UNIVERSIDAD
DE LOS ANDES
VENEZUELA

PROYECTO DE GRADO

Presentado ante la ilustre Universidad de Los Andes como requisito final
para obtener el Título de Ingeniero de Sistemas

Aplicación Móvil para la Gestión de información y Manejo del Ganado Bovino, en las Fincas Ganaderas del estado Mérida.

Br.Adolfo José Contreras Castillo
Tutor: Dr. Gerard Páez Monzón

Mérida, Diciembre 2020.

Indice general

Indice de figuras

Indice de tablas

Resumen

El presente proyecto tiene como finalidad desarrollar una aplicación móvil para la gestión de información y manejo de ganado bovino en las fincas ganaderas del estado Mérida, con el fin de automatizar el proceso de registro de animales, alimentación, vacunas y tratamientos que se llevan a cabo en las unidades de producción ganadera, así como también diseñar una interfaz gráfica agradable a los usuarios y de fácil manejo. Se realizó un análisis de la situación actual que los productores y veterinarios poseen, utilizando la entrevista informal y escrita, con la cual resultó factible desarrollar un modelo mínimo viable que agilice cada uno de los procesos que se llevan a cabo relacionados con los rebaños de las unidades de producción. Para la implementación del frontend y el backend de la aplicación se utilizaron los frameworks React Native y Django respectivamente.

Palabras clave: Aplicación Móvil, Sector Pecuario, Ganado, Fincas.

Capítulo 1

Introducción

El desarrollo de las tecnologías de la información y comunicación (TIC) le han permitido a la sociedad actual una dinámica particular que conduce a profundos cambios estructurales en toda las naciones, por tanto, hoy en día es innegable la presencia de los avances tecnológicos y de informática en las diversas áreas de la actividad humana y en particular de la vida cotidiana.

En consecuencia, las nuevas tecnologías han traído consigo la creación de aplicaciones orientadas a la producción agrícola. Que buscan incrementar el apoyo técnico en el campo ya sea para la producción de cultivos o manejo de ganadería, permitiéndole al productor tomar decisiones de manera eficiente.

Ante las consideraciones de lo expresado, el presente trabajo tiene como propósito fundamental incorporar la utilización de las tecnologías de información y comunicación en el sector pecuario de Venezuela, como una manera de mejorar la productividad y la competitividad con los demás países de Latinoamérica mediante el uso de una aplicación móviles que permita gestionar la información y manejo del ganado bovino, en las fincas ganaderas del estado Mérida.

1.1. Antecedentes

Los antecedentes de esta investigación son trabajos, investigaciones y artículos relacionados con aplicaciones móviles dedicadas al manejo agropecuario donde se involucran en el sector productores agrícolas, médicos veterinarios, institutos de investigación y universidades.

Federico Henze, Guillermo Vilas, Ignacio Rohr, María Eugenia Etchemendy. (2013), [10], en su trabajo especial de grado proponen un Sistema para la gestión de establecimientos ganaderos denominado AGROSOFT, que tiene como objetivo desarrollar un sistema que otorgue valor agregado a la gestión de los establecimientos ganaderos y poder tener seguimiento y control de las actividades que se realizan a diario. Universidad ORT Uruguay, Uruguay. Desarrollado como una aplicación de escritorio.

El sistema permite manejar las actividades que en los establecimientos se llevan a cabo, como lo es la trazabilidad del ganado, desde el nacimiento hasta la muerte, compras, ventas, control de empleados y contabilidad en general.

Se presenta el trabajo de David Bastidas Vargas y Liliana Esther Machuca Villegas. (2014), [15], Aplicación Móvil para el Seguimiento y Control de las Siembras de arrocera LA ESMERALDA S.A. ALESA MÓVIL, posee como objetivo diseñar e implementar una aplicación móvil que permita recopilar, controlar y analizar información sobre el estado de las siembras de arroz, para la toma de decisiones en aras de mejorar la productividad. Universidad del Valle, Santiago de Cali, Colombia.

La aplicación se estructura en un módulo de control que se encarga de gestionar el estado actual de las siembras y la formulación de insumos necesarios para su evolución con un módulo de indicadores que se encarga de visualizar informes con el fin de identificar y conocer el estado y evolución de las siembras.

El estudio citado se encuentra relacionado con el presente trabajo por cuanto demuestra la necesidad palpable de contar con una aplicación que permita a los productores del campo manejar de forma organizada sus unidades de producción para

la toma de decisiones.

Según Luiyiana Pérez y Rolando Lasso. (2018),[13], describen en su artículo un Sistema de Información Empresarial de Fincas Ganaderas para pequeñas y medianas empresas en la Provincia de Los Santos. Este artículo presenta una solución móvil para la gestión empresarial de fincas ganaderas, articulado con el programa de trazabilidad bovina. Este sistema fue desarrollado con dos interfaces, una aplicación de escritorio para los administradores del Ministerio de Desarrollo Agropecuario y la otra una App para los productores. El sistema consiste en llevar el registro general de la finca, composición ganadera o inventario. Que permiten ser consultados desde la aplicación de escritorio. Universidad Tecnológica de Panamá, Panamá. La utilidad de este trabajo se caracteriza por tener similitud a la aplicación propuesta sobre el registro de trazabilidad del ganado, pero se diferencia porque no toman encuesta el contacto que deben tener los productores con profesionales del sector pecuario.

Para el desarrollo de este proyecto, se estudiaron algunas aplicaciones que hacen vida en el sector ganadero, que se describen a continuación:

- GANSOFT APP: Programa diseñado para el control de registros de rebaños, desarrollada para plataformas móviles por la empresa llamada Tecnología de Procesos, C.A. Maracaibo, Edo. Zulia, Venezuela, (2018). Es una herramienta para optimizar el manejo de múltiples sistemas de producción animal. Permite mantener la información de sus animales ordenada y precisa.

- CONTROL GANADERO: Fue desarrollada por un grupo internacional de consultores, veterinarios, ingenieros y productores, Colombia, (2019). Esta aplicación lleva reportes de la producción del ganado en fincas, registrando productos; cárnicos, leche, cabezas de ganado, engorde, reproducción etc.

Los resultados de la investigación antes descrita, comprobaron que, la incorporación de aplicaciones móviles en fincas productoras de ganado permite generar grandes beneficios, como lo es manejar los registros de las unidades de producción de manera

eficiente, permitiendo que se tomen decisiones adecuadas y permitan aumentar la producción de carne, leche y derivados.

1.2. Planteamiento del Problema

En la actualidad el uso de las tecnologías de información y comunicación han permitido cambiar el rumbo de nuestras vidas en diferentes áreas, logrando mejorar el bienestar del ser humano, por esta razón la utilización de las herramientas tecnológicas van desde las necesidades que cada uno de las personas tienen, hasta especializarse en diferentes contextos. Por esta razón la ciencia y la tecnología están cambiando el rumbo de la producción alimentaria de todo país que haga uso de ella, haciendo que aumente la producción en los distintos rubros que produce.

En este contexto de ideas surge la necesidad de tecnificar el manejo de la ganadería en Venezuela ya que la gran mayoría de las unidades de producción no se encuentran tecnificadas; por esta razón, es innovador hacer uso de las tecnologías móviles como medida que permita tecnificar el sector ganadero.

En la producción de ganado vacuno han de destacarse, procesos claves como lo es, la administración de información y control sanitario en rebaños, que la mayoría de las unidades de producción no toman en cuenta y los productores del sector que lo hacen lo manejan con fichas manuscritas que en la actualidad ya son obsoletas. Debido al problema planteado se propone como solución el crear una aplicación móvil que permita gestionar la información y manejo del ganado de forma rápida y fácil.

El Desarrollo de la aplicación con enfoque pecuario permitirá ser estrategia tecnológica y de cambio en Venezuela generando un aumento en la tecnificación de fincas dedicadas a la producción ganadera, mejorando el manejo organizacional del sector, beneficiando a los productores de forma directa, ya que tendrán a la mano desde un teléfono inteligente el poder registrar y consultar información en el control de vacunas, períodos de reproducción, buen manejo en la alimentación de los animales y conexión con profesionales del sector. Todo eso con el fin de que el sector sea más

productivo.

1.3. Objetivos del Proyecto

1.3.1. Objetivo General

Desarrollar una Aplicación Móvil para la Gestión de información y Manejo de ganado Bovino, en las fincas ganaderas del estado Mérida.

1.3.2. Objetivos Específicos

- Investigar la información necesaria relacionada con el uso de tecnologías de información y comunicación en el sector ganadero.
- Analizar la información recopilada para que pueda ser usada en la creación de la aplicación móvil.
- Diseñar la aplicación móvil, que se encargará de alojar registros de usuarios, registro de animales, control sanitario y contacto entre productores y médicos veterinarios.
- Elaborar una base de datos para el almacenamiento de información de las unidades de producción ganadera.
- Implementar la aplicación móvil para que pueda ser utilizada por los productores y médicos veterinarios.
- Realizar pruebas y correcciones necesarias a la aplicación móvil, con la finalidad de verificar su buen funcionamiento.

1.4. Justificación del Proyecto

El proyecto es considerado como una herramienta estratégica de organización operacional desde el punto de vista tecnológico donde la aplicación móvil pasa a

convertirse en un enorme valor del sector ganadero que favorezca a los productores en llevar el control de sus fincas de una forma innovadora, creativa y eficiente.

Además sirve para estimular a los productores a que tecnifiquen la manera de llevar la administración de las unidades de producción, adquiriendo conocimiento profesional de manera directa con expertos del área, generando un aumento en la producción de ganado bovino.

En este sentido la aplicación móvil para la gestión de información y manejo del ganado bovino se justifica porque el empleo de esta estrategia permitirá promover la gestión de las fincas en el estado Mérida de manera eficiente, adaptada a las características de nuestra región y de esta manera contribuir al aumento productivo que hace falta en Venezuela.

De igual manera, el estudio se considera relevante desde el punto de vista productivo, porque a través del uso de una aplicación móvil, los productores y profesionales del ramo pecuario trabajarán en conjunto y no de forma aislada, generando al productor tecnificar sus fincas.

En atención a esta razón, el trabajo representa un aporte al sector ganadero, porque el uso de la aplicación móvil favorece la actividad continua del campo, como forma de desarrollar y aumentar la producción alimentaria.

1.5. Metodología

Para dar respuesta al problema planteado es necesario una metodología de investigación, que permita hacer referencia o descripción de las diferentes unidades de análisis de investigación.
La metodología del proyecto incluye el nivel, diseño de la investigación, población y muestra, las técnicas y los instrumentos de recolección de datos así como la técnica de procesamiento y análisis de datos que serán utilizados para llevar a cabo la indagación.

1.5.1. Nivel de la investigación

Según el nivel, el proyecto se ubica en una investigación descriptiva, ya que se realiza un análisis del problema planteado con la finalidad de describir y explicar sus causas y efecto.
Al respecto Fidias G. Arias (2016),[6], plantea que la misma consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento.(Pg. 24).

1.5.2. Diseño de la investigación

La presente investigación corresponde a un diseño de campo, ya que el investigador recolecta los datos de interés directamente de la realidad, en tal sentido Arias (2016), señala:

La investigación de campo es aquella que consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes, de allí su carácter de investigación no experimental. (Pg. 31).

Por otra parte, se enmarca bajo la modalidad de proyecto de investigación-acción ya que es un tipo de investigación aplicada, destinada a encontrar soluciones a problemas que tenga un grupo, una comunidad o una organización. Es por ello que esta investigación es factible, ya que se está dando una solución al problema presentado en el sector pecuario.

1.5.3. Población

La población se define como el conjunto de individuos, objetos o medidas que poseen algunas características comunes observables en un lugar y en un momento determinado. Cuando se vaya a llevar a cabo alguna investigación debe de tenerse en cuenta algunas características esenciales al seleccionarse la población bajo estudio.

Al respecto Arias (2016),la describe como: Un conjunto de elementos con características comunes para los cuales serán extensivas las conclusiones de la investigación. (Pg. 81). La población objeto de estudio en la investigación estará conformado por los productores de ganado bovino y Médicos Veterinarios que hacen vida en Venezuela.

1.5.4. Muestra

La muestra se define como un subconjunto representativo finito que se extrae de la población accesible, (Arias, 2016, pg. 83). Debido a que se cuenta con una población grande, como lo es el caso de productores ganaderos y veterinarios en Venezuela. Es necesario tomar una representación de ella, porque su observación total y directa es difícil de realizar. A esa parte específica de la población se le denomina muestra, mediante la cual se busca conocer aproximadamente las características de la población.

En resumen, los instrumentos de recolección de datos se les aplicó a 10 productores del sector ganadero del municipio Antonio Pinto Salinas del estado Mérida y 3 médicos veterinarios.

1.5.5. Técnicas e instrumentos de recolección de datos

Las técnicas de recolección de datos son lineamientos que direccionan la recolección de la información, al respecto Fidias G. Arias (2016), define la técnica de recolección de datos como las distintas formas o maneras de obtener la información. Las técnicas utilizadas en el estudio son la encuesta escrita, la entrevista informal, el análisis documental y de contenidos.

El instrumento de investigación es el medio que se utiliza para medir el comportamiento de las unidades de análisis. En este estudio se utiliza el cuestionario. (Ver

anexo).

1.6. Alcance

El presente proyecto tiene como alcance realizar un análisis, diseño y desarrollo de una aplicación móvil que sirva como herramienta para la gestión de información y manejo del ganado bovino en las Fincas Ganaderas del estado Mérida, Venezuela. Donde los productores y profesionales (Veterinarios) lleven la administración de los rebaños desde un teléfono inteligente con sistema operativo Android.

Capítulo 2

Marco Teórico

El marco teórico se define como un conjunto de conceptos teóricos, permitiendo que el investigador señale los recursos documentales y literarios recabando información de diferentes fuentes acerca de lo que está estudiando.

2.1. Ganadería y cría en Venezuela

La ganadería es una actividad económica que consiste en la crianza de animales para su aprovechamiento. Como fuente de alimentos básicos para la seguridad alimentaria, alrededor de una décima parte de la población mundial tiene algún grado de vinculación del sector ganadero y de cría, por tanto, los sistemas de producción pecuaria son considerados como la estrategia demográfica, social, económica y cultural más apropiada para mantener el bienestar de las comunidades especialmente las rurales, debido a que es la única actividad que puede simultáneamente proveer seguridad en el sustento diario, conservar ecosistemas, colaborar en las estrategias de poblamiento nacional y satisfacer los valores culturales y tradiciones.

En nuestro país el sector pecuario proporciona alrededor de dos quintos del valor total de la producción agropecuaria, predominando el ganado vacuno de doble propósito (carne y leche), le sigue la cría de ganado porcino, aviar y, en menor escala, el ganado caprino y ovino.

La ganadería en Venezuela se puede diferenciar según sus técnicas y sus fines. Puede ser extensiva: refiriéndose a grandes extensiones de tierras hatos y el libre pastoreo de grandes rebaños de ganado bovino y caballar; como también es intensiva: cuando se invierte capital en el establecimiento de potreros, el mejoramiento de los pastos y la atención de la calidad genética y la salud de los animales.

2.2. Uso de Registros y manejo de la información en la Ganadería de Venezuela

Según [12], La información es la herramienta fundamental para la elaboración y uso de los registros en la ganadería. La información se convierte en una magnífica herramienta que le permite al ganadero diagnosticar su situación actual, conocer volúmenes de producción, limitantes y establecer el monto estimado de las inversiones y el margen de rentabilidad de la finca. De allí pues, que se haga uso de todas las técnicas para recolectar la información y elaborar los registros, entre ellas la observación es una de las técnicas usada con más frecuencia, para recolectar datos que luego serán procesados y convertidos en información a través de los registros.

En todo sistema productivo es importante que se conozca el comportamiento de cada eslabón de la cadena, por lo cual se debe registrar todos y cada uno de los eventos que allí ocurran, no escapa de ello el sistema productivo doble propósito en la ganadería bovina, en la cual actualmente es casi inexistente el uso de registros. Si bien es cierto que el estado es el responsable de mantener un sistema de registros actualizados, no menos cierto es que los datos o información inicial deben ser aportada por los ganaderos y debe provenir de cada uno de los eventos que ocurren o de las actividades que se realizan, por lo que es responsabilidad directa de cada uno de los propietarios de las fincas que los registros que maneja sean confiables.

2.2.1. Captación y uso de la información

El uso de registros implica un proceso de concientización de parte del productor para llevar de manera controlada y planificada las actividades diaria de la finca, ya

que de ese modo se podrá hacer un diagnóstico de la situación actual de la finca, y en base a ello buscar el mejoramiento del proceso actual u orientarse en la aplicación de un sistema útil de manejo y control de finca como puede ser un registro.

2.2.2. Tipos de fuentes para la recolección de datos

Primaria: La finca proporciona información histórica que se obtiene a través de los registros, siempre y cuando se hayan llevado correctamente; de no ser así únicamente se contará con la memoria del propietario y empleados que hayan permanecido por algún tiempo en la explotación y tengan conocimientos de las actividades desarrolladas en la misma y sus parámetros de producción.

Secundaria: Información obtenida de fincas vecinas que cuentan con un sistema de producción igual o similar, resultados de estudios regionales realizados por algunas instituciones privadas o públicas y valores promedios de producción establecidos en los mercados locales.

2.2.3. Registros y características

Para obtener un registro que sirva para los análisis de las unidades de producción, es necesario contar con registros que sean fáciles de entender, mantener y actualizar. Partiendo de ello es claro que los registros deben ser lo más sencillos y específicos posibles, aportando información útil que facilite el manejo de la finca y ante todo aportando información que pueda ser manejada por el productor.

2.2.4. Uso de los registros en la ganadería venezolana

Actualmente, la falta de registros ha sido un problema grave que ha tenido la ganadería en Venezuela y que ha ido creciendo a través de los años debido a los problemas que el sector viene atravesando, como lo es la falta de inversión, escasez y altos costos de insumos agrícolas. Todo esto ha causado que los productores no tomen con importancia el llevar registros de las unidades de producción.

Debido a los problemas anteriores, estamos en la necesidad de incentivar el uso de los registros en la ganadería venezolana, como manera de contrarrestar la problemática, el cual permitiría tratar a fondo el desconocimiento o falta de información por parte de los dueños de fincas, es decir los registros son una manera útil, práctica, económica y confiable para mejorar significativamente el manejo productivo y reproductivo para la toma de decisiones.

Este planteamiento genera la necesidad de ampliar y mejorar el uso de registros en las fincas, utilizando aplicaciones móviles como parte de una mejora de los sistemas ganaderos en Venezuela, mejorando los esquemas organizacionales y paradigmas actuales.

2.3. Condiciones fisiológicas del ganado bovino

Según [5], define las condiciones fisiológicas de la siguiente manera:

Becerro(a): Vacuno con edad comprendida entre el nacimiento y el destete (aprox 6 - 8 meses).

Maute(a): Vacuno con edad comprendida entre el destete e inicio de actividad sexual (150 - 250 kg PV).

Novilla: Hembra vacuna con edad comprendida entre el inicio de la actividad sexual y el primer parto (divergencia entre autores).

Torete: Macho vacuno luego de la pubertad (>250 kg PV).

Novillo: Macho vacuno castrado (>250 kg PV).

Vaca: Hembra vacuno adulta, considerada así luego de su primer parto.

Toro: Macho vacuno adulto.

2.4. Las TIC en la Ganadería

- Permiten disminuir costos.
- Expandir el alcance de los mercados.
- Llegan a un público remoto y muchas veces disperso.
- Optimizan los contenidos de relevancia para el sector.
- Facilitan un mejoramiento continuo de la calidad y cantidad de la información.
- Permiten ofrecer nuevos servicios a los usuarios aprovechando la tecnología.
- Disminuyen los costos de los sistemas de información
- Agilizan trámites

2.5. Aplicación Móvil

Es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Las aplicaciones permiten al usuario efectuar un conjunto de tareas de cualquier tipo profesional, de ocio, educativas, de acceso a servicios, etc., facilitando las gestiones o actividades a desarrollar.

Por lo general, se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros.

2.5.1. Características de los tipos de aplicaciones móviles

Diferencias entre los principales tipos de aplicaciones móviles, [4] .

Web App

Una Web App es uno de los tipos de aplicaciones móviles desarrolladas con lenguajes muy conocidos por los programadores, tales como CSS, Javascript o HTML.

La Web App, se puede programar independiente del sistema operativo en el que se va a utilizar la aplicación, así se pueden ejecutar las aplicaciones en distintos dispositivos, sin la necesidad de crear varias aplicaciones.

Estas aplicaciones se ejecutan dentro del propio navegador web del dispositivo mediante una URL.

Ventajas de la Web App:

- A diferencia de las nativas, no necesitan aprobación externa para su publicación.
- Proceso de desarrollo bastante fácil y económico.
- El usuario siempre tiene opción de acceder a la última versión.
- El código base es reutilizable en distintas plataformas.
- Se pueden reutilizar sitios responsive ya diseñados.

Desventajas de la Web App:

- Acceso limitado a los elementos del hardware del dispositivo.
- Experiencia de usuario peor que la opción de la app nativa.
- Hace falta conexión a Internet.
- Se necesita hacer un mayor esfuerzo para su promoción.

Aplicación nativa:
Una aplicación nativa es aquella que se crea de forma específica para un sistema operativo determinado.

Hay que ser consciente de que cada una de las plataformas, ya sea iOS, Android, o Windows Phone, tienen un sistema distinto, por lo tanto si se desea que las aplicación esté disponible en todas las plataformas, es necesario crear varias apps con el lenguaje del sistema operativo seleccionado.

La principal ventaja de este tipo de aplicaciones, es que no necesitan conexión a Internet para funcionar. La instalación y descarga de las aplicaciones nativas, se hace por medio de las tiendas de aplicaciones.

Ventajas de las aplicaciones nativas:

- Envío de avisos o de notificaciones a los usuarios.
- Proporciona un acceso completo al dispositivo.
- Tiene visibilidad en App Store.
- Gran experiencia de usuario.
- Actualizaciones constantes de la app.

Desventajas de las aplicaciones nativas.

- Tendencia a ser más cara que otras opciones durante la fase de desarrollo.
- Distintas herramientas, habilidades e idiomas para cada plataforma de destino
- No se puede reutilizar entre las distintas plataformas el código del cliente.

Web App Nativa:

Los tipos de aplicaciones móviles, App Nativa y Web App, se funden para crear una aplicación híbrida que recibe el nombre de Web App Nativa.

La Web App Nativa reúne lo mejor de la App Nativa y de la Web App, esta App híbrida se desarrolla con lenguajes propios de las web App, es decir, con JavaScript, HTML y CSS. Con Web App nativa es posible agrupar los códigos y distribuirla en

App store.

Ventajas de la Web App Nativa:

- La instalación es nativa pero la construcción se hace con los lenguajes de programación CSS, JavaScript y HTML.
- Se puede acceder a parte del hardware del dispositivo.
- Utiliza el mismo código base para distintas plataformas.
- Tienes la posibilidad de distribuirla en las tiendas de Android e Ios.

Desventajas de la Web App Nativa:

- El diseño virtual no siempre está en relación con el sistema operativo en el que va a ser mostrada.
- La experiencia de usuario está más cerca de aplicación web que de la app nativa.

2.5.2. Ventajas de las aplicaciones móviles:

- Actualmente las aplicaciones móviles nos permiten tener información al alcance de nuestros dedos.
- Permiten tener acceso al correo, a las noticias, a la mensajería, música, tiempo, etc. Y evitan cargar con el portátil a todas partes.
- Experiencia atractiva de los usuarios por la comodidad y la funcionalidad de estas, ofreciendo un acceso completo al dispositivo, en software y hardware.
- Facilidad para realizar compras de productos y servicios.
- Son productivas a nivel empresarial mejorando el flujo de información para el cumplimiento de objetivos ya que favorecen un crecimiento de la productividad en el día a día.
- Canales directos para realizar notificaciones o realizar gestiones sencillas de forma inmediata.

2.5.3. Desventajas de las aplicaciones móviles

- Poseen costos de desarrollo y mantenimiento.
- Los estándares de calidad son muy altos y no todas las aplicaciones pueden introducirse en el mercado tan fácilmente.
- Logística compleja porque la distribución de las aplicaciones depende de las tiendas o empresas en línea. Como por ejemplo: Apple Store, Play Store.
- Espacio limitado y consumo elevado, esto se debe a que las App ocupan cierto espacio de memoria en los dispositivos móviles.

2.6. Android

Android [1] es un sistema operativo móvil desarrollado por Google, basado en el Kernel de Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores.

2.7. Que es un Servidor

Un servidor es un computador que está al servicio de otros computadores, dispositivos electrónicos (impresoras, móviles, etc.) y personas a los que suministra información. Todos estos sujetos que reciben la información de un servidor se conocen como clientes. Son computadores más potentes de lo normal y se encuentra siempre encendido, ya que si está apagado o da algún error, no será capaz de cumplir con su función.

2.7.1. Tipos de servidores

Hay multitud de tipos de servidores. Cada uno tiene las características adecuadas para las tareas específicas que realiza. Aquí tienes los más habituales.

[1] https://www.android.com/

Servidor de correo

También conocido como Mail Server, es un computador que envía, recibe y almacena los mensajes de correo electrónico o emails.

Servidor proxy

Este tipo de servidor actúa como intermediario entre el servidor y el cliente. De este modo, el servidor no conoce la identidad del cliente. Se utiliza para mejorar la privacidad del usuario. Es habitual en los navegadores de Internet.

Servidor FTP

Es el acrónimo de File Transfer Protocol, que sería protocolo de transferencia de archivos en español. Se utiliza para enviar archivos de un computador a un servidor o para descargarlos desde el servidor al computador.

Servidor de bases de datos

Ofrece servicios de almacenamiento y gestión de bases de datos a sus clientes. Permite almacenar grandes cantidades de información, como datos bancarios. Utilizan clúster de servidores, que es un conjunto de servidores de bases de datos.

Servidor web

Almacena todos los archivos propios de una página web (texto, imagen, vídeo, etc.) y los muestra a los clientes a través de los navegadores. Utiliza el protocolo HTTP (Hipertext Transfer Protocol).

2.8. Base de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

2.8.1. Ventajas del uso de bases de datos

- Independencia de los datos respecto a los programas. Permite modificar los datos, excepto el código de aplicaciones.
- Menor redundancia, es decir, no es necesaria la repetición de datos. Solamente se indica la manera en la que se relacionan éstos.
- La base de datos facilita al usuario obtener más información debido a la facilidad que provee esta estructura para proveer datos a los usuarios.
- Integridad de los datos, lo que genera mayor dificultad de perder la información o de realizar incoherencias con los datos.
- Mayor seguridad en los datos. Al permitir restringir el acceso a los usuarios, cada tipo de éstos tendrá la posibilidad de acceder a ciertos elementos.
- Coherencia de los resultados. Al recolectar y almacenar la información una sola vez, en los procedimientos se utilizan los mismos datos, razón por la que los resultados son coherentes.

2.8.2. PostgreSQL

PostgreSQL [2] es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Posee una sólida reputación por su arquitectura ya que garantiza en los datos, confiabilidad, integridad, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer constantemente soluciones innovadoras y de alto rendimiento. PostgreSQL se ejecuta en todos los sistemas operativos principales, ha sido compatible con ACID desde 2001 y tiene complementos potentes como el popular extensor de base de datos geoespacial PostGIS.

[2] https://www.postgresql.org/

A continuación se muestra varias características que se encuentran en PostgreSQL:

Tipos de datos

- **Primitivas:** entero, numérico, cadena, booleano
- **Estructurado:** fecha / hora, matriz, rango, UUID
- **Documento:** JSON / JSONB, XML, valor-clave (Hstore)
- **Geometría:** Punto, Línea, Círculo, Polígono
- **Personalizaciones:** Compuestas, Tipos Personalizados.

Integridad de los datos

- ÚNICO, NO NULO
- Llaves primarias
- Llaves extranjeras

Restricciones de exclusión

- Cerraduras explícitas, cerraduras consultivas.
- Concurrencia, rendimiento.
- Indexación: B-tree, Multicolumn, Expresiones, Parcial.
- Indexación avanzada: GiST, SP-Gist, KNN Gist, GIN, BRIN, índices de cobertura, filtros Bloom.
- Planificador / optimizador de consultas sofisticado, análisis de solo índice, estadísticas de varias columnas.
- Transacciones, transacciones anidadas (a través de puntos guardados).
- Control de concurrencia multi-versión (MVCC).

- Paralelización de consultas de lectura y creación de índices de árbol.
- Particionamiento de tablas.
- Todos los niveles de aislamiento de transacciones definidos en el estándar SQL, incluido Serializable.
- Recopilación de expresiones Just-in-time (JIT).
- Confiabilidad, Recuperación de Desastres.
- Registro de escritura anticipada (WAL).
- Replicación: asíncrona, síncrona, lógica.
- Recuperación de punto en el tiempo (PITR), recursos activos.
- Espacios de tabla.

Seguridad

- Autenticación: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, certificado y más
- Sistema robusto de control de acceso
- Seguridad de columnas y filas

Extensibilidad

- Funciones y procedimientos almacenados.
- Lenguajes de procedimiento: PL / PGSQL, Perl, Python (y muchos más)
- Contenedores de datos externos: conéctese a otras bases de datos o flujos con una interfaz SQL estándar
- Muchas extensiones que proporcionan funcionalidad adicional, incluyendo PostGIS
- Internacionalización, Búsqueda de texto

- Soporte para conjuntos de caracteres internacionales, por ejemplo, a través de colaciones de UCI.
- Búsqueda de texto completo.

Se ha demostrado que PostgreSQL es altamente escalable tanto por la gran cantidad de datos que puede administrar como por la cantidad de usuarios concurrentes que puede acomodar. Existen clústeres de PostgreSQL activos en entornos de producción que administran muchos terabytes de datos y sistemas especializados que administran petabytes.

2.8.3. Modelo Entidad-Relación

Es un método del que disponemos para diseñar esquemas que posteriormente debemos de implementar en un gestor de bases de datos. Este modelo se representa a través de diagramas que ayuda a entender los datos y como se relacionan entre ellos y está formado por varios elementos como los son.

Entidad: Las entidades representan cosas u objetos (ya sean reales o abstractos), que se diferencian claramente entre sí. Las entidades se representan en un diagrama con un rectángulo.

Atributos: Los atributos definen o identifican las características de entidad. Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...). Los atributos se representan como una elipse que descienden de una entidad.

Relación: Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable. Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas.

Relaciones de cardinalidad: Cardinalidad es el número de entidades con la cual otra entidad puede asociar mediante una relación binaria; la cardinalidad puede

ser:

- **Uno a uno:** Una entidad se relaciona únicamente con otra y viceversa.
- **Uno a muchos o muchos a uno:** determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez.
- **Muchos a Muchos:** determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa.

Claves

Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad). Se clasifican en:

- **Superclave:** aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.
- **Clave primaria:** identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad.
- **Clave externa o clave foránea:** este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave.

2.8.4. Modelo Relacional

En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único.

El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura

en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro.

El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R. Los diseños de bases de datos a menudo se realizan en el modelo E-R, y después se traducen al modelo relacional.

2.9. Herramientas tecnológicas para el desarrollo de la aplicación

2.9.1. HTML 5

HTML5 [3] es un lenguaje markup (de hecho, las siglas de HTML significan Hyper Text Markup Language) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es la quinta revisión del estándar que fue creado en 1990. Con HTML5 tenemos la posibilidad de consumir menos recursos. Con HTML5, también entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

2.9.2. CSS

Las hojas de estilo en cascada (CSS) [4] son un mecanismo simple para agregar estilo (por ejemplo, fuentes, colores, espaciado) a los documentos web.

2.9.3. JavaScript

JavaScript [5] es un lenguaje de programación ligero e interpretado que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

[3] https://www.w3.org/html/
[4] https://www.w3.org/Style/CSS/
[5] https://developer.mozilla.org/es/docs/Web/JavaScript

2.9.4. Python

Python [6] es un lenguaje de programación de propósito general que se puede usar para una amplia variedad de aplicaciones. Un gran lenguaje para principiantes debido a su legibilidad y otros elementos estructurales diseñados para que sea fácil de entender, Python no se limita al uso básico. De hecho, potencia algunas de las aplicaciones y sitios web más complejos del mundo.

Python es un lenguaje interpretado, lo que significa que los programas escritos en Python no necesitan compilarse de antemano para poder ejecutarse, por lo que es fácil probar pequeños fragmentos de código y hacer que el código escrito en Python sea más fácil de mover entre las plataformas.

2.9.5. Django

Django [7] es un framework web de Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de la molestia del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es de código abierto.

2.9.6. React

React [8] es una biblioteca de JavaScript para construir interfaces de usuario, React te ayuda a crear interfaces de usuario interactivas de forma sencilla. Diseña vistas simples para cada estado en tu aplicación, y React se encargará de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien.

[6] https://www.python.org/
[7] https://www.djangoproject.com/
[8] https://es.reactjs.org/

2.9.7. React Native

React Native [9] es un framework que permite el desarrollo de proyectos de software. Es una alternativa al desarrollo de aplicaciones nativas, te permite crear aplicaciones móviles usando solo JavaScript. Utiliza el mismo diseño que React, lo que le permite componer una interfaz de usuario móvil rica utilizando componentes declarativos.

2.9.8. Heroku

Heroku es uno de los PaaS (por sus siglas es una plataforma como servicio) más utilizados en la actualidad en entornos empresariales por su fuerte enfoque en resolver el despliegue de una aplicación. Ademas te permite manejar los servidores y sus configuraciones, escalamiento y la administración. A Heroku solo le dices qué lenguaje de backend estás utilizando o qué base de datos vas a utilizar y te preocupas únicamente por el desarrollo de tu aplicación [14], Heroku fue el servicio utilizado para el hospedaje de la aplicación propuesta.

2.9.9. Postman

Postman es una plataforma de colaboración para el desarrollo de API[10]. Postman se utiliza, sobre todo, para el *testing* de API REST. Gracias a esta herramienta, además de testear, consumir y depurar API REST, se puede monitorizar, escribir pruebas automatizadas, documentarlas y simularlas [11].

Características de Postman [8]:

- **Crear Peticiones**, te permite crear y enviar peticiones http a servicios REST mediante una interfaz gráfica. Estas peticiones pueden ser guardadas y reproducidas a posteriori.

- **Definir Colecciones**, mediante Postman podemos agrupar las APIs en colecciones. En estas colecciones podemos definir el modelo de autentificación de las

[9] http://www.reactnative.com/
[10] https://www.postman.com/

APIs para que se añada en cada petición. De igual manera podemos ejecutar un conjunto de pruebas, así como definir variables para la colección.

- **Gestionar la Documentación**, genera documentación basada en las API y colecciones que hemos creado en la herramienta.

- **Entorno Colaborativo**, permite compartir las API para un equipo entre varias personas. Para ello se apoya en una herramienta de colaborativa en Cloud.

- **Genera código de invocación**, dado un API es capaz de generar el código de invocación para diferentes lenguajes de programación: C, cURL, C#, Go, Java, JavaScript, NodeJS, Objective-C, PHP, Python, Ruby, Shell, Swift.

- **Establecer variables**, con Postman podemos crear variables locales y globales que posteriormente utilicemos dentro de nuestras invocaciones o pruebas.

- **Soporta Ciclo Vida API management**, desde Postman podemos gestionar el ciclo de vida del API Management, desde la conceptualización del API, la definición del API, el desarrollo del API y la monitorización y mantenimiento del API.

- **Crear mockups**, mediante Postman podemos crear un servidor de mockups o sandbox para que se puedan testear nuestras API antes de que estas estén desarrolladas.

2.10. Estudio de Metodologías para el desarrollo de aplicaciones móviles

La mayor parte de los proyectos de desarrollo de aplicaciones para móviles requieren de un método de desarrollo común para organizar sus tareas. A continuación se describen una serie de metodologías Agile.

2.10.1. Scrum

La metodología Scrum fue desarrollada específicamente para la gestión de proyectos agiles.

Roles dentro de un equipo Scrum:

- **Scrum Master:** es la persona encargada de gestionar el proyecto Scrum, quedando al servicio del equipo. Se ha de asegurar de que el equipo entiende y adopta la forma de trabajo Scrum.
- **Dueño del producto o product owner:** gestiona las características que ha de tener el producto final. Organiza la prioridad de las tareas, y se ha de asegurar de que el equipo entiende todos los elementos del trabajo que han de realizar.
- **Equipo de desarrollo o development team:** está compuesto por un equipo de 3 a 9 miembros, sin ninguna jerarquía determinada, que se auto gestionan. Tareas de Scrum
- **Lista de producto o product backlog:** es una lista de los requisitos del producto. El dueño del producto es el único responsable de su contenido y el orden de prioridades de las tareas. Puede actualizarse en cualquier momento.
- **Lista de pendientes del sprint o sprint backlog:** elementos de la lista de producto seleccionados para el próximo sprint.
- **Incremento o increment:** elementos de la lista de productos completados durante un sprint. Fases de Scrum
- **Sprint:** es la parte central de Scrum. Es un período de tiempo de 1 a 4 semanas, durante el cual se crea un incremento del producto. Cada nuevo sprint comienza inmediatamente después del anterior.
- **Reunión de planificación de sprint o sprint planning meeting:** en una reunión de unas 8 horas (para sprints de 4 semanas), el equipo Scrum al completo planificará lo que se va a hacer en el próximo sprint.

- **Objetivo del sprint o sprint goal:** es la meta que se ha establecido, y que se puede alcanzar completando la lista de producto.
- **Scrum diario o daily Scrum:** reunión diaria de 15 minutos donde el equipo de desarrollo crea un plan de trabajo para el día.
- **Revisión de sprint o sprint review:** reunión informal de 4 horas (para sprints de 4 semanas) en la que se inspecciona el incremento y se cambia la lista de producto si se considera necesario. En estas reuniones colaboran el equipo Scrum y cualquier otro interesado, y entre otros temas se tratan la revisión del presupuesto y de la línea de tiempo.
- **Retrospectiva de sprint o sprint retrospective:** es una reunión para el equipo Scrum en la cual se proponen mejoras para el siguiente Sprint. Tiene lugar después de la revisión de sprint, y una duración de 3 horas (para sprints de 4 semanas).

Scrum tiene como problema es que la documentación derivada no suele ser muy extensa o rigurosa, lo que puede provocar problemas en la fase de mantenimiento. En esta metodología se da prioridad a la comunicación verbal y al desarrollo de software funcional sobre la redacción de documentación.

Desventajas

- Algunos miembros del equipo pueden saltar pasos importantes en el camino rápido para llegar al sprint final.
- El cliente siempre va a esperar los informes con la fecha exacta, y muchas veces los va a pedir antes, cuando capaz no pudiste avanzar en nada.
- Demasiadas Reuniones para poco avance, a veces es muy estresante reunirse demasiadas veces por el mismo tema, algunos van perdiendo el interés en el proyecto.
- Si una persona renuncia o hay algún cambio es complicado remplazar ese rol ya que es la persona que se lleva el conocimiento específico afectando a todo el proyecto.

2.10.2. Mobile-D

La metodología Mobile-D [11] está compuesta por cinco fases:

Fase de Exploración: Esta fase es la encargada de la planificación y educción de requisitos del proyecto, donde tendremos la visión completa del alcance del proyecto y también todas las funcionalidades del producto.

Fase de inicialización: La fase de inicialización es la implicada en conseguir el éxito en las próximas fases del proyecto, donde se preparará y verificará todo el desarrollo y todos los recursos que se necesitarían. Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

Fase de producción: En la fase de producción, se vuelve a repetir la programación de los tres días, iterativamente hasta montar (implementar) las funcionalidades que se desean. Aquí usamos el desarrollo dirigido por pruebas (TDD), para verificar el correcto funcionamiento de los desarrollos.

Fase de estabilización: Se llevarán a cabo las últimas acciones de integración donde se verificará el completo funcionamiento del sistema en conjunto. De toda la metodología, esta es la fase más importante de todas ya que es la que nos asegura la estabilización del desarrollo. También se puede incluir en esta fase, toda la producción de documentación.

Fase de pruebas: Es la fase encargada del testeo de la aplicación una vez terminada. Se deben realizar todas las pruebas necesarias para tener una versión estable y final. En esta fase, si nos encontramos con algún tipo de error, se debe proceder a su arreglo pero nunca se han de realizar desarrollos nuevos de última hora, ya que nos haría romper todo el ciclo.

[11] http://agile.vtt.fi/mobiled.html

2.10.3. Metodología para el Desarrollo de Aplicaciones Móviles, MDAM:

La metodología se encuentra enmarcada en cinco fases como se muestra en la figura, denominadas: análisis, diseño, desarrollo, pruebas de funcionamiento y entrega. A continuación se describe cada una de las actividades que intervienen en el desarrollo de la propuesta.

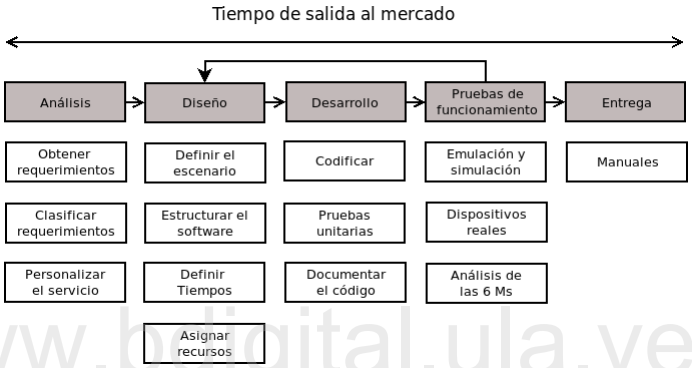


Figura 2.1: Diagrama de la metodología para el desarrollo de aplicaciones móviles.

Análisis: En esta fase se analizan las peticiones o requerimientos de las personas o entidad para la cual se desarrolla el servicio móvil, el propósito es definir las características del mundo o entorno de la aplicación. Se realizan tres tareas: obtener requerimientos, clasificar los requerimientos y personalizar el servicio.

Diseño: El objetivo de esta etapa es plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa al integrar aspectos técnicos, funcionales, sociales y económicos. A esta fase se retorna si no se obtiene lo deseado en la etapa de pruebas de funcionamiento.

Desarrollo: El objetivo de esta fase es implementar el diseño en un producto de software. En esta etapa se realizan las siguientes actividades: Codificar, pruebas unitarias, documentar el código.

Pruebas de funcionamiento: El objetivo de esta fase es verificar el funcionamiento de la aplicación en diferentes escenarios y condiciones; para esto se realizan las siguientes tareas: Emulación y simulación, dispositivos reales, análisis de las 6 Ms.

Entrega: Terminada la depuración de la aplicación y atendidos todos los requerimientos de última hora del cliente se da por finalizada la aplicación y se procede a la entrega del ejecutable, el código fuente, la documentación y el manual del sistema.

2.10.4. Metodología seleccionada para el desarrollo del producto

Después de realizar un estudio de las metodologías para el desarrollo de software en el inciso anterior, se decidió elegir la Metodología para el Desarrollo de Aplicaciones Móviles, MDAM. Por ser la que mejor se adapta al proyecto propuesto.

Capítulo 3

Análisis y Diseño

En esta fase se analizan los requerimientos de los diferentes usuarios para la cual se desarrolla la aplicación móvil, con el objetivo de plasmar mediante diagramas y esquemas la solución de sus necesidades.

3.1. Requisitos de Usuario.

- El usuario debe manejar registros y consultas de los animales bovinos que se encuentran en la unidad de producción.
- Los usuarios manejan los datos del animal mediante un número, nombre, raza, fecha de nacimiento, peso, color.
- Los productores manejan varias unidades de producción.
- Los productores llevan un manejo de vacunas, tratamientos y alimentación de los animales.
- Los productores necesitan poder consultar a veterinarios.
- Los veterinarios quieren llevar un registro de los clientes que asesoran.
- Los veterinarios desde la aplicación quieren registrar y consultar la trazabilidad de animales que mantienen un tratamiento médico.

3.2. Clasificación de requerimientos.

En esta sección se clasifican los requerimientos por entorno, mundo, funcionales y no funcionales.

3.2.1. Requerimientos de entorno.

Las características mínimas que los teléfonos deben cumplir como requisitos de hardware y software para la ejecución de la aplicación son los siguientes.

- Sistema operativo: Android de v4.4 Kit Kat.
- RAM: 512GB
- Acceso de red: plataforma compatible con TCP/IP
- Pantalla: 3.5 pulgadas

3.2.2. El mundo.

En esta clasificación se describen los requerimientos que involucran la comunicación hombre-máquina. El cual podrían definir dos tipos de interfaces.

La interfaz física:

- La interacción del usuario con la aplicación se realiza mediante una pantalla táctil, permitiendo activar las diferentes funcionalidades de la aplicación, [16].

La interfaz virtual o interfaz gráfica (GUI): En relación a la interfaz de la aplicación se ha tenido en cuenta las siguientes características:

- Facilidad de comprensión y manejo.
- Diseño de pantallas donde los usuarios puedan acceder a la aplicación, con diferentes elemento distribuidos como formularios, entradas de texto y botones que permitan activar funcionalidades de la aplicación.

3.2.3. Requisitos Funcionales.

Los requisitos funcionales permiten describir los servicios que se esperan de la aplicación.

1. La aplicación debe permitir a los productores y veterinarios registrarse.
2. Los productores y veterinarios tendrán una cuenta de usuario.
3. La aplicación permite que los usuarios puedan modificar la información de su perfil y cambiar contraseña.
4. La aplicación permitirá a los productores registrar la información básica de su unidad de producción.
5. El sistema permitirá a los productores registrar los animales con sus respectivos datos que se manejan en la finca.
6. El sistema debe permitir que los productores consulten veterinarios que se encuentren registrados en la aplicación.
7. La aplicación permite a los productores almacenar información sobre el control sanitario de los animales.
8. La aplicación debe manejar registros sobre la alimentación de los animales.
9. Los productores podrán consultar, modificar y eliminar en la aplicación información de los rebaños.
10. La aplicación permite a los veterinarios almacenar información de los productores que asesora al momento de realizar un expediente médico de un animal.

3.2.4. Requisitos No funcionales.

Describen restricciones sobre los requisitos funcionales.

1. La aplicación deberá ejecutarse correctamente en dispositivos móviles con sistema operativo Android v4.4 Kit Kat o superior.

2. La aplicación móvil se estructura en dos partes el *frontend* y *backend.*

3. El estilo arquitectónico de la aplicación móvil se basa en Cliente-Servidor, REST y Redux.

4. La aplicación no debe tardar más de 8 segundos en responder a las interacciones del usuario.

3.3. Personalizar el servicio.

Esta sección permite analizar aspectos de la cotidianidad del cliente como preferencias, costumbres y particularidades del usuario, con el propósito de garantizar la aceptación del servicio.

Características Personales: Edad: 20 y 60 años.

Ocupación: productores y veterinarios dedicados a la producción de ganadería.

Conocimientos y habilidades:

1. Disponer de un teléfono inteligente con sistema operativo Android y conocimientos básicos en su manejo.

2. Uso frecuente del dispositivo móvil

3. Manejo de rebaños sin ningún tipo de registro digital y manual.

Motivaciones:

1. Los diferentes usuarios poseen interés en innovar el manejo ganadero con la tecnología móvil.

Necesidades:

1. Los productores y veterinarios necesitan llevar el registro de animales de manera sencilla.

Color de interfaz de usuario: Se eligió el color verde que se perciben entre amarillo y el azul. Es el color principal de las diferentes pantallas de la aplicación ya que es el color característico de la vegetación y puede definirse como aquel que asemeja a la coloración de las hojas de la hierba fresca o de la piedra esmeralda [18].

Nombre de la aplicación: El nombre de la aplicación se deriva de la raza Guzerá o Guzerat es una raza brasileña de ganado doméstico . Se deriva del cruce de ganado indio Kankrej , importado a Brasil desde 1870 en adelante, con ganado taurino crioulo local de origen europeo [17], debido a esta raza de doble propósito se género el nombre GUZERAPP.

Logo de la aplicación: En la figura 3.1 se puede observar el logo de la aplicación el cual describe el rostro de un animal bovino con una pequeña silueta de su alimento fresco.

Figura 3.1: Logo de la aplicación móvil.

3.4. Diseño.

El objetivo de esta etapa es realizar el diseño de la aplicación mediante diagramas o esquemas, a esta fase se retorna si no se obtiene lo deseado en la etapa de pruebas de funcionamiento.

3.4.1. Definir el escenario.

las aplicación móvil se puede ejecutar bajo el escenario siguiente:

Conectado: Los dispositivos que ejecuten la aplicación deben estar siempre conectados a internet para su correcto funcionamiento, también se almacenan datos en el móvil, la sincronización se realiza mediante la validación de formularios, se utiliza el Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol, HTTP), ya que la aplicación deberá comunicarse con el API REST.

3.4.2. Estructurar el software.

La estructura de la aplicación se describe utilizando como perspectivas del sistema los diagramas de casos de uso, el cual permite describir la funcionalidad completa de la aplicación móvil para la gestión de información y manejo de ganado bovino adaptados a los requisitos planteados que el sistema debe cumplir mediante la interacción de los actores (agentes externos) y los casos de uso (acciones). En la figura 3.2 se muestra el diagrama de casos de uso general con sus respectivos actores. también se describe en la figura 3.14 como se estructura la base de datos mediante un modelo entidad relación.

3.4.3. Definición de Actores/Roles de la aplicación.

Los actores que tendrán interacción con la aplicación se definen de la siguiente manera:

- **Usuario:** el usuario tiene acceso a las diferentes características que el sistema ofrece dependiendo del rol que tome al momento de registrarse, puede ser un **productor** o **médico veterinario** pero no ambos a la vez y heredan las propiedades del actor usuario. a continuación se describen los roles que el usuario puede tener.

    1. **Productor:** Este rol representa a una persona que desee registrarse o ya se encuentre registrado, con un perfil de productor pecuario.

2. **Médico Veterinario:** Este rol representa al profesional que tengan o no registro en la aplicación, con un perfil de médico veterinario.

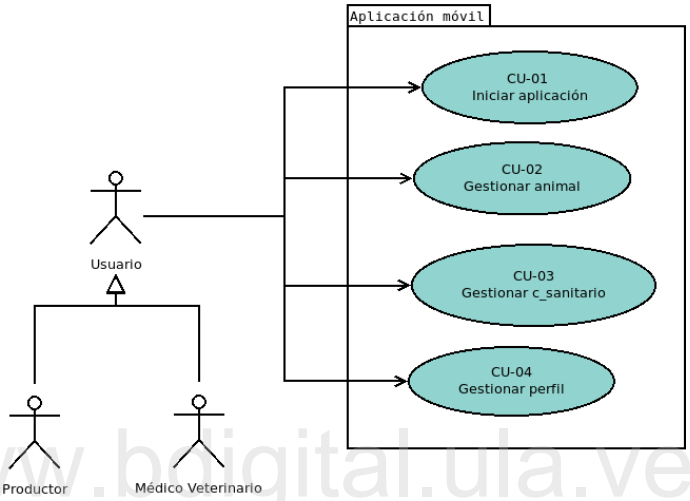


Figura 3.2: Diagrama de casos de uso general de la aplicación móvil.

3.4.4. Diagramas de casos de uso.

A continuación de describen a detalle los diferentes casos de uso que conforman la aplicación móvil.

Iniciar aplicación

En la figura 3.3 se aprecia los diferentes casos de uso que componen iniciar aplicación y se describen de manera textual el comportamiento de cada uno de ellos:

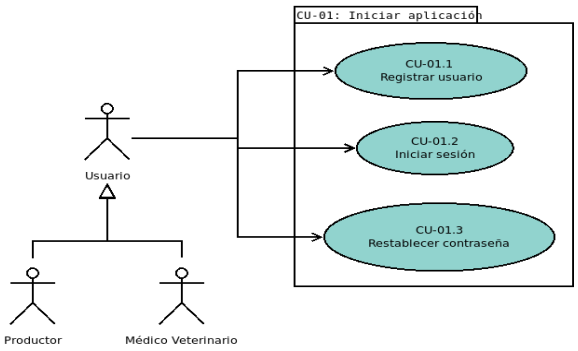


Figura 3.3: Estructura del CU-01 para iniciar la aplicación los usuarios.

Tabla 3.1: Descripción detallada del caso de uso CU-01 Iniciar aplicación.

Nombre	CU-01 Iniciar Aplicación.
Descripción	Permite al productor y veterinario registrarse, iniciar sesión o restablecer contraseña en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario no debe tener sus datos registrados en la aplicación para crear una cuenta. 2. Para iniciar sesión en la aplicación el usuario debe tener una cuenta registra.
Flujo Normal	3. El sistema solicita los datos del usuario ya sean para registrar un nuevo usuario o iniciar sesión al momento de utilizar el sistema. 4. El usuario puede iniciar sesión en la aplicación después de tener una cuenta registrada. 5. Luego de que el sistema verifique los datos del usuario para iniciar sesión, la aplicación le permitirá tener acceso a las diferentes funcionalidades dependiendo del rol. 6. El usuario puede restablecer la contraseña si posee una cuenta registrada en la aplicación en caso de olvidarla.
Flujo alternativo	5.1 La aplicación genera error de acceso al intentar iniciara sesión usuarios no registrados. 6.1 Si la validación del usuario es incorrecta se podrá recuperar la contraseña.
Post-condiciones	7. Dependiendo del rol que tenga el usuario se iniciara la aplicación con las diferentes funcionalidades.

Tabla 3.2: Descripción detallada del caso de uso CU-01.1 Registrar Usuario.

Nombre	CU-01.1 Registrar Usuario.
Descripción	Este caso de uso sucede al momento de que existan actores que deseen registrarse en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	Ninguna.
Flujo Normal	1. El usuario selecciona registrarse en la aplicación. 2. La aplicación muestra un formulario de registro. 3. El usuario debe ingresar los diferentes datos que la aplicación le solicita en el formulario de registro. 4. El usuario debe elegir uno de los roles entre productor o veterinario al momento de crear la cuenta. 5. Luego de ingresado los datos, el usuario debe seleccionar el botón registrar. 6. La aplicación comprueba si los datos son correctos para poder guardarlos. 7. Una vez guardado los datos, la aplicación envía un mensaje por pantalla que el usuario se registró con éxito.
Flujo alternativo	6.1 La aplicación al momento de comprobar la validez de los datos genera un error si el usuario ingresa valores con un formato desconocido, en caso de que esto ocurra el sistema se dirige al paso 3.
Post-condiciones	8. La aplicación genera una cuenta al usuario para que pueda iniciar sesión.

Tabla 3.3: Descripción detallada del caso de uso CU-01.2 Iniciar Sesión.

Nombre	CU-01.2 Iniciar Sesión.
Descripción	El usuario inicia sesión en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. Ninguna.
Flujo Normal	2. El usuario solicita a la aplicación iniciar sesión. 3. La aplicación solicita al usuario que introduzca el nombre de usuario y su contraseña. 4. El usuario introduce su nombre de usuario y contraseña. 5. La aplicación comprueba la validez de los datos introducidos. 6. Si los datos son correctos la aplicación muestra la ventana inicial de la cuenta perteneciente al rol correspondiente.
Flujo alternativo	5.1 Si la aplicación comprueba que el nombre de usuario y la contraseña son inválidos mostrará un mensaje de error y regresa al paso 3. 5.2 Si el usuario no accede a la aplicación. Se muestra un mensaje por pantalla que intente de nuevo el acceso y regresará al paso 3.
Post-condiciones	7. Si la aplicación comprueba que el nombre de usuario y la contraseña son correctos inicia sesión e ingresa al sistema.

Tabla 3.4: Descripción detallada del caso de uso CU-01.3 Restablecer Contraseña.

Nombre	CU-01.3 Restablecer Contraseña.
Descripción	Cuando el usuario ha olvidado la contraseña de inicio de sesión este caso de uso permite restablecerla para poder ingresar nuevamente a la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario registrado a olvidado la contraseña.
Flujo Normal	2. El usuario indica a la aplicación que olvidó la contraseña de inicio de sesión. 3. La aplicación genera por pantalla un formulario de recuperación de contraseña donde el usuario debe ingresar su correo electrónico. 4. El usuario introduce al formulario su correo electrónico. 5. La aplicación verifica si el correo electrónico es correcto y envía un código de recuperación. 6. Se solicita al usuario el código de recuperación de contraseña que la aplicación le ha enviado a su correo. 7. El usuario ingresa el código a la aplicación. 8. La aplicación comprueba que el código ingresado sea correcto. 9. La aplicación muestra un formulario donde se solicita al usuario ingresar una nueva contraseña y su respectiva confirmación. 10. La aplicación valida la nueva contraseña que será utilizada por el usuario y genera un mensaje por pantalla que el restablecimiento de contraseña fue realizado con éxito.
Flujo alternativo	5.1 Si el usuario ingresa un correo electrónico de manera incorrecta con un formato inválido, la aplicación lo indicará mediante un mensaje por pantalla y regresara al paso 2. 5.2 Si el correo electrónico no se encuentra almacenado en la aplicación regresa al paso 2. 8.1 Si el usuario ingresa el código de recuperación de manera incorrecta la aplicación genera un mensaje de error y regresa al paso 5.
Post-condiciones	11. El usuario ha restablecido su contraseña para iniciar sesión en la aplicación.

Gestionar animal

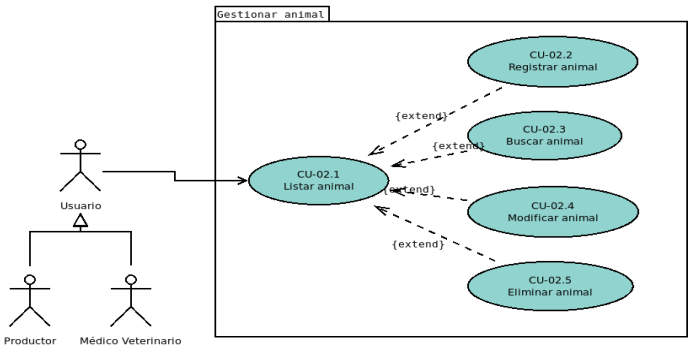


Figura 3.4: Estructura del CU-02 para gestionar animal.

Tabla 3.6: Descripción detallada del caso de uso CU-02.1 Listar animal.

Nombre	CU-02.1 Listar animal.
Descripción	Permite a los usuarios listar los animales registrados en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario elige en la aplicación la opción de una finca registrada. 3. La aplicación muestra una lista de los animales registrados. 4. El usuario puede elegir un animal de la lista al seleccionarlo manualmente. 5. La aplicación ofrece al usuario buscar un animal mediante un campo de texto. 6. La aplicación también ofrece registrar nuevos animales a la lista. 7. El usuario selecciona la opción de su preferencia.
Flujo alternativo	3.1 La aplicación genera un mensaje al usuario que no existen animales registrados en caso de estar la lista vacía.
Post-condiciones	8. El usuario consulta la lista de animales registrados en la aplicación.

Tabla 3.5: Descripción detallada del caso de uso CU-02 Gestionar animal.

Nombre	CU-02 Gestionar animal.
Descripción	Permite a los productores llevar el control de cada animal de manera organizada, permitiendo listar, registrar, buscar, modificar y eliminar los animales en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. Los actores deben haber iniciado sesión en la aplicación.
Flujo Normal	2. El sistema lista los animales que se encuentran registrados. 3. La aplicación permite registrar nuevos animales. 4. La aplicación permite buscar animales por número. 5. La aplicación muestra los detalles del animal consultado por el actor. 6. Los usuarios pueden eliminar animales que se encuentren registrados. 7. La aplicación permite modificar datos de animales ya registrados.
Flujo alternativo	3.1 En caso de registrar un animal con un número existente el sistema no permitiría agregarlo solo modificar los datos existentes.
Post-condiciones	8. El actor termina el manejo de gestion animal por otra funcionalidad de la aplicación.

Tabla 3.7: Descripción detallada del caso de uso CU-02.2 Registrar animal.

Nombre	CU-02.2 Registrar animal.
Descripción	Permite a los usuarios registrar nuevos animales en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario utiliza en la aplicación el CU-02.1 listar animal. 3. El usuario selecciona la opción **Registrar animal**. 4. La aplicación despliega un formulario con los campos a llenar. 5. El usuario debe ingresar los datos solicitados en el formulario. 6. El sistema mostrará en la interfaz un boton que permita guardar el registro. 7. El usuario elige guardar. 8. La aplicación debe mostrar un mensaje por pantalla que el animal fue guardado con éxito.
Flujo alternativo	5.1 La aplicación genera un mensaje de alerta al usuario si los campo del formulario tienen datos inválidos. Y regresa al paso 4. 8.1 La aplicación debe mostrar un mensaje de error si no se pudo registrar el animal. Y regresa al paso 4.
Post-condiciones	9. El animal queda registrado en la aplicación.

Tabla 3.8: Descripción detallada del caso de uso CU-02.3 Buscar animal.

Nombre	CU-02.3 Buscar animal.
Descripción	Permite a los usuarios buscar animales en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El actor debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario utiliza en la aplicación el CU-02.1 listar animal. 3. El usuario elige en la aplicación la opción de buscar animal. 4. La aplicación muestra un campo de texto para la consulta de animales registrados. 5. El actor digitara en el campo de texto los criterios de búsqueda. 6. La aplicación valida los datos introducidos. 7. La aplicación hace verificación de la existencia del registro consultado y despliega: 7.1. Listado con él o los animales que coinciden con la búsqueda. 7.2. ningún animal encontrado en la búsqueda. 8. Si en la búsqueda se genera más de un animal el usuario debe: 8.1. seleccionar un animal manualmente. 8.2. Redefinir la búsqueda.
Flujo alternativo	7. La aplicación debe mostrar un mensaje por pantalla si no se encuentra coincidencia del animal con datos ingresados. Y regresa al paso 3.
Post-condiciones	9. El resultado de la búsqueda es mostrada por pantalla si se encuentra el animal.

Tabla 3.9: Descripción detallada del caso de uso CU-02.4 Modificar animal.

Nombre	CU-02.4 Modificar animal.
Descripción	Permite a los usuarios modificar los datos de animales en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. Después que el usuario selecciona un animal, la aplicación desplegará las siguientes opciones: 2.1. Modificar animal. 2.2. Eliminar animal. 3. El usuario elige modificar animal. 4. La aplicación debe mostrar los datos existentes del animal en un formulario para que puedan ser modificados. 5. El usuario cambia los registros del animal. 6. La aplicación mostrará el botón de guardar. 7. El usuario selecciona guardar. 8. La aplicación debe mostrar un mensaje por pantalla que los datos fueron modificados con éxito.
Flujo alternativo	8.1. En caso de que la aplicación no realice la modificación del animal se mostrará por pantalla un mensaje que el registro no pudo ser modificado. Y regresa al paso 2.
Post-condiciones	9. El registro del animal se modificó en la aplicación.

Tabla 3.10: Descripción detallada del caso de uso CU-02.5 Eliminar animal.

Nombre	CU-02.5 Eliminar animal.
Descripción	Permite a los usuarios eliminar animales en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. Después que el usuario selecciona un animal, la aplicación desplegará las siguientes opciones: 2.1. Modificar animal. 2.2. Eliminar animal. 3. El usuario elige eliminar animal. 4. La aplicación desplegará por pantalla un mensaje con las siguientes opciones: 4.1. Aceptar. 4.2. Cancelar. 5. El actor elige aceptar. 6. La aplicación eliminará el registro del animal. 7. La aplicación debe mostrar por pantalla un mensaje de que el animal fue eliminado.
Flujo alternativo	5.2. Si el usuario elige cancelar. La aplicación regresa al paso 2. 7.1. En caso de que la aplicación no elimine el registro del animal se mostrará por pantalla un mensaje que el animal no fue eliminado. Y regresa al paso 2.
Post-condiciones	8. El registro del animal se eliminó de la aplicación.

Gestionar control sanitario

En la figura 3.5 se aprecia los diferentes casos de uso que componen gestionar el control sanitario del animal y se describen de manera textual el comportamiento de cada uno de ellos:

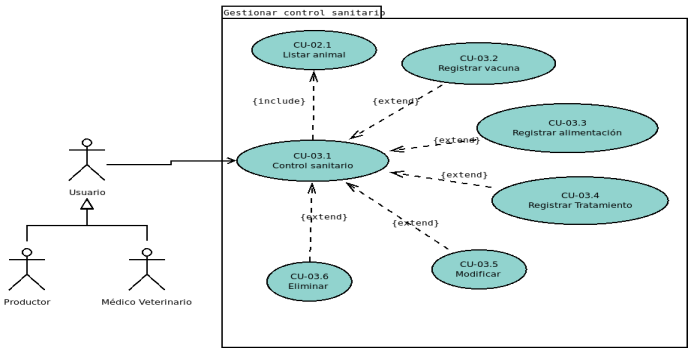


Figura 3.5: Estructura del CU-03 para gestionar el control sanitario.

Tabla 3.11: Descripción detallada del caso de uso CU-03 Gestionar control sanitario.

Nombre	CU-03 Gestionar control sanitario.
Descripción	Permite a los usuarios gestionar el control sanitario de animales en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. La aplicación muestra una lista de los animales registrados. 3. El usuario puede elegir de la lista un animal. 4. El sistema ofrece al usuario elegir una opción entre: 4.1. Registrar vacuna. 4.2. Registrar alimentación. 4.3. Registrar tratamiento. 5. El sistema ofrece al usuario las opciones siguientes para vacunas, alimentación y tratamiento: 5.1. Modificar. 5.2. Eliminar. 6. El usuario luego de seleccionar una de las opciones podrá asignar datos de sanidad al animal.
Flujo alternativo	Ninguno.
Post-condiciones	6. La aplicación retorna por pantalla el registro de vacunas, alimentación o tratamientos.

Tabla 3.12: Descripción detallada del caso de uso CU-03.2 Registrar vacuna.

Nombre	CU-03.2 Registrar vacuna.
Descripción	Permite a los usuarios registrar vacunas en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario elige en la aplicación la opción de control sanitario. 3. La aplicación ofrece al usuario seleccionar una de las siguientes opciones: 3.1. Registrar vacuna. 3.2. Registrar Tratamiento. 3.3. Registrar alimentación 3.4. modificar. 3.5. eliminar. 4. El usuario elige en la aplicación la opción **Registrar vacuna**. 5. La aplicación muestra una lista de vacunas registradas y desplegará una opción para registrar una nueva vacuna. 6. El usuario selecciona la opción de registrar nueva vacuna. 7. La aplicación desplegará un formulario con campos a llenar sobre los datos de la vacuna. 8. El usuario debe ingresar los datos solicitados en el formulario. 9. El sistema mostrará en la interfaz el boton de guardar vacuna. 10. La aplicación debe mostrar un mensaje por pantalla que la vacuna fue registrada con éxito.
Flujo alternativo	8.1. La aplicación genera un mensaje de alerta al usuario si los campo del formulario tienen datos inválidos. Y regresa al paso 7. 10.1. La aplicación debe mostrar un mensaje de error si no se pudo registrar la vacuna. Y regresa al paso 4.
Post-condiciones	11. La vacuna queda registrada en la aplicación.

Tabla 3.13: Descripción detallada del caso de uso CU-03.3 Registrar alimentación.

Nombre	CU-03.3 Registrar alimentación.
Descripción	Permite a los usuarios registrar alimentación en la aplicación.
Actor	Usuario (Productor).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario elige en la aplicación la opción de control sanitario. 3. La aplicación ofrece al usuario seleccionar una de las siguientes opciones: 3.1. Registrar vacuna. 3.2. Registrar Tratamiento. 3.3. Registrar alimentación 3.4. modificar. 3.5. eliminar. 4. El usuario elige en la aplicación la opción **Registrar alimentación**. 5. La aplicación muestra una lista de alimentación por animal registrado y desplegará una opción para registrar un nuevo registro de alimentación. 6. El usuario selecciona la opción de registrar alimentación. 7. La aplicación desplegará un formulario con campos a llenar sobre los datos de la alimentación. 8. El usuario debe ingresar los datos solicitados en el formulario. 9. El sistema mostrará en la interfaz el botón de guardar alimentación. 10. La aplicación debe mostrar un mensaje por pantalla que la alimentación fue registrada con éxito.
Flujo alternativo	8.1. La aplicación genera un mensaje de alerta al usuario si los campo del formulario tienen datos inválidos. Y regresa al paso 7. 10.1. La aplicación debe mostrar un mensaje de error si no se pudo registrar la alimentación. Y regresa al paso 4.
Post-condiciones	11. La vacuna queda registrada en la aplicación.

Tabla 3.14: Descripción detallada del caso de uso CU-03.4 Registrar Tratamiento.

Nombre	CU-03.4 Registrar tratamiento.
Descripción	Permite a los usuarios registrar un tratamiento en la aplicación.
Actor	Usuario (Productor).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario elige en la aplicación la opción de control sanitario. 3. La aplicación ofrece al usuario seleccionar una de las siguientes opciones: 3.1. Registrar vacuna. 3.2. Registrar Tratamiento. 3.3. Registrar alimentación 3.4. modificar. 3.5. eliminar. 4. El usuario elige en la aplicación la opción **Registrar tratamiento**. 5. La aplicación muestra una lista de tratamientos registrados y desplegará una opción para registrar un nuevo registro de tratamiento. 6. El usuario selecciona la opción de registrar tratamiento. 7. La aplicación desplegará un formulario con campos a llenar sobre los datos del tratamiento. 8. El usuario debe ingresar los datos solicitados en el formulario. 9. El sistema mostrará en la interfaz el botón de guardar tratamiento. 10. La aplicación debe mostrar un mensaje por pantalla que el tratamiento fue registrado con éxito.
Flujo alternativo	8.1. La aplicación genera un mensaje de alerta al usuario si los campo del formulario tienen datos inválidos. Y regresa al paso 7. 10.1. La aplicación debe mostrar un mensaje de error si no se pudo registrar el tratamiento. Y regresa al paso 4.
Post-condiciones	11. El tratamiento queda registrada en la aplicación.

Tabla 3.15: Descripción detallada del caso de uso CU-03.5 Modificar.

Nombre	CU-03.5 Modificar.
Descripción	Permite a los usuarios modificar los datos de vacuna, alimentación o tratamiento en la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario selecciona una de las opciones de sanidad animal como vacunas, alimentación o tratamientos. 3. El sistema ofrece al usuario elegir una opción entre: 3.1. Modificar. 3.2. Eliminar. 4. El usuario elige modificar. 5. La aplicación debe mostrar los datos existentes de la opción elegida ya sea vacuna, alimentación o tratamiento en un formulario para que puedan ser modificados. 6. El usuario cambia los registros en el formulario. 7. La aplicación mostrará los botones siguientes: 7.1. Guardar. 7.2. Cancelar. 8. El usuario selecciona guardar. 9. La aplicación debe mostrar un mensaje por pantalla que los datos fueron modificados con éxito.
Flujo alternativo	7.2. Si el usuario elige cancelar. La aplicación regresa al paso 3. 9.1. En caso de que la aplicación no realice la modificación de la vacuna se mostrará por pantalla un mensaje de que no pudo ser modificada. Y regresa al paso 3.
Post-condiciones	10. El registro de la opción que elija el usuario se modificó en la aplicación.

Tabla 3.16: Descripción detallada del caso de uso CU-03.6 Eliminar.

Nombre	CU-03.6 Eliminar.
Descripción	Permite a los usuarios eliminar una de las opciones que el usuario seleccione ya sean vacunas, tratamiento o alimentación, de la aplicación.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. La aplicación muestra un menú a elegir entre las vacunas, alimentación o tratamientos. 3. El usuario selecciona una de las opciones. 4. La aplicación muestra una lista de las vacunas, alimentación o tratamientos registrados. 5. El usuario selecciona un registro de la lista. 6. El sistema ofrece al usuario elegir una opción entre: 6.1 Modificar. 6.2 Eliminar. 7. El usuario elige eliminar. 8. La aplicación envía un mensaje de comprobación por pantalla si está seguro de eliminar el registro. 9. La aplicación desplegará los siguientes botones: 9.1 Aceptar. 9.2 Cancelar. 10. El usuario selecciona aceptar. 11. La aplicación debe eliminar el registro seleccionado y mostrar un mensaje por pantalla que se eliminó con éxito.
Flujo alternativo	9.2. Si el usuario elige cancelar. La aplicación regresa al paso 3. 11.1. En caso de que la aplicación no pueda eliminar el registro se mostrará por pantalla un mensaje que la vacuna no pudo ser eliminada. Y regresa al paso 3.
Post-condiciones	12. El registro se eliminó de la aplicación.

Gestionar perfil

En la figura 3.6 se aprecia los diferentes casos de uso que componen el caso de uso general para gestionar el perfil de los usuarios y se describen de manera textual el comportamiento de cada uno de ellos:

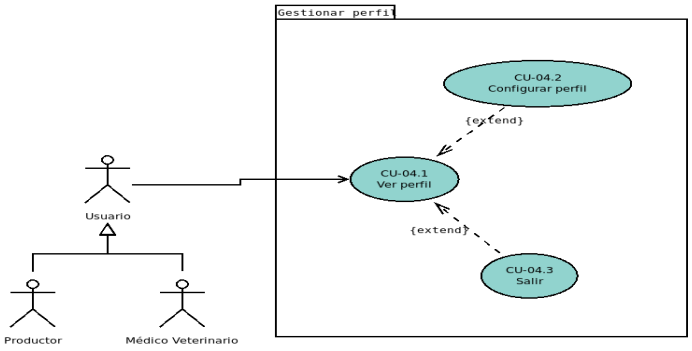


Figura 3.6: Estructura del CU-04 gestionar perfil de los usuarios.

Tabla 3.17: Descripción detallada del caso de uso CU-04 Gestionar perfil.

Nombre	CU-04 Gestionar perfil
Descripción	Permite a los usuarios manejar y configurar su perfil.
Actor	Usuario (Productor, Médico Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. La aplicación ofrece a los usuarios ver y configurar su perfil. 3. La aplicación permite ver los veterinarios que se encuentran registrados en la aplicación. 4. La aplicación permite cerrar sesión.
Flujo alternativo	Ninguno.
Post-condiciones	5. El usuario puede configurar el perfil.

Tabla 3.18: Descripción detallada del caso de uso CU-04.2 Configurar perfil.

Nombre	CU-04.2 Configurar perfil.
Descripción	Permite a los usuarios configurar su perfil.
Actor	Usuario (Productor, Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario selecciona Perfil en el menú principal de la aplicación. 3. La aplicación le mostrará sus datos de usuario. 4. La aplicación permitirá modificar datos como son su nombre de inicio de sesión y su contraseña. 5. La aplicación mostrará un botón de aceptar si realiza cambios del perfil. 6. Los datos se modificarán en la base de datos.
Flujo alternativo	6.1. Si los datos no son correctos se mostrará un mensaje de error debajo de los campos correspondientes.
Post-condiciones	7. La información del perfil de usuario queda actualizada en la aplicación.

Tabla 3.19: Descripción detallada del caso de uso CU-04.3 Salir.

Nombre	CU-04.3 Salir.
Descripción	Permite a los usuarios salir de la aplicación.
Actor	Usuario (Productor, Veterinario).
Pre-condiciones	1. El usuario debe tener activa su respectiva sesión.
Flujo Normal	2. El usuario selecciona Salir en el menú principal de la aplicación. 3. La aplicación cierra la sesión de usuario. 4. La aplicación se redirecciona al CU-01 Iniciar aplicación.
Flujo alternativo	Ninguno.
Post-condiciones	5. La sesión de usuario queda cerrada.

3.4.5. Diagramas de secuencias.

En esta sección se ilustran algunos diagramas de secuencias para notar la comunicación y funcionalidad de la aplicación propuesta.

El diagrama 3.7 describe la secuencia que todo usuario no registrado debe cumplir para crear una cuenta en la aplicación.

El diagrama 3.8 muestra la funcionalidad de iniciar sesión, este diagrama de secuencia describe la interacción del usuario al momento de interactuar con la aplicación, y el comportamiento ideal que el servidor toma al momento de autenticarse el usuario.

En el diagrama de la Figura 3.10, esta la secuencia de acciones que el usuario debe realizar para salir de la aplicación.

Si el usuario que interactúa con la aplicación tiene un rol de productor, la aplicación le permite registrar animales por finca, ver diagrama 3.11.

Si el usuario tiene animales registrados en la unidad de producción puede realizar las operaciones de modificar y eliminar animal, el cual se describen en el diagrama de secuencia 3.12 y el diagrama 3.13

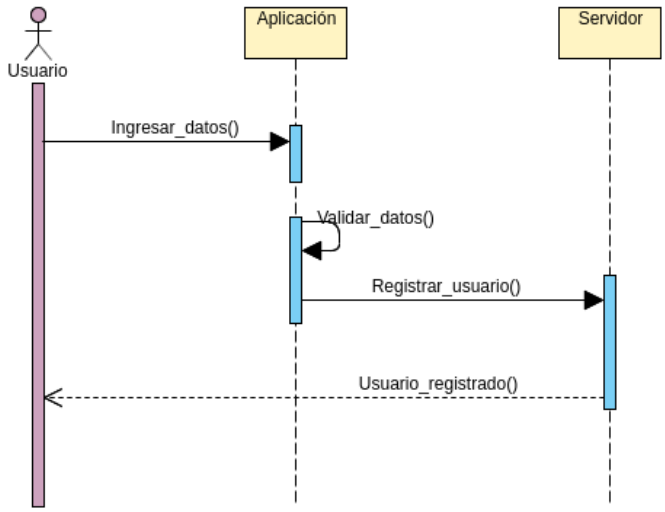


Figura 3.7: Diagrama registrar usuario.

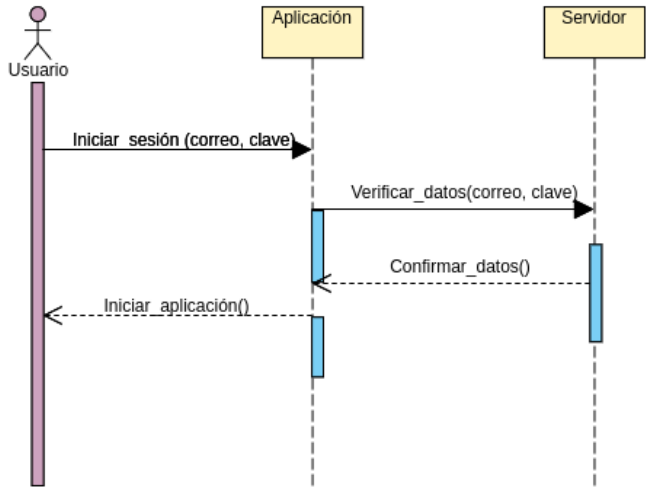


Figura 3.8: Diagrama de iniciar sesión.

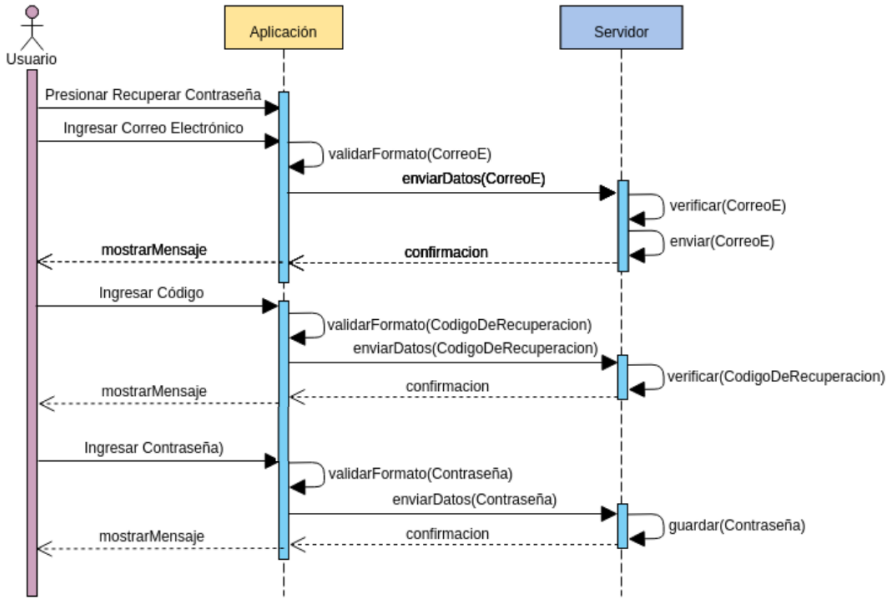


Figura 3.9: Diagrama de recuperación de contraseña.

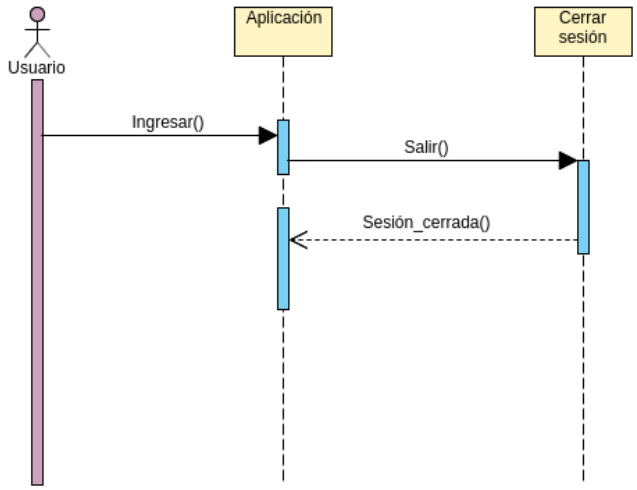


Figura 3.10: Diagrama de cerrar sesión.

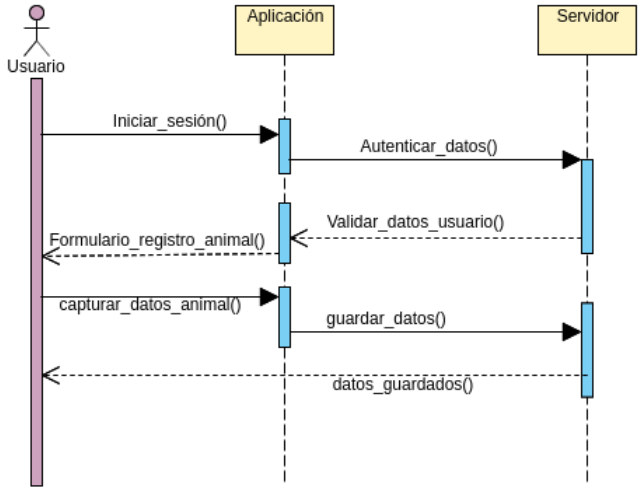


Figura 3.11: Diagrama registro de animal.

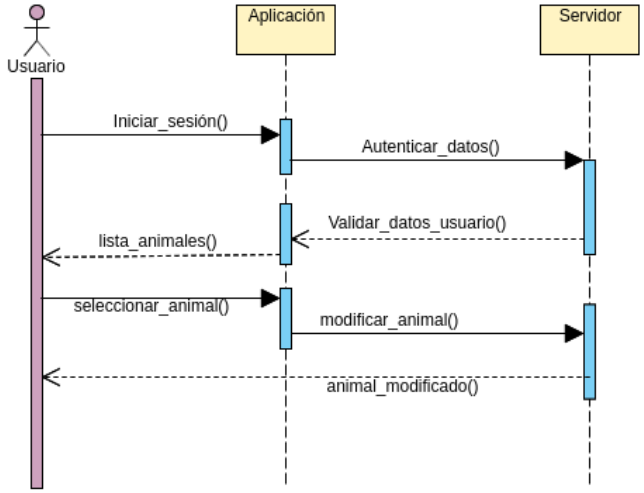


Figura 3.12: Diagrama para modificar animal.

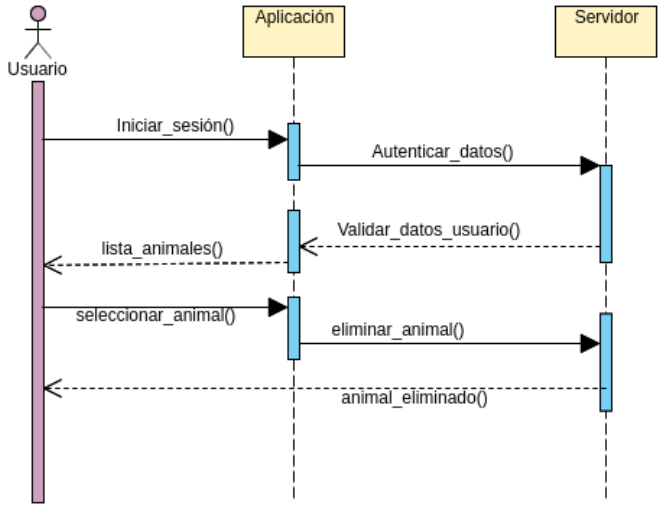


Figura 3.13: Diagrama para eliminar animal.

3.4.6. Diseño de la base de datos

A continuación se representa el modelo de la base de datos que la aplicación utilizara, representados por un conjunto de entidades y relaciones.

Una vez culminado el modelo ERE se realizó el diseño lógico de la base datos, para poder ser implementado en un SGBD, para esto fue necesario construir el siguiente modelo relacional utilizando el mapeado ERE a relacional.

Modelo relacional: Para la representación del modelo relacional se utiliza la notación R(A1, A2, ..., An) conocida también como extensión de relación [19], donde la letra R especifica el nombre de la relación y las letras A1, A2, ..., An sus atributos. Los atributos clave se representan subrayados y la clave foránea que representa la relación entre entidades se representan con sus nombres en cursiva.

Usuario (contraseña, correo, nombre, apellido, teléfono, dirección)
Productor (*correoP*)
Veterinario (*correoV*, rmv)
Cliente (id, nombre, dirección, *correoV*)
Hist_medica (id, num_animal, sexo, *clienteId*)

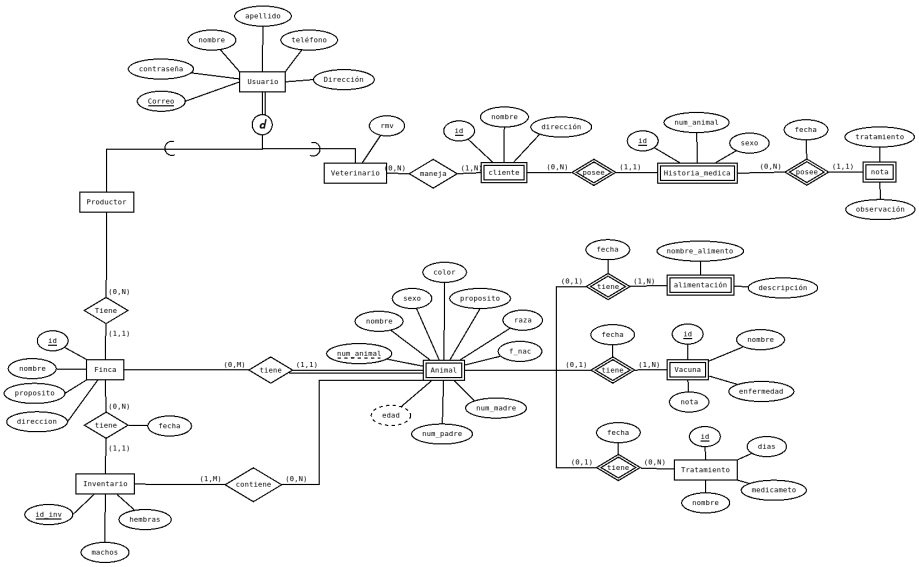


Figura 3.14: Modelo Entidad Relación Extendido.

Nota (id, fecha, tratamiento, observación, *historiaId*)
Finca (id, nombre, propósito, dirección, *idp*)
Animal(id, *id_finca*, num_animal, nombre, sexo, color, propósito, raza, f_nacimiento, edad, num_madre, num_padre)
Inventario (id_inventario, *id_animal*, machos, hembras, *id_finca*)
Alimentación (id, nombre, descripción, fecha, *id_animal*)
Vacuna(id, nombre, enfermedad, nota, fecha, *id_animal*)
Tratamiento(id, medicamento, nombre, días, fecha, *id_animal*)

Normalización de la base de datos: Una vez concluido el modelo relacional es necesario realizar el proceso de normalización de la base de datos el cual se llevará hasta la 3FN, con el fin de minimizar anomalías en la base de datos.

Primera forma normal (1FN): Una relación está en 1FN si todos los atributos de la relación tienen valores atómicos.

las relaciones tienen atributos con valores atómicos, pero se aprecia que existen redundancia de datos ya que existen atributos que pueden tomar valores que se re-

piten para distintas tuplas de una misma relación, por lo tanto es necesario dividir estos datos en relaciones separadas para resolver este problema.

Relación Animal:

La relación Animal presenta problemas de redundancia ya que los atributos sexo, color, propósito y raza tienen valores que se pueden repetir para distintos animales, para resolver este problema es necesario descomponer la relación en las siguientes relaciones.

- **Sexo**(*id*, tipo)
- **Color**(*id*, nombre)
- **Propósito**(*id*, nombre)
- **Raza**(*id*, nombre)
- **Animal**(id, *id_finca*, num_animal, nombre, *idsexo*,*idcolor*, *idpropósito*, *idraza*, f_nacimiento, edad, num_madre, num_padre)

De esta manera toda las relaciones cumplen con la primera forma normal 1FN.

Segunda forma normal (2FN): Una relación está en 2FN si cumple la 1FN y todo atributo que no forma parte de la clave depende funcionalmente de toda la clave y no de una parte de ella. Esto quiere decir que toda relación que posee solo un atributo clave cumple son la segunda forma normal. Por lo tanto todas las relaciones cumplen con la regla 2FN.

Tercera forma normal (3FN): Una relación está en 3FN si cumple la 2FN y todo atributo que no forma parte de la clave no depende funcionalmente de un atributo no clave. Por lo tanto las relaciones que no poseen más de un atributo no clave cumplen de manera directa con la 3FN.

Relación Hist_medica:

Hist_medica (id, num_animal, sexo, *clienteId*)

Relación Animal:

Animal(id, *id_finca*, num_animal, nombre, *idsexo*,*idcolor*, *idpropósito*, *idraza*, f_nacimiento, edad, num_madre, num_padre)

En las relaciones HIST_MEDICA y ANIMAL el atributo *num_animal* determina funcionalmente al resto de los atributos que no son clave en la relación, pero cabe resaltar que el atributo num_animal es clave candidata, solo aparecerá una vez por cada tupla de la relación, por lo tanto no se generarán problemas de actualización que es la causa común que se genera en las relaciones que no cumplen la tercera forma normal, debido a esto no fue necesario descomponer la relación. Por lo tanto se concluye que el conjunto de relaciones cumplen con la 3FN.

Modelo Relacional Normalizado:

Usuario (contraseña, correo, nombre, apellido, teléfono, dirección)
Productor (*correoP*)
Veterinario (*correoV*, rmv)
Cliente (id, nombre, dirección, *correoV*)
Hist_medica (id, num_animal, sexo, *clienteId*)
Nota (id, fecha, tratamiento, observación, *historiaId*)
Finca (id, nombre, propósito, dirección, *idp*)
Sexo(*id*, tipo)
Color(*id*, nombre)
Propósito(*id*, nombre)
Raza(*id*, nombre)
Animal(id, *id_finca*, num_animal, nombre, *idsexo*,*idcolor*, *idpropósito*, *idraza*, f_nacimiento, edad, num_madre, num_padre)
Inventario (id_inventario, *id_animal*, machos, hembras, *id_finca*)
Alimentación (id, nombre, descripción, fecha, *id_animal*)
Vacuna(id, nombre, enfermedad, nota, fecha, *id_animal*)
Tratamiento(id, medicamento, nombre, días, fecha, *id_animal*)

Esquema Físico:

En la figura 3.15 se muestra el esquema físico de la base de datos representado por un diagrama UML. En la imagen se pueden observar las tablas y sus respectivos atributos, así como las relaciones entre las ellas.

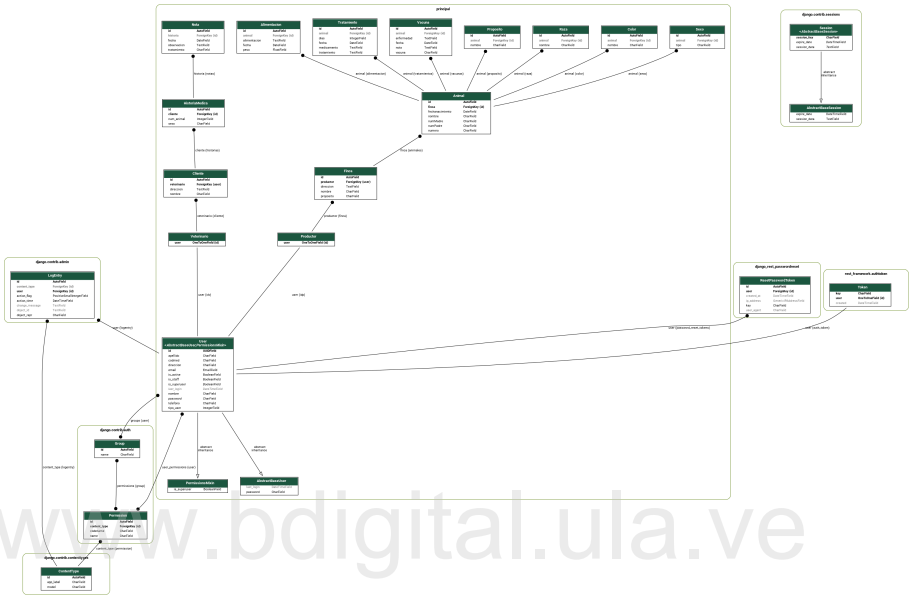


Figura 3.15: Diagrama UML del esquema físico de la base de datos.

3.4.7. Diseño Técnico

La arquitectura del sistema propuesto estará estructurada por una aplicación móvil y una API REST. Utilizando los siguientes estilos arquitectónicos.

Cliente-Sevidor:

La funcionalidad del sistema se basa en la arquitectura cliente-servidor. Ya que los dispositivos cliente proporcionan una interfaz de usuario para productores o veterinarios, permitiendo que realicen solicitudes al servidor y mostrar los resultados que el servidor devuelve. El servidor espera que lleguen las solicitudes de los clientes y luego responden a ellos [7].

Los clientes suelen estar ubicados en teléfonos inteligentes con la aplicación instalada, mientras el servidor se encuentra en otra parte de la red, generalmente en máquinas más potentes, donde se encuentra alojada el API REST que es la encargada de manejar las funcionalidades de procesamiento de datos. En la figura 3.16 se describe la arquitectura cliente-servidor.

Figura 3.16: Arquitectura Cliente-Servidor.

REST:

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP (uno de los protocolos más antiguos) y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON. En esta arquitectura la información en el servidor se organiza por recursos, cada recurso se representa mediante un punto final (end-point o url). Para manipular cada recurso se utilizan los verbos GET para obtener recursos, POST que permite crear recursos, PUT modificar y DELETE borrar un recurso. La utilidad de la arquitectura REST para el desarrollo del sistema es que permite separar el cliente del servidor.

Cliente:

Del lado del cliente la aplicación propuesta se basa en la arquitectura Redux el cual está basada en el patrón de diseño Flux. Se encarga en cierta manera de desacoplar el estado global de una aplicación web (en Front-End) de la parte visual, es decir los componentes [1]. El estado normalmente se trata de los datos que puede recibir a través del API REST (por ejemplo lista de animales), también se refiere al estado de la interfaz gráfica en determinados momentos. En la figura 3.17 se puede apreciar a detalle la arquitectura redux.

La arquitectura Redux se base en los siguientes actores:

1. **Una única "fuente de la verdad":** Flux propone que haya varios Stores

para almacenar el estado. Sin embargo Redux simplifica esto utilizando un único Store. Todo el estado queda almacenado en un árbol. En JavaScript esto se conseguiría con un objeto JavaScript

2. **El estado es de sólo lectura.** No podemos modificar el estado directamente, sólo podemos leer de él para representarlo en la vista y si queremos modificarlo, lo tenemos que hacer a través de acciones. Una acción es simplemente un objeto JavaScript que incluye al menos un atributo type que indica el tipo de acción que estamos emitiendo y en caso de que haya datos asociados al cambio o modificación, un atributo payload.

3. **Cambios con funciones puras.** Ya que no podemos modificar el estado directamente (tiene que ser a través de acciones) y el estado está almacenado en un único Store, para especificar cómo realizar los cambios en el árbol del estado utilizamos funciones puras llamadas Reducers. Una función pura es simplemente una función que ante los mismos datos de entrada devuelve el mismo resultado. De ésta manera es más sencillo depurar y encontrar errores, y es más fácil testear. El reducer es simplemente eso, una función que recibe dos parámetros, el estado inicial y una acción y dependiendo del tipo de acción realizará una operación u otra en el estado. Siempre de manera inmutable, no podemos modificar el estado, si no crear una copia a partir del anterior. De esta forma es más fácil rastrear posibles errores.

Servidor:

La arquitectura utilizada para el servidor está basada en Django que es conocido como un Framework MTV. El patrón de diseño Model/Template/View se describe de manera resumida a continuación.

- **M** significa "Model"(Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

- **T** significa "Template"(Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.

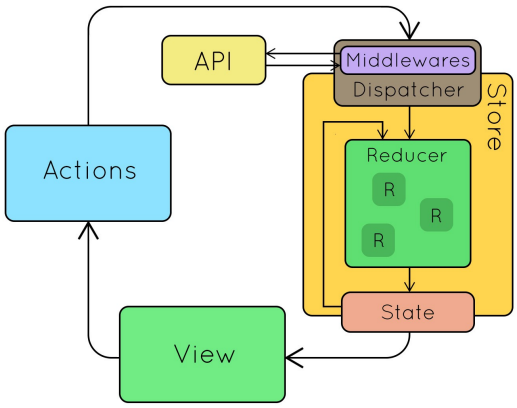


Figura 3.17: Arquitectura del Cliente.

- **V** significa "View"(Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelos y las plantillas.

Debido a que el servidor ejecuta un API REST, la vista será la encargada de enviar las respuestas al cliente ya que cada una tiene una ruta asociada a las solicitudes que los usuarios realicen, por lo que el motor de plantillas de django será sustituida por la librería Django_rest_framework que será la encargada de renderizar los datos en formato JSON. Es por esto que la arquitectura del servidor se puede representar tal y como se muestra en la figura 3.18

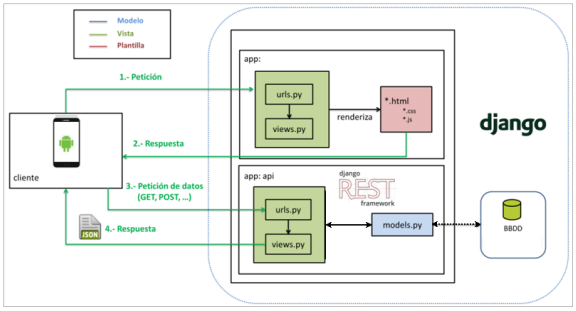


Figura 3.18: Arquitectura del Servidor.

3.4.8. Patrón de diseño

La aplicación se basa en el patrón de diseño modelo/vista/plantillas o (MVT) por parte del servidor el cual esta basado en el modelo/vista/controlador o MVC [9]., ya que su estructura se basa en separar los datos de la aplicación, la interfaz de usuario, y la lógica de control en diferentes módulos. Por parte del cliente se utilizaron los patrones de diseño de React Native el cual fueron tomados de las fuentes [2] y [3].

Patrón Componente Contenedor: Los componentes contenedores se encargan de manipular los datos, de mantener un estado y de pasar propiedades (props) a otros componentes, como los componentes de presentación, que son componentes que no mantienen ningún estado y que se ocupan únicamente de mostrar datos en la interfaz de usuario.

Patrón Componente de Orden Superior: Los componentes de orden superior (HOC) son un patrón de React que permiten la encapsulación de un componente en otro componente. Mientras los componentes de React normales, transforman propiedades en interfaz de usuario, los HOC transforman un componente en un nuevo componente. Este patrón permite la reutilización de la lógica entre componentes, agregando funcionalidades a componentes ya existentes.

Patrón Render Props: Este patrón de React permite compartir código entre componentes utilizando una función, que devuelve otro componente, y que es pasada como propiedad entre componentes. De este modo, se puede delegar de un componente a otro lo que se va a renderizar utilizando funciones.

Patrones de Navegación: Los patrones de navegación establecen la manera en la que se deben llevar a cabo las transiciones entre pantallas de la aplicación. A continuación se describen los utilizados en el desarrollo.

- **Patrón Stack Navigator:** Este es el patrón de navegación más común de las aplicaciones móviles, en la cual la transición entre pantallas se lleva a cabo agregando cada nueva pantalla en la cima de un contenedor llamado pila (donde

el primero en entrar será el último en salir) y, si el usuario desea volver a la pantallas anterior, simplemente se saca la última pantalla de la pila.

- **Patrón Switch Navigator:** El propósito de este patrón es montar una sola pantallas a la vez, por lo que cada pantallas se monta cuando se navega hacia esta y se desmonta cuando se navega a una nueva pantalla. Este patrón fue utilizado para navegar de la pila de navegación de autenticación de usuarios a la pila principal de usuario ya autenticados.

- **Patrón Drawer Navigation:** Este patrón incluye un menú desplegable en la parte izquierda de la pantalla desde la cual el usuario podrá acceder mediante el botón que se muestra en la parte superior izquierda y navegar a otras pantallas. Los componentes de cada pantalla no se montan hasta que son seleccionados en el menú.

- **Patrón Bottom Tab Navigation:** En este patrón se incluye una barra en la parte inferior de la pantalla desde la cual el usuario podrá navegar a otras pantallas. Los componentes de cada pantalla no se montan hasta que son seleccionados en la barra por primera vez.

3.4.9. Definición de tiempos.

Para definir el tiempo de duración para cada una de las etapas restantes de la metodología para el desarrollo de aplicaciones móviles, fue necesario describirlo mediante un diagrama de Gantt mostrado en la figura 3.19. En este diagrama la línea de tiempo está dividida en semanas y se puede apreciar que existen actividades que se desarrollan en paralelo independientemente de la etapa a la que pertenezca.

3.4.10. Recursos.

Como recursos tecnológicos fueron utilizados los siguientes equipos:

- Laptop NB-3300 Siragon, Intel Pentium 2030M, 4GB DDR3, Linux Mint 19.2.

- Teléfono Samsung S5, Snapdragon 2.5GHz, 2 GB, Android 6.0.

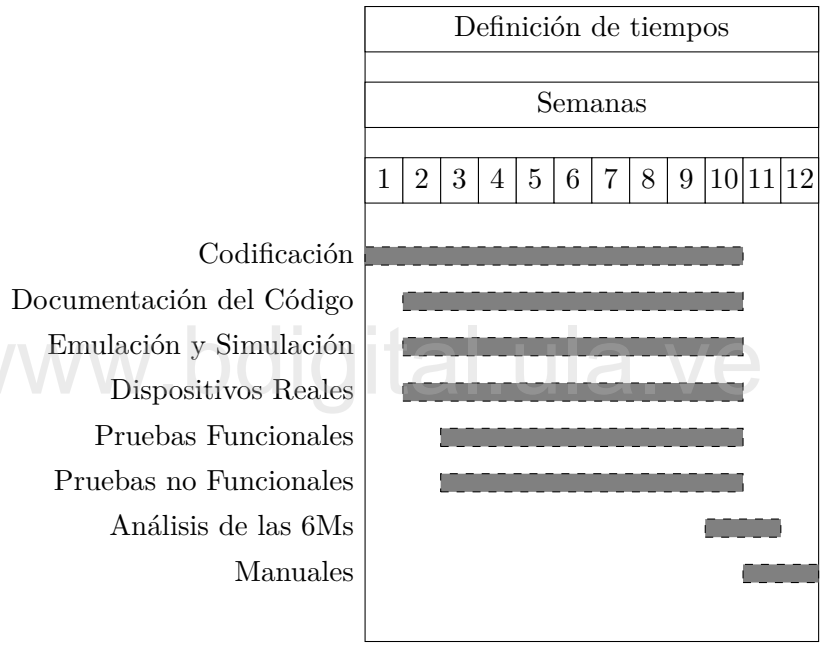


Figura 3.19: Diagrama de tiempo para las etapas faltantes de la metodología (MDAM).

Capítulo 4

Codificación y Pruebas

4.1. Desarrollo

Luego de diseñar la aplicación, se debe proceder a codificar con distintos paquetes que luego van a ser utilizados de acuerdo a los requerimientos técnicos, ellos tienen la tarea de realizar pruebas del funcionamiento de la aplicación y permitir documentar el código.

4.1.1. Codificar.

Para la codificación de la aplicación fue necesario utilizar las siguientes tecnologías:

Desarrollo *Frontend*:

1. **react-native:** Es un *framework* que permite desarrollar e implementar apps nativas para dispositivos móviles utilizando Javascript. en esencia, es un conjunto de componentes React.

2. **react:** Es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

3. **npm:** Es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.

4. **JavaScript:** Es un lenguaje interpretado, orientado a objetos, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat.

5. **Kit de desarrollo de software de Android (SDK):** Es un conjunto de herramientas de desarrollo. Comprende un depurador de código y un simulador de teléfono basado en QEMU.

Desarrollo *Backend*:

1. ***Framework* Django:** Es un *framework* de aplicaciones web gratuito y de código abierto (open source) escrito en Python.

2. **Python:** Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.

3. **Django rest *framework*[1]:** Es una librería que permite construir un API REST sobre Django de forma sencilla. Ofreciendo una alta gama de métodos y funciones para el manejo, definición y control de nuestros recursos(endpoints).

4. **PostgreSQL:** Es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto.

5. **Heroku[2]:** Es una plataforma en la nube que permite a las empresas construir, entregar, supervisar aplicaciones y alojarlas en la nube.

4.1.2. Pruebas Unitarias.

En esta sección se realizan las pruebas unitarias por caso de uso para comprobar que el comportamiento de la aplicación es el correcto.

[1]https://www.django-rest-framework.org/
[2]https://www.heroku.com/

ENTRADA	RESULTADO ESPERADO	RESULTADO
Datos de entrada con correo correcto y contraseña con una longitud de 8 dígitos.	La aplicación registra al nuevo usuario.	SATISFACTORIO
Datos de entrada sin correo y contraseña con una longitud de 8 dígitos.	La aplicación genera un mensaje al usuario que ingrese un correo correcto	SATISFACTORIO
Datos de entrada con correo sin normalizar y contraseña correcta.	La aplicación normaliza el correo y registra el usuario.	SATISFACTORIO
Datos de entrada con correo ya registrado y contraseña correcta.	La aplicación notifica que el correo ya esta siendo utilizado por otro usuario.	SATISFACTORIO

Tabla 4.1: Prueba del CU-01.1 Registrar Usuario.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Correo y contraseã válidos.	La aplicación inicia sesión de acuerdo al rol del usuario	SATISFACTORIO
Correo inválido y contraseña correcta.	La aplicación genera un mensaje al usuario que no se pudo iniciar sesión.	SATISFACTORIO
Correo válido y contraseña inválida.	La aplicación genera un mensaje al usuario que no se pudo iniciar sesión.	SATISFACTORIO

Tabla 4.2: Prueba del CU-01.2 Iniciar Sesión.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Ingresar todos los parámetros de entrada del animal	La aplicación registra el animal	SATISFACTORIO
Ingresar no todos los parámetros de entrada del animal	La aplicación genera un mensaje al usuario que tiene datos inválidos.	SATISFACTORIO
Ingresar todos los parámetros de entrada del animal	La aplicación genera un error al usuario si no se pudo registrar el animal.	SATISFACTORIO

Tabla 4.4: Prueba del CU-02.2 Registrar Animal.

ENTRADA	RESULTADO ESPERADO	RESULTADO
El usuario ingresa un correo válido.	La aplicación envía un código de restablecimiento de contraseña al correo que debe ser utilizado para cambiar la contraseña.	SATISFACTORIO
El usuario ingresa el código de restablecimiento válido y la nueva contraseña.	La aplicación guarda la nueva contraseña y regresa a la pantalla de iniciar sesión.	SATISFACTORIO
El usuario ingresa un correo que no se encuentra asociado a ningún usuario.	La aplicación genera un mensaje al usuario que el correo no pertenece a ningún usuario.	SATISFACTORIO
El usuario ingresa un código de restablecimiento de contraseña inválido.	La aplicación genera un mensaje al usuario que el código es inválido.	SATISFACTORIO

Tabla 4.3: Prueba del caso de uso Restablecer Contraseña.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Seleccionar modificar un animal de la lista e ingresar los parámetros a cambiar del animal	La aplicación modifica el animal	SATISFACTORIO
Ingresar los parámetros a modificar de manera incorrecta	La aplicación genera un mensaje que el animal no pudo ser modificado.	SATISFACTORIO

Tabla 4.5: Prueba del CU-02.4 Modificar Animal.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Ingresar todo los parámetros válidos de la alimentación.	La aplicación debe mostrar un mensaje al usuario que la alimentación se registro con éxito.	SATISFACTORIO
Ingresar parámetros inválidos de alimentación	La aplicación genera un mensaje que se ingresaron datos inválidos.	SATISFACTORIO

Tabla 4.8: Prueba del CU-03.3 Registrar Alimentación.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Seleccionar eliminar un animal de la lista.	La aplicación genera un mensaje con las opciones de aceptar o cancelar, si elije aceptar el animal se elimina de la lista.	SATISFACTORIO
Seleccionar eliminar un animal de la lista.	La aplicación genera un mensaje con la opción aceptar o cancelar, si elije cancelar el animal no se elimina de la lista.	SATISFACTORIO

Tabla 4.6: Prueba del CU-02.5 Eliminar Animal.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Ingresar todo los parámetros válidos de la vacuna	La aplicación debe mostrar un mensaje al usuario que la vacuna se registro con éxito.	SATISFACTORIO
Ingresar parámetros inválidos de la vacuna	La aplicación genera un mensaje que se ingresaron datos inválidos.	SATISFACTORIO

Tabla 4.7: Prueba CU-03.2 Registrar Vacuna.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Seleccionar eliminar ya sea una vacuna, alimentación o tratamiento de la lista.	La aplicación envía un mensaje al usuario si esta seguro de elimina los datos con las opciones de aceptar o cancelar.	SATISFACTORIO
El usuario selecciona la opción aceptar al momento de eliminar una vacuna, alimentación o tratamiento.	La aplicación elimina el registro seleccionado.	SATISFACTORIO

Tabla 4.11: Prueba del CU-03.6 Eliminar.

ENTRADA	RESULTADO ESPERADO	RESULTADO
El usuario presiona en perfil del menú principal de la aplicación.	La aplicación muestra el perfil del usuario.	SATISFACTORIO

Tabla 4.12: Prueba del CU-04.1 Ver Perfil.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Ingresar todo los parámetros válidos del tratamiento.	La aplicación debe mostrar un mensaje al usuario que el tratamiento se registro con éxito.	SATISFACTORIO
Ingresar parámetros inválidos del tratamiento	La aplicación genera un mensaje que se ingresaron datos inválidos.	SATISFACTORIO

Tabla 4.9: Prueba del CU-03.4 Registrar Tratamiento.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Seleccionar modificar ya sea una vacuna, alimentación o tratamiento de la lista e ingresar los parámetros a modificar.	La aplicación modifica ya sea la vacuna, alimentación o tratamiento, deacuerdo a la opción que el usuario elija.	SATISFACTORIO
Ingresar los parámetros a modificar de manera incorrecta en cualquiera de la opciones que el usuario seleccione entre vacuna, alimentación o tratamiento.	La aplicación genera un mensaje al usuario que no pudo modificar los datos.	SATISFACTORIO

Tabla 4.10: Prueba del CU-03.5 Modificar.

ENTRADA	RESULTADO ESPERADO	RESULTADO
El usuario en perfil selecciona editar perfil, e ingresa parámetros válidos.	La aplicación modifica el perfil del usuario.	SATISFACTORIO
ENTRADA	**RESULTADO ESPERADO**	**RESULTADO**
El usuario en perfil selecciona editar perfil, e ingresa parámetros inválidos.	La aplicación muestra un mensaje que se ingresaron parámetros inválidos.	SATISFACTORIO

Tabla 4.13: Prueba del CU-04.2 Configurar Perfil.

ENTRADA	RESULTADO ESPERADO	RESULTADO
Click en salir de la aplicación.	La aplicación debe cerrar sesión de usuario.	SATISFACTORIO

Tabla 4.14: Prueba del CU-04.3 Salir de la aplicación.

4.1.3. Documentar Código.

Para la documentación del frontend y backend se utilizaron las siguientes herramientas.

1. **git**[3]**:** Es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

2. **GitHub:** Se utilizó GitHub como un servicio de hosting de repositorios almacenados en la nube. Esencialmente, permitió hacer más fácil el control y colaboración de la aplicación.

3. **Pylint:** Se integra con vsCode es un verificador de código fuente, error y calidad para el lenguaje de programación Python.

4. **JSDoc:** Es una sintaxis para agregar documentación de la API al código fuente de JavaScript

4.2. Pruebas de funcionamiento.

Las pruebas de funcionamiento del sistema completo se llevarán a cabo utilizando los métodos y herramientas que se describen a continuación.

4.2.1. Emulación y simulación

Emulador Android V3.5.3[4]**:** Permitió simular dispositivos Android en una computadora para probar la aplicación en diferentes dispositivos y niveles de API de Android sin necesidad de contar con los dispositivos físicos.

Genymotion V3.0.3[5]**:** Este emulador de Android permite realizar pruebas de ejecución sobre la aplicación desarrollada, ya que ofrece la posibilidad de ejecutar

[3]https://git-scm.com/
[4]https://developer.android.com/
[5]https://www.genymotion.com/

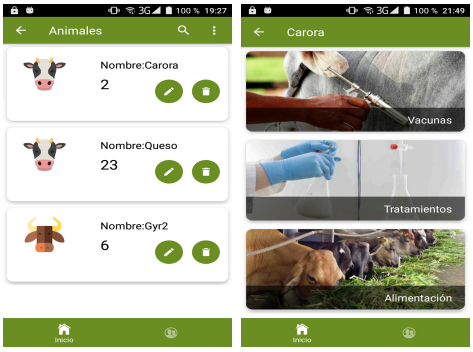
escenarios iguales a un dispositivo real mediante sensores de hardware (GPS, red, multitáctil.).

4.2.2. Dispositivos reales

En esta sección se realizaron pruebas de la aplicación instalándola en dispositivos físicos. Estas pruebas se llevaron a cabo para evaluar el comportamiento de las diferentes funcionalidades de la aplicación, en la figura 4.1 se puede observar alguna de las pruebas realizadas al momento de registrar y editar fincas, mientras que en la figura 4.2 se observa el manejo sanitario de animales.

(a) (b)

Figura 4.1: Pruebas de registro (a) y edición de finca (b).



(a) (b)

Figura 4.2: Pruebas para listar animales (a) y manejo de sanidad animal (b).

Se utilizaron los teléfonos Huawei-Y221 y Samsung Galaxy J7 Pro.

4.2.3. Pruebas del API

Durante el desarrollo del API REST se llevaron a cabo pruebas de su comportamiento utilizando la herramienta Postman para enviar peticiones al API y evaluar si la respuesta que retorna es la correcta para el cliente. En la figura 4.3 y 4.4 se observan una de las pruebas manuales realizadas donde se aprecia la url y los parámetros que serían enviados al API en el cuerpo de la petición.

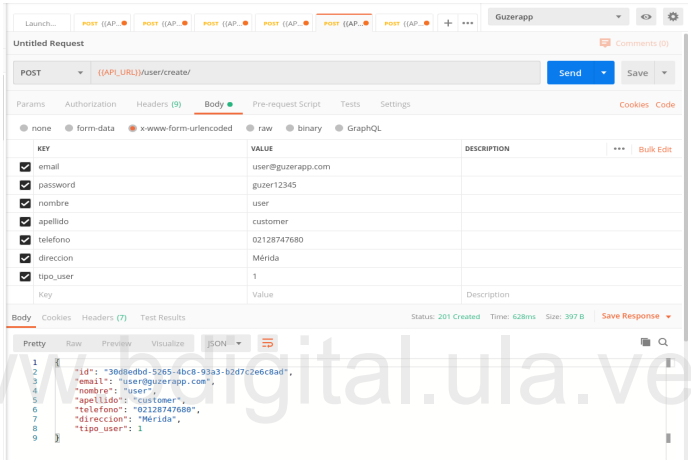


Figura 4.3: Pueba manual para registrar un usuario.

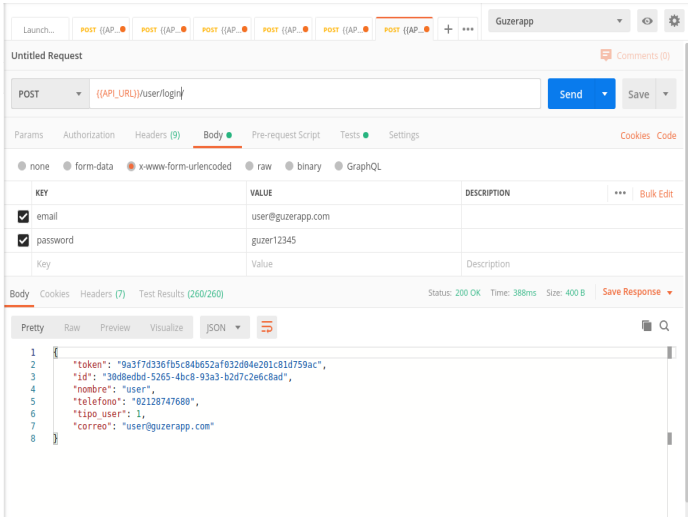


Figura 4.4: Pueba manual para autenticación de usuario.

En las figuras figura 4.5 se muestran una de las pruebas automatizadas realizadas en Postman. Donde se ejecutan automáticamente cada una de las peticiones configuradas manualmente así como los scripts de prueba, al final la herramienta de postman muestra los resultados, indicando las pruebas que pasaron, los que fallaron, el código de respuesta y el tiempo que el API tarda para responder.

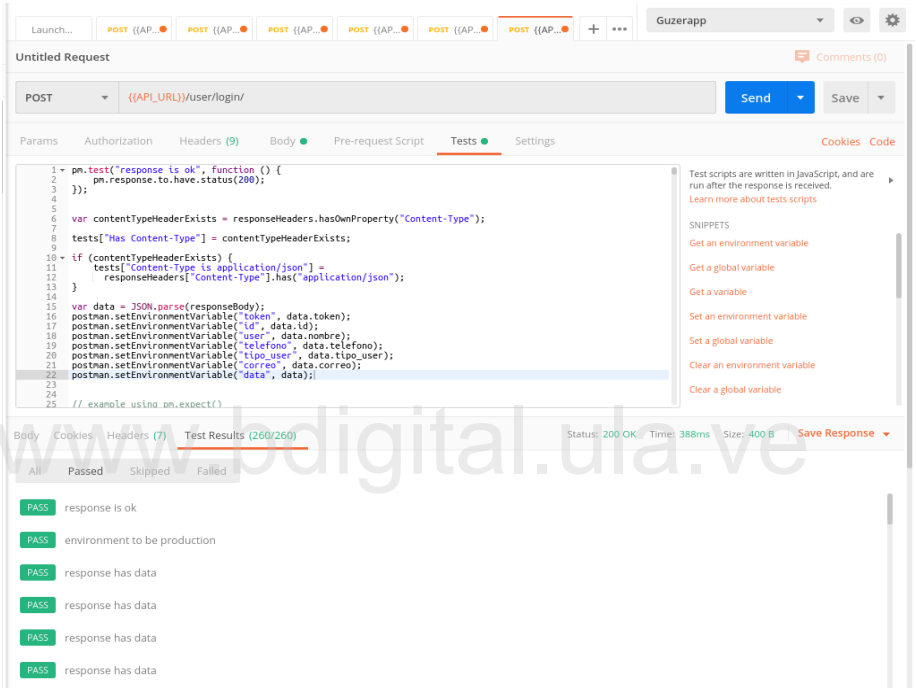


Figura 4.5: *Script* de pruebas automatizadas.

4.2.4. Pruebas de la aplicación

Las pruebas de funcionamiento para la aplicación móvil se llevarón a cabo utilizando las herramientas de React Developer Tools para comprobar el comportamiento de la herencia de componentes, Redux DevTools que permitió tener un seguimiento al estado de Redux, estas herramientas son incluidas en React Native Debugger. Para realizar estas pruebas fue necesario tener una conexión cableada con el dispositivo móvil y estar en ejecución el servidor de React Native. En la figura 4.6 se aprecia el cambio de estado en los componentes

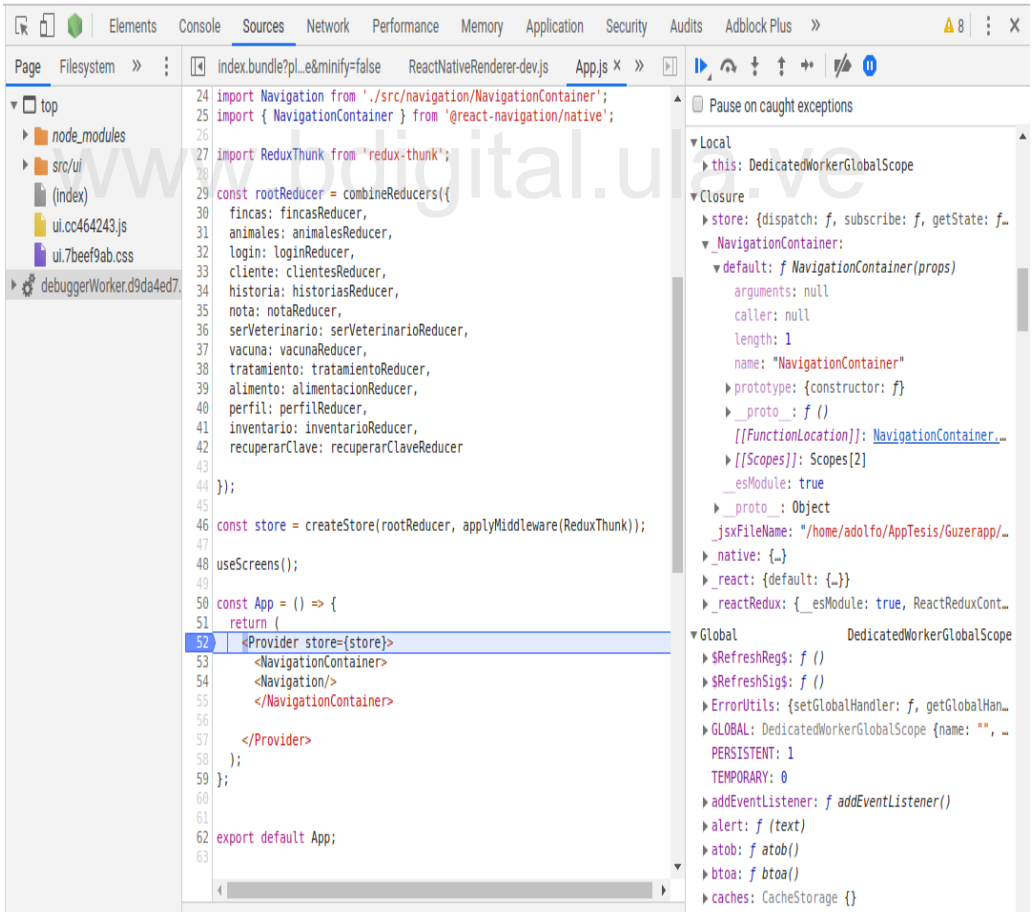


Figura 4.6: Depuración de la aplicación utilizando React Developer Tools.

4.2.5. Análisis de las 6Ms

A continuación se evalúa el potencial de éxito de la aplicación haciendo un análisis de las 6Ms.

Atributo	**Justificación**
Momento	La aplicación propuesta estará disponible en cualquier instante de tiempo siempre y cuando el dispositivo móvil tenga acceso a internet, esto se debe a la arquitectura que posee el sistema cliente-servidor.
Movilidad	Los productores y veterinarios tendran movilidad a cualquier lugar para realizar operaciones de registro, consulta, edición de animales, siempre y cuando el móvil tenga servicio de internet.
Dinero	La aplicación móvil tiene un fin lucrativo, a pesar de no encontrarse en tiendas de app, cumple con los requisitos mínimos que los usuarios del sector pecuario necesitan. la monetizacion se podría realizar incluyendo publicidad a personas jurídicas del sector pecuario.
Yo	El servicio es personalizable ya que los usuarios podrán elegir uno de los roles entre productor y veterinario al momento de registrarse, esto genera que la aplicación tome operaciones de interacción distintas.
Máquina	La aplicación podrá ser ejecutada por teléfonos que tengan el sistema operativo android V4.2 y versiones superiores.
Multiusuario	La aplicación podrá ser usada por múltiples usuarios de manera simultanea, donde cada usuario pueda tener un rol entre productor o veterinario.

Tabla 4.15: Evaluación de las 6 M's para la aplicación propuesta.

4.3. Entrega.

En la fase de entrega se desarrolló el manual de usuario de la aplicación móvil Guzerapp. En las figura 4.7 y 4.8 se aprecian algunas de las instrucciones que deben de realizar los usuarios para manejar la aplicación.

4.3.1. Manuales

Figura 1: Pantalla inicial.

Manual de Usuarios:

1) En la figura 1 se aprecia la pantalla inicial de la aplicación donde los usuarios podrán iniciar sesión.
   A) Campo de correo electrónico.
   B) Campo para la contraseña.
   C) Botón para iniciar sesión de usuario.
   D) Botón para recuperar contraseña.
   E) Botón para registrar nuevo usuario.

2) En la figura 2 se describe el menú de opciones para el productor manejar finca.
   A) Botón que accede a la lista de animales.
   B) Botón de acceso al inventario de la finca.
   C) Botón que regresa a lista de fincas.
   D) Botón que muestra lista de veterinarios.

3) En la figura 3 se listan los animales de la finca donde se aprecia un avatar del animal y el numero, y en la parte superior una barra de búsqueda.

Figura 2: Pantalla de administración de finca.

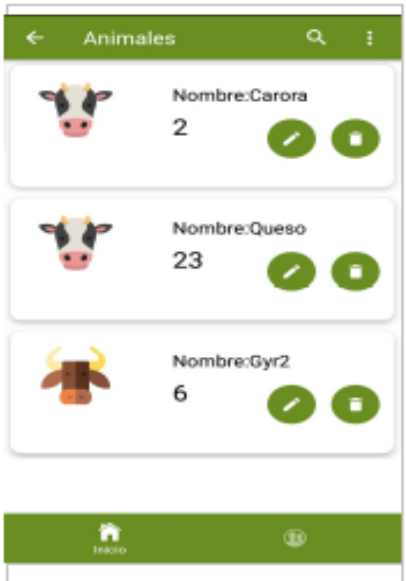


Figura 3: Pantalla que lista animales

Figura 4.7: Manual de usuarios(1).

Perfil de Veterinario:

1) En la figura 4 se aprecia el menú principal al momento de iniciar sesión el rol veterinario donde encontrara los botón siguientes.

A) Botón para regresar a la lista de clientes en caso de encontrarse en historias medicas o notas.
B) Botón para realizar configuraciones del perfil.
C) Salir de la aplicación.

2) En la figura 5 se describe la lista de clientes que asesora el veterinario.
A) Menú principal del perfil de usuario
B) Botón que permite agregar un nuevo cliente.
C) Botón para editar el cliente.
D) Botón para eliminar el cliente.

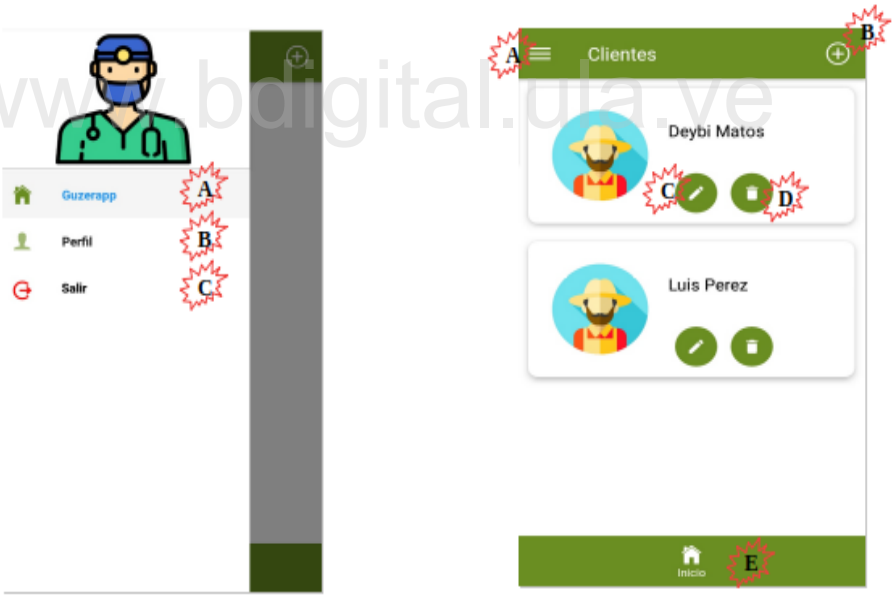


Figura 4.8: Manual de usuarios(2).

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones.

Para llevar a cabo el proyecto fue necesario realizar entrevistas a productores pecuarios del municipio Antonio Pinto Salinas del estado Mérida, permitiendo capturar información que sirviera de base a un estudio de requisitos. También se realizaron investigaciones documentales sobre el uso de tecnologías de información y comunicación en el sector ganadero, para establecer criterios de diseño e implementación. Adicionalmente, se realizó la implementación de una base de datos que permitiera almacenar los datos relevantes que manejan los productores en sus unidades de producción, para luego implementar la aplicación móvil con enfoque pecuario. Después de terminar la aplicación se realizó la integración de la base de datos mediante un API-REST permitiendo realizar pruebas y correcciones necesarias para su buen funcionamiento.

Una vez concluido este trabajo y cumpliendo con todos los objetivos pautados para dicha investigación, es importante recalcar la importancia de las aplicaciones móviles para el avance en la producción alimentaria de cualquier país. Por lo tanto la aplicación móvil Guzerapp no pretende quedar limitada al funcionamiento que está desarrollada actualmente, ya que su diseño e implementación permite agregar nuevas funcionalidades a medida que el sector pecuario así lo demanden.

5.2. Trabajos futuros.

1. Desarrollar un módulo que permita medir el área de forraje mediante el uso del GPS.

2. Desarrollar un módulo que genere notificaciones a los usuarios sobre los eventos de reproducción.

3. Desarrollar un módulo que permita realizar reportes del clima donde se encuentre la unidad de producción.

4. Desarrollar un chat a la aplicación que permita tener comunicación directa los veterinarios y productores.

Capítulo 6

Anexos.

6.1. Instrumento de recolección de Datos.

Cuestionario dirigido a los productores de ganado bovino. Propósito

El presente cuestionario tiene como finalidad de recibir información sobre la situación y percepción que tienen los productores de ganado bovino del municipio Antonio Pinto Salinas del estado Mérida. En cuanto los conocimientos que poseen en el manejo de tecnologías de información y comunicación para la gestión y manejo de rebaños de sus fincas. El instrumento aplicado tiene carácter anónimo, de ahí que su opinión pueda ser la más sincera y libre. Debido a la importancia y seriedad del estudio, se le agradece ser lo más claro y objetivo al responder cada uno de los planteamientos del cuestionario.

1. ¿Cuenta usted actualmente con un teléfono inteligente con sistema operativo Android? SI **8** NO **2**

   Análisis: En la figura 6.1 se puede apreciar que de los 10 encuestados un 80 % de ellos cuentan con un teléfono inteligente con sistema operativo Android. Lo cual demuestra que la mayoría de ellos pueden usarlos para ejecutar aplicaciones referentes al sector pecuario.

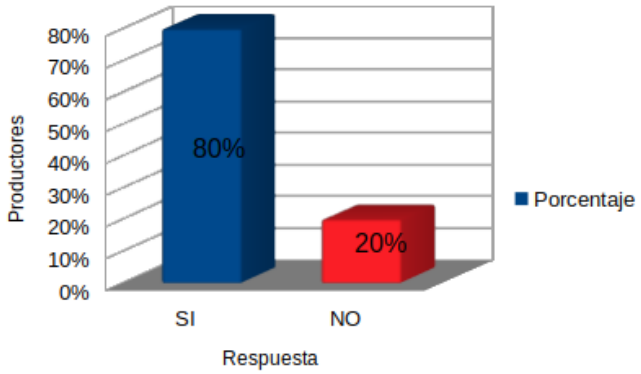


Figura 6.1: Diagrama de resultados para la pregunta 1.

2. ¿Usted tiene en su teléfono acceso a Internet? SI **7** NO **3**

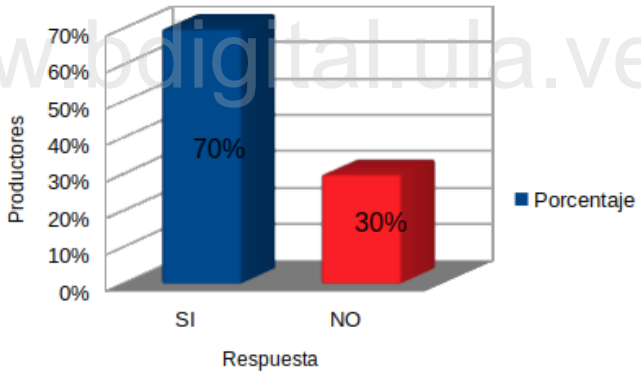


Figura 6.2: Diagrama de resultados para la pregunta 2.

Análisis: En la figura 6.2 se aprecia que un 70 % de los encuestados manifiestan tener acceso a Internet, el cual es un porcentaje importante que puede ejecutar aplicaciones en linea desde sus diferentes unidades de producción.

3. En las labores diarias referente al manejo de rebaños de la finca, ¿Toma apuntes o notas de lo que haces? NUNCA **2** POCAS VECES **3** SIEMPRE **5**

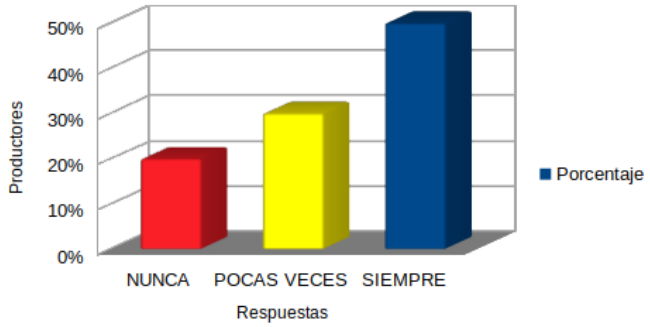


Figura 6.3: Diagrama de resultados para la pregunta 3.

Análisis: La figura 6.3 representa que un 20 % de los encuestados nunca toman registro de los rebaños en las fincas, un 30 % pocas veces lo hacen y la mayoría representada en un 50 % siempre toman nota del manejo animal de forma manual, que en la mayoría de los casos son difíciles de consultar a tiempo.

4. ¿El manejo del ganado en su finca es asesorado por un Médico Veterinario? NUNCA **5** POCAS VECES **2** SIEMPRE **3**

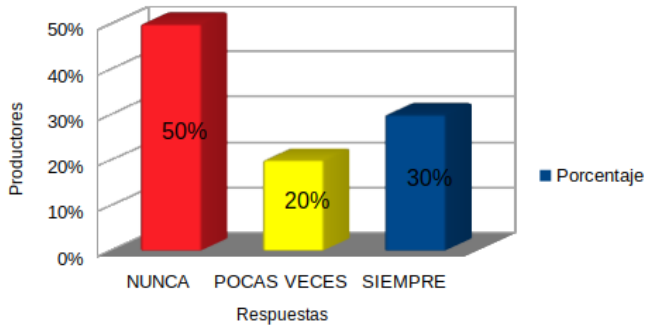


Figura 6.4: Diagrama de resultados para la pregunta 4.

Análisis: La figura 6.4 representa que el 50 % de los productores nunca consultan a un medico veterinario, esto se debe a la distancia a las que se encuentran de las unidades de producción y falta de información para ubicarlos a tiempo. un 20 % pocas veces lo hacen y el 30 % siempre tiene un medico asesor.

5. ¿Piensa usted. Qué llevar un registro del ganado en la finca mediante una

aplicación móvil mejorará el manejo del rebaño y la toma de decisiones? SI **10** NO_

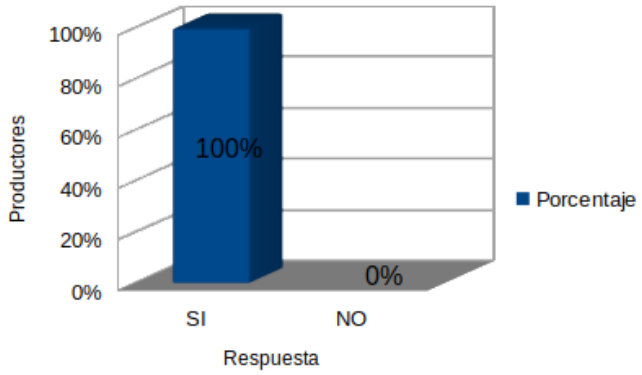


Figura 6.5: Diagrama de resultados para la pregunta 5.

Análisis: La figura 6.5 representa que el 100 % de los productores están de acuerdo que llevar de manera automatizada los registros del ganado desde una aplicación móvil mejorará la toma de decisiones en sus unidades de producción.

6. ¿Ha impactado el NO manejar de manera óptima los registros del ganado en pérdidas económicas? SI **10** NO_

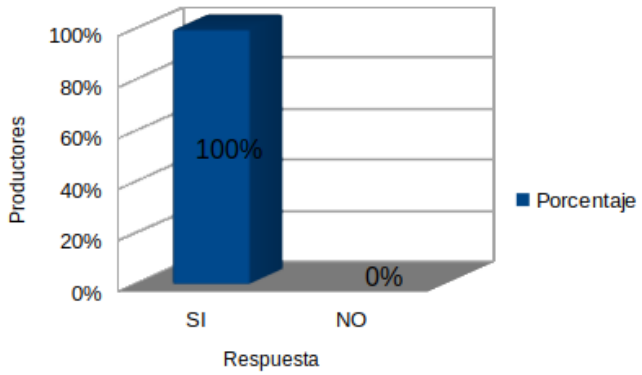


Figura 6.6: Diagrama de resultados para la pregunta 6.

Análisis: La figura 6.6 representa que el 100 % de los productores están de acuerdo que el no llevar registros de los animales en las unidades de produc-

ción les ha ocasionado perdidas económicas, lo cual evidencia que es factible desarrollar una aplicación móvil que cumpla con esta función y permita reducir perdidas a los productores de ganado.

7. ¿Está usted. A favor de tecnificar las fincas productoras de ganado utilizando una aplicación móvil que aloje los registros de trazabilidad animal? SI **10** NO_

Figura 6.7: Diagrama de resultados para la pregunta 7.

Análisis: La figura 6.7 se puede observar que el 100 % de los productores tienen el deseo de mejorar el manejo de los animales utilizando una aplicación móvil que aloje la trazabilidad, permitiendo ahorrarles tiempo cuando deseen consultar información de su interés.

8. ¿Registra en fichas manuscritas los procesos de vacunación, alimentación y tratamientos aplicados al animal? SI **7** NO **3**

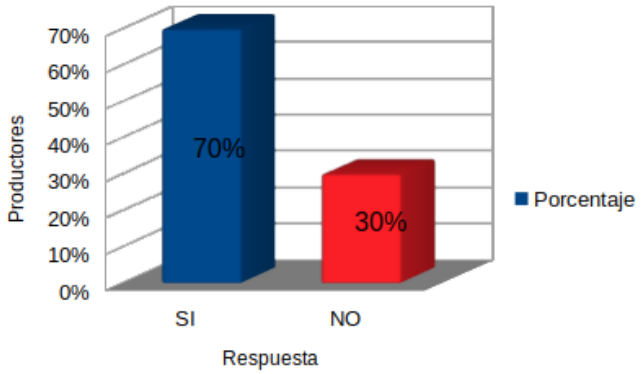


Figura 6.8: Diagrama de resultados para la pregunta 8.

Análisis: La figura 6.8 se aprecia que el 100 % de los productores poseen un control de vacunas, alimentación y tratamientos en sus unidades de producción animal, pero solo un 30 % no maneja registro de estas actividades y los productores que lo hacen utilizan fichas manuscritas, que en la mayoría de los casos se extravían y son difíciles de recuperar.

Cuestionario dirigido a Médicos Veterinarios.

Propósito

El presente cuestionario tiene como finalidad recibir información sobre la situación y percepción que tienen los Médicos Veterinarios del municipio Antonio Pinto Salinas del estado Mérida, en realación al manejo animal que realizan. El instrumento aplicado tiene carácter anónimo, de ahí que su opinión pueda ser la más sincera y libre. Debido a la importancia y seriedad del estudio, se le agradece ser lo más claro y objetivo al responder cada uno de los planteamientos.

1. ¿Cuenta usted actualmente con un teléfono inteligente con sistema operativo Android? SI **3** NO_

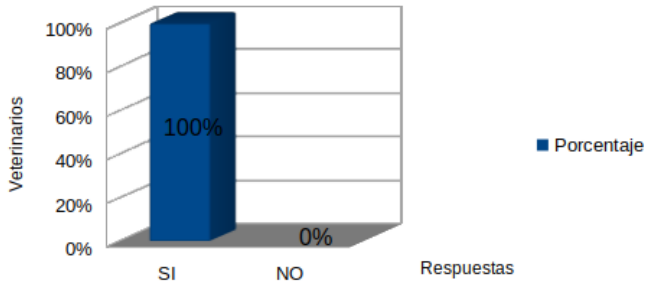


Figura 6.9: Diagrama de resultados para la pregunta 1.

Análisis: Para la inferencia numérica obtenida del análisis del instrumento aplicado a los veterinarios se aprecia en la 6.9 que el 100 % tiene un teléfono inteligente con sistema operativo Android, lo que nos revela que poseen el dispositivo para ejecutar aplicaciones móviles.

2. ¿Usted tiene en su teléfono acceso a Internet al momento de realizar visitas al campo? SI **2** NO_ POCAS VECES **1**

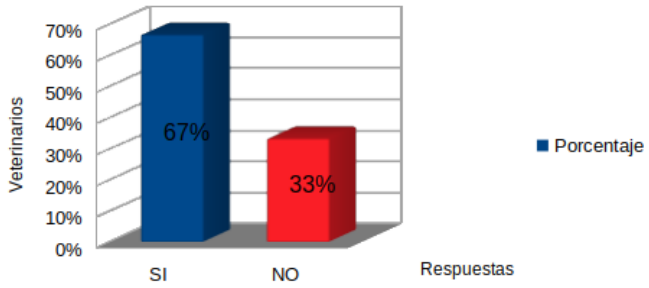


Figura 6.10: Diagrama de resultados para la pregunta 2.

Análisis: En la figura 6.10 se pudo detectar que el 67 % de los encuestados siempre tienen acceso a Internet en sus teléfonos al momento de realizar visitas al campo, la cual nos permite inferir que pueden ejecutar aplicaciones móviles en linea. Y un 33 % pocas veces tienen acceso al servicio de Internet.

3. ¿Asesora a productores de ganado? SI **3** NO_

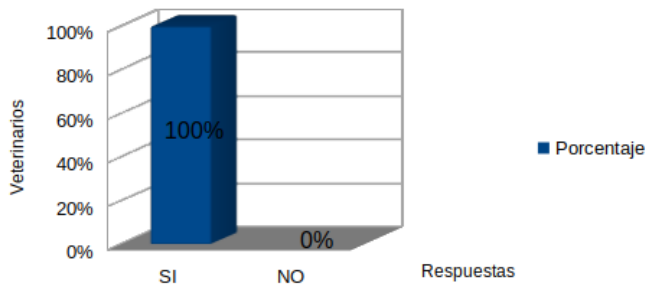


Figura 6.11: Diagrama de resultados para la pregunta 3.

Análisis: De acuerdo a los resultados que se observan en la figura 6.11 el 100 % de los veterinarios asesora a productores de ganado.

4. En las labores que realizas referente al manejo del ganado en fincas, ¿Toma apuntes o notas de lo que haces? NUNCA_ POCAS VECES **3** SIEMPRE_

Figura 6.12: Diagrama de resultados para la pregunta 4.

Análisis: De los datos obtenidos en la figura 6.12 el 100 % de los encuestados manifestaron que pocas veces toman nota de lo que hacen en las unidades de producción. Ya que consideran tedioso llevar archivos manuscritos al campo laboral.

5. ¿Manejas registro de los animales al momento de aplicarles un plan sanitario? NUNCA_ POCAS VECES **3** SIEMPRE_

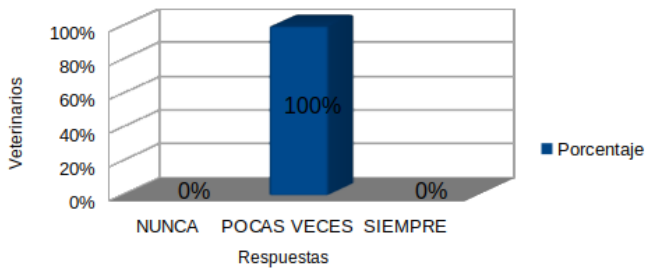


Figura 6.13: Diagrama de resultados para la pregunta 5.

Análisis: En la figura 6.13 el 100 % de los veterinarios pocas veces toman registro de los animales que tratan con un plan sanitario en las unidades de producción. Esto evidencia que al momento de retomar por segunda vez un plan sanitario sobre los animales no cuentan con registros históricos del animal a tratar haciendo difícil tomar decisiones a tiempo.

6. ¿Piensa usted. Qué llevar un registro de rebaños de las fincas que asesoras, mejorará el manejo de animales y la toma de decisiones en los médicos veterinarios? SI **3** NO_

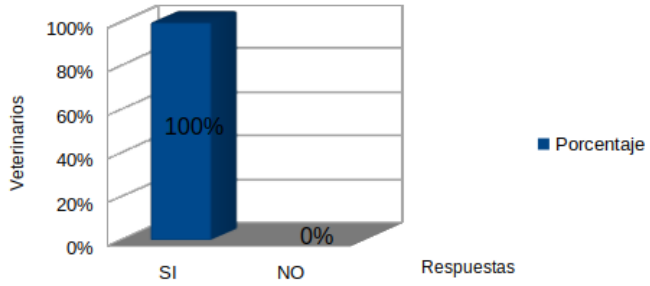


Figura 6.14: Diagrama de resultados para la pregunta 6.

Análisis: Los datos que refleja la figura 6.14 demuestra que el 100 % de los veterinarios encuestados están de acuerdo que llevar el registro de los rebaños garantiza tomar decisiones a tiempo, mejorando el manejo del animal.

7. ¿Está usted. A favor de tecnificar el manejo de ganado utilizando una aplicación móvil que aloje los registros de trazabilidad animal? SI **3** NO_

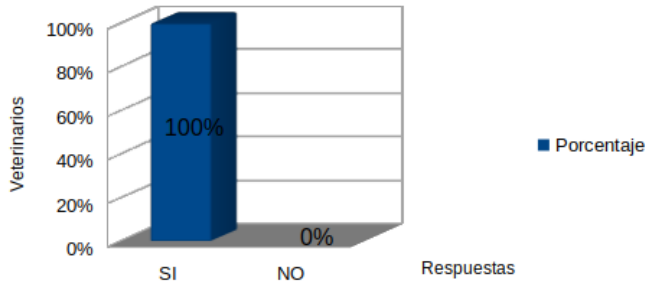


Figura 6.15: Diagrama de resultados para la pregunta 7.

Análisis: De acuerdo a los resultados de la figura 6.15 el 100 % de los veterinarios están de acuerdo que al utilizar una aplicación móvil que aloje registros de trazabilidad animal, permitirá tecnificar el manejo del ganado.

8. ¿Cree usted que el uso de una aplicación móvil para la gestión de información y manejo del ganado bovino en las fincas ganaderas del estado Mérida serviría de apoyo a los médicos veterinarios? SI **3** NO_

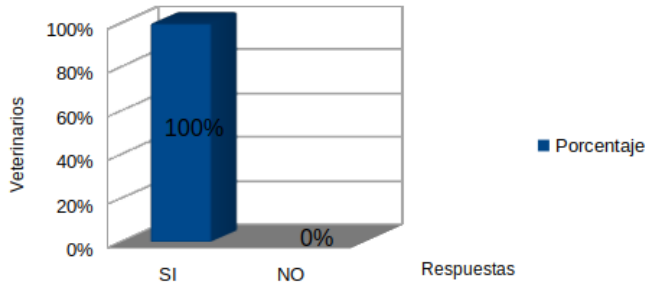


Figura 6.16: Diagrama de resultados para la pregunta 8.

Análisis: En la figura 6.16 se aprecia que el 100 % de los veterinarios consideran que al usar una aplicación móvil para el manejo animal en el estado Mérida servirá de apoyo a sus labores diarias.

9. ¿Apoyaría la iniciativa de utilizar una aplicación móvil para mejorar el manejo y producción ganadera en el estado Mérida? SI **3** NO_

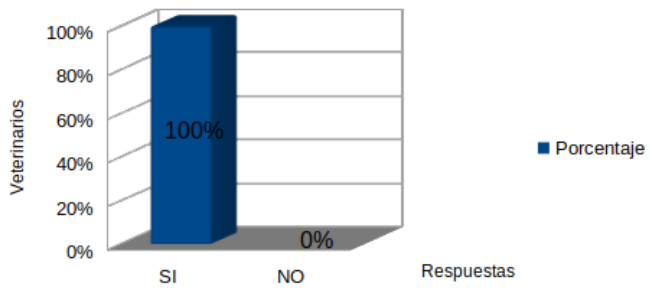


Figura 6.17: Diagrama de resultados para la pregunta 9.

Análisis: En la figura 6.17 refleja que el 100 % de los encuestados apoyaría la iniciativa de utilizar una aplicación que mejore el manejo y producción ganadera en el estado Mérida.

Bibliografía

[1] Redux. [Web; accedido el 13-08-2019]. URL https://carlosazaustre.es/como-funciona-redux-conceptos-basicos.

[2] React. [Web; accedido el 22-06-2019]. URL https://es.reactjs.org/docs/higher.html.

[3] React. [Web; accedido el 22-06-2019]. URL https://es.reactjs.org/docs/render-props.html.

[4] Illya Alvarado. Características de los tipos de aplicaciones móviles. 2019. URL https://ceroideas.es/principales-caracteristicas-de-los-tipos-de-aplicaciones-moviles/.

[5] Luis Alberto Depablos Alviárez. Sistemas de producción con bovinos de carne. 2009. URL http://www.ucv.ve/fileadmin/user_upload/facultad_agronomia/Clase_III.pdf.

[6] Fidias G. Arias. *El Proyecto de Investigación*. Editorial Episteme, 2016.

[7] Enciclopedia Británica. Arquitectura cliente-servidor. [Web; accedido el 12-01-2020]. URL https://www.britannica.com/technology/client-server-architecture.

[8] V. Cuervo. Qué es postman. [Web; accedido el 20-10-2019]. URL http://www.arquitectoit.com/postman/que-es-postman/.

[9] Universidad de Alicante. Modelo vista controlador (mvc). [Web; accedido el 20-11-2019]. URL https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html.

[10] Ignacio Rohr María Eugenia Etchemendy Federico Henze, Guillermo Vilas. Tesis de grado. agrosoft, sistema para la gestión de establecimientos ganaderos, 2013. Universidad ORT Uruguay,Facultad de Ingeniería.

[11] A. Lopez. Qué es postman. [Web; accedido el 18-10-2019]. URL https://openwebinars.net/blog/que-es-postman/.

[12] Enayarix Castillo y Noris Roa A. Luis Guevara. Inia. ceniap. fisiología de la reproducción animal. venezuela. 2009. URL https://www.engormix.com/ganaderia-carne/articulos/uso-registros-manejo-informacion-t27802.htm.

[13] Rolando Lasso Luiyiana Pérez. App móvil: Sistema de información empresarial de fincas ganaderas para pymes en la provincia de los santos, panamá. 2018.

[14] Platzi. ¿qué es heroku y para qué me sirve?. [Web; accedido el 18-10-2019]. URL https://platzi.com/blog/que-es-heroku-y-para-que-me-sirve/.

[15] David Bastidas Vargas. Tesis de grado. aplicación móvil para el seguimiento y control de las siembras de arrocera la esmeralda s.a. alesa móvil, gaproa, 2014. Universidad del Valle, Santiago de Cali, Colombia.

[16] R. Martinez M Vera, P. Rodríguez. Diseño y desarrollo de interfaces con interacción física utilizando dispositivos móviles. 2017.

[17] Wikipedia. Guzerá. [Web; accedido el 15-10-2019]. URL https://en.wikipedia.org/wiki/Guzer%C3%A1.

[18] Wikipedia. Verde. [Web; accedido el 28-11-2019]. URL https://es.wikipedia.org/wiki/Verde.

[19] Ramez Elmasri y Shamkant B. Navathe. *Fundamentos de Sistemas de Bases de Datos*. PEARSON EDUCACION S.A., Madrid, 2007.