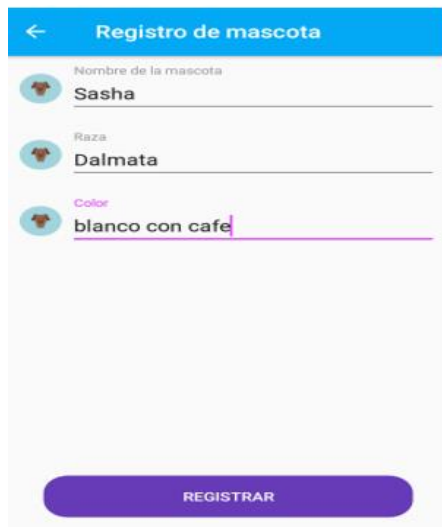
Figura 71. Inicio de sesión y almacenamiento en Firebase.
Elaborado por: Los autores.

4.2.3.3. Pruebas de registro de mascota

Una vez el usuario se registra e inicia sesión, tiene la opción de registrar los datos principales de su mascota como el nombre, raza y color, así se tendrá un control de las características de la misma registrado con la cuenta de su dueño.

En la Figura 72, cuando el usuario inicia sesión, puede ingresar los datos de su mascota para ser almacenados en la base de datos.

Registro de la mascota



Registro de mascota

Nombre de la mascota
Sasha

Raza
Dalmata

Color
blanco con cafe

REGISTRAR

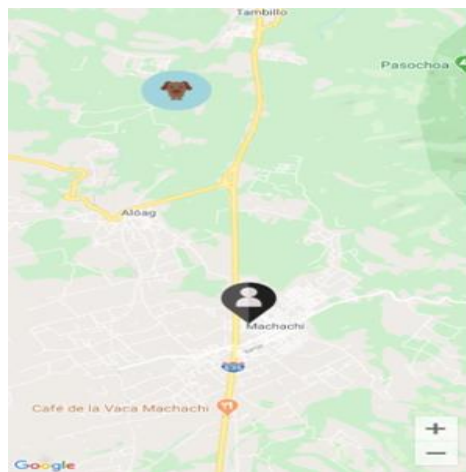
```
KHM1bw86AkNpwyzLpPzbRtBcKv1  
dogColor: "blanco con cafe"  
dogRace: "Dalmata"  
email: "dominio_mascota@Sasha.cc"  
id: "KHM1bw86AkNpwyzLpPzbRtBcKv1"  
nombreMascota: "Sasha"
```

Figura 72. Registro de datos de mascota.
Elaborado por: Los autores.

4.2.3.4. Pruebas de localización y visualización de la frecuencia cardiaca

En la Figura 73, se muestran las pulsaciones de la mascota y se aprecia el posicionamiento tanto de la mascota como del dueño.

Localización de la mascota



```
mascotas_activas + X  
OMMwvGtEZ3N48NtqglPyjhAn9rt1  
g: "6rbjh51xvj"  
0: -0.27209  
1: -78.5375
```

Figura 73. Localización y frecuencia cardiaca.
Elaborado por: Los autores.

En la Figura 74, se muestra el dato obtenido de la frecuencia cardiaca de la mascota cargado en la nube.

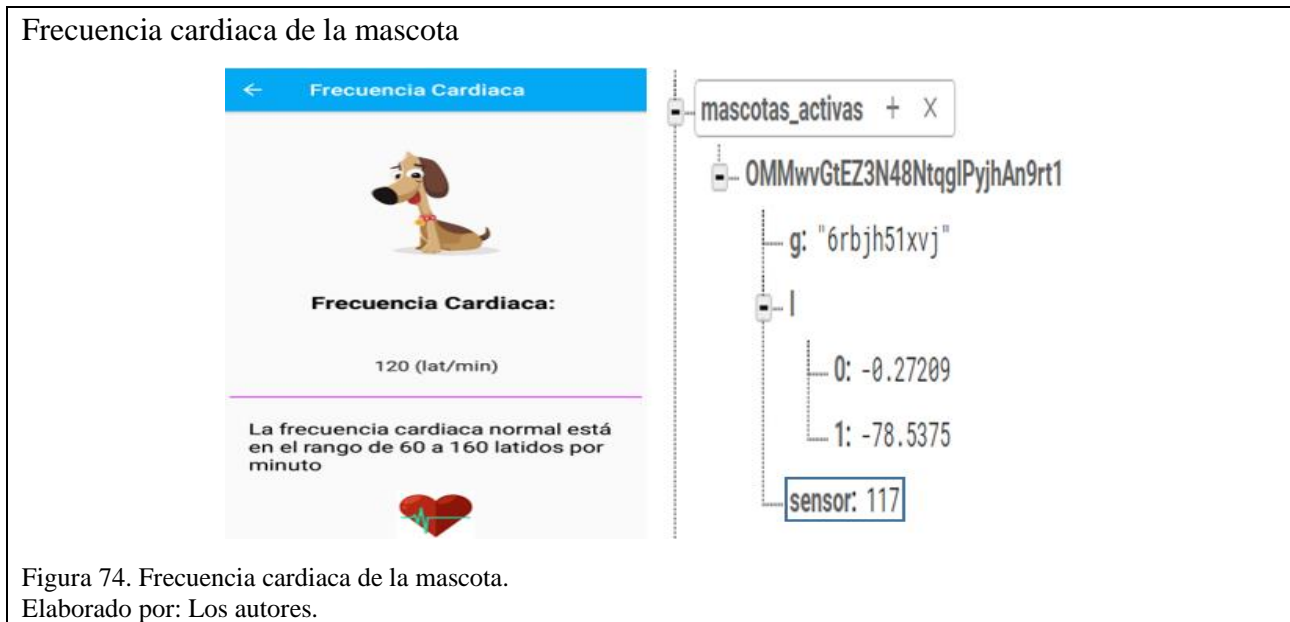


Figura 74. Frecuencia cardiaca de la mascota.
Elaborado por: Los autores.

4.2.3.5. Pruebas de validación de campos de la App

Se muestra la validación de los campos, en caso de que el usuario ingrese a la aplicación sin completar los datos, o si la longitud de la contraseña no cumple con un mínimo de seis caracteres.

4.2.4. Pruebas y resultados del prototipo

Se realizará una comprobación de resultados en base a una tabla que contendrá valores obtenidos del módulo GPS y del sensor de frecuencia cardiaca.

4.2.4.1. Resultados del GPS

En los resultados se tomaron en cuenta diez puntos geográficos para obtener el valor de latitud y longitud además se procedió a compararlos con los valores entregados por el prototipo a la aplicación móvil. Para realizar el cálculo, primero se tomaron datos a partir de las coordenadas geográficas que mostró la aplicación móvil de la posición donde se encontraba la mascota

inicialmente, para posterior cotejarlos con los puntos geográficos reales con ayuda de una herramienta con mejor precisión, en este caso se utilizó la API de Google, se realizó una tabla con ambos valores donde se aplicó la ecuación de Haversine que sirve para hallar la distancia entre dos ubicaciones geográficas en la tierra, encontrando el margen de error con respecto a la posición inicial de la mascota localizada.

$$d = 2R * \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{lat1 - lat0}{2} \right) + \cos(lat0) * \cos(lat1) * \sin^2 \left(\frac{long1 - long0}{2} \right)} \right)$$

Ecuación 3. Ecuación de Haversine.

Para analizar la Tabla 14, se tomaron diez muestras de latitud y longitud captadas por el GPS del prototipo colocado en la mascota en relación con la ubicación real posteriormente identificada en el mapa, para así determinar un índice de error en cada ejemplo.

Tabla 14. Resultado de las muestras.

N.º	Prototipo		Real		Distancia	
	Latitud 0	Longitud 0	Latitud 1	Longitud 1	Distancia (km)	Distancia (m)
1	-0,110121	-78,297097	-0,110191	-78,297075	0,008158914	8,16
2	-0,110502	-78,296383	-0,110498	-78,29651	0,014128784	14,13
3	-0,097316	-78,289665	-0,097348	-78,28956	0,012205741	12,21
4	-0,084568	-78,292363	-0,084608	-78,292456	0,011257341	11,26
5	-0,085736	-78,292439	-0,085704	-78,29234	0,01156926	11,57
6	-0,063103	-78,289301	-0,063082	-78,289397	0,010927461	10,93
7	-0,064305	-78,289687	-0,064337	-78,289623	0,007956462	7,96
8	-0,064884	-78,287595	-0,064938	-78,287541	0,008491802	8,49
9	-0,077894	-78,298549	-0,077905	-78,298453	0,010744478	10,74
10	-0,078033	-78,297777	-0,078108	-78,297691	0,012688698	12,69
Total						10,81

Nota: Se muestran los valores tomados de latitud, longitud y la distancia obtenida.

Elaborado por: Los autores.

La distancia obtenida del prototipo en varias ocasiones asciende en otras capta un valor más estable, teniendo en cuenta las condiciones climáticas como los fuertes vientos, los días lluviosos y la

precisión propia que ofrece el módulo, se obtiene un error de 10,81 [m] de la posición inicial con respecto a la posición real.

4.2.4.2. Resultados del sensor de frecuencia cardiaca

Para el sensor de frecuencia cardiaca se tomaron cinco muestras en cinco estados diferentes según la energía de la mascota, para obtener datos en cada una de esas situaciones para posteriormente comprobarlas con las pulsaciones que capta el sensor y las pulsaciones tomadas de forma manual a la mascota. Para tal fin se optó por recolectar la información en la Tabla 15, misma que se muestra a continuación.

Tabla 15. Resultados de pulsaciones según el estado energético.

Estado	Valor Aproximado (lat/min)	Valor Real (lat/min)	Porcentajes
Reposo	60	76	21,05%
Caminata	83	100	17%
Caminata Rápida	119	130	8,46%
Ejercicio	174	180	3,33%
Descanso	110	118	6,8%

Nota: Valores de frecuencia cardiaca aproximados y reales.
Elaborado por: Los autores.

En la Figura 75, se muestra el margen de error dado por el sensor de la frecuencia cardiaca, para lo cual se tomaron de muestra cinco estados diferentes de la mascota, los resultados se muestran a continuación:

Para el primer estado en reposo, existe un margen de error del 21,05% con respecto al valor tomado manualmente, para el segundo estado en caminata se obtuvo un error del 17%, para el tercer estado en caminata rápida se registró un margen de error del 8,46%, para el cuarto estado realizando ejercicio el error que se obtiene es del 6%, finalmente para el quinto estado cuando ya la mascota se encuentra descansando el margen de error es del 6,8%, para tal fin se complementó con una gráfica en la que se muestran ambos casos y se interpretan mejor los resultados del margen de error.

Gráfica de frecuencia cardiaca

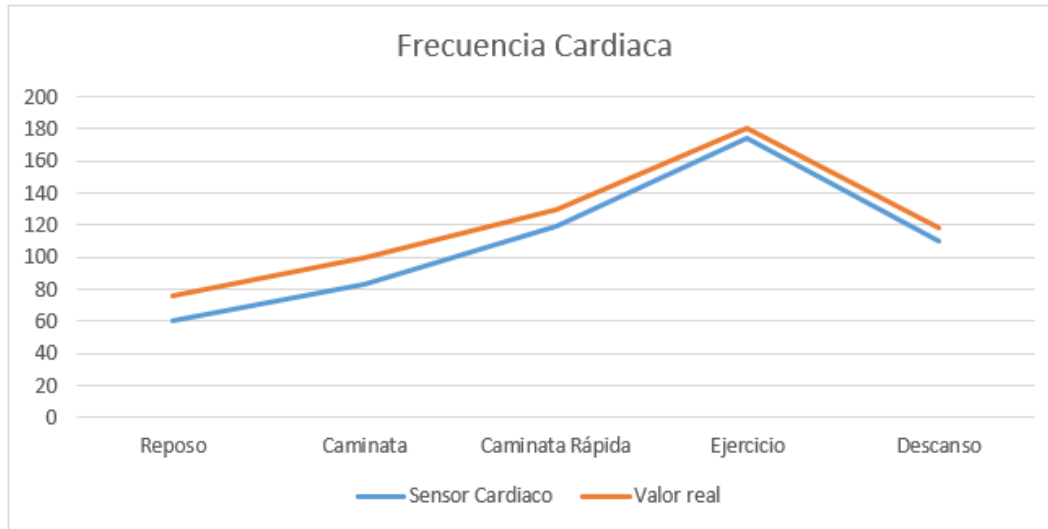


Figura 75. Gráfica de frecuencia cardiaca.
Elaborado por: Los autores.

4.2.4.3. Resultados de elección del prototipo (hardware)

La implementación del prototipo final se elaboró en base a medidas tomadas con respecto a mascotas de raza mediana, es por eso que el tamaño del prototipo final no sobrepasa los 10 [cm] de lado y lado. En un principio se experimentó con una placa y un protoboard de mayores dimensiones, hasta que finalmente se imprimió una placa PCB para sustituir al protoboard.

En la Figura 76, se muestran las medidas de los prototipos, en la izquierda se observa el prototipo inicial con las medidas de 10 [cm] x 8 [cm], en el lado derecho se observa el prototipo final con medidas más considerables con respecto al prototipo de la izquierda. Finalmente se optó por escoger componentes de dimensiones más reducidas para lograr un tamaño menor y que resulte más cómodo para la mascota.

Medidas del prototipo

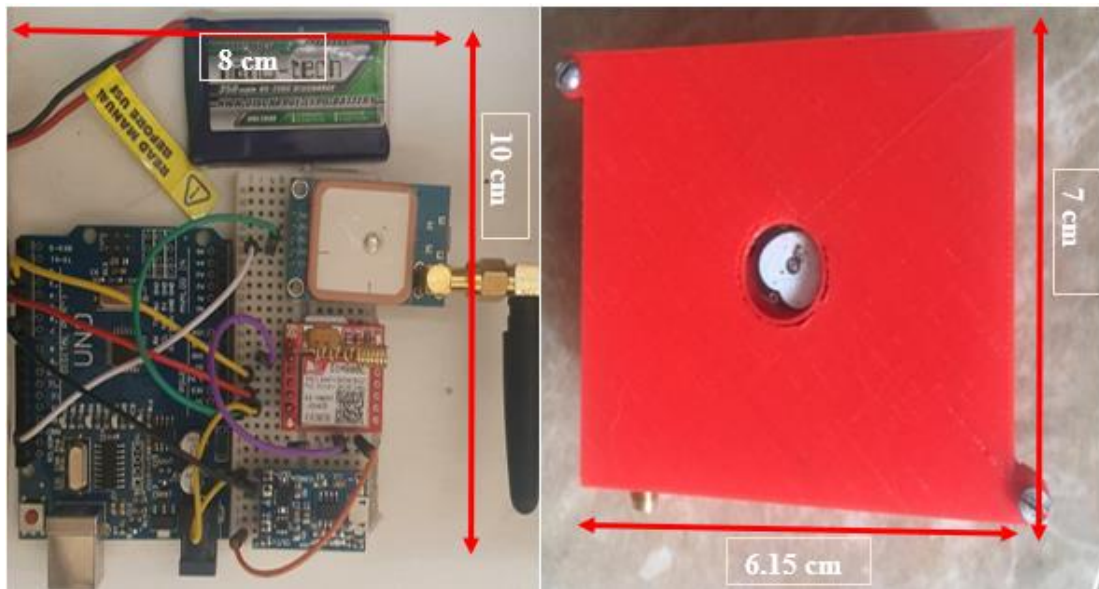


Figura 76. Medidas del prototipo para la elección.
Elaborado por: Los autores.

4.2.4.4. Resultados de perímetro del GPS.

Para las pruebas de delimitación del perímetro de la mascota, se estableció un rango de 500 metros desde la posición actual del dueño que disponga de la aplicación móvil, si la mascota excede esa distancia se notificará mediante una alerta advirtiendo que la mascota está alejándose más de lo debido de una zona segura.

En la Figura 77, se muestra la mascota saliendo del rango establecido de 500 metros por lo que la aplicación móvil envía una alerta advirtiendo al dueño de la mascota que la misma se encuentra en una distancia mayor a la permitida y está en riesgo de perderse.

Alerta de perímetro de la mascota

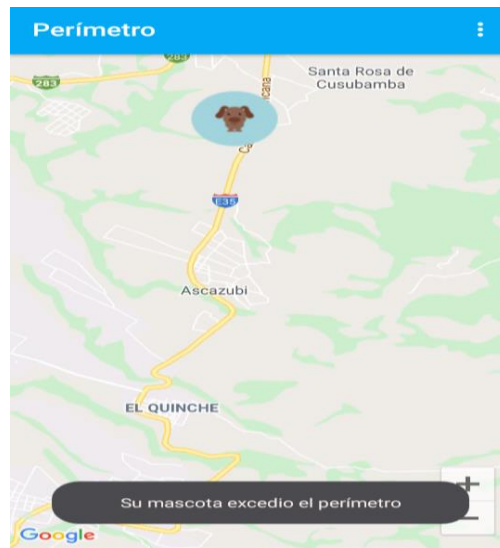


Figura 77. Alerta de perímetro de la mascota.
Elaborado por: Los autores.

En la Figura 78, se observan las validaciones para el inicio de sesión, si no coinciden los campos, también se valida la longitud de al menos seis caracteres para elegir una contraseña, seguido de la validación de que es obligatorio llenar los campos de email, contraseña y finalmente el de la obligación de ingresar todos los datos.

Validaciones en diferentes escenarios

The figure displays four screenshots of a mobile application interface, arranged in a 2x2 grid, illustrating different validation scenarios. Each screenshot features a blue header bar with a back arrow and a title, a cartoon dog illustration, and input fields for email and password.

- Top Left (Login de usuario):** The email field contains "bryan." and the password field contains ".....". A grey error message box at the bottom states "La contraseña o el password son incorrectos". The "INGRESAR" button is visible.
- Top Right (Login de usuario):** The email field contains "bryan_torres@hotmail.com" and the password field contains "....". A grey error message box at the bottom states "La contraseña debe tener mas de 6 caracteres". The "INGRESAR" button is visible.
- Bottom Left (Login de usuario):** The email field contains "bryan.2" and the password field is empty. A grey error message box at the bottom states "La contraseña y el email son obligatorios". The "INGRESAR" button is visible.
- Bottom Right (Registro de mascota):** This screen is for pet registration. It has three input fields: "Nombre de la mascota", "Raza", and "Color", all of which are empty. A grey error message box at the bottom states "Ingrese todos los campos". The "REGISTRAR" button is visible.

Figura 78. Validación de campos en diferentes escenarios.
Elaborado por: Los autores.

CONCLUSIONES

- Para analizar la precisión del sensor GPS, se realizó la toma de diez muestras en diferentes puntos geográficos con el prototipo instalado en la mascota para obtener el margen de error absoluto resultante del módulo, para posteriormente compararlo con los valores adquiridos de la herramienta de localización de la aplicación móvil. Se precisó un margen de error en la distancia de 10,81[m] con respecto a la ubicación real de la mascota.
- El margen de error relativo que genera el sensor de la frecuencia cardíaca es de 14.02%, promedio tomado desde cinco diferentes estados de ánimo de la mascota. Con los datos reales tomados se verifica la falla en cada uno de sus estados, el margen de error del sensor en caso de que el animal se encuentre en descanso es de 21,05%, lo que representa a 16 pulsaciones no medidas, si el estado del animal se encuentra en caminata su error es de 17%, que representa a 17 pulsos no captados durante la caminata, en la caminata rápida se toma un margen de error de 8,46%, lo que representa 11 pulsaciones no captadas por el sensor, cuando el animal se encuentra haciendo ejercicio el margen de error es del 3,33%, lo que representa 6 pulsaciones no logradas durante el ejercicio, y el último estado en descanso con un margen de error del 6,8 %, que representa 8 pulsaciones no captadas, como se observa, la precisión del sensor cuando el animal se encuentra en una actividad física resulta más exacta que cuando la mascota se encuentra en un estado de reposo y los latidos del corazón no son tan perceptibles por el sensor.
- Para la integración de los componentes, se optó por la impresión de una placa PCB puesto que inicialmente se realizó la prueba del prototipo en un protoboard y con un microcontrolador de tamaño mayor al implementando finalmente, por motivos de reducción de espacio debido a que el prototipo está destinado para las mascotas. La utilización de una

placa PCB ayudó a minimizar el tamaño final y lograr unas medidas de 7 [cm] x 6.15 [cm], reduciendo el prototipo en 3 [cm] de alto y 4 [cm] de ancho.

- Se concluyó que la plataforma Firebase al no utilizar bases de datos relacionales, emplea una estructura diferente compuesta por un árbol JSON que almacena los datos en la nube. Este tipo de estructura está orientada para realizar proyectos como el propuesto en el presente trabajo, ya que tiene la capacidad de almacenar datos en tiempo real y una de sus ventajas está en el desarrollo de aplicaciones móviles que se sincronizan directamente con esta plataforma. Todo el almacenamiento de la aplicación se coordina directamente con las Cloud Functions de Google y no se tiene que alojar la aplicación en un servidor externo ya que todo se administra desde la misma aplicación.

RECOMENDACIONES

- Es importante que exista una buena cobertura de red GSM con una operadora telefónica, para obtener una mejor precisión en la toma de datos. Es por ese motivo que se recomienda una implementación extra en el medio de comunicación para una mejor cobertura.
- La utilización de otro tipo de tecnologías mejorará de alguna manera la estructura del prototipo en tamaño, así como en funcionamiento, llegando a obtener datos aún más precisos.
- El prototipo se elaboró para mascotas de raza mediana y grande, es por ese motivo que no se tuvo ningún problema con los componentes que se utilizó para el desarrollo de este, sin embargo, con nuevos requerimientos se debería optar por un prototipo para perros de raza pequeña.
- Se recomienda experimentar con nuevas plataformas que brinden servicios en la nube similares a Firebase, para así obtener mayores prestaciones en el mercado.

LISTA DE REFERENCIAS

Artículos Académicos

- Ávila Mejía, O. (2011). *Computación en la nube*. Universidad Autónoma Metropolitana, Mexico.
- Bonilla Fabela, I., Tavizon Salazar, A., Morales Escobar, M., Guajardo Muñoz, L. T., & Laines Alamnia, C. I. (2016). *IOT, EL INTERNET DE LAS COSAS Y LA INNOVACIÓN DE SUS APLICACIONES*. Universidad Autónoma de Nuevo León, Monterrey.
- Fernández, G. M. (11 de Marzo de 2008). *El fraude en roaming: estrategias de ataque y de*. Obtenido de http://www.iirsa.org/admin_iirsa_web/Uploads/Documents/taller_roaming_pp_fraude.pdf
- Gómez Ruiz, C. A. (2015). *LOCALIZACIÓN EN ESPACIOS INTERIORES DE DISPOSITIVOS MÓVILES POR MEDIO DE TECNOLOGÍAS WIFI, BASE PARA EL DESARROLLO DE APLICACIONES SOCIALES*. Universidad Agustiniana, Bogotá.
- Khawas, C., & Shah, P. (06 de 2018). *Aplicación de Firebase en el desarrollo de aplicaciones de Android a Estudiar*. Obtenido de https://www.researchgate.net/publication/325791990_Application_of_Firebase_in_Android_App_Development-A_Study
- M.K. Sudarshan a, *. S.-X. (2005). *Stray dog population demographics in Jodhpur, India following a population control/rabies vaccination program*. London: ELSEVIER.
- Mete, M. R. (03 de 2014). VALOR ACTUAL NETO Y TASA DE RETORNO: SU UTILIDAD COMO HERRAMIENTAS PARA EL ANÁLISIS Y EVALUACIÓN DE PROYECTOS DE INVERSIÓN. *FIDES ET RADIO*, 7, 67-85. Recuperado el 28 de 05 de 2020
- Stella de la Torre1, C. F.-R. (2018). *Resultados del I Censo Ciudadano de Perros Abandonados en Quito*. Quito.
- Vera Romero, C. A., Barbosa Jaimes, J. E., & Pabón González, D. C. (2017). La Tecnología ZigBee estudio de las características de la capa física. *Scientia Et Technica*.

Imágenes

- AFEL. (02 de 06 de 2020). *Sensor Pulso Cardíaco Corazón*. Obtenido de <https://afel.cl/producto/sensor-pulso-cardiaco-corazon/>
- Benitez Silva, C. J. (2018). *PROTOTIPO DE COLLAR DE MONITOREO PARA GANADO USANDO TECNOLOGÍA LPWAN*. Universidad del Norte , Barranquilla.
- Chávez Castrillón, F. R., & Yuquilema León, C. G. (2013). *APLICACIONES CON MINICOMPUTADORES RASPBERRY PI PROVISTO DE MÓDULO GPS Y ACELERÓMETRO PARA CONTROL DE VELOCIDAD Y POSICIONAMIENTO*. ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL , Guayaquil.
- Coley, G. (2014). *BeagleBone Black System Reference Manual*. Dallas.

- Flupion. (15 de 03 de 2018). *PROYECTOS DE TECNOLOGÍA*. Obtenido de <https://elblogdelprofesordetecnologia.blogspot.com/2016/02/el-buzzer-o-zumbador.html>
- Leiva, C. (2017). *Gestión de flota y control de conductas de*. Universidad Tecnológica Nacional, Paraná.
- Molina Riviera, Y. J., Sandoval Cardona, J., & Toledo Franco, S. A. (2012). *SISTEMA OPERATIVO ANDROID: CARACTERÍSTICAS Y FUNCIONALIDAD PARA DISPOSITIVOS MÓVILES*. PEREIRA.
- Programarfacil.com*. (15 de 05 de 2020). Obtenido de *Programarfacil.com*: <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino>
- Torrente Artero, Ó. (2013). *ARDUINO Curso práctico de formación* (1 ed.). Ciudad de México: Alfaomega.
- TP-LINK. (28 de 05 de 2020). *Routers Inalámbricos*. Obtenido de <https://www.tp-link.com/ar/home-networking/wifi-router/tl-wr940n/>
- Yuquilema Muñoz, M. D. (2017). *ESTUDIO DE UN SISTEMA DE MONITOREO Y CONTROL DE POSICIONAMIENTO VÍA GPS APLICADO A MASCOTAS*. Universidad de Guayaquil, Guayaquil.

Libros

- Huerta, E., Mangiaterra, A., & Noguera, G. (2005). *GPS Posicionamiento Satelital*. Rosario: UNR Editora.
- Press, M. (2016). *Xamarin*. Washington.
- Tomasi, W. (2003). *SISTEMAS DE COMUNICACIONES ELECTRÓNICAS* (Vol. 4). Phoenix: Pearson Hall.

Revistas

- Ávila Mejía, O. (2012). Android. *Contactos*(83), 45-47.
- Garandi, E., Clemares, N., & Moltoni, A. (2015). Collar con tecnología GPS para monitoreo animal. *Instituto de Ingeniería Rural*.
- Lorefice Sparacino, G. (2003). TECNOLOGÍA INALÁMBRICA BLUETOOTH SOBRE LOS SERVICIOS DE COMUNICACIONES EN LOS ÁMBITOS SOCIAL Y EMPRESARIAL. *Telématique*.

Sitios Web

- Android Studio FAQs. (08 de 12 de 2016). *Android Studio: ventajas, desventajas y principales características*. Obtenido de <https://androidstudiofaqs.com/conceptos/ventajas-desventajas-android-studio>
- AWS. (2020). *Informática en la nube con AWS*. Recuperado el 12 de 06 de 2020, de <https://aws.amazon.com/es/what-is-aws/>

- Besteiros, M. (21 de febrero de 2019). *Constantes vitales de un perro*. Obtenido de Experto Animal: <https://www.expertoanimal.com/constantas-vitales-de-un-perro-24083.html#:~:text=debemos%20desinfectarlo%20bien,-,Frecuencia%20card%C3%ADaca%20de%20un%20perro,ritmo%20ser%C3%A1%20algo%20m%C3%A1s%20lento.>
- Casas, A. (25 de febrero de 2019). *PCWorld*. Obtenido de <https://www.pcworld.es/articulos/smartphones/iphone-vs-android-cuota-de-mercado-3692825/#:~:text=StatCounter%20informa%20que%20durante%20el,apenas%20un%202%2C85%20%25.>
- Castellanos, G. (09 de Septiembre de 2018). Población de perros callejeros en Quito sigue en aumento . *El Comercio*.
- Demian, J. (15 de 08 de 2018). *AWS vs Firebase: ¿es incluso una pelea justa?* Obtenido de <https://dashbird.io/blog/aws-lambda-vs-firebase/>
- Developers. (01 de 05 de 2020). *Introducción a Android Studio*. Obtenido de <https://developer.android.com/studio/intro?hl=es-419>
- Enagora . (2017). *Desventajas de Xamarin, cuando no interesa usarlo*. Recuperado el 05 de 06 de 2020, de <https://www.enagora.com/noticias/desventajas-de-xamarin-cuando-no-interesa-usarlo/>
- Enagora. (2017). *Desarrollar en Xamarin, estas son sus ventajas*. Recuperado el 05 de 06 de 2020, de <https://www.enagora.com/noticias/desarrollar-xamarin-ventajas/>
- Fabián Alexis. (24 de 01 de 2020). *Diferencia clave: Firebase vs MongoDB*. Obtenido de <https://es.bccrwp.org/compare/difference-between-firebase-and-mongodb/>
- Fernández, R. F. (10 de Abril de 2020). *Google Activate*. Obtenido de Universidad Complutense de Madrid: <https://learndigital.withgoogle.com/activate>
- fundaciónesplai. (2015). *GUÍA DE INICIACIÓN A APP INVENTOR*. Recuperado el 05 de 06 de 2020, de <https://codeweek.eu/docs/spain/guia-iniciacion-app-inventor.pdf>
- GOBIERNO DE LA REPÚBLICA DEL ECUADOR. (29 de 05 de 2020). *AGENCIA DE REGULACIÓN Y CONTROL DE LAS TELECOMUNICACIONES*. Obtenido de <https://www.arcotel.gob.ec/1-preguntas-generales/>
- Hernández Blanco, J. (s.f.). *FUNDAMENTOS DEL GPS Y APLICACIONES CON NAVEGADORES*. Recuperado el 10 de 04 de 2020, de <http://www.eweb.unex.es/eweb/exgrafica/Tema14.pdf>
- Ingeniería Mecafenix. (15 de 10 de 2018). *La enciclopedia de la ingeniería*. Obtenido de <https://www.ingmecafenix.com/electronica/el-buzzer/>
- Lazaro, F. (09 de 03 de 2017). *5 Ventajas y desventajas de Firebase en 2017*. Obtenido de <https://obux.wordpress.com/2017/03/09/5-ventajas-y-desventajas-de-firebase-en-2017/>
- ManagementSolutions. (2012). *La nube: oportunidades y retos para los integrantes de la cadena de valor*. Obtenido de <https://www.managementsolutions.com/sites/default/files/publicaciones/esp/La-nube.pdf>

- MAPIT. (24 de 01 de 2017). *¿Qué tipos de localizadores GPS existen?* Obtenido de <https://mapit.me/2017/01/24/tipos-localizadores-gps/>
- PANDORAFMS. (s.f.). *LPWAN COMO BASE DE COMUNICACION IOT* . Obtenido de LPWAN COMO BASE DE COMUNICACION IOT : <https://pandorafms.com/blog/es/que-es-lpwan/>
- Pérez Hernandez, M. (15 de 11 de 2017). *DISEÑA BASES DE DATOS OFIMATICAS*. Obtenido de <https://mape309site.wordpress.com/2017/11/15/ventajas-y-desventajas-de-mysql-oracle-visual-foxpro-y-access/>
- Proserquisa. (05 de 10 de 2016). *Curso de Arduino Lección 1*. Recuperado el 18 de 04 de 2020, de <http://cursoarduino.proserquisa.com/2016/10/05/introduccion/>
- Sánchez, J. (2004). *MySQL guía rápida (versión Windows)*. Cartagena. Recuperado el 10 de 06 de 2020, de <http://www.cartagena99.com/recursos/programacion/apuntes/mysql.pdf>
- Sternheim, I. (2012). *Perros Callejeros en los Países Bajos*. Isis.
- UA MASTER MOVILES. (2012). *Introducción a Android*. Universidad de Alicante, Alicante.
- UNED. (2011). *Controladores industriales de diseño de alto nivel*. Recuperado el 17 de 04 de 2020, de http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE5_3_1.pdf
- Viviana. (29 de 10 de 2018). *MySQL vs MongoDB*. Obtenido de <https://guiadev.com/mysql-vs-mongodb/>

Tesis

- Abrajan Arias, C. S. (2020). *DISEÑO Y CONSTRUCCIÓN DE UN ROBOT SEGUIDOR DE LÍNEA EVASOR DE OBSTÁCULOS EMPLEANDO ARDUINO NANO*. Instituto Superior Tecnológico Vida Nueva, Quito.
- Aguaguíña, A. P. (2017). *Proliferación de perros callejeros y la contaminación ambiental. Caso: Barrio La Ecuatoriana, Quito, Provincia de Pichincha*. Quito.
- Alambiaga Pascual, J. (2017). *ADQUISICIÓN Y PROCESADO DE INFORMACIÓN DE POSICIONAMIENTO GPS MEDIANTE DISPOSITIVO INALÁMBRICO BASADO EN ARDUINO*. Universidad Politécnica de Valencia, Valencia.
- Ayala Delgado, C. A., & Lemos Londoño, D. A. (2019). *SISTEMA DE MONITOREO PARA PACIENTES EN ZONA DE CIRUGÍA UTILIZANDO GEOLOCALIZACIÓN*. UNIVERSIDAD CATÓLICA DE COLOMBIA, Bogota.
- Báez , M., Borrego, Á., Cordero, J., Cruz, L., González, M., Hernández, F., . . . Zapata, Á. (2019). *Introducción a Android*. Madrid: E.M.E. doi: 978-84-96285-39-5
- Barneda Faudot, I. (2008). *ZIGBEE APLICADO A LA TRANSMISIÓN DE DATOS DE SENSORES BIOMÉDICOS*. Universidad Autónoma de Barcelona, Barcelona.

- Bedoya, Y., Salazar, C., & Muñoz, J. (2013). *IMPLEMENTACIÓN, CONTROL Y MONITOREO DE UN SISTEMA DE SEGURIDAD VEHICULAR POR REDES GSM/GPRS*. Pereira.
- Casanova Matera, L. (2002). *TOPOGRAFÍA PLANA*. Universidad de los Andes , Merida.
- Casco, S. M. (2014). *Raspberry Pi, Arduino y Beaglebone Black Comparación y Aplicaciones*. Universidad Católica Nuestra Señora de la Asunción, Asunción.
- Gavilanes Bayas, A. J. (2018). *SISTEMA ELECTROÓNICO DE DETECCIÓN Y RASTREO DE MASCOTAS*. Universidad Técnica de Ambato, Ambato.
- León Quiñones, A. J. (2005). *EL SERVICIO EUROPEO DE NAVEGACIÓN POR COMPLEMENTO GEOESTACIONARIO -EGNOS-*. Tesis, Universidad de San Carlos de Guatemala, Ciudad de Guatemala.
- López, E. A. (2016). *PERROS CALLEJEROS: UN PROBLEMA DE SALUBRIDAD Y DE SEGURIDAD PARA LOS HABITANTES*. Tamaulipas.
- Quintero Martínez, A., & Rodríguez Guerrero, K. (2004). *TECNOLOGÍA SATELITAL APLICADA A GPS: GLOBAL POSITIONING SYSTEM*. Universidad Tecnológica de Bolívar, Cartagena.
- Romarís Caamaño, C. (2019). *Aplicación Web para la Gestión de Adopción y Localización de Mascotas Perdidas*. Universidad Da Coruña, La Coruña.

ANEXOS

1. Artículos