



UNIVERSIDAD TÉCNICA DE BABAHOYO

**FACULTAD DE ADMINISTRACIÓN, FINANZAS E
INFORMÁTICA**

PROCESO DE TITULACIÓN

MAYO - OCTUBRE 2018

PROPUESTA TECNOLÓGICA DE GRADO O DE FIN DE CARRERA

PROPUESTA TECNOLÓGICA

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN SISTEMAS

TEMA:

**Sistema móvil para controlar la posición en tiempo real del ganado de la Finca
Sartenejal del Cantón Baba.**

EGRESADA:

Lisi Katherine Carpio Santos

TUTOR:

Ing. Miguel Ángel Zúñiga Sánchez

AÑO 2018

DEDICATORIA

A:

Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Mis padres Luis Carpio y Obdulia Santos, por darme la vida, quererme mucho, creer en mí y porque siempre me apoyaron. Padres gracias por darme una carrera para mi futuro, todo esto se los debo a ustedes.

Mis hermanos, Jorge Luis y María Auxiliadora, por estar conmigo y apoyarme siempre, los quiero mucho.

Mis sobrinos, Sergio y Alexander, para que vean en mí un ejemplo a seguir.

Todos mis amigos, que nos apoyamos mutuamente en nuestra formación profesional, compartiendo los buenos y malos momentos.

Todos los Ingenieros, Ing. José Sandoya Villafuerte, Ing. Omar Montecé Moreno, Ing. María Gonzales Valero y Ing. Miguel Zúñiga Sánchez por su gran apoyo y motivación para la culminación de nuestro estudio profesionales y para la elaboración de esta propuesta tecnológica.

Todos aquellos familiares e ingeniero que no recordé al momento de escribir esto. Ustedes saben quiénes son.

AGRADECIMIENTO

A:

Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Mis padres Luis Carpio y Obdulia Santos por apoyarme en todo momento, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo, por ser un excelente ejemplo de vida a seguir.

Mis hermanos por ser parte importante de mi vida y representar la unidad familiar. A Jorge y María por ser un ejemplo de desarrollo profesional a seguir y por llenar mi vida de amor y alegrías cuando más lo he necesitado.

La confianza, apoyo y dedicación de tiempo a mis profesores: Ing. José Sandoya Villafuerte, Ing. Omar Montecé Moreno, Ing. María Gonzales Valero y Ing. Miguel Zúñiga Sánchez. Por haber compartido conmigo sus conocimientos y sobre todo su amistad.

Mis amigos por confiar y creer en mi y haber hecho de mi etapa universitaria un trayecto de vivencias que nunca olvidaré.

ÍNDICE GENERAL

DEDICATORIA	II
AGRADECIMIENTO	III
ÍNDICE GENERAL	IV
ÍNDICES DE FIGURAS	VII
ÍNDICES DE TABLAS	VIII
ÍNDICES DE ILUSTRACIONES	VIII
INTRODUCCIÓN	1
CAPÍTULO I	3
1. Diagnóstico de Necesidades y Requerimientos	3
1.1. Ámbito de Aplicación	3
1.1.1. Descripción del Contexto y Hechos de Interés.	3
1.1.1.1. Localización del problema objeto de estudio.	4
1.1.1.2. Problema general.	5
1.2. Establecimiento de Requerimientos	5
1.2.1. Administrador del Sistema (Administrador de la finca)	5
1.2.2. Ganadero	5
1.2.3. Funcionamiento del Prototipo.....	6
1.2.4. Requerimiento para el Servidor.	7
1.2.5. Requerimiento para el Cliente.	7
1.3. Justificación del Requerimiento a Satisfacer	8
CAPÍTULO II.....	10
2. Desarrollo del Prototipo.....	10
2.1. Definición del Prototipo Tecnológico	10
2.2. Fundamentación teórica del prototipo.....	10
2.2.1. Metodología.	11
2.2.2. Arquitectura.	12

2.2.2.1.	Funciones del cliente.....	12
2.2.2.2.	Funciones del servidor.....	12
2.2.2.3.	Características.....	13
2.2.3.	Tecnología.....	13
2.2.3.1.	Android.....	13
2.2.3.1.1.	Evoluciones en las versiones de Android.....	13
2.2.3.1.2.	Arquitectura de Android.....	14
2.2.3.2.	Android Studio.....	15
2.2.3.2.1.	Ventajas de Android Studio.....	15
2.2.3.2.2.	Desventajas de Android Studio.....	16
2.2.3.3.	Arduino.....	16
2.2.3.3.1.	Ventajas de Arduino.....	17
2.2.3.4.	Internet de las Cosas.....	18
2.2.3.5.	NodeMCu ESP8266.....	18
2.2.3.5.1.	Esquema de los PIN NodeMCu ESP8266.....	19
2.2.3.6.	Sistema de Posicionamiento Global (GPS).....	20
2.2.3.7.	Módulo SIM 808 GPS, GPRS, GSM.....	22
2.2.3.7.1.	Características GSM.....	22
2.2.3.7.2.	Características GPS.....	23
2.2.3.8.	Sensor de Temperatura LM-35.....	24
2.2.3.8.1.	Características.....	25
2.2.3.9.	Firebase.....	25
2.2.3.9.1.	Base de datos Realtime.....	26
2.2.3.9.2.	Autenticación.....	26
2.2.3.9.3.	Almacenamiento.....	27
2.2.3.9.4.	Hosting.....	27
2.2.3.9.5.	Cloud Functions (BETA).....	28

2.2.3.9.6. Laboratorio de tests para Android.	28
2.2.3.9.7. Informes sobre fallos.	29
2.2.3.9.8. Monitoreo de rendimiento.	29
2.2.3.9.9. Notificaciones.	30
2.2.3.9.10. Configuración remota.	30
2.2.3.9.11. Enlaces dinámicos.	31
2.2.3.9.12. Admob.	31
2.3. Objetivos del Prototipo.	32
2.3.1. Objetivos Generales.	32
2.3.2. Objetivos Específicos.	32
2.4. Diseño del Prototipo.	33
2.4.1. Fase de Análisis.	33
2.4.2. Fase de Diseño.	33
2.4.2.1. Diagrama de Caso de uso.	35
2.4.2.2. Caso de Uso del Administrador.	35
2.4.2.3. Caso de Uso del Ganadero.	36
2.4.2.4. Diagrama de Actividad del Administrador.	36
2.4.2.5. Diagrama de Actividad del Ganadero.	37
2.4.2.6. Diagrama de Actividad del GPS.	37
2.4.2.7. Diagrama de Secuencia del Administrador.	38
2.4.2.8. Diagrama de Secuencia del Ganadero.	39
2.4.2.9. Diagrama de Clase.	40
2.4.2.10. Diccionario de Datos.	40
2.4.2.11. Json de la Base de Datos.	41
2.4.3. Fase de Desarrollo.	44
2.4.3.1. Android Studio.	44
2.4.3.2. Arduino.	44

2.4.3.3. Firebase.....	45
2.5. Ejecución y/o Ensamblaje del Prototipo	46
2.5.1. Código.....	55
2.5.1.1. Android.....	55
2.5.1.1.1. Clase Principal.....	55
2.5.1.1.2. Clase Registró Ganado.	58
2.5.1.2. Arduino.....	59
CAPÍTULO III.....	62
3. Evaluación del prototipo.....	62
3.1. Plan de Evaluación	62
3.1.1. Funcionalidad y Facilidad de uso.	62
3.1.2. Estabilidad.....	63
3.1.3. Compatibilidad.....	64
3.1.4. Interoperabilidad	65
3.2. Resultados de la Evaluación.....	66
3.2.1. Análisis de Resultados.	66
CONCLUSIONES	67
RECOMENDACIONES	68
REFERENCIAS.....	69

ÍNDICES DE FIGURAS

<i>Figura 1.</i> Arquitectura de Android.	14
<i>Figura 2.</i> Android Studio.....	15
<i>Figura 3.</i> Arduino	17
<i>Figura 4.</i> Internet de las Cosas	18
<i>Figura 5.</i> NodeMCu.....	19
<i>Figura 6.</i> Esquemas NodeMCu ESP8266	19
<i>Figura 7.</i> El sistema GPS.....	21

<i>Figura 8.</i> La forma en que el sistema GPS.....	21
<i>Figura 9.</i> Módulo SIM808.....	22
<i>Figura 10.</i> Esquema SIM808.....	23
<i>Figura 11.</i> Esquema del sensor LM-35	24
<i>Figura 12.</i> Firebase.....	25
<i>Figura 13.</i> Database.....	26
<i>Figura 14.</i> Authentication.....	26
<i>Figura 15.</i> Storage	27
<i>Figura 16.</i> Cloud Functions.....	28
<i>Figura 17.</i> Informes sobre fallos	29
<i>Figura 18.</i> Monitoreo de rendimiento	29
<i>Figura 19.</i> Notificaciones	30
<i>Figura 20.</i> Configuración remota	30
<i>Figura 21.</i> Enlaces dinámicos.....	31
<i>Figura 22.</i> Admob	31

ÍNDICES DE TABLAS

Tabla 1 Tabla Animal	40
Tabla 2 Tabla Coordenadas	40
Tabla 3 Tabla Temperatura.....	41
Tabla 4 Funcionalidad y Facilidad de uso.	62
Tabla 5 Estabilidad	63
Tabla 6 Compatibilidad.....	64
Tabla 7 Interoperabilidad.....	65
Tabla 8 Resultados de la evaluación.....	66

ÍNDICES DE ILUSTRACIONES

<i>Ilustración 1.</i> Mapa de la Finca Sartenejal	4
<i>Ilustración 2.</i> Requerimiento de Software Servidor	7
<i>Ilustración 3.</i> Requerimiento de Hardware Servidor.....	7
<i>Ilustración 4.</i> Requerimiento de Hardware y Software Cliente.....	7

<i>Ilustración 5.</i> Funcionalidad del Administrador	34
<i>Ilustración 6.</i> Funcionalidad del Ganadero.....	34
<i>Ilustración 7.</i> Diagrama de Caso de Uso General.....	35
<i>Ilustración 8.</i> Caso de Uso del Administrador.....	35
<i>Ilustración 9.</i> Caso de Uso del Ganadero	36
<i>Ilustración 10.</i> Diagrama de Actividad Administrador	36
<i>Ilustración 11.</i> Diagrama de Actividad Ganadero	37
<i>Ilustración 12.</i> Diagrama de Actividad GPS	37
<i>Ilustración 13.</i> Diagrama de Secuencia Administrador.....	38
<i>Ilustración 14.</i> Diagrama de Secuencia Ganadero.....	39
<i>Ilustración 15.</i> Diagrama de Clase.....	40
<i>Ilustración 16.</i> Interfaz de Android	44
<i>Ilustración 17.</i> Interfaz de Arduino	44
<i>Ilustración 18.</i> Interfaz de Firebase	45
<i>Ilustración 19.</i> Inicio de Sesión	46
<i>Ilustración 20.</i> Interfaz inicial.....	47
<i>Ilustración 21.</i> Registro	47
<i>Ilustración 22.</i> Ingreso del Ganado.....	48
<i>Ilustración 23.</i> Consulta de Ganado.....	48
<i>Ilustración 24.</i> Lista del Ganado.....	49
<i>Ilustración 25.</i> Limitación del Terreno	49
<i>Ilustración 26.</i> Formulario para modificar datos	50
<i>Ilustración 27.</i> Ventana para Eliminar.....	50
<i>Ilustración 28.</i> Ventana de Monitoreo	51
<i>Ilustración 29.</i> Monitoreo de la ubicación del Ganado.....	51
<i>Ilustración 30.</i> Monitoreo de la temperatura del ganado	52
<i>Ilustración 31.</i> Gráfico Temperatura del Ganado.....	52
<i>Ilustración 32.</i> Mensaje de Alerta Temperatura	53
<i>Ilustración 33.</i> Alerta Fuera de Recinto.....	53
<i>Ilustración 34.</i> Diagrama de Conexión Dispositivo GPS	54
<i>Ilustración 35.</i> Dispositivo GPS.	54
<i>Ilustración 36.</i> Código Android Principal Parte 1	55
<i>Ilustración 37.</i> Código Android Principal Parte 2	55
<i>Ilustración 38.</i> Código Android Principal Parte 3	56

<i>Ilustración 39.</i> Código Android Principal Parte 4	56
<i>Ilustración 40.</i> Código Android Principal Parte 5	57
<i>Ilustración 41.</i> Código Android Principal Parte 6	57
<i>Ilustración 42.</i> Código registro Ganado Parte 1	58
<i>Ilustración 43.</i> Código registro Ganado Parte 2	58
<i>Ilustración 44.</i> Código de Arduino Parte 1.....	59
<i>Ilustración 45.</i> Código de Arduino Parte 2.....	59
<i>Ilustración 46.</i> Código de Arduino Parte 3.....	60
<i>Ilustración 47.</i> Código de Arduino Parte 4.....	60
<i>Ilustración 48.</i> Código de Arduino Parte 5.....	61
<i>Ilustración 49.</i> Código de Arduino Parte 6.....	61

INTRODUCCIÓN

La ganadería es un sector clave para economía del país, el ganado ofrece diversos derivados prima como el cuero, la carne y la leche. Aunque de esos materiales se deriva subproductos como la mantequilla, queso, carteras de cueros, yogurt, etc. La ganadería en Ecuador se ve afectada por diversos factores como: mercados, precios, clima, delincuencia.

En el año 2011, debido a la ley de no portar armas impuesta por el expresidente Rafael Correa Delgado, los ganaderos estaban indefensos frente a los constante robos. En el año 2015 la Federación Nacional de Ganadero dió a conocer que el sector ganadero podrá portar arma dentro del lugar donde desarrollen sus actividades ganaderas.

En Agosto de 2016 el Ministerio de Agricultura y Ganadería (MAG), Ministerio de Ambiente (MAE), Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO), con el apoyo de Fondo para el Medio Ambiente Mundial(GEF) lanzan el proyecto “Ganadería Climáticamente Inteligente” para revertir la degradación de la tierra y reducir los riesgos de desertificación en las provincias como Guayas, Santa Elena, Manabí, Imbabura, Loja, Morona Santiago y Napo, estará vigente hasta el año 2020. En marzo del presente año lanzaron una plataforma para gestión del conocimiento del proyecto.

Debido a que la tecnología está creciendo a pasos agigantados, en la ganadería se está utilizando la tecnología para mejorar el ámbito productivo en el ganado como en la producción de leche, queso, etc.

No obstante, en la finca Sartenejal del cantón baba, existe inseguridad en los potreros del predio, provocando que la delincuencia cometa sus fechorías, dejando una gran pérdida al ganadero ya que las reses son su sustento diario.

Con este dispositivo GPS se erradicará los robos, pérdida del tiempo al ganadero al momento de buscar una vaca extraviada, debido a que el sistema brindará la ubicación en tiempo real del ganado.

La presente propuesta plantea desarrollar un Sistema Móvil para Controlar la Posición en Tiempo Real del Ganado de la finca Sartenejal del Cantón Baba, para mantener al ganadero informado sobre la ubicación de sus reses.

Debido a que se trabajará con la línea de investigación de Desarrollo de Sistema de la Información, Comunicación y Emprendimientos Empresariales y Tecnológicos. Con la sublínea de Desarrollo de Sistema Informáticos.

CAPÍTULO I

1. Diagnóstico de Necesidades y Requerimientos

1.1. Ámbito de Aplicación

1.1.1. Descripción del Contexto y Hechos de Interés.

En la finca Sartenejal no cuenta con un sistema para la seguridad y control de los potreros del ganado bovino confiable que permita conocer la ubicación en tiempo real del animal cuando salen a pastar, si se pasan de los linderos de la finca o saber su temperatura. También no hay un sistema codificado confiable de cada res, que permita estar al tanto del peso, color, nombre, etc.; de manera fácil, automática y eficaz, llevada la información en tiempo real.

Por el desconocimiento de tecnologías que ayuden a mejorar la seguridad del ganado bovino, en la actualidad los ganaderos de la finca Sartenejal tienen una gran labor en darse cuenta de las reses en los potreros para que no les hagan falta, en caso que se les extraviará una vaca tienen que buscar en cada lote del terreno o pasarse a los terrenos aledaños, así provocando pérdida de tiempo en sus trabajos agrícolas. Deben controlarlas cada cierta hora para que la delincuencia no se aproveché de la ocasión.

Al momento de buscar información de un animal es posible que no la encuentre, debido que el ganadero lleva el registro en hojas, cuadernos, etc., provocando pérdida de información de las reses.

Debido a las reformas agrarias en el Ecuador y al crecimiento en la población de animales bovinos, se hace imprescindible diseñar un sistema móvil para la seguridad y control del ganado, que permita al ganadero tener más confianza al momento que el ganado esté en la sabana y tener la información de sus reses de manera segura.

1.1.1.1. Localización del problema objeto de estudio.

La investigación se localiza geográficamente en la provincia de Los Ríos; cantón Baba; la limitación de lindero de la finca Sartenejal son: por el Norte limita con las fincas Las Brisas, Loma Redonda y Rabo de Buey con 1722.42 metros; por el Sur con la finca Jujanillo con 448.72 metros; por el Este con la finca Rabo de Buey con 1930.56 metros; por el Oeste con las fincas Las Cañas y Las Brisas con 1778.30 metros.



Ilustración 1. Mapa de la Finca Sartenejal
Fuente: Tomada de Google Maps.

1.1.1.2. Problema general.

¿Cómo ayuda el desarrollo de un Sistema Móvil para Controlar la Posición en Tiempo Real del Ganado de la Finca Sartenejal del Cantón Baba, para disminuir la inseguridad en los potreros?

1.2. Establecimiento de Requerimientos

Después de ver analizado el problema que surge en la finca Sartenejal podemos definir los requerimientos necesarios para resolver el inconveniente del predio. El prototipo será una aplicación que funcionará en un entorno móvil que permitirá monitorear, controlar y registrar al ganado bovino en los potreros de la finca. Funciones de los siguientes roles:

1.2.1. Administrador del Sistema (Administrador de la finca)

- Modifica información
- Registra el ganado de la finca
- Búsqueda de información del ganado
- Monitorea al ganado en tiempo real
- Consulta temperatura del ganado en tiempo real
- Elimina información del ganado de la finca

1.2.2. Ganadero

- Monitorea al ganado en tiempo real
- Consulta temperatura del ganado en tiempo real

1.2.3. Funcionamiento del Prototipo

Esta aplicación dará apoyo a las siguientes funciones:

- **Dentro o fuera del recinto.** - El ganadero definirá toda el área de la finca o recintos. El sistema notificará cuando una vaca se salga de los límites de la finca.
- **Alertas por temperatura.** - El dispositivo contará con sensor de temperatura. El sistema notificará cuando el animal tenga alta o baja la temperatura y también se puede divisar la temperatura que tenga la vaca.
- **Monitorear en tiempo real del ganado.** - Con el dispositivo GPS podrás saber cuál es la ubicación actual del animal en cualquier lugar y en cualquier momento. También el ganadero podrá ver su recorrido o actividad durante el día.
- **Registro del animal.** - El sistema móvil permitirá el registro de cada animal y propietario.
- Gestión de actividades de cada animal.
- Implementación de un collar GPS al ganado.

El dispositivo electrónico de geolocalización está compuesto por una tarjeta de Arduino, módulo SIM808 GSM y GPS, en la cual permitirá ver la posición en tiempo real del ganado. Para la realización del sistema móvil es necesario tener un computador con las siguientes especificaciones:

1.2.4. Requerimiento para el Servidor.

Nº	Descripción	Observaciones	Prioridad
1	Sistema Operativo Windows 10	Disponible	Alta
2	Android Studio versión actual	Instalar	Alta
3	Arduino versión 1.8.6	Instalar	Alta
4	Firebase	Web	Alta
7	Java 8	Instalar	Alta
8	Librerías necesarias para Arduino y Android	Instalar	Alta

Ilustración 2. Requerimiento de Software Servidor

Fuente: Elaborado por la Autora.

Nº	Descripción	Observaciones	Prioridad
1	Procesador Intel Core i5	Disponible	Alta
2	Memoria RAM 8GB mínimo	Disponible	Alta
3	Disco Duro 100GB disponible	Disponible	Alta

Ilustración 3. Requerimiento de Hardware Servidor

Fuente: Elaborado por la Autora.

1.2.5. Requerimiento para el Cliente.

Nº	Descripción	Observaciones	Prioridad
1	Android versión de 4.4 en adelante	Disponible	Alta
2	Memoria RAM 512 MB mínimo	Disponible	Alta
3	Almacenamiento 850 MB disponible	Disponible	Alta
4	Acceso a internet	Disponible	Alta

Ilustración 4. Requerimiento de Hardware y Software Cliente

Fuente: Elaborado por la Autora.

1.3. Justificación del Requerimiento a Satisfacer

En la finca Sartenejal del cantón Baba, la inseguridad en los potreros tienen en zozobra a los ganaderos debido a la delincuencia, el pastoreo lo realizan en forma tradicional, al momento que una vaca se extravía el ganadero tiene una gran labor de andar buscando al animal en cada rincón de la finca o irse a terrenos aledaños.

A través de una entrevista con el ganadero de la finca, se conoció que el problema principal que existe es la inseguridad en los potreros y la pérdida del ganado, al momento que el dueño de las reses requiera datos del animal, no la encuentra debido a que los registros lo llevan en hoja o en cuadernos.

Con el desarrollo del sistema móvil se pretende mejorar las falencias que existe en la finca Sartenejal, con el objetivo de resolver el problema de pérdida del ganado, pérdida económica, etc.

La presente propuesta tiene como objetivo principal mantener informado al ganadero sobre la ubicación en tiempo real del ganado, registros, temperatura y control de las reses dentro de la finca.

Mantener el control sobre la ubicación del ganado bovino siempre ha sido una prioridad para los ganaderos. Hoy en día gracias a la tecnología, con la creación de un sistema móvil para el control de las reses, el ganadero podrá saber la ubicación exacta de su ganado en tiempo real, desde cualquier lugar y momento, a través de un dispositivo GPS colocado en cada animal.

El equipo rastreador GPS para ganado incluye un collar inteligente con sensor de temperatura, una batería recargable y una aplicación para controlar la actividad del animal de manera sencilla desde cualquier Smartphone con sistema operativo Android.

La plataforma permite además gestionar la ganadería mediante una ficha independiente para cada animal en la que se indica información como el código, nombre, la fecha de nacimiento, el color y la raza.

La localización y monitorización del animal permite reducir las bajas. El ganadero es avisado si se da cualquier incidencia. Permite ahorrar tiempo y combustible al conocer la ubicación exacta del animal. Mejora la productividad de la explotación, se notificará automáticamente de cualquier incidencia, lo que permite reducir el número de bajas por enfermedad y robos.

CAPÍTULO II

2. Desarrollo del Prototipo

2.1. Definición del Prototipo Tecnológico

Sistema móvil para controlar la posición en tiempo real del ganado de la finca Sartenejal del cantón Baba, será una aplicación móvil, es un prototipo que cumple las necesidades de los ganaderos, las cuales fueron encontradas mediante entrevista directa con el ganadero de la finca.

El prototipo será una aplicación móvil, la cual funcionará a través de un collar GPS que se alojará en cada uno de los animales. El lenguaje de programación en la que se desarrollará la aplicación será en Android Studio.

Robledo (2017) afirma: “Android Studio está basado en IntelliJ IDEA. Este IDE es multiplataforma permitiendo su instalación de forma sencilla tanto en Windows como en Linux o Mac” (p. 9).

La base de datos para almacenar la información será Firebase “es una plataforma para construir aplicaciones móviles y web, se encarga del manejo de la infraestructura permitiendo que el desarrollador se enfoque en otros aspectos de la aplicación” Catalán (2015).

Las herramientas de software que se utilizaran en el desarrollo de prototipo son de libre distribución. El sistema móvil contará con un circuito electrónico, que utilizará la tecnología de hardware libre Arduino y GPS que se encargará en el control de ubicación del ganado.

2.2. Fundamentación teórica del prototipo

En la realización del sistema móvil utilizará la tecnología de desarrollo basado en Android Studio y la tecnología de geolocalización GPS, se aprovecharán herramientas como Arduino para

la programación correspondiente del SIM 808 GPS-GSM-GRPS y técnicas que apruebe la utilización de manera dinámicas y nada compleja el uso por parte del ganadero de la finca Sartenejal.

El objetivo principal que tendrá el sistema móvil para controlar la posición en tiempo real del ganado de la finca Sartenejal del cantón Baba y también beneficiar al ganadero de la región.

2.2.1. Metodología.

La metodología orientada a objeto es la que se utilizará para el desarrollo del prototipo, ya que no comprende los procesos como funciones, sino que arma modelos basados en componentes, es decir, cada componente es independiente del otro. Esto nos permite que el código sea reutilizable.

“La Metodología Orientada a Objeto es la técnica utilizada para modularizar el problema encontrado con el fin de llevar a una solución adecuada para el problema” (Drake, pág. 19).

Entre las fases de esta metodología tenemos:

“**Análisis Orientado a Objetos (AOO).** – Este método explora los requerimientos de las clases y objetos del problema” (Drake, pág. 21).

“**Diseño Orientado a Objetos (DOO).** – Diseñar un programa para identificar los modelos de que se compone, mediante componentes que representan conjuntamente los datos y las operaciones de una” (Drake, pág. 21).

Programación Orientada a Objetos (POO). - Es la manera que tiene para expresarse de las clases de un programa y describe la información y procedimiento de un conjunto de objetos, representa una instancia independiente de la clase cada uno de ellos. (Drake, pág. 21)

2.2.2. Arquitectura.

La arquitectura del prototipo será Cliente/Servidor, es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. (Morales, 2011, pág. 18)

2.2.2.1. Funciones del cliente.

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos. (Morales, 2011, pág. 18)

2.2.2.2. Funciones del servidor.

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo. (Morales, 2011, pág. 18)

2.2.2.3. Características.

- Servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente. (Morales, 2011, pág. 18)

2.2.3. Tecnología.

2.2.3.1. Android.

Según (Porto & María Merino, 2017) android es un sistema operativo que se emplea en dispositivos móviles y electrónicos, como smartphones, tabletas y relojes inteligentes, aunque el software en la actualidad se usa en automóviles, televisores y otras máquinas.

2.2.3.1.1. Evoluciones en las versiones de Android.

- Android 1.1 Petit Four
- Android 1.5 Cupcake
- Android 1.6 Donut
- Android 2.0 / 2.1 Eclair
- Android 2.2 Froyo
- Android 2.3 / 2.3.3 Gingerbread
- Android 3.0 / 3.1 / 3.2 Honeycomb
- Android 4.0 / 4.0.3 IceCreamSandwich

- Android 4.1. / 4.2 / 4.3 Jelly Bean
- Android 4.4 KitKat
- Android 5.0 / 5.1 Lollipop
- Android 6.0 Marshmallow
- Android 7.0 Nougat
- Android 8.0 Oreo

2.2.3.1.2. *Arquitectura de Android.*



Figura 1. Arquitectura de Android.

Fuente: Elaborado por (Cedillo. I., 2012)

<http://androidayuda.com/wp-content/uploads/2012/03/Arquitectura-Android.jpg>

2.2.3.2. *Android Studio.*

Robledo (2017) afirma: “Android Studio está basado en IntelliJ IDEA. Este IDE es multiplataforma permitiendo su instalación de forma sencilla tanto en Windows como en Linux o Mac” (p. 9).

Las funciones de Android son las siguientes:

- Utiliza un sistema Gradle flexible para la compilación
- Un emulador con varias funciones y rápido
- Un entorno accesible para desarrollar aplicaciones móviles
- Con Instant Run nos permite seguir ejecutando la app mientras hacemos cambios sin instalar de nuevo APK
- Gran cantidad de herramientas para un mejor desarrollo
- Es compatibilidad con C++ y NDK
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine. (Developer, 2018)



Figura 2. Android Studio

Fuente: <https://okhosting.com/resources/uploads/2016/05/Android-Studio.png>

2.2.3.2.1. *Ventajas de Android Studio.*

- Una ejecución rápida.
- Permite en tiempo real la ejecución de la app en el emulador.

- Permite compilar la app en nuestro celular móvil.
- Puede utilizar herramientas de tools como Firease.
- Tiene todas las herramientas necesarias para desarrollar cualquier IDE. (AS, 2016)

2.2.3.2.2. *Desventajas de Android Studio.*

- Para su instalación requiere de una buena computado como mínimo de 4G de RAM.
- La batería se descarga rápido. (AS, 2016)

2.2.3.3. *Arduino.*

Es una plataforma de código abierto y hardware libre, formado por una placa con un sencillo microcontrolador y un entorno de desarrollo para realizar sus programas, que posibilita recibir información de varios sensores externos y accionar sobre ello. El hecho de estar soportado sobre código abierto brinda grandes posibilidades a los desarrolladores para crear proyectos que corran sobre ellos, ya que no es necesario pagar la licencia para hacer modificaciones o utilizar el producto asociado a esta. (Pérez Guzmán & González Rivero , 2012, pág. 7)

Arduino se basa en una placa con un microcontrolador Atmel AVR, diversos puertos de entrada/salida y un entorno de desarrollo (IDE) diseñado para facilitar el uso de la electrónica en proyectos multidisciplinarios. Es uno de los buques insignia del hardware libre de bajo costo y su éxito ha sido tal, que podemos ver desde impresoras 3D diseñadas con varios microcontroladores Arduino, hasta medidores de pulso cardíaco. (Pérez Guzmán & González Rivero , 2012, pág. 7)

Los microcontroladores AVR presentan una arquitectura tipo Harvard 8-bit RISC modificada desarrollada por Atmel en 1996 y es una de las primeras familias en implementar

un chip de memoria flash para el almacenamiento del programa en lugar de ROM, EPROM o EEPROM. Los más usados por Arduino dentro de esta familia son el Atmega168, Atmega328, Atmega1280 y ATmega8 por su sencillez y bajo costo. (Pérez Guzmán & González Rivero , 2012, pág. 7)

El lenguaje de programación de Arduino es una implementación de Wiring, una plataforma de computación física, que a su vez se basa en Processing, un entorno de programación multimedia. Las placas pueden ser montadas a mano o adquirirse ensambladas. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. (Pérez & González, 2012, pág. 8)



Figura 3. Arduino

Fuente: <https://img.blogs.es/bloglenovo/wp-content/uploads/2016/06/arduino-cursos-730x497.jpg>

2.2.3.3.1. *Ventajas de Arduino.*

- Las placas Arduino son baratas comparadas con otras plataformas microcontroladoras.
- El software de Arduino es multiplataforma debido a que se ejecutan en sistemas operativos como Windows, Macintosh OSX y GNU/Linux.
- El entorno de programación de arduino es fácil de usar.
- Es un software de código abierto. (Arduino, 2018)

“El ESP8266, diseñado y fabricado por Espressif Systems, contiene todos los elementos cruciales de la computadora moderna: CPU, RAM, redes (wifi) e incluso un moderno sistema operativo y SDK” (Yuan, 2017).

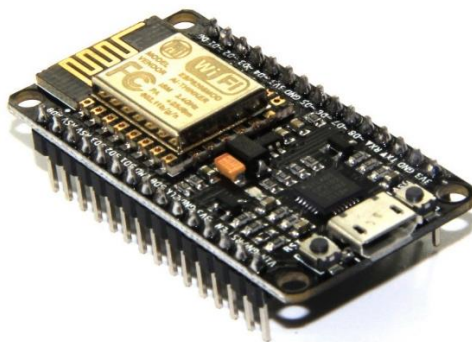


Figura 5. NodeMCu

Fuente: https://files.readme.io/4a75924-2_bp_blogspot_com_-9-vKnWKXmg8_VYITf9HWkEI_AAAAAAAAAEe0_3th9OfSKb5k_s1600_NodeMCU-development-board-for-ESP-12E-from-ESP8266-esp-12E-Lua-IoT-programable-diy-rc-toy_jpg.png

2.2.3.5.1. Esquema de los PIN NodeMCu ESP8266.

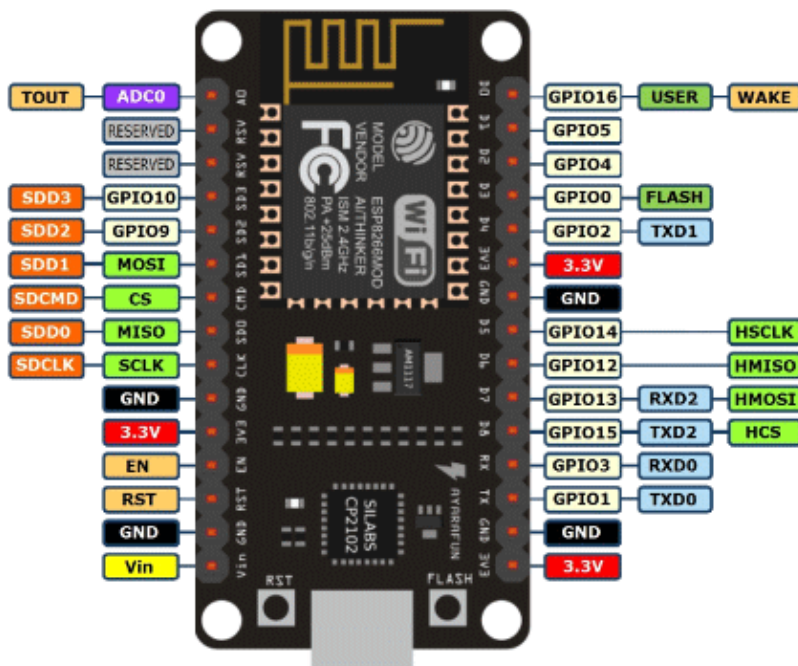


Figura 6. Esquemas NodeMCu ESP8266

Fuente: https://programarfacil.com/wp-content/uploads/2017/03/nodemcu_pineado.png

2.2.3.6. Sistema de Posicionamiento Global (GPS).

Un GPS (por sus siglas en inglés: Global Positioning System) es un sistema de posicionamiento global, a través del cual es posible obtener información muy diversa acerca del objeto que porta la antena (por ejemplo, su altitud, latitud y longitud, su velocidad, la hora UTC 2 a la que está siendo tomada la lectura, la fecha, el número de satélites disponibles para la lectura, etcétera). (Corona, Abarca, & Mares, 2014, pág. 91)

El GPS fue creado por el Departamento de Defensa de Estados Unidos de América en 1970, aunque en un principio no estuvo disponible para todo el público, sino hasta 1995, año en que empezó su comercialización. Este sistema es operado por una serie de satélites llamado constelación. (Corona, Abarca, & Mares, 2014, pág. 91)

Para su operación, el sistema GPS está dividido en tres segmentos:

El segmento de espacio está compuesto por la constelación; en sus inicios, el sistema contaba con 24 satélites, número que se ha incrementado al paso del tiempo, con el fin de mejorar la señal recibida por el GPS.

El segmento de control está compuesto por las oficinas alrededor del mundo encargadas de dar seguimiento y mantenimiento al sistema. Su matriz se encuentra en Colorado, Estados Unidos de América

El segmento de usuario se refiere a cualquier receptor GPS que sea capaz de recibir la señal. (Corona, Abarca, & Mares, 2014, pág. 92)

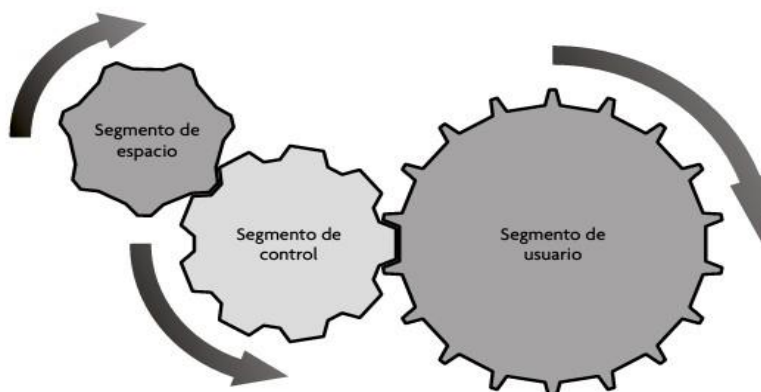


Figura 7. El sistema GPS

Fuente: Elaborado por (Corona, Abarca, & Mares, 2014, pág. 92)

La forma en que el sistema GPS ubica a un usuario es a través de la triangulación, la cual se realiza mediante el uso de tres satélites, donde cada uno de estos determina a qué distancia se encuentra el usuario y traza un círculo con ese valor de radio; el punto en donde se intersecan esos tres radios constituye el punto en donde está ubicado el usuario. (Corona, Abarca, & Mares, 2014, pág. 92)

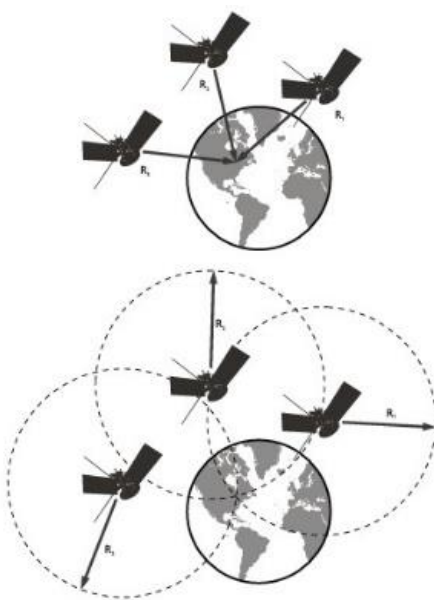


Figura 8. La forma en que el sistema GPS.

Fuente: Elaborado por (Corona, Abarca, & Mares, 2014, pág. 92)

2.2.3.7. *Módulo SIM 808 GPS, GPRS, GSM.*

El SIM808 es un módulo con dos funciones principales. Su diseño se creó a partir del módulo GSM / GPS SIM808 de SIMCOM. Es compatible con GSM / GPRS de cuatro bandas. Combina la tecnología GPS para obtener la posición en latitud y longitud. (URIARTE, 2017)



Figura 9. Módulo SIM808

Fuente: https://http2.mlstatic.com/kit-modulo-shield-sim808-con-antenas-gsm-y-gps-ssdielect-D_NQ_NP_176315-MCO25240780427_122016-F.jpg

Su diseño incorpora un modo de consumo de baja energía y puede conectarse con sistemas de energía a base de baterías de litio. Es compatible con A-GPS. El módulo se controla mediante comandos AT mediante una interfaz de comunicación serial, funciona con una lógica de voltaje de 3.3V y/o 5V. (URIARTE, 2017)

2.2.3.7.1. *Características GSM.*

- El Quad-band 850/900/1800 / 1900MHz su conexión es red mundial GSM con cualquier SIM 2G.
- Envía y recibe mensajes SMS

- Se puede realizar y recibir llamadas
- PWM / Buzzer control motor vibratorio
- Interfaz de comandos AT con detección de “transmisión automática”

2.2.3.7.2. Características GPS.

- Tiene 66 canales de adquisición y 22 de seguimiento
- Contiene GPS L1 C/A code
- Tiene un alcance de 2.5 metros. (Anonimo, s.f.)

El chip que se utilizara en el prototipo es de la red de telefonía Claro.

Esquema

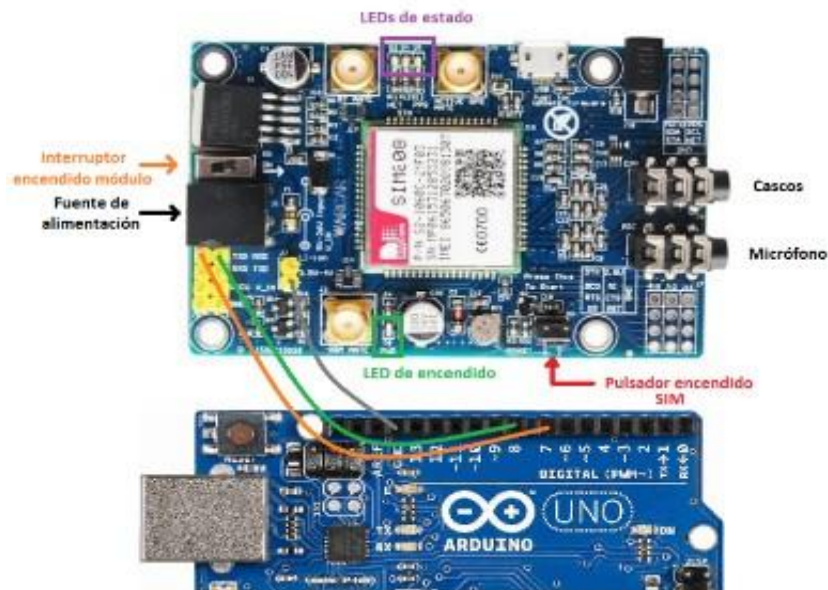


Figura 10. Esquema SIM808

Fuente: <https://www.prometec.net/wp-content/uploads/2017/01/Conexion-Arduino-SIM808-768x560.png>

2.2.3.8. Sensor de Temperatura LM-35.

Según (E-Marmolejo, 2017) “El LM35 es un circuito electrónico sensor que puede medir temperatura. Su salida es analógica, es decir, te proporciona un voltaje proporcional a la temperatura. El sensor tiene un rango desde -55°C a 150°C ”.

Su popularidad se debe a la facilidad con la que se puede medir la temperatura. Incluso no es necesario de un microprocesador o microcontrolador para medir la temperatura. Dado que el sensor LM35 es analógico, basta con medir con un multímetro, el voltaje a salida del sensor. (E-Marmolejo, 2017)

Para convertir el voltaje a la temperatura, el LM35 proporciona 10mV por cada grado centígrado. También cabe señalar que ese sensor se puede usar sin offset, es decir que, si medimos 20mV a la salida, estaremos midiendo 2°C . (E-Marmolejo, 2017)

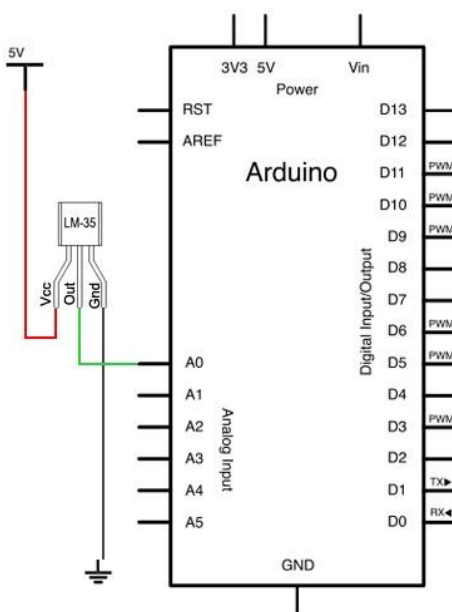


Figura 11. Esquema del sensor LM-35

Fuente: <https://www.luisllamas.es/wp-content/uploads/2015/07/arduino-LM35-esquema-electrico.png>

2.2.3.8.1. Características.

- Calibrado en forma directa en grados Celsius.
- Escala de factor lineal.
- Opera con entre 4 y 30 volts de alimentación.
- Bajo autocalentamiento. (Corona, Abarca, & Mares, 2014, pág. 189)

2.2.3.9. Firebase.

Catalán (2015) afirma: “Es una plataforma de backend para construir aplicaciones móviles y web, se encarga del manejo de la infraestructura permitiendo que el desarrollador se enfoque en otros aspectos de la aplicación.”.

La plataforma está subida en la nube y está disponible para diferentes plataformas como iOS, Android y web. Contiene diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades. Firebase se inició cuando Google la compró en 2014. (Pérez M. , 2016)

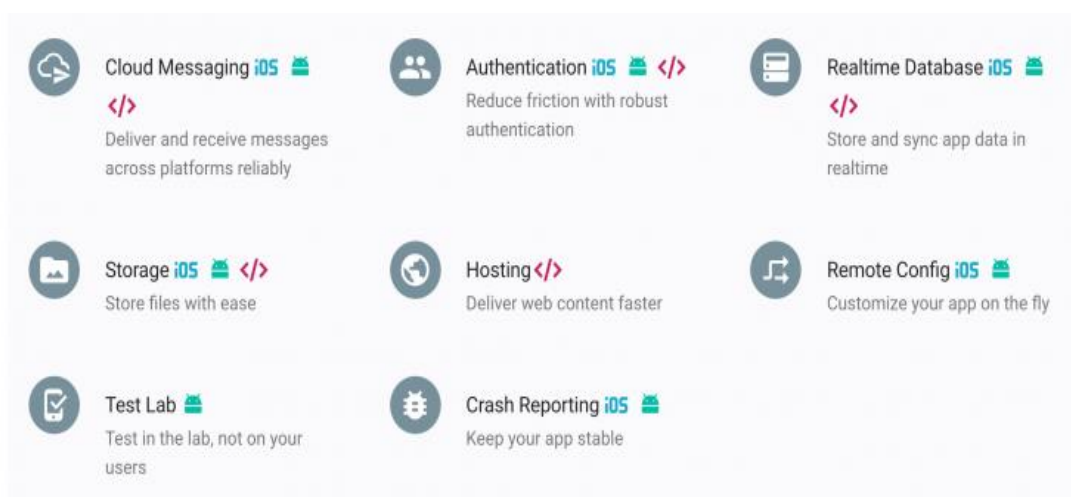


Figura 12. Firebase

Fuente: <https://elandroidelibre.espanol.com/wp-content/uploads/2016/05/Screen-Shot-2016-05-19-at-00.19.39-750x299.png>

2.2.3.9.1. Base de datos Realtime.

Con la base en tiempo real de Firebase podrás guardar todos los datos que requiera tu aplicación. Los datos se almacenan en formato JSON y se pueden agregar reglas para permitir requests con token o solo desde una URL, por ejemplo. (Ruiz, 2017)

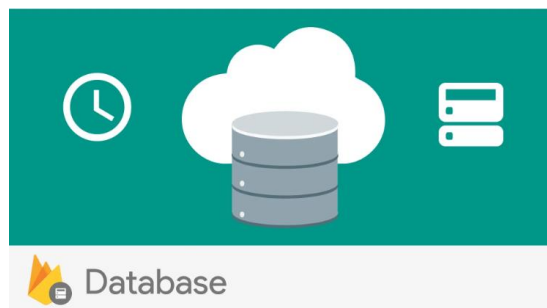


Figura 13. Database

Fuente: https://i.ytimg.com/vi_webp/U5aeM5dvUpA/maxresdefault.webp

2.2.3.9.2. Autenticación.

Es un servicio que nos simplifica el inicio de sesión y la gestión de la misma en nuestra aplicación. Si la usamos en aplicaciones web es extremadamente fácil de configurar, tenemos el proveedor de Google, Correo/Contraseña, Teléfono, Facebook, Twitter, GitHub, Anónimo, sólo es un paso más en el caso de las redes. (Ruiz, 2017)

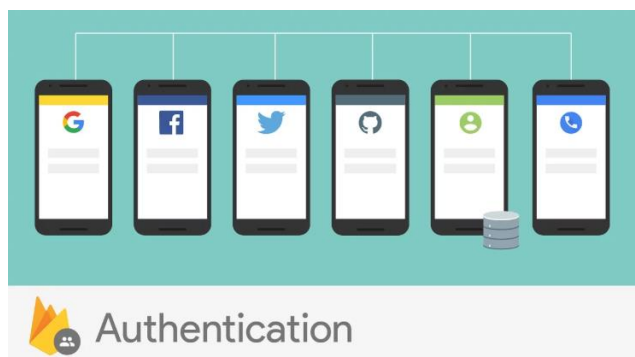


Figura 14. Authentication

Fuente: https://i.ytimg.com/vi_webp/8sGY55yxica/maxresdefault.webp

2.2.3.9.3. Almacenamiento.

Este servicio es muy bueno para aplicaciones que requieran guardar archivos del usuario. También nos sirve si queremos subir estáticos ya que existe un botón desde la interfaz o podemos programar algo. En mi caso lo he usado para subir imágenes desde un formulario y no he tenido ningún tipo de problemas. (Ruiz, 2017)

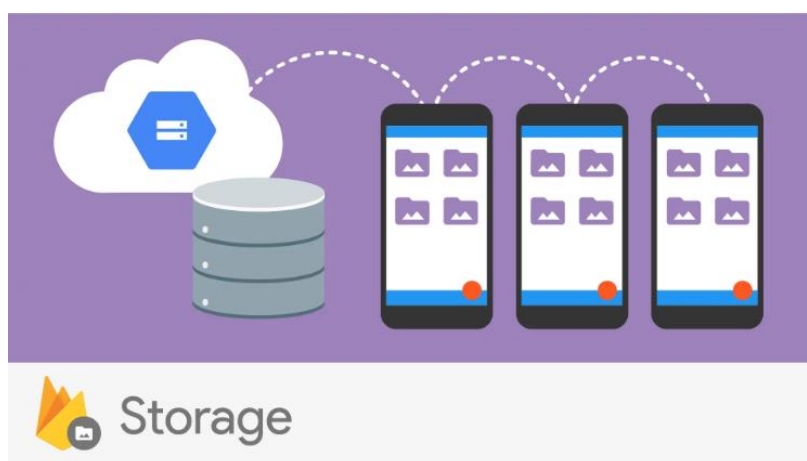


Figura 15. Storage

Fuente: https://i.ytimg.com/vi_webp/_tyjqozrEPY/maxresdefault.webp

2.2.3.9.4. Hosting.

Tiene una colección estáticos, podemos subir una aplicación y esta automáticamente contará con SSL y HTTP2. Si tenemos una app con Angular o Firebase podemos hacer un build desde nuestros ordenadores y subir estos archivos generados y nos funcionarán sin problemas. Si necesitamos un Backend hay que hacer uso de las Cloud Functions. (Ruiz, 2017)

También cabe destacar que al hacer deploy de tu aplicación esta hará parte del CDN de Firebase y se replicará en servidores a lo largo de todo el mundo, disminuyendo el tiempo de transferencia desde estos a el ordenador de tu visitante. (Ruiz, 2017)

2.2.3.9.5. *Cloud Functions (BETA).*

Cloud Functions es un producto bastante reciente de Firebase que tiene como objetivo la transformación de nuestro código del backend en pequeñas piezas del mismo (funciones). Estas funciones son creadas en NodeJS y se suben a la vez que hacemos deploy. (Ruiz, 2017)

Al crearse generan una URL a la que podemos llamar desde AJAX para que se ejecute el código pertinente. Firebase nos provee de unos triggers para comunicarnos con ciertas funciones de sus productos, así como un trigger de HTTP, pero si lo piensas al ser un archivo común de Node podemos añadir cualquier módulo y hacer cosas increíbles. (Ruiz, 2017)

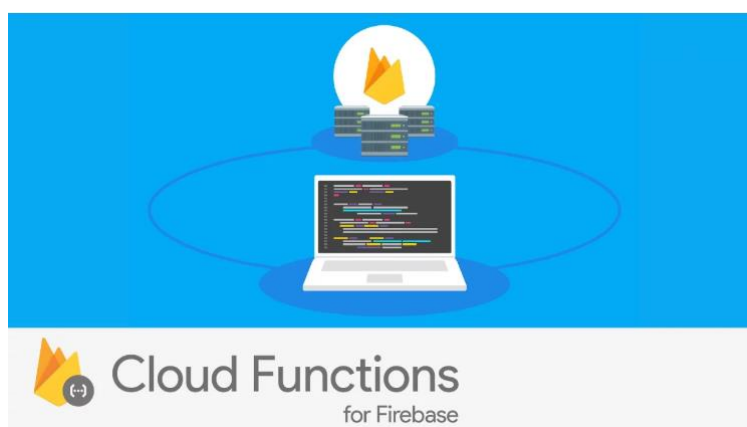


Figura 16. Cloud Functions

Fuente: https://i.ytimg.com/vi_webp/vr0Gfvp5v1A/maxresdefault.webp

2.2.3.9.6. *Laboratorio de tests para Android.*

Los desarrolladores de Android se habrán planteado la encrucijada que produce muchas veces a la hora de probar una aplicación. Existen múltiples modelos de dispositivos con diferentes versiones y diseño del OS, así como diferentes sensores. Este servicio nos ofrece la posibilidad de probar los tests de nuestra aplicación en los entornos que configuremos. (Ruiz, 2017).

2.2.3.9.7. Informes sobre fallos.

“Con esta utilidad podemos detectar los errores que aparezcan en nuestras aplicaciones de iOS y Android, así como poder erradicarlos a tiempo. Esta herramienta dio paso a Crashlytics de Fabric” (Ruiz, 2017).



Figura 17. Informes sobre fallos

Fuente: https://i.ytimg.com/vi_webp/OsWZ3vcInq8/maxresdefault.webp

2.2.3.9.8. Monitoreo de rendimiento.

Esta herramienta te ayudará a medir el rendimiento de tu aplicación y el tiempo que los usuarios pasan durante los diferentes procesos de la misma con el fin de que puedas detectar problemas en el tiempo de carga y prevenirlos antes de que causen cualquier otro tipo de problema o que los usuarios desinstalen tu aplicación. (Ruiz, 2017)

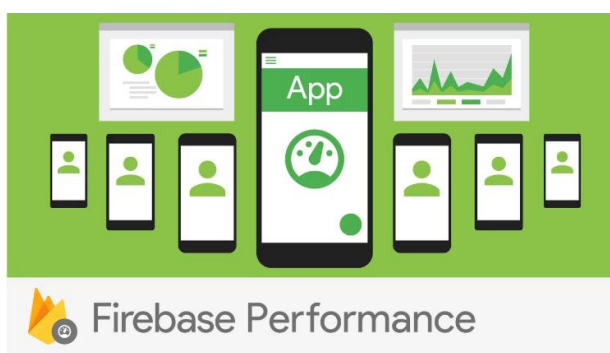


Figura 18. Monitoreo de rendimiento

Fuente: https://i.ytimg.com/vi_webp/0EHSPFvH7vk/maxresdefault.webp

2.2.3.9.9. *Notificaciones.*

“Este servicio, como su nombre nos indica, nos permitirá gestionar el envío de notificaciones a nuestros usuarios con la diferencia de que estas podrán ser programadas acorde a diferentes parámetros” (Ruiz, 2017).



Figura 19. Notificaciones

Fuente: https://i.ytimg.com/vi_webp/R3cbfZKfmkE/maxresdefault.webp

2.2.3.9.10. *Configuración remota.*

“Este añadido nos permitirá básicamente modificar el comportamiento y la apariencia de nuestra aplicación sin que nuestros usuarios tengan que tocar nada” (Ruiz, 2017).

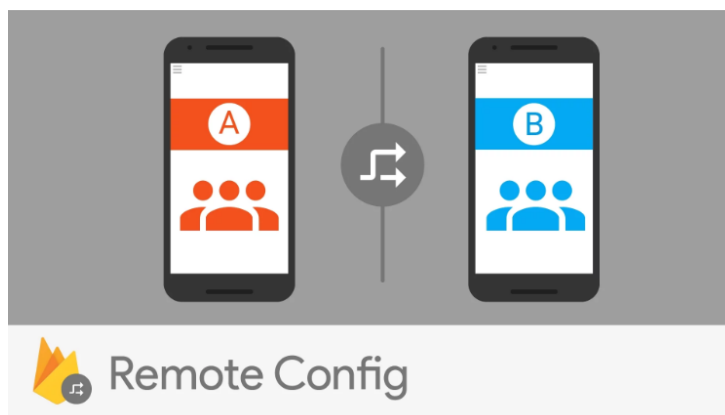


Figura 20. Configuración remota

Fuente: https://i.ytimg.com/vi_webp/_CXXVFPO6f0/maxresdefault.webp

2.2.3.9.11. Enlaces dinámicos.

“Con los enlaces dinámicos podremos asegurarnos que a pesar de las redirecciones que supone el proceso de instalar una aplicación desde un anuncio, lleguemos a la vista deseada, ganando así una experiencia que el usuario agradecerá” (Ruiz, 2017).



Figura 21. Enlaces dinámicos

Fuente: https://i.ytimg.com/vi_webp/LvY1JMcrPF8/maxresdefault.webp

2.2.3.9.12. Admob.

"Mediante este servicio podremos colocar publicidad en distintos lados de nuestra aplicación. Muchas aplicaciones gratuitas como pueden ser algunos juegos llevan implementado este sistema de publicidad o similares" (Ruiz, 2017).



Figura 22. Admob

Fuente: https://www.gstatic.com/mobilesdk/160505_mobilesdk/zerostate/2x/amb.png

2.3. Objetivos del Prototipo

2.3.1. Objetivos Generales.

Desarrollar un sistema móvil para vigilar la geolocalización en tiempo real del ganado de la finca Sartenejal del cantón Baba.

2.3.2. Objetivos Específicos.

- Conocer referencias teóricas sobre las herramientas con las que se desarrollará el sistema móvil, para así brindarles una buena seguridad a los ganaderos de la finca Sartenejal del cantón Baba.
- Ofrecer una herramienta de vigilancia para establecer la ubicación en tiempo real del ganado.
- Verificar la ubicación del ganado, con el uso de Android, Arduino y Firebase.

2.4. Diseño del Prototipo.

El ganadero será encargado de monitorear al ganado en la finca Sartenejal. Para ello se procedió a automatizar la mayoría de los procesos de control en la finca a través de un collar GPS que será incorporado en cada animal con el fin de saber su ubicación, temperatura y en la aplicación llevar un buen control en los registros de cada uno de ellos. Al momento que un animal pasa los límites del terreno el sistema notificará al ganadero que vaca salió del recinto.

Se utilizará la tecnología adecuada para que el uso de la aplicación sea fácil y amigable para que supla las necesidades que tiene el ganadero. Toda la información que manipule el sistema será almacenada en una base de datos.

2.4.1. Fase de Análisis.

El ganadero de la finca tiene como objetivo principal dar más seguridad al ganado de la región, para evitar los robos y pérdida de tiempo al momento que estén buscando una vaca extraviada. El ganadero a cualquiera hora del día puede ver la ubicación de su ganado, ver la temperatura para así ver si no se encuentran enfermos.

Durante la entrevista con el ganadero se pudo deducir que muchas veces se extravían las vacas provocando pérdida de tiempo y los delincuentes abusan de la inseguridad del ganado en el potrero. También hay pérdida de animal al momento de no detectar a tiempo que están enfermos.

2.4.2. Fase de Diseño.

La aplicación móvil que se ha desarrollado además de automatizar los registros del ganado, registro de propietario, también se encarga de notificar la temperatura del ganado y cuando se

pasan los límites de la finca, a través de un collar GPS, lo que facilita al ganadero de monitorearla a cualquier hora sin ir a los potreros.



Ilustración 5. Funcionalidad del Administrador
Fuente: Elaborado por la Autora.

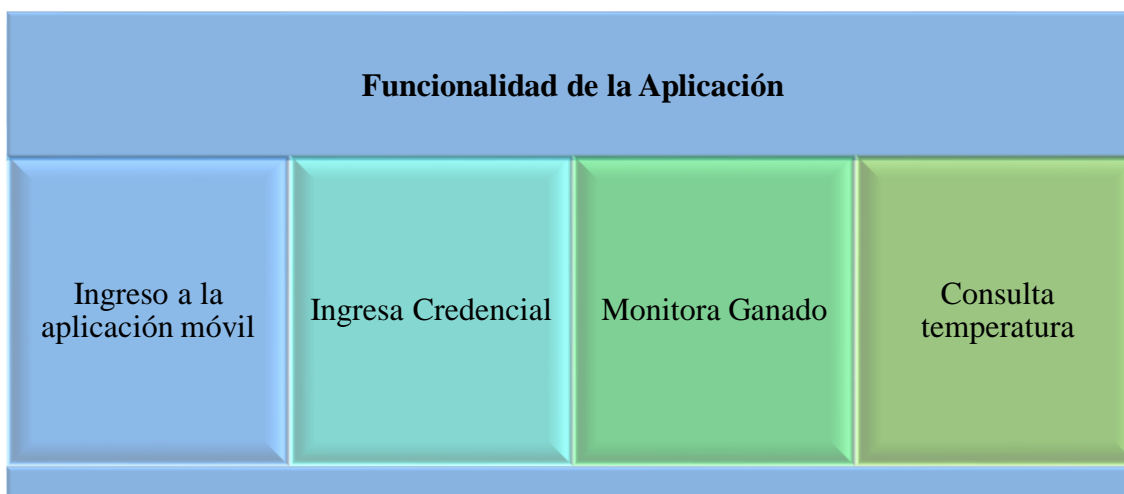


Ilustración 6. Funcionalidad del Ganadero
Fuente: Elaborado por la Autora.

2.4.2.1. Diagrama de Caso de uso.

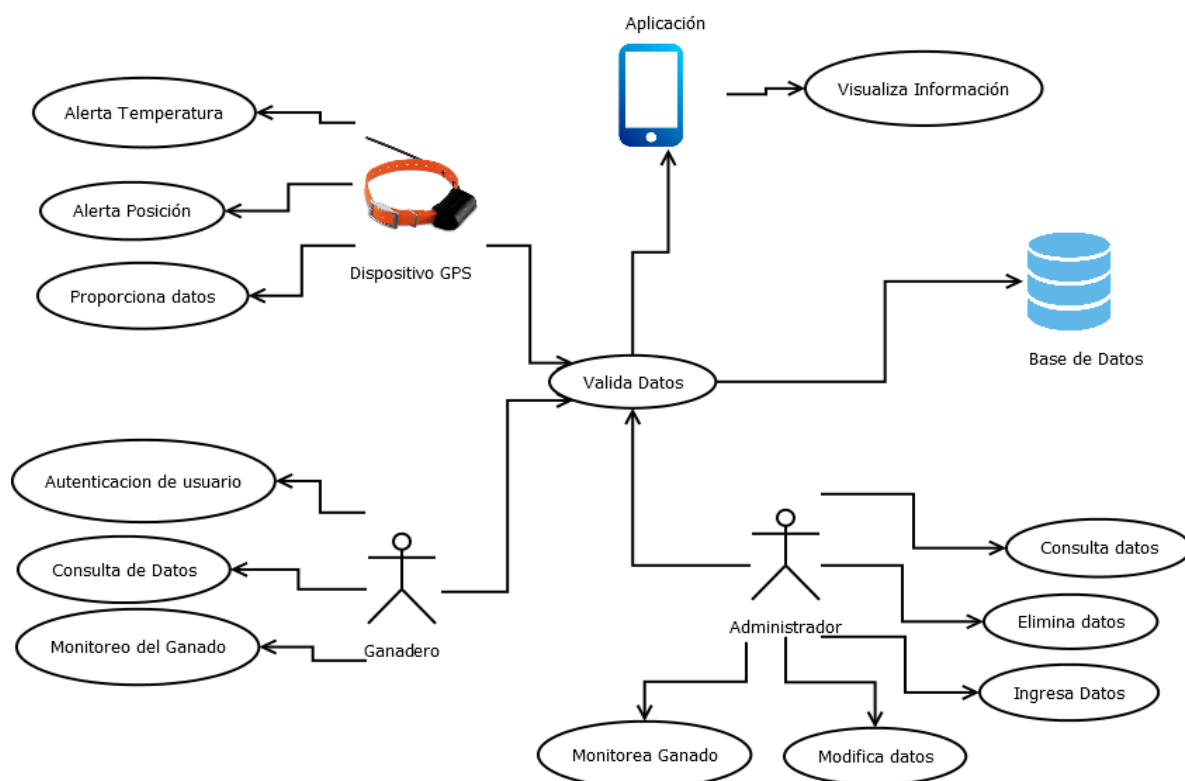


Ilustración 7. Diagrama de Caso de Uso General
Fuente: Elaborado por la Autora.

2.4.2.2. Caso de Uso del Administrador.

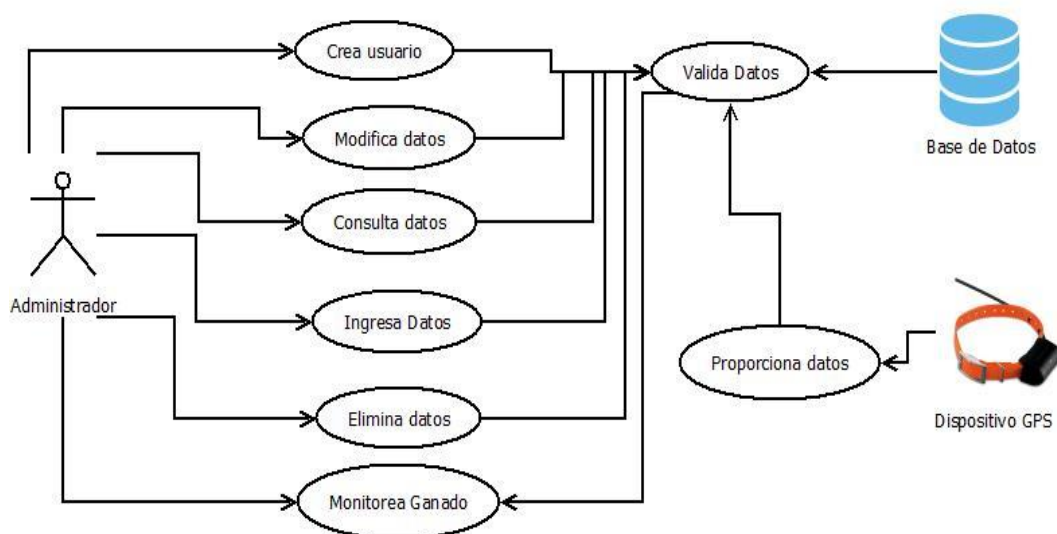


Ilustración 8. Caso de Uso del Administrador
Fuente: Elaborado por la Autora.

2.4.2.3. Caso de Uso del Ganadero.

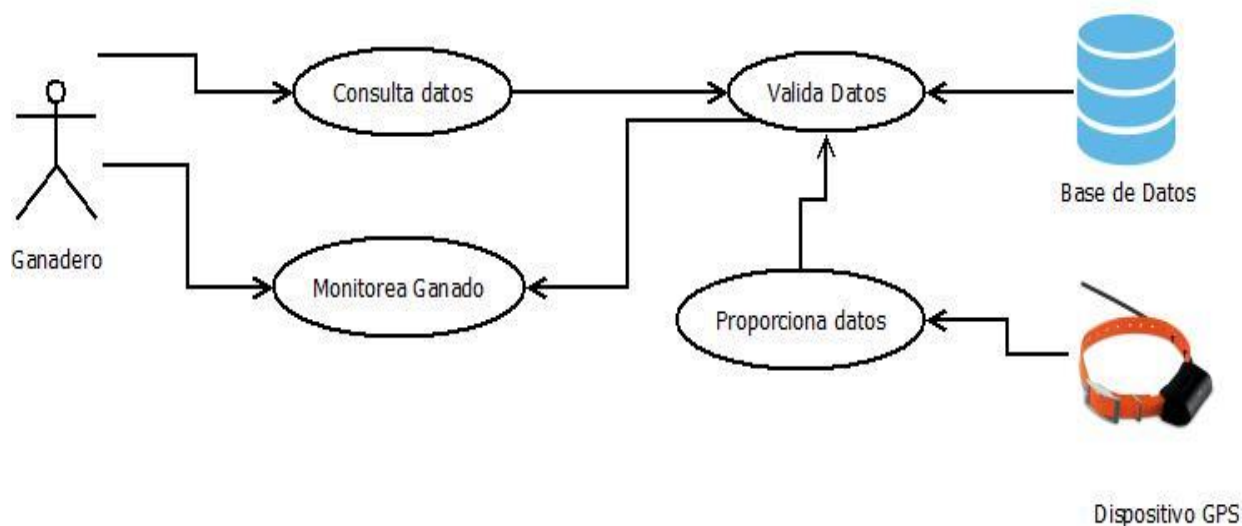


Ilustración 9. Caso de Uso del Ganadero
Fuente: Elaborado por la Autora.

2.4.2.4. Diagrama de Actividad del Administrador.

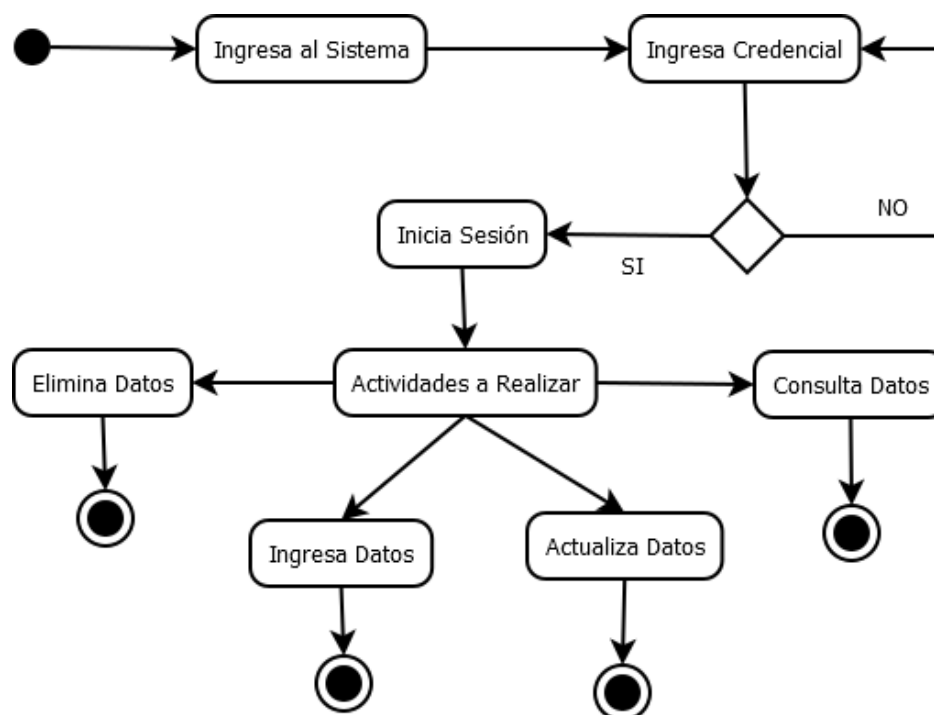


Ilustración 10. Diagrama de Actividad Administrador
Fuente: Elaborado por la Autora.

2.4.2.5. Diagrama de Actividad del Ganadero.

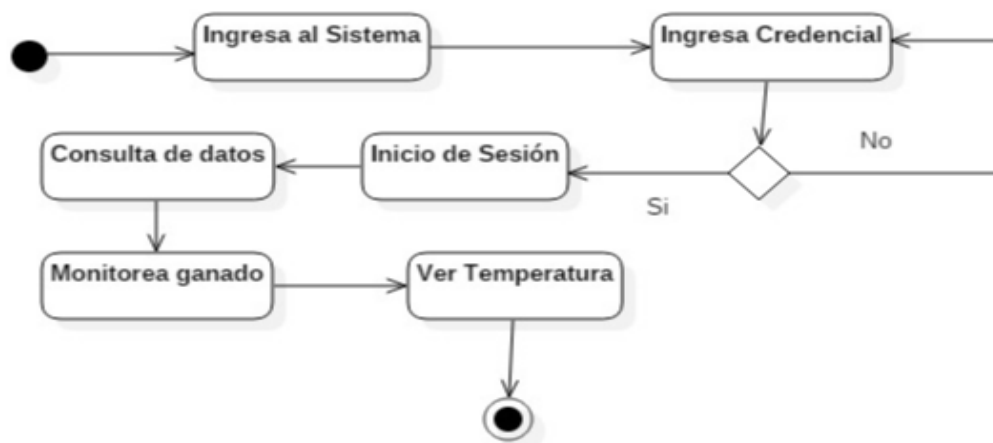


Ilustración 11. Diagrama de Actividad Ganadero
Fuente: Elaborado por la Autora.

2.4.2.6. Diagrama de Actividad del GPS.

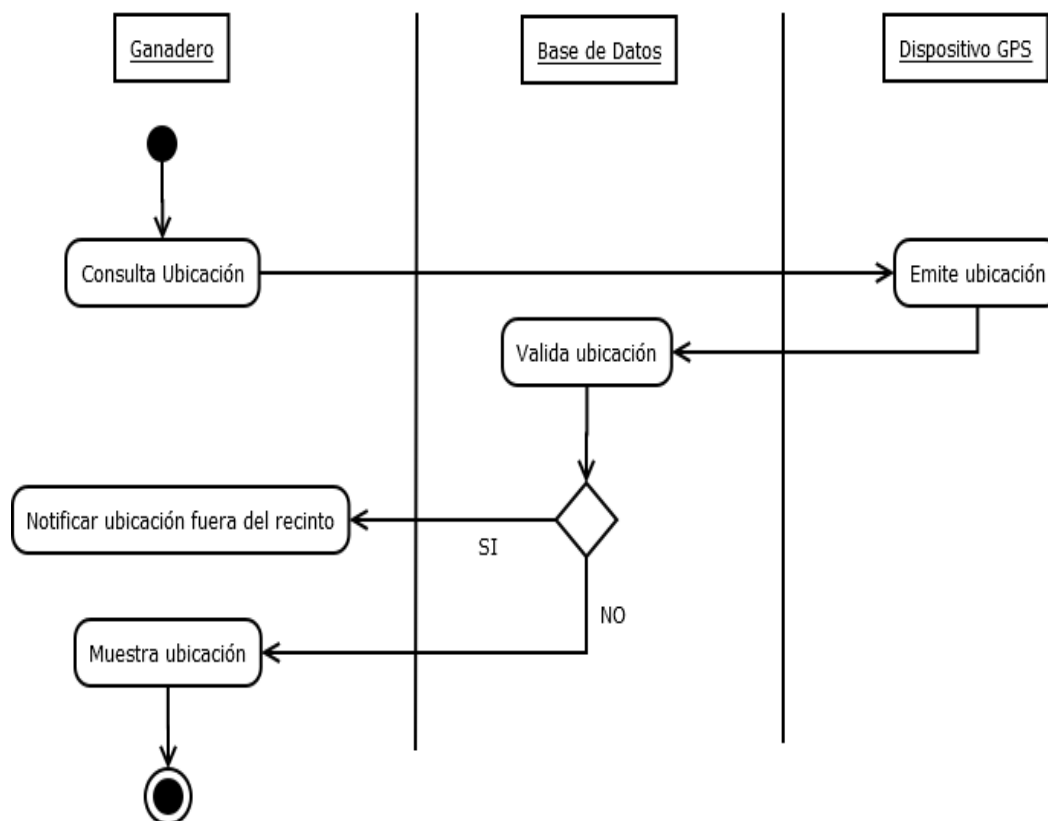


Ilustración 12. Diagrama de Actividad GPS
Fuente: Elaborado por la Autora.

2.4.2.7. Diagrama de Secuencia del Administrador.

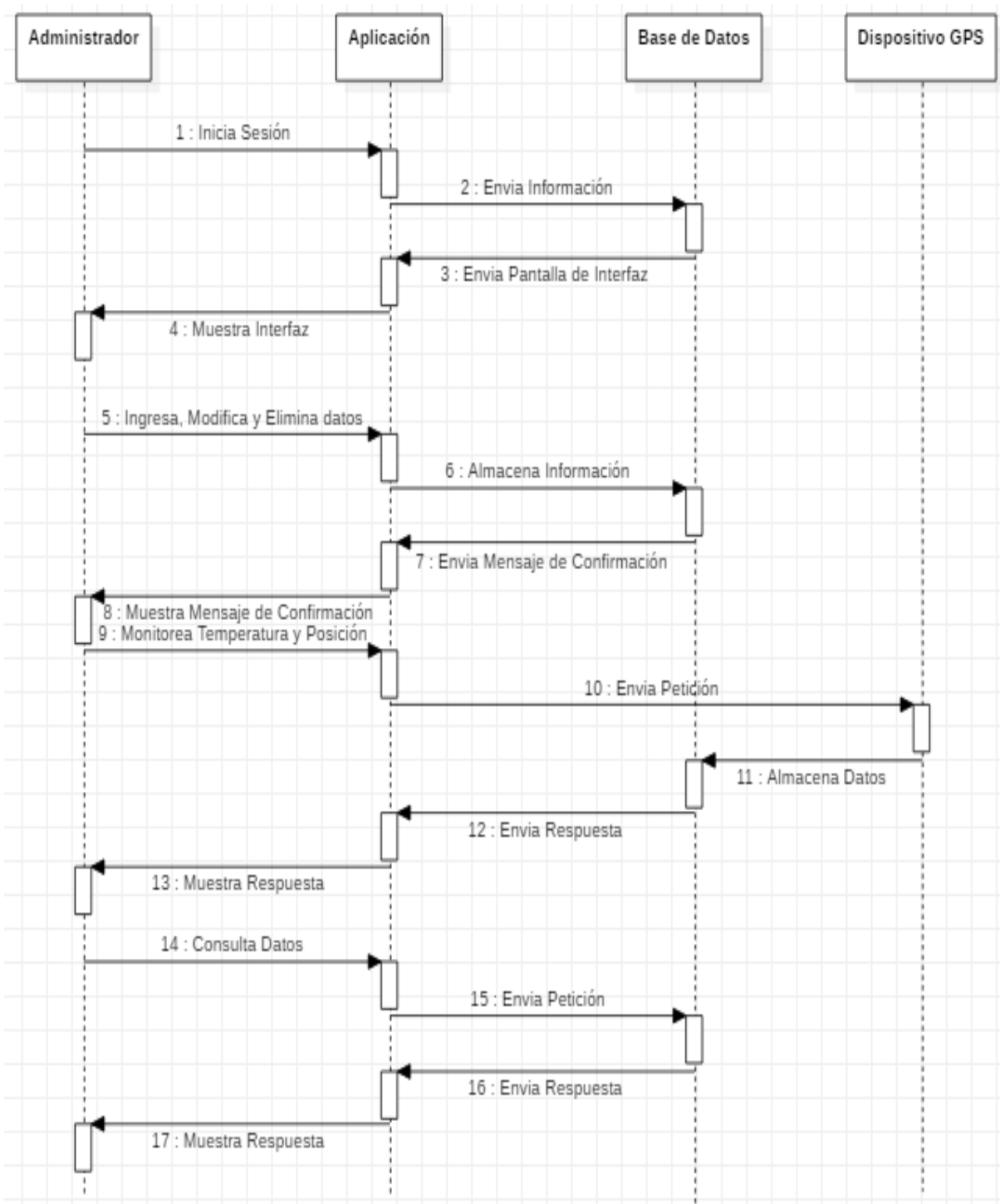


Ilustración 13. Diagrama de Secuencia Administrador
Fuente: Elaborado por la Autora.

2.4.2.8. Diagrama de Secuencia del Ganadero.

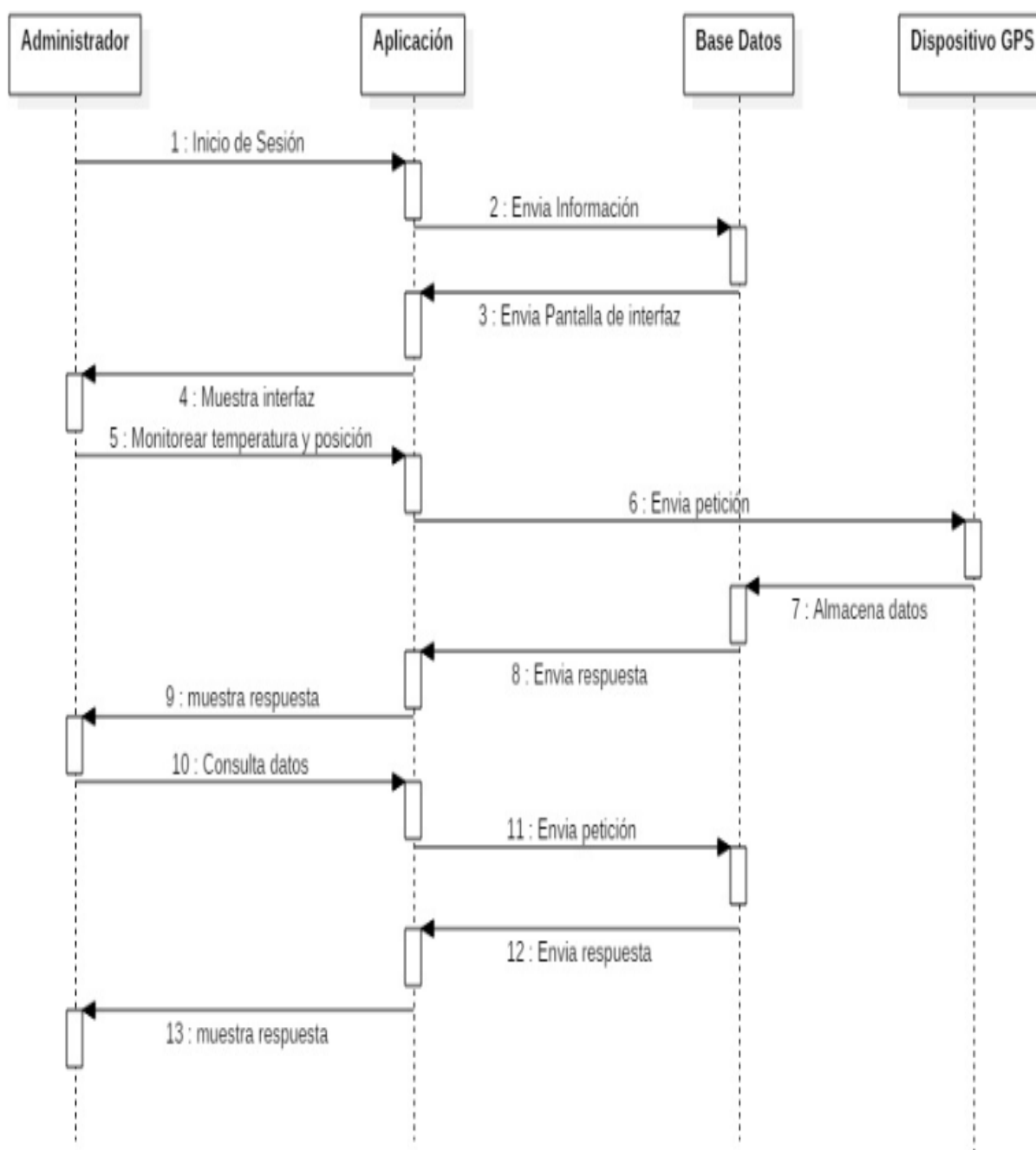


Ilustración 14. Diagrama de Secuencia Ganadero.

Fuente: Elaborado por la Autora.

2.4.2.9. Diagrama de Clase.

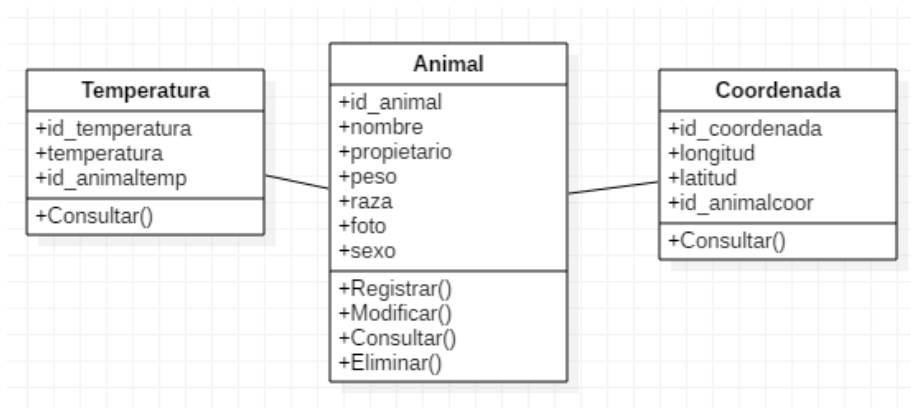


Ilustración 15. Diagrama de Clase
Fuente: Elaborado por la Autora.

2.4.2.10. Diccionario de Datos.

Tabla 1

Tabla Animal

Atributos	Tipo	Longitud	Restricciones	Nulo
Id_animal	Varchar	15	Primary Key	No
Nombre	Varchar	40		No
Peso	Varchar	15		No
Raza	Varchar	30		No
Fecha_nacimiento	Date			No
Foto	Varchar	200		No

Nota: Descripción de la tabla animal. Fuente: Elaborado por la Autora.

Tabla 2

Tabla Coordenadas

Atributos	Tipo	Longitud	Restricciones	Nulo
Id_coordenadas	Int	4	Primary Key	No
Longitud	Varchar	100		No
Latitud	Varchar	100		No

Nota: Descripción de la tabla coordenada. Fuente: Elaborado por la Autora.

Tabla 3

Tabla Temperatura

Atributos	Tipo	Longitud	Restricciones	Nulo
Id_temperatura	Int	4	Primary Key	No
Temperatura	Varchar	30		No

Nota: Descripción de la tabla temperatura. Fuente: Elaborado por la Autora.

2.4.2.11. Json de la Base de Datos.

```
{
  "animal" : {
    "-LMmrqHTGWo7q38zYFmK" : {
      "id_animal" : "0993476176",
      "nombreGanado" : "Pelucona",
      "nombrePropietario" : "Domingo Santos",
      "peso" : "12 Arrobas",
      "raza" : "Lechera",
      "sexo" : "Hembra"
    } },
  "coordenada" : {
    "-LN1KUZasKMyOs_Hn68d" : {
      "latitud" : -1.64256,
      "longitud" : -79.583633
    } },
  "-LN1LRlagFF6FCMow8p-" : {
```

```
"latitud" : -1.642595,  
"longitud" : -79.583634  
},  
"-LN1LtyMBD9pTG3K5TT4" : {  
  "latitud" : -1.64283,  
  "longitud" : -79.582812  
} },  
"sensorTemperatura " : {  
  "-LNI_RAney--J4e8JbVX" : {  
    "temperatura" : 26  
  },  
  "-LNI_SFWSf2AMBoo6UW3" : {  
    "temperatura" : 26  
  },  
  "-LNkwjJvVdiqUOVL8vns" : {  
    "temperatura" : 22  
  },  
  "-LNkw16QO-9aI4BUG21O" : {  
    "temperatura" : 22  
  },  
  "-LNkwrRdyTXreVPDDCB2" : {  
    "temperatura" : 23  
  },  
}
```

```
"-LNkwsEoew-Tubt0ptp9" : {  
  "temperatura" : 22  
},  
"-LNkwt0m-YD0Uq9_uwI8" : {  
  "temperatura" : 23  
},  
"-LNkxKbeBplBx_r837rA" : {  
  "temperatura" : 24  
},  
"-LNkxezw2cy2EAZmrdHI" : {  
  "temperatura" : 23  
},  
"-LNkyG3dRs1D1jE6SU-r" : {  
  "temperatura" : 23  
},  
"-LNkyGrKXPEbE1cIegaM" : {  
  "temperatura" : 24  
},  
"-LNkyHdESZpz7yqx8kjX" : {  
  "temperatura" : 24  
}  
}  
}
```

2.4.3. Fase de Desarrollo.

Las herramientas que se utilizara para desarrollar el prototipo son las siguientes:

2.4.3.1. *Android Studio.*

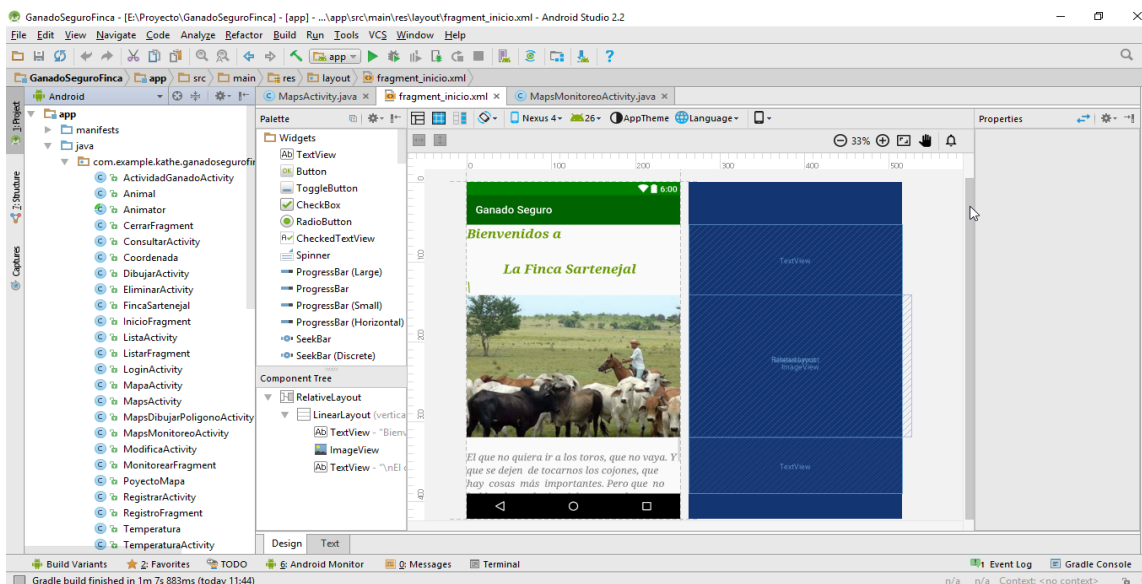


Ilustración 16. Interfaz de Android
Fuente: Elaborado por la Autora.

2.4.3.2. *Arduino.*

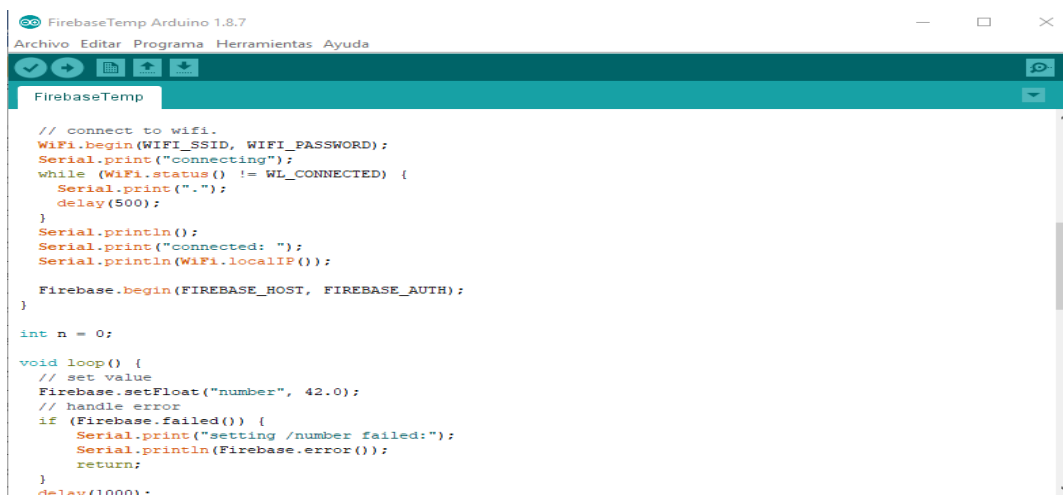


Ilustración 17. Interfaz de Arduino
Fuente: Elaborado por la Autora.

2.4.3.3. *Firestore.*

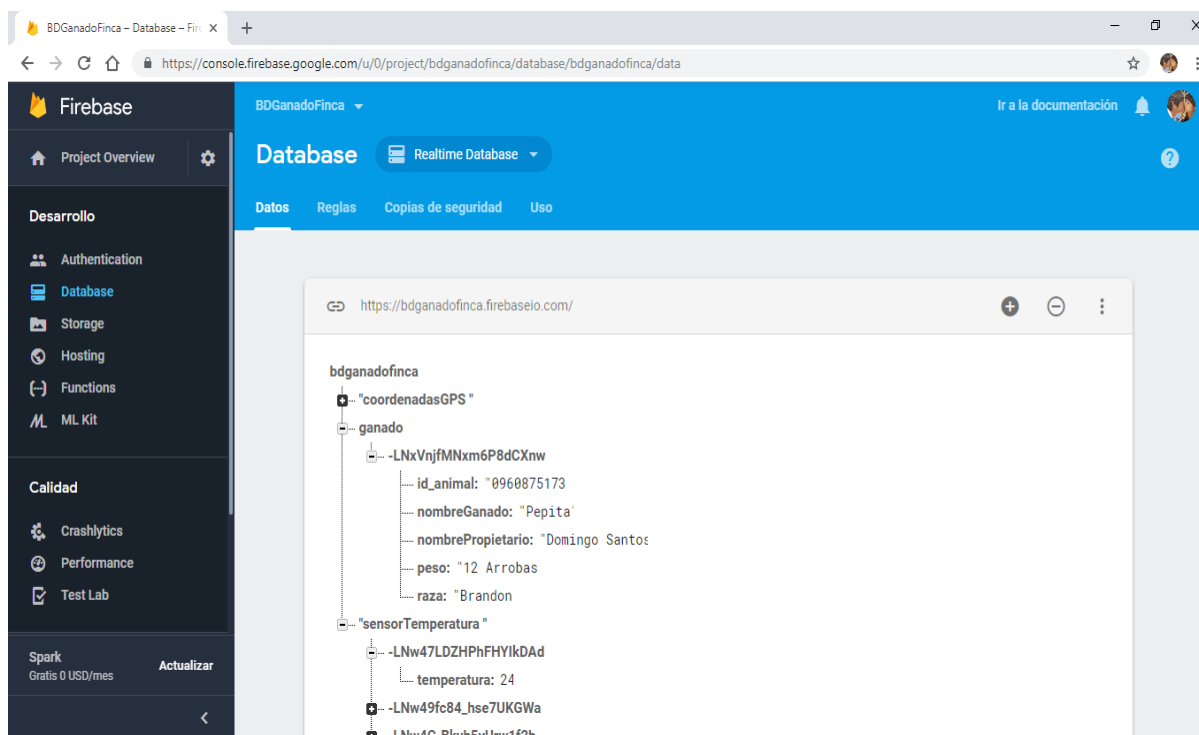


Ilustración 18. Interfaz de Firebase
Fuente: Elaborado por la Autora.

2.5. Ejecución y/o Ensamblaje del Prototipo

El ingreso a la aplicación móvil será a través de una cuenta de Google.

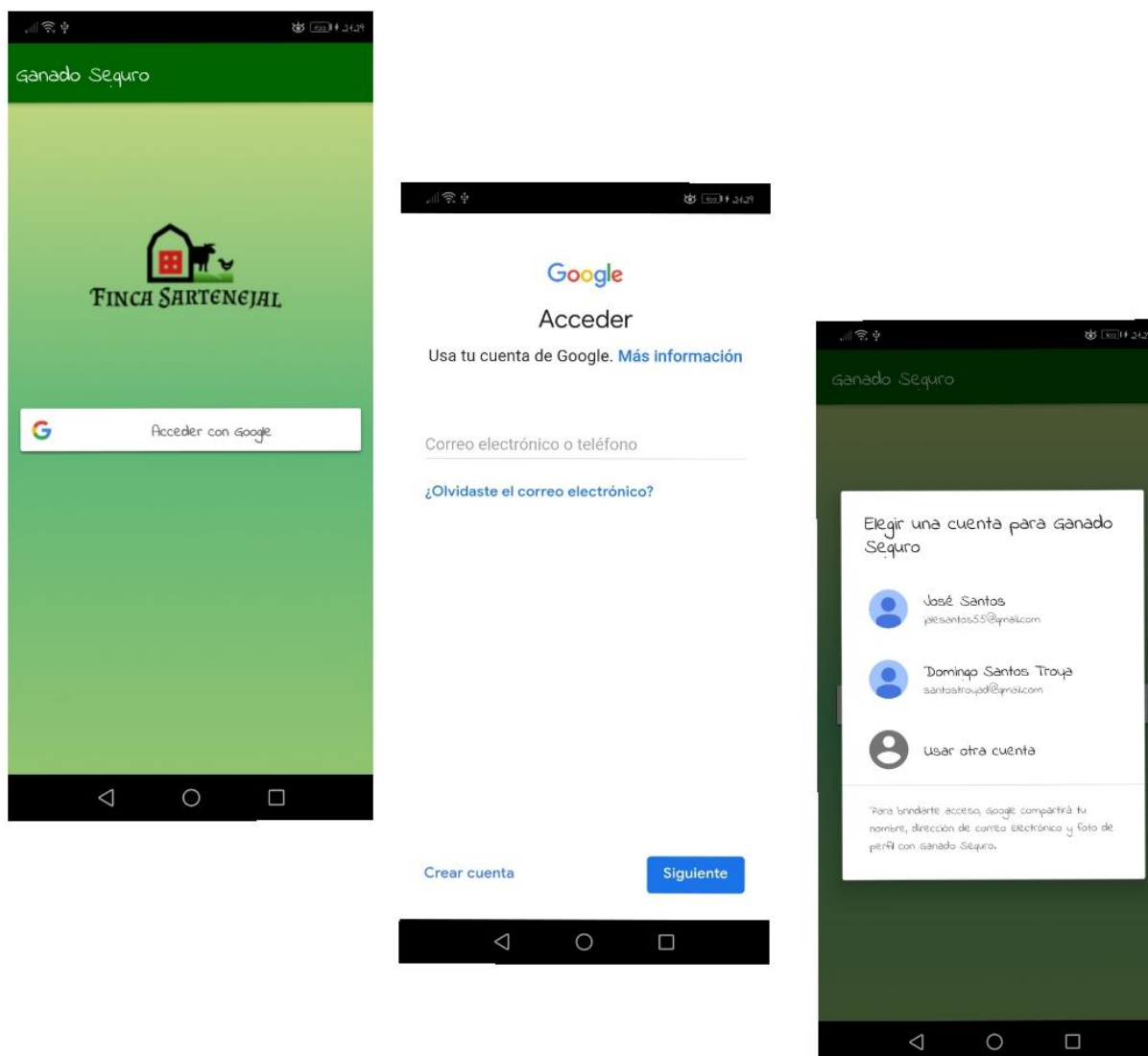


Ilustración 19. Inicio de Sesión
Fuente: Elaborado por la Autora.

En cuanto se haya iniciado sesión muestra la pantalla de inicio.

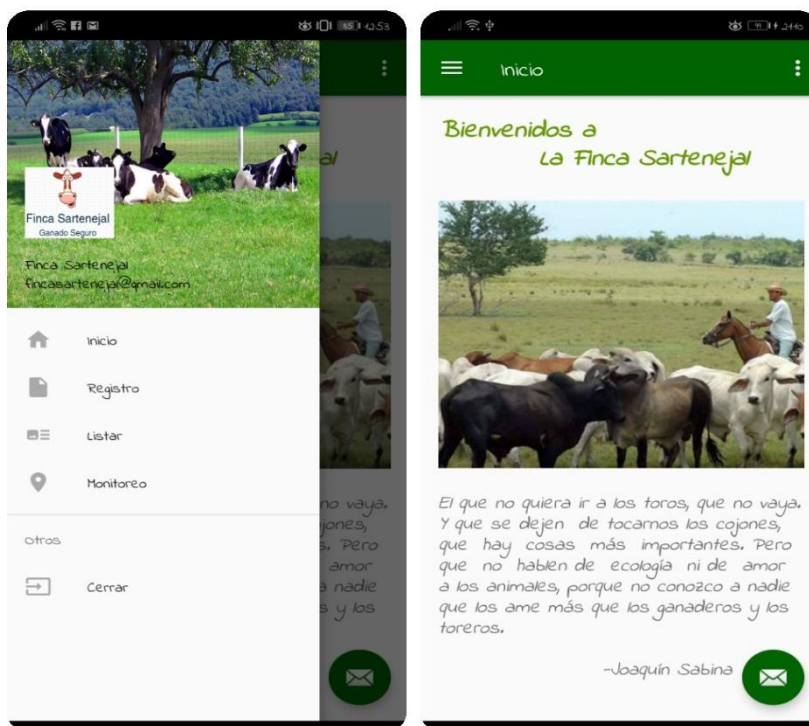


Ilustración 20. Interfaz inicial
Fuente: Elaborado por la Autora.

En la pestaña de Registro tendrá las siguientes funciones: ingreso, consulta, enlista, modifica y elimina al ganado. También delimita el terreno de la finca.



Ilustración 21. Registro
Fuente: Elaborado por la Autora.

Formulario para el ingreso del ganado de la finca Sartenejal.

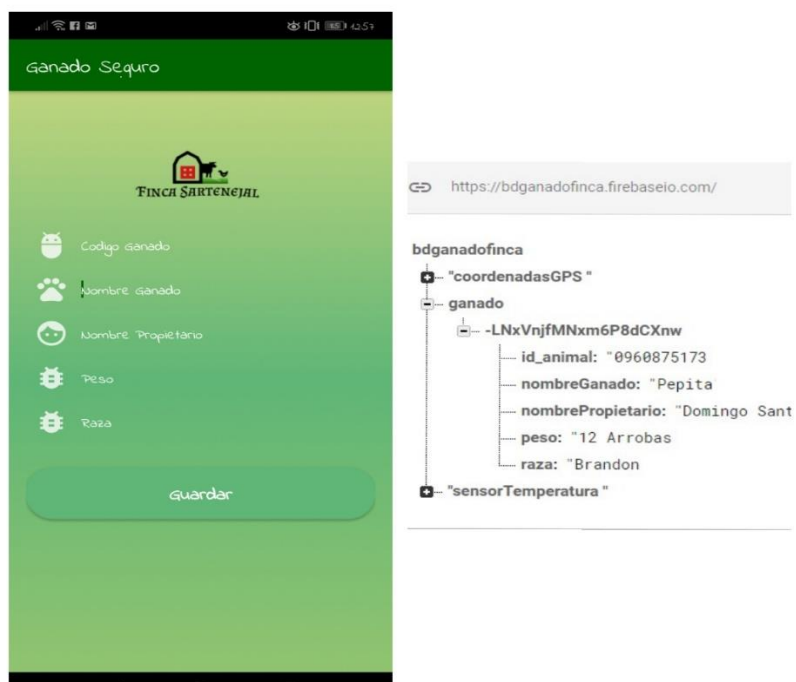


Ilustración 22. Ingreso del Ganado
Fuente: Elaborado por la Autora.

Ventana de consulta del ganado de la finca a través del número incorporado en él collar.



Ilustración 23. Consulta de Ganado
Fuente: Elaborado por la Autora.

Muestra una lista de los nombres del ganado ingresado.

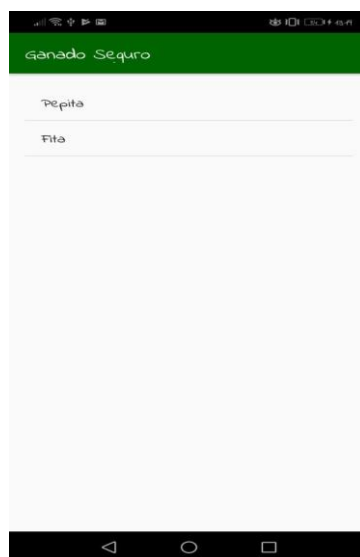


Ilustración 24. Lista del Ganado
Fuente: Elaborado por la Autora

En esta ventana se dibuja el polígono de la limitación del terreno.



Ilustración 25. Limitación del Terreno
Fuente: Elaborado por la Autora.

Formulario para modificar datos del ganado de la finca, primero me hace una consulta por el número del chip incorporado en el collar y me devuelve en cada campo los datos de la res consultada para su debida modificación.



Ilustración 26. Formulario para modificar datos
Fuente: Elaborado por la Autora.

En esta ventana se eliminará el ganado a través del número de chip incorporado en él collar.

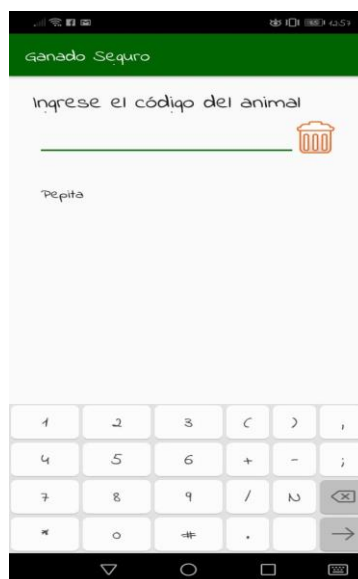


Ilustración 27. Ventana para Eliminar
Fuente: Elaborado por la Autora.

Pestaña de monitoreo contendrá lo siguiente:

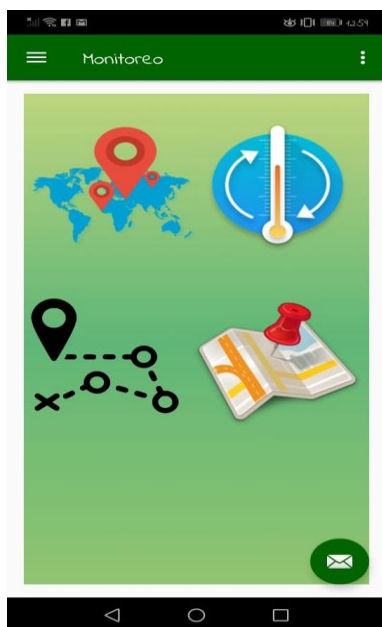


Ilustración 28. Ventana de Monitoreo
Fuente: Elaborado por la Autora.



Sartenejal.

En esta pestaña me permitirá ver la ubicación en tiempo real del ganado de la finca



Ilustración 29. Monitoreo de la ubicación del Ganado.
Fuente: Elaborado por la Autora.



En esta pestaña se enlista la temperatura de la res en tiempo real.

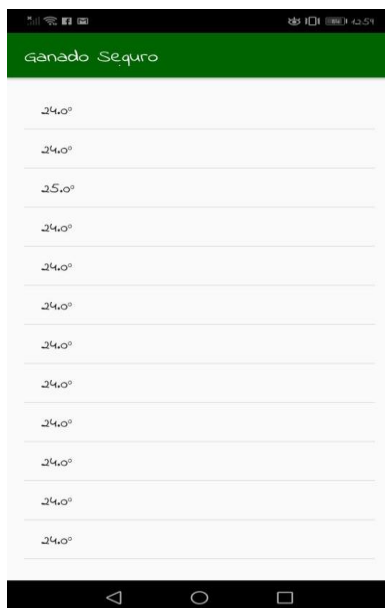


Ilustración 30. Monitoreo de la temperatura del ganado
Fuente: Elaborado por la Autora.

Al momento que la temperatura del ganado baje o suba se notificará al ganadero a través de un mensaje de texto que la temperatura no es adecuada para la res, ya que la temperatura normal es la siguiente:

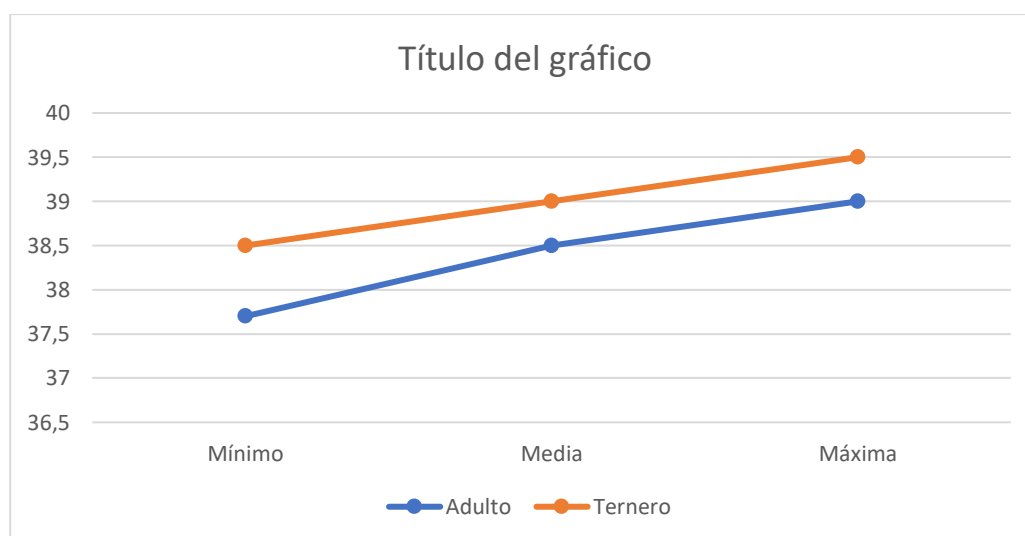


Ilustración 31. Gráfico Temperatura del Ganado.
Fuente: Elaborado por la Autora.

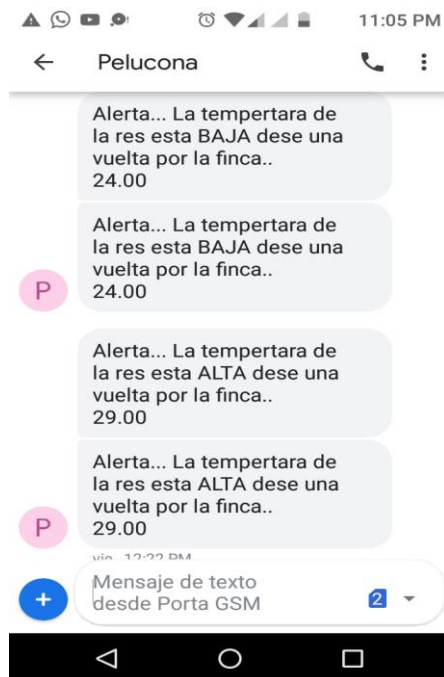


Ilustración 32. Mensaje de Alerta Temperatura
Fuente: Elaborado por la Autora.

Al momento que el ganado se pase de lindero, le llegara una notificación al ganadero alertándole que la res se encuentra fuera de la finca.

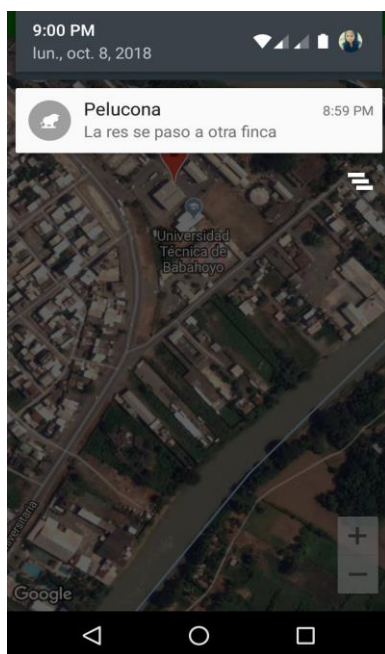


Ilustración 33. Alerta Fuera de Recinto.
Fuente: Elaborado por la Autora.

Diagrama de conexión del circuito del dispositivo GPS.

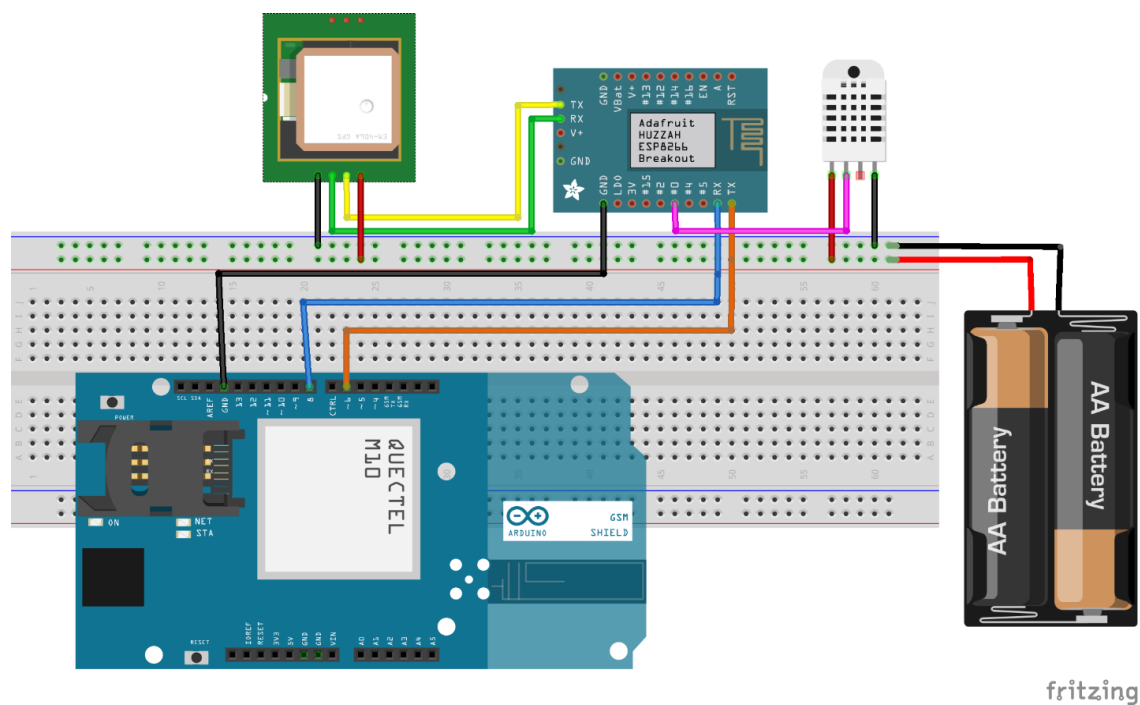


Ilustración 34. Diagrama de Conexión Dispositivo GPS
Fuente: Elaborado por la Autora.



Ilustración 35. Dispositivo GPS.
Fuente: Elaborado por la Autora.

2.5.1. Código.

2.5.1.1. Android.

2.5.1.1.1. Clase Principal.

```

FincaSartenejal.java x
package com.example.tifi.ganadosegurofs;

import ...

public class FincaSartenejal extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener, GoogleApiClient.OnConnectionFailedListener {

    private GoogleApiClient googleApiClient;
    private FirebaseAuth firebaseAuth;
    private FirebaseAuth.AuthStateListener authStateListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_finca_sartenejal);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FragmentManager fm = getSupportFragmentManager();
        fm.beginTransaction().replace(R.id.content_finca_sartenejal, new InicioFragment()).commit();
        getSupportActionBar().setTitle("Inicio");

        GoogleSignInOptions gso = new GoogleSignInOptions
            .Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestEmail().build();

        googleApiClient = new GoogleApiClient.Builder(this)
            .enableAutoManage(this, this)
            .addApi(Auth.GOOGLE_SIGN_IN_API, gso).build();
    }
}

```

Ilustración 36. Código Android Principal Parte 1

Fuente: Elaborado por la Autora.

```

    firebaseAuth = FirebaseAuth.getInstance();

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener((view) -> {
        Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    });

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
    drawer.setDrawerListener(toggle);
    toggle.syncState();

    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}
}

```

Ilustración 37. Código Android Principal Parte 2

Fuente: Elaborado por la Autora.

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.finca_sartenejal, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        firebaseAuth.signOut();
        Auth.GoogleSignInApi.signOut(googleApiClient).setResultCallback((ResultCallback) (status) -> {
            if (status.isSuccess()){
                goLogInScreen();
            }else {
                Toast.makeText(getApplicationContext(), "no se puede cerrar sesión", Toast.LENGTH_SHORT).show();
            }
        });
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

Ilustración 38. Código Android Principal Parte 3
Fuente: Elaborado por la Autora.

```

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    FragmentManager fm = getSupportFragmentManager();

    if (id == R.id.nav_inicio) {
        fm.beginTransaction().replace(R.id.content_finca_sartenejal, new InicioFragment()).commit();
        getSupportActionBar().setTitle("Inicio");
    } else if (id == R.id.nav_registro) {
        fm.beginTransaction().replace(R.id.content_finca_sartenejal, new RegistroFragment()).commit();
        getSupportActionBar().setTitle("Registro");
    } else if (id == R.id.nav_listar) {
        fm.beginTransaction().replace(R.id.content_finca_sartenejal, new ListarFragment()).commit();
        getSupportActionBar().setTitle("Listar");
    } else if (id == R.id.nav_notificacion) {
        fm.beginTransaction().replace(R.id.content_finca_sartenejal, new NotificarFragment()).commit();
        getSupportActionBar().setTitle("Notificaciones");
    }
}

```

Ilustración 39. Código Android Principal Parte 4
Fuente: Elaborado por la Autora.

```

} else if (id == R.id.nav_ubicacion) {

    fm.beginTransaction().replace(R.id.content_finca_sartenejal, new NavegarFragment()).commit();
    getSupportActionBar().setTitle("Monitoreo");

} else if (id == R.id.nav_temperatura) {

    fm.beginTransaction().replace(R.id.content_finca_sartenejal, new TemperaturaFragment()).commit();
    getSupportActionBar().setTitle("Temperatura");

} else if (id == R.id.nav_cerrar) {
    firebaseAuth.signOut();

    Auth.GoogleSignInApi.revokeAccess(googleApiClient).setResultCallback(new ResultCallback<Status>() {
        @Override
        public void onResult(@NonNull Status status) {
            if (status.isSuccess()){
                goLogInScreen();
            } else {
                Toast.makeText(getApplicationContext(), "No hay conexion", Toast.LENGTH_SHORT).show();
            }
        }
    });
}

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
}

```

Ilustración 40. Código Android Principal Parte 5
Fuente: Elaborado por la Autora.

```

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
}

private void goLogInScreen(){

    Intent intent = new Intent(this, LoginActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
}
}

```

Ilustración 41. Código Android Principal Parte 6
Fuente: Elaborado por la Autora.

2.5.1.1.2. Clase Registró Ganado.

```

MainActivity.java x
package com.example.tifi.ganadosegurofs;

import ...

public class MainActivity extends AppCompatActivity {
    EditText codigo, nombreganado, nombrepropietario, peso, raza, sexo;
    Button guardar;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        databaseReference = FirebaseDatabase.getInstance().getReference();
        codigo = (EditText) findViewById(R.id.editCodigoGanado);
        nombreganado = (EditText) findViewById(R.id.editNombreGanado);
        nombrepropietario = (EditText) findViewById(R.id.editNombrePropietario);
        peso = (EditText) findViewById(R.id.editPeso);
        raza = (EditText) findViewById(R.id.editRaza);
        sexo = (EditText) findViewById(R.id.editSexo);
        guardar = (Button) findViewById(R.id.btnGuardar);

        guardar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String codigogana = codigo.getText().toString();
                String nombregana = nombreganado.getText().toString();
                String nombrepropie = nombrepropietario.getText().toString();
                String pesogana = peso.getText().toString();
                String razagana = raza.getText().toString();
                String sexogana = sexo.getText().toString();
            }
        });
    }
}

```

Ilustración 42. Código registro Ganado Parte 1

Fuente: Elaborado por la Autora.

```

        if ((!TextUtils.isEmpty( codigogana ))||(!TextUtils.isEmpty( nombregana )) || (!TextUtils.isEmpty( nombrepropie ))
            || (!TextUtils.isEmpty( pesogana )) || (!TextUtils.isEmpty( razagana )) || (!TextUtils.isEmpty( sexogana ))){
            String id = databaseReference.push().getKey();
            Animal animal = new Animal(codigogana, nombregana, nombrepropie, pesogana, razagana,sexogana);
            databaseReference.child("animal").child(id).setValue(animal);
            Toast.makeText(getApplicationContext(), "Animal Registrado",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Debe llenar todos los campos",
                Toast.LENGTH_SHORT).show();
        }
        codigo.setText("");
        nombreganado.setText("");
        nombrepropietario.setText("");
        peso.setText("");
        raza.setText("");
        sexo.setText("");
    }
}
}
}

```

Ilustración 43. Código registro Ganado Parte 2

Fuente: Elaborado por la Autora.

2.5.1.2. Arduino.



```

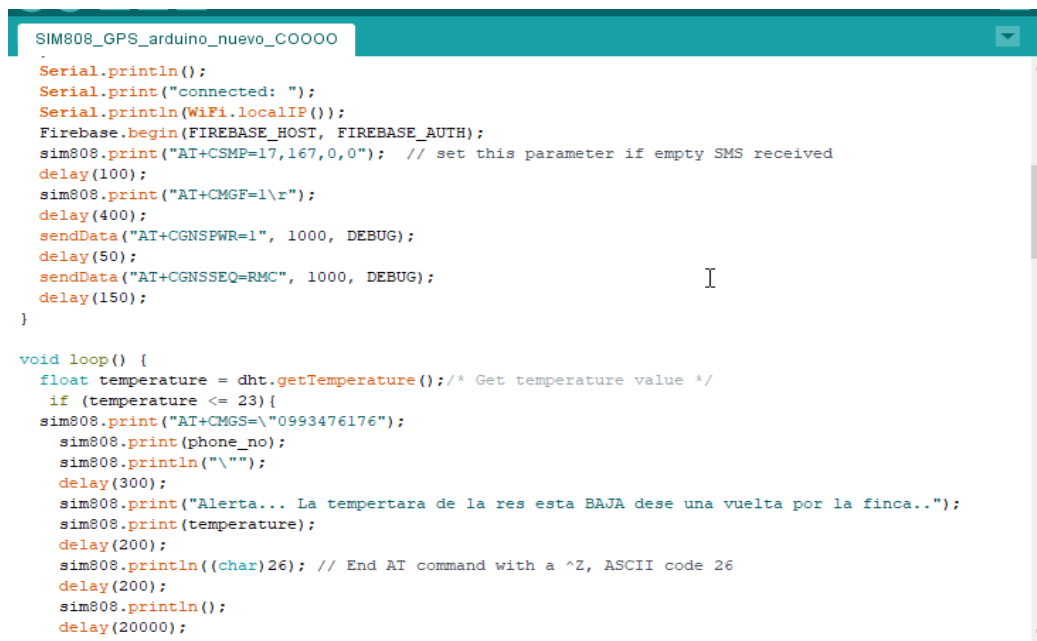
SIM808_GPS_arduino_nuevo_COOOO
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <Firebase.h>
SoftwareSerial sim808(D7, D8);
#include <DHT.h>
DHT dht;
char phone_no[] = "0993476176"; // replace with your phone no.
String data[5];
#define DEBUG true
String state, timegps, latitude, longitude;
#define FIREBASE_HOST "XXXXXXXXX.firebaseio.com"
#define FIREBASE_AUTH "XXXXXXXXX"
#define WIFI_SSID "Tecnologic"
#define WIFI_PASSWORD "XXXXXXXXX"

void setup() {
  sim808.begin(9600);
  Serial.begin(9600);
  delay(50);
  dht.setup(D1);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
}

```

Ilustración 44. Código de Arduino Parte 1.

Fuente: Elaborado por la Autora.



```

SIM808_GPS_arduino_nuevo_COOOO
Serial.println();
Serial.print("connected: ");
Serial.println(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
sim808.print("AT+CSMP=17,167,0,0"); // set this parameter if empty SMS received
delay(100);
sim808.print("AT+CMGF=1\r");
delay(400);
sendData("AT+CGNSPWR=1", 1000, DEBUG);
delay(50);
sendData("AT+CGNSSEQ=RMC", 1000, DEBUG);
delay(150);
}

void loop() {
  float temperature = dht.getTemperature(); /* Get temperature value */
  if (temperature <= 23){
    sim808.print("AT+CMGS=\"0993476176");
    sim808.print(phone_no);
    sim808.println("\n");
    delay(300);
    sim808.print("Alerta... La tempertara de la res esta BAJA dese una vuelta por la finca..");
    sim808.print(temperature);
    delay(200);
    sim808.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(200);
    sim808.println();
    delay(20000);
  }
}

```

Ilustración 45. Código de Arduino Parte 2.

Fuente: Elaborado por la Autora.

```

SIM808_GPS_arduino_nuevo_COOOO
    delay(20000);
    sim808.flush();
} if (temperature >= 30)
sim808.print("AT+CMGS=\"0993476176\"");
sim808.print(phone_no);
sim808.println("\");
delay(300);
sim808.print("Alerta... La tempertara de la res esta ALTA dese una vuelta por la finca..");
sim808.print(temperature);
delay(200);
sim808.println((char)26); // End AT command with a ^Z, ASCII code 26
delay(200);
sim808.println();
delay(20000);
sim808.flush();
}
Serial.println(temperature, 1);
delay(1000);
StaticJsonBuffer < 200 > jsonBuffer;
JsonObject & root1 = jsonBuffer.createObject ();
root1 ["temperatura"] = temperature;
Firebase.push("/sensorTemperatura " , root1);
if (Firebase.failed()) {
    Serial.print("setting /temperatura failed:");
    Serial.println(Firebase.error());
    return;
}
delay(1000);
sendTabData("AT+CGNSINF", 1000, DEBUG);

```

Ilustración 46. Código de Arduino Parte 3
Fuente: Elaborado por la Autora.

```

SIM808_GPS_arduino_nuevo_COOOO
.
delay(1000);
sendTabData("AT+CGNSINF", 1000, DEBUG);
if (state != 0) {
    Serial.println("Latitude : " + latitude);
    Serial.println("Longitude : " + longitudude);
    StaticJsonBuffer < 200 > jsonBuffer;
    JsonObject & root = jsonBuffer.createObject ();
    root ["latitud"] = latitude;
    root ["longitud"] = longitudude;
    Firebase.push("/coordenadasGPS " , root);

    if (Firebase.failed()) {
        Serial.print("setting /coordenada failed:");
        Serial.println(Firebase.error());
        return;
    }
}
delay(1000);
sim808.print("AT+CMGS=\"");
sim808.print(phone_no);
sim808.println("\");
delay(300);
sim808.print("http://maps.google.com/maps?q=loc:");
sim808.print(latitude);
sim808.print(",");
sim808.print (longitude);
delay(200);
sim808.println((char)26); // End AT command with a ^Z, ASCII code 26
delay(200);

```

Ilustración 47. Código de Arduino Parte 4.
Fuente: Elaborado por la Autora.

```

SIM808_GPS_arduino_nuevo_CO000 $
    delay(20000);
    sim808.flush();
} else {
    Serial.println("GPS Initialising...");
}
}

void sendTabData(String command , const int timeout , boolean debug) {
    String data[5] = {"", "", "", "", ""};
    state = data[1];
    timegps = data[2];
    latitude = data[3];
    longitude = data[4];
    sim808.println(command);
    long int time = millis();
    int i = 0;
    while ((time + timeout) > millis()) {
        while (sim808.available()) {
            char c = sim808.read();
            if (c != ',') {
                data[i] += c;
                delay(100);
            } else {
                i++;
            }
        }
        if (i == 5) {
            delay(100);
            goto exitL;
        }
    }
}

```

Ilustración 48. Código de Arduino Parte 5.
Fuente: Elaborado por la Autora.

```

SIM808_GPS_arduino_nuevo_CO000 $
    goto exitL;
}
}
} exitL:
if (debug) {
    state = data[1];
    timegps = data[2];
    latitude = data[3];
    longitude = data[4];
}
}
String sendData (String command , const int timeout , boolean debug) {
    String response = "";
    sim808.println(command);
    long int time = millis();
    int i = 0;

    while ( (time + timeout) > millis()) {
        while (sim808.available()) {
            char c = sim808.read();
            response += c;
        }
    }
    if (debug) {
        Serial.print(response);
    }
    return response;
}
}

```

Ilustración 49. Código de Arduino Parte 6.
Fuente: Elaborado por la Autora.

CAPÍTULO III

3. Evaluación del prototipo

3.1. Plan de Evaluación

3.1.1. Funcionalidad y Facilidad de uso.

Tabla 4

Funcionalidad y Facilidad de uso.

Destinatario	Docente Guía	Fecha	Actividades Desarrolladas	Observación	Cambios en el Sistema
Usuario	Ing. Miguel Ángel Zúñiga Sánchez	03/09/2018	Prueba del uso del prototipo	Funciona correctamente	Ninguno
		04/09/2018		Inicio de Sesión correctamente	Ninguno
		05/09/2018		Visualización de la información	Tamaño de la fuente y color de la interfaz
		06/09/2018	Prueba del funcionamiento de la aplicación móvil	Ingreso de Información correctamente	Ninguno
		07/09/2018		Accede a la información correctamente	Ninguno
		10/09/2018		Muestra información según lo esperado	Ninguno
		11/09/2018	Prueba de la aplicación móvil en diferentes dispositivos	La aplicación funciona como se los esperaba	Ninguno

Nota: Descripción de la funcionalidad y facilidad del uso del prototipo. Fuente: Elaborado por la Autora.

3.1.2. Estabilidad.

Tabla 5

Estabilidad

Destinatario	Docente Guía	Fecha	Actividades Desarrolladas	Observación	Cambios en el Sistema
Base de Datos	Ing. Miguel Ángel Zúñiga Sánchez	12/09/2018	Prueba de persistencia de datos	Mediante Firebase los datos se muestra en tiempo real	Ninguno
Aplicación Móvil		13/09/2018	Prueba de instalación de la aplicación	La aplicación no ejecuta correctamente en sistemas operativo Android con versiones menores a 4.4	Ninguno
Aplicación Móvil		14/09/2018	Prueba de funcionalidad	La aplicación funcionó según lo esperado	Ninguno
Circuito		17/09/2018	Prueba del collar GPS	El dispositivo funcionó según lo esperado	Ninguno

Nota: Descripción de estabilidad del prototipo. Fuente: Elaborado por la Autora.

3.1.3. Compatibilidad.

Tabla 6

Compatibilidad

Destinatario	Docente Guía	Fecha	Actividades Desarrolladas	Observación	Cambios en el Sistema
Aplicación Móvil	Ing. Miguel Ángel Zúñiga Sánchez	18/09/2018	Prueba de velocidad	La velocidad de ejecución de la aplicación móvil depende de la versión del sistema operativo Android	Ninguno
Dispositivo GPS		19/09/2018	Prueba de envío de mensaje	El dispositivo no emitía mensaje	El chip incorporado en el SIM808 debe tener saldo
		20/09/2018		El dispositivo funcionó según lo esperado	Ninguno

Nota: Descripción de Compatibilidad del prototipo. Fuente: Elaborado por la Autora.

3.1.4. Interoperabilidad

Tabla 7

Interoperabilidad

Destinatario	Docente Guía	Fecha	Actividades Desarrolladas	Observación	Cambios en el Sistema
Sistema		21/09/2018	Test de intercambio de datos entre la base de datos y la aplicación móvil	Los datos se muestran en tiempo real	Ninguno
Dispositivo GPS	Ing. Miguel Ángel Zúñiga Sánchez	22/09/2018	Prueba de funcionalidad	Emite datos correctamente	Ninguno
Aplicación Móvil		23/09/2018	Prueba de funcionalidad	La aplicación funcionó correctamente	Ninguno
Sistema		24/09/2018	Prueba de funcionamiento	El sistema funcionó según lo esperado	Ninguno

Nota: Descripción de Interoperabilidad del prototipo. Fuente: Elaborado por la Autora.

3.2. Resultados de la Evaluación

Tabla 8

Resultados de la evaluación

Plan de Evaluación	Aceptación	Rechazo
Funcionalidad y factibilidad de uso	93%	7%
Estabilidad	95%	5%
Compatibilidad	92%	8%
Interoperabilidad	97%	3%
Resultados de la evaluación	94.25%	5.75%

Nota: Descripción del resultado de evaluación del prototipo. Fuente: Elaborado por la Autora.

3.2.1. Análisis de Resultados.

En la etapa de evaluación se realizó diversas pruebas de funcionamiento del sistema móvil y del dispositivo GPS en tiempo real, dando resultados satisfactorios al final. Donde se mostró ciertas inconsistencias que fueron solucionadas para una buena funcionalidad del prototipo.

Con un resultado de 92% de aceptación en la etapa de compatibilidad, donde se mejoró el diseño de la interfaz para una buena visualización al ganadero y se cambiaron código en el dispositivo GPS para su correcto funcionamiento, en la etapa de interoperabilidad obtuvo un 95% de aceptación demostrando que funcionó según lo esperado.

En los resultados obtenidos de funcionalidad en el prototipo fue de 94.5% de aceptación, con un índice de rechazo del 5.75%, demostrando el cumplimiento de los objetivos planteado para mejorar la seguridad en la finca Sartenejal.

CONCLUSIONES

Durante este periodo de desarrollo del prototipo podemos concluir:

- Una vez desarrollado el sistema se ha podido determinar que el módulo SIM808 tiene un margen de conmutación de 1 a 3m, es un margen mínimo para la correcta ubicación del ganado.
- El sistema móvil puede ser utilizado desde cualquier smartphone que cuente con el sistema operativo Android y posea el servicio de internet.
- El sistema ofrecerá a los ganaderos de la región, la ubicación exacta, fuera del recinto y temperatura del ganado en tiempo real.
- Una vez evaluado el prototipo por parte de los ganaderos, concluyeron que el mismo tuvo una acogida positiva de parte de ellos.
- Se pudo determinar que el uso de la tecnología puede llevar a obtener mejores resultados de parte de los ganaderos, se contará con la ayuda de mensaje de texto, llamadas y geolocalización en caso de existir inconveniente con las reses.

RECOMENDACIONES

Para finalizar con lo propuesto recomendamos lo siguiente:

- Tener una capacitación óptima del funcionamiento del sistema móvil a todos los ganaderos de la finca que vayan a utilizar la aplicación.
- Es imprescindible que el ganadero le de mantenimiento al dispositivo GPS que contendrá una batería recargable.
- Se debe tener una cuenta de Google para su respectiva autenticación y establecer las seguridades necesarias para evitar el fraude o robo de la información del ganado.
- Se sugiere un ancho de banda adecuado y contratación de mensajería móvil para un correcto funcionamiento de la aplicación.
- Se deben realizar reuniones con los ganaderos mensual, trimestral y anualmente para establecer los problemas y sugerencias que ellos han tenido en la utilización de esta aplicación, para realizar mejoras en el sistema y ofrecer un mejor servicio de calidad a los clientes.

REFERENCIAS

- Anonimo. (s.f.). *ElectroniLab*. Obtenido de <https://electronilab.co/tienda/adafruit-fona-808-modulo-celular-gsm-gps-sim-808/>
- Arduino. (2018). *Arduino*. Obtenido de <http://arduino.cl/que-es-arduino/>
- AS. (08 de 12 de 2016). AS. Obtenido de <https://androidstudiofaqs.com/conceptos/ventajas-desventajas-android-studio>
- Catalán, A. (26 de 10 de 2015). *Medium*. Obtenido de Mi primer app con Firebase y Android : <https://medium.com/@ykro/mi-primer-app-con-firebase-y-android-parte-1-fd0b7d717e0b>
- Corona, L., Abarca, G., & Mares, J. (2014). *Sensores y actuadores*. México: PATRIA S.A.
- Developer. (25 de 04 de 2018). *Developer*. Obtenido de <https://developer.android.com/studio/intro/?hl=es-419>
- Drake, J. (s.f.). *Programacion Orientado a Objeto*. Obtenido de https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I1_OrientadoObjeto.pdf
- E-Marmolejo, R. (25 de Noviembre de 2017). *Hetpro*. Obtenido de <https://hetpro-store.com/TUTORIALES/lm35/>
- Morales, A. G. (2011). *MANUAL DESARROLLO DE ELEMENTOS SOFTWARE PARAGESTIÓN DE SISTEMAS*. Madrid: EditorialCep.
- Pérez Guzmán, R. E., & González Rivero , O. (2012). *PROTOTIPO DE ADQUISICIÓN DE SEÑALES BIOLÓGICAS UTILIZANDO ARDUINO*. La Habana: Universitaria.
- Pérez, M. (14 de Octubre de 2016). *IESB Shool*. Obtenido de Firebase, qué es y para qué sirve la plataforma de Google: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>
- Pérez, R. E., & González, O. (2012). *PROTOTIPO DE ADQUISICIÓN DE SEÑALES BIOLÓGICAS UTILIZANDO ARDUINO*. La Habana: Universitaria.
- Rincón, L. (02 de 06 de 2010). *SlideShare*. Obtenido de <https://es.slideshare.net/LadyRincon/phpmyadmin>

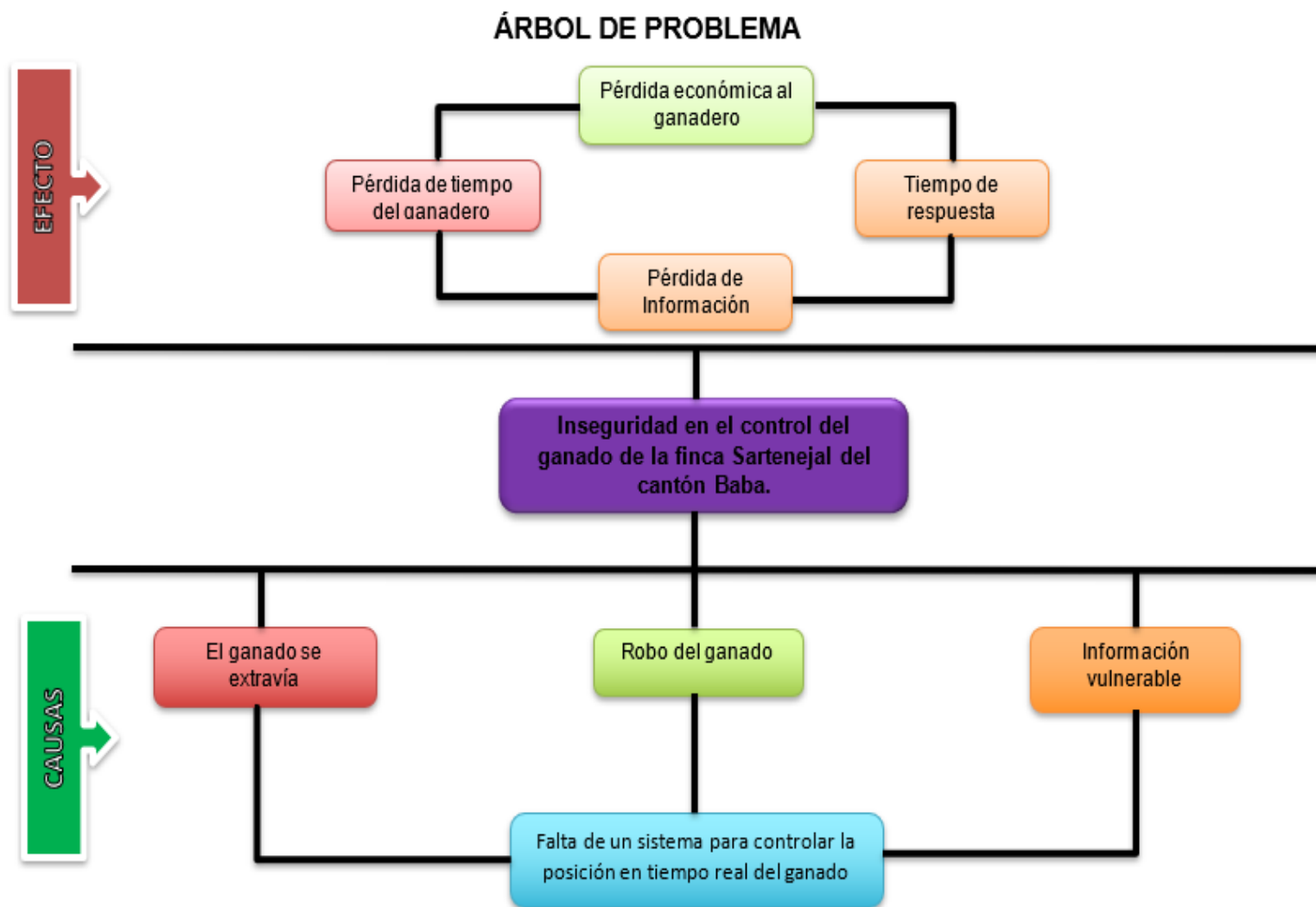
Robledo, D. (2017). *Desarrollo de Aplicaciones para Android I*. España: Ministerio de Educación de España.

Ruiz, M. (9 de Agosto de 2017). *OpenWebinars*. Obtenido de ¿Qué es Firebase de Google?: <https://openwebinars.net/blog/que-es-firebase-de-google/>

URIARTE, I. (2017). *SIM808 MÓDULO GSM GPRS GPS CON ANTENA GPS*. Obtenido de <https://hetpro-store.com/sim808-modulo-gsm-gprs-gps-con-antena-gps/>

Vergara, J. (28 de 06 de 2016). *CoriaWeb*. Obtenido de <https://www.coriaweb.hosting/nos-ofrece-phpmyadmin/>

ΑΝΕΚΚΟΣ



PRESUPUESTO

Ítem	Cantidad	Precio Unitario	Total
Módulo SIM808 con Antena	1	\$ 45,00	\$ 45,00
NodeMCu ESP8266	1	\$ 13,00	\$ 13,00
Sensor de Temperatura LM35	1	\$ 1,90	\$ 1,90
Batería 18650 3.7V AA	3	\$ 4,00	\$ 12,00
Cables Macho Hembra	10	\$ 0,20	\$ 2,00
Porta Baterías de 2	1	\$ 1,00	\$ 1,00
Porta Batería de 1	2	\$ 0,50	\$ 1,00
Switch On / Off	2	\$ 0,60	\$ 1,20
Collar	1	\$ 7,00	\$ 7,00
Caja porta circuito	1	\$ 5,00	\$ 5,00
Total			\$ 89,10

Nota: El prototipo tuvo un costo de \$89,10 por cada dispositivo GPS.

Denuncia de cuatrerros en Girón

Los moradores del cantón Girón están preocupados ante los constantes robos de ganado en el sector

Redacción - 22 de Mayo de 2018. 12:52

202 veces



Tags:

ganado
giron
cuatrerros
azuay

LO ÚLTIMO DE REDACCIÓN

➤ El huracán Michael amenaza a Cuba y Florida

➤ Bolsonaro y Haddad alistan estrategias en Brasil

➤ Mejor jugador del mundo se conocerá el 3 de diciembre

➤ Orquesta Batahola cumple 21 años de trayectoria

➤ Alertan de necesidad de mitigar el cambio climático

➤ Menor es golpeado



Foto: Archivo

La semana pasada los vecinos denunciaron un robo de cinco cabezas de ganado en la comunidad Léntag, parroquia La Asunción.

Para sorpresa de los dueños, días más tarde los ganados fueron encontrados el sector de Tablón y Cachi; por razones desconocidas los antisociales los abandonaron el ganado.

Los moradores del Girón contaron que los robos de ganado son frecuentes en varias parroquias, piden a la Policía Nacional que incremente los patrullajes y que ponga atención a las camionetas sin placas.

Personal de la **Policía Nacional** y autoridades dijeron estar conscientes de la problemática y piden denunciar estos hechos y registrar los animales para tener un control de la zona.

Dos detenidos con ganado robado

JLPE, 23, 2018 |



INFRACTORES. Dos personas fueron detenidas mientras transportaban dos reses que horas antes fueron robadas en una finca de La Concordia.

La actitud sospechosa de dos ocupantes de una camioneta roja de placa ABP-417, cuyo cajón de madera iba cubierto totalmente con una carpa, que circulaba en dirección La Concordia-Santo Domingo, hizo que los agentes policiales detuvieran la marcha del automotor, cerca a la estación del Iniap.

Al revisar el vehículo los uniformados se percataron que en la cajón iban dos reses maniatadas entre sí, por lo que solicitaron la guía de movilización a los ocupantes de la camioneta, quienes dijeron que no la tenían y que estaban realizando un flete hasta Santo Domingo.

Como Angiel B., de 28 años, y Renán Z., de 35, fueron identificados los ocupantes de la camioneta donde se movilizaba el ganado, quienes por disposición del Fiscal de turno fueron detenidos para investigaciones. En la cabina del carro los policías encontraron varios machetes, linternas y teléfonos celulares.

Tres vacas robadas

Uno de los gendarmes informó que mientras realizaban el procedimiento de aprehensión, llegó al lugar Wilson Cobeña, quien reconoció como suyas las dos cabezas de ganado que horas antes se las habían robado desde su propiedad.

"El agricultor dijo que desde su finca ubicada en la vía La Independencia-Fuente Guito, sector "El Salazar", se habían sustraído tres vacas, mostró los ferros (marcas) y documentos que justifican su legítima propiedad", explicó el uniformado.

La Policía considera que presuntamente el otro semoviente robado los delincuentes lo habían dejado escondido en algún lugar para posteriormente transportarlo sin ser descubiertos. UJAR

Reses robadas eran vendidas en dos provincias



SANTO DOMINGO: Comerciantes llegan cada semana a la feria de ganado, en esta ciudad.

ECUADOR | Gisela Guerrero | 4 de agosto, 2018



Santo Domingo -

Las reses robadas de fincas en Santo Domingo y en El Carmen (Manabí), eran supuestamente vendidas en forma clandestina en las inmediaciones de ferias de ganado de Santo Domingo y en cameles de la provincia de Guayas.

Así lo indicó una fuente de la Unidad de Investigación contra el Delito de Abigasto (UICA) de la Policía tras el operativo desplegado el jueves 2 en 20 inmuebles de las provincias de Santo Domingo de los Tsáchilas, Manabí y Los Ríos.

Ese operativo dejó 16 detenidos, entre esos un juez de Garantías Penales y dos policías de Santo Domingo y de Buena Fe (Los Ríos).

Quedaron doce personas con prisión preventiva y cuatro con medidas sustitutivas. Entre ellos se encuentran dos policías en servicio activo y el juez de Garantías Penales que fue aprehendido para investigaciones por presunta asociación ilícita y que por tener fuero de corte será direccionado a una Fiscalía provincial.

Cuando se realizó el allanamiento los uniformados desconocían que el detenido era juez, él les dijo su estatus cuando lo llevaron a la Fiscalía.

La Policía investiga el posible vínculo del magistrado y los dos policías con la presunta banda. Los uniformados implicados habrían entregado información para que se puedan cometer los delitos, según Denilo Carvajal, jefe de la UICA Zona 4.

MÁ

Ni
l
d
V

M
s
N
m

Tr
s
s
j
P

VE

ÚLT

Is
B
m
m

Con
por
oct
Gua

Fuji
vid
un l

VE

LO

Tras varias denuncias de abigeato, la UICA detectó que se mantenía el mismo patrón de robo. Se realizaban en dos camiones en los que se embarcaban cerca de 30 reses por robo, que representaban cerca de \$30.000 de pérdida para los dueños de las semovientes.

Los delincuentes estudiaban las fincas que tenían ganado de engorde y acudían armados, amedrentaban a las víctimas, todo en la noche, dijo Carvajal. "Tenían un equipo de avanzada que avisaban que las vías estén sin vigilancia para mover el ganado", citó.

Los agentes también detectaron que quienes tenían antecedentes eran absueltos u obtenían medidas sustitutivas pese a lo grave de las denuncias.

En un caso, una mujer que denunció de robo a un implicado terminó siendo acusada de fraude procesal, por rectificar en su testimonio que no fue amedrentada con un destomillador cuando fue atacada por uno de los hoy detenidos. (f)

Apuntes

Abigeato

Operativo

Se realizó con 20 allanamientos a inmuebles, 16 en Santo Domingo y 4 en Manabí y Los Ríos encontrando 10.000 municiones, que al parecer eran comercializadas como abastecimiento para otros delincuentes.

Antecedentes

Tres de los dieciséis detenidos tienen antecedentes por abigeato, robo, porte de armas y tráfico ilícito de sustancias, indicó la Policía Nacional.

Cinco cuatrereros robaban ganado en Los Chillos

SEP 06, 2018 |



BANDA. Cuatro de los cinco sospechosos tienen antecedentes penales.

El robo de ganado vacuno tiene tras las rejas a cinco personas, que presuntamente integraban una banda dedicada al abigeato.

Todo se planificaba desde Imbabura. Ahí se radicaba el líder de la organización delictiva, quien también fue detenido durante la operación 'Impacto 330'.

Durante los últimos tres meses, la Unidad de Investigación Contra el Delito de Abigeato (UIACA) estaba tras los pasos de los sospechosos.

Según el reporte de delitos, se reportaron robos de ganado en las poblaciones de Sangolquí, Amagualá y en barrios alejados del sur de Quito.

El mayor Marco Moreno, jefe de la UIACA, dijo que esta organización habría ocasionado un perjuicio económico de al menos 20 000 dólares a pequeños y medianos ganaderos.

EL DATO

En el operativo se incautó un camión, municiones y una tablet.

Los antisociales amedrentaban a las víctimas con armas de fuego mientras robaban las vacas. Posteriormente, las cabezas de ganado eran embarcadas en un camión y las llevaban a camales clandestinos para faenarlas. Algunas eran comercializadas vivas.

La Policía efectuó cuatro allanamientos para capturar a los cinco cuatrereros. Cuatro de ellos presentan antecedentes penales por asociación ilícita y robo.

Los implicados violentaban los cercos durante las noches, pero previamente hacían un trabajo de inteligencia delictual que consistió en identificar rutas de escape y limpiar los terrenos para que el traslado de los animales sea más ágil. @UCI