

Diseño y Desarrollo de un Software de Gestión y Seguimiento de Trazabilidad de Ganado Vacuno

Autor: Mario A. Rosales (mariorosales941@gmail.com)

Director: Ing. Lucila Romero (lucila.rb@gmail.com)

Asesor Temático:

- Ing. Agrónomo Jorge Cinquini
 - Med. Veterinario Juan Carlos Garcia
-

Informe Final

Índice general

1	Introducción	9
2	Análisis de Requerimientos	10
1	Introducción	10
2	Ingeniería de Requerimientos	10
2.1	Introducción	10
2.2	Actividades de la Ingeniería de Requerimientos	12
2.3	Interesados o Stakeholders	12
2.4	Ingeniería de Requerimientos: Técnicas y Herramientas	13
3	Requerimientos Identificados	14
3.1	Requerimientos Funcionales.	14
3.2	Requerimientos No Funcionales	16
4	Representación de los requerimientos recolectados	17
4.1	Casos de Uso	18
5	Análisis y Definición de Framework y Tecnologías Complementarias	19
5.1	Lenguajes de Programación Back-End	20
5.2	Lenguajes de Programación Front-End	21
5.3	Frameworks Back-End	21
5.4	Frameworks Front-End	24
5.5	Bases de Datos	26
5.6	Framewokrs para Testing	31
5.7	Manejadores de Dependencias	31
5.8	Bibliotecas	32
5.9	Sistemas de Control de Versiones	32
3	Diseño del Sistema	33
1	Etapa 2 - Diseño del Sistema	33
1.1	Introducción	33
1.2	Arquitectura del Sistema	33
1.3	Descomposición del Sistema en Módulos	36
1.4	Diseño físico de la Base de Datos	38
1.5	Diseño de la Interfaz Gráfica de Usuario	39
1.5.1	Aspectos fundamentales a la hora de Diseñar la Interfaz	40
4	Desarrollo	43
1	Introducción	43
2	Desarrollo del Sistema	43
2.1	Página de Bienvenida e Inicio de Sesión	44
2.2	Página Principal	46
2.3	Funcionalidades Administrativas y de Configuración	48
2.3.1	Administración de Usuarios	49
2.3.2	Administración de Campos y Divisiones	51

2.3.3	Administración de Caravanas	56
2.3.4	Administración de Razas, Categorías y Sub Categorías	57
2.3.5	Administración de Tipos de Actividades Veterinarias	60
2.3.6	Administración de Clientes y Proveedores	61
2.4	Funcionalidades Operativas	61
2.4.1	Modulo Compras	62
2.4.2	Módulo Cambios de Categorías	64
2.4.3	Módulo Traslados	65
2.4.4	Módulo Mortandad	66
2.4.5	Módulo Actividades Veterinarias	67
2.4.6	Módulo Ventas	67
2.5	Reportes	68
2.5.1	Reporte de Usuarios	69
2.5.2	Reporte de Vacunos	71
2.5.3	Reporte de Operaciones	71
3	Resultados	72
3.1	Objetivos y Alcance	72
3.2	Requerimientos	73
3.3	Interfaz Gráfica de Usuario	74
5	Pruebas	76
6	Conclusiones y Trabajos Futuros	77
7	Apéndice	78
1	Diagramas de Casos de Uso	78
2	Fichas Textuales de Casos de Uso	82

Índice de figuras

2.1 Tabla Comparativa de Motores de Base de Datos.	30
3.1 Diseño del Sistema: Diagrama de Clases.	37
3.2 Diseño del Sistema: Diseño Físico de la Base de Datos.	38
3.3 Diseño del Sistema: Diseño Adaptable.	40
3.4 Diseño del Sistema: Inicio de Sesión.	42
3.5 Diseño del Sistema: Formulario de Compra.	42
4.1 Página de Bienvenida.	44
4.2 Página de Inicio de Sesión.	45
4.3 Página Principal.	47
4.4 Últimos movimientos: Compras.	47
4.5 Página principal de usuarios.	50
4.6 Formulario de registro de usuarios.	50
4.7 Página principal de campos.	52
4.8 Formulario de registro de campos.	52
4.9 Página principal de divisiones.	53
4.10 Formulario de registro de divisiones.	54
4.11 Página principal de relaciones entre campos y divisiones.	55
4.12 Formulario de registro de relaciones entre campos y divisiones.	55
4.13 Página principal de Caravanas.	56
4.14 Formulario de registro de nuevas caravanas.	57
4.15 Página principal de categorías.	58
4.16 Formulario de registro de nuevas categorías.	59
4.17 Página principal de relaciones entre categorías y sub categorías.	59
4.18 Formulario de registro de relaciones entre categorías y sub categorías.	60
4.19 Datos de la compra.	62
4.20 Agregar ítem a la compra.	63
4.21 Ítems a la compra.	63
4.22 Caravanas.	64
4.23 Observaciones.	64
4.24 Reportes.	69
4.25 Reporte de usuarios por perfil.	70
4.26 Reporte de usuarios por campo.	70
4.27 Reporte de usuarios por campo.	71
4.28 Reporte de usuarios por campo.	72
7.1 Diagrama de Casos de Uso del Actor Administrador.	78
7.2 Diagrama de Casos de Uso del Actor Encargado - Parte 1.	79

7.3	Diagrama de Casos de Uso del Actor Encargado - Parte 2	80
7.4	Diagrama de Casos de Uso del Actor Veterinario	81
7.5	CU01: Ingresar al Sistema.	82
7.6	CU02: Recuperar Contraseña.	82
7.7	CU03: Registrar Usuario.	83
7.8	CU04: Consultar Campo.	84
7.9	CU05: Consultar Perfil.	84
7.10	CU06: Modificar Usuario.	85
7.11	CU07: Eliminar Usuario.	85
7.12	CU08: Consultar Usuario.	86
7.13	CU09: Registrar Campo.	86
7.14	CU10: Dividir Campo.	87
7.15	CU11: Modificar Campo.	87
7.16	CU12: Eliminar Campo.	88
7.17	CU13: Registrar Raza.	88
7.18	CU14: Modificar Raza.	89
7.19	CU15: Eliminar Raza.	89
7.20	CU16: Consultar Raza.	90
7.21	CU17: Registrar Operación.	90
7.22	CU18: Registrar Compra.	91
7.23	CU19: Registrar Cambio de Categoría.	92
7.24	CU20: Registrar Traslado.	93
7.25	CU21: Registrar Venta.	94
7.26	CU22: Registrar Mortandad.	94
7.27	CU23: Registrar Nacimiento.	95
7.28	CU24: Generar Reporte.	95
7.29	CU25: Reporte de Actividad.	96
7.30	CU26: Reporte de Operación.	97
7.31	CU27: Reporte de Stock.	98
7.32	CU28: Consultar Actividad.	98
7.33	CU29: Consultar Operación.	99
7.34	CU30: Consultar Stock.	99
7.35	CU31: Registrar Actividad.	100
7.36	CU32: Modificar Actividad.	100
7.37	CU33: Eliminar Actividad.	101
7.38	CU34: Registrar Evento.	101
7.39	CU35: Notificar Evento.	102
7.40	CU36: Modificar Compra.	102
7.41	CU37: Consultar Compra.	103
7.42	CU38: Eliminar Compra.	103
7.43	CU39: Modificar Cambio de Categoría.	104
7.44	CU40: Consultar Cambio de Categoría.	104
7.45	CU41: Eliminar Cambio de Categoría.	105
7.46	CU42: Modificar Traslado.	105
7.47	CU43: Consultar Traslado.	106
7.48	CU44: Eliminar Traslado.	106
7.49	CU45: Modificar Venta.	107

7.50 CU46: Consultar Venta.	107
7.51 CU47: Eliminar Venta.	108
7.52 CU48: Modificar Mortandad.	108
7.53 CU49: Consultar Mortandad.	109
7.54 CU50: Eliminar Mortandad.	110
7.55 CU51: Modificar Nacimiento.	110
7.56 CU52: Consultar Nacimiento.	111
7.57 CU53: Eliminar Nacimiento.	111
7.58 CU54: Registrar Aborto.	112
7.59 CU55: Modificar Aborto.	112
7.60 CU56: Consultar Aborto.	113
7.61 CU57: Eliminar Aborto.	113
7.62 CU58: Registrar Destete.	114
7.63 CU59: Modificar Destete.	114
7.64 CU60: Consultar Destete.	115
7.65 CU61: Eliminar Destete.	115

Índice de tablas

Listings

Capítulo 1

Introducción

Capítulo 2

Análisis de Requerimientos

1. Introducción

Introducción al capítulo... no sé si es necesario o no.. preguntar...!

2. Ingeniería de Requerimientos

2.1. Introducción

En *Ingeniería de Software*, la *Ingeniería de Requerimientos* comprende todas las tareas relacionadas con la determinación de las necesidades o las condiciones que un software debe satisfacer, tomando en cuenta los diversos requisitos de las partes interesadas. El propósito de esta disciplina es hacer que los mismos alcancen un estado óptimo antes de alcanzar la fase de diseño, en todo proyecto.

La *Ingeniería de Requerimientos* cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

Si bien existen muchas definiciones para requerimiento, ha continuación se presenta la definición que aparece en el glosario de la *IEEE*:

- *Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.*
- *Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.*
- *Una representación documentada de una condición o capacidad, como en los casos anteriores.*

Sommerville (2005) afirma que cuando hablamos de requerimientos debemos saber que existen requerimientos funcionales y no funcionales.

- **Requerimientos Funcionales:** son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar ante entradas particulares y de cómo se debe comportar en diversas situaciones.
- **Requerimientos No Funcionales:** son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen las restricciones de tiempo, sobre el proceso de desarrollo y estándares.

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, debe presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las propiedades de los requerimientos, según la IEEE:

- **Necesario:** un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.
- **Conciso:** un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarla en un futuro.
- **Completo:** un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- **Consistente:** un requerimiento es consistente si no es contradictorio con otro requerimiento.
- **No ambiguo:** un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje utilizado en su definición, no debe causar confusiones al lector.
- **Verificable:** un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

A continuación se darán algunas definiciones para ingeniería de requerimientos:

- *Ingeniería de Requerimientos* ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software. (Pressman, 2006: 155).
- *Ingeniería de Requerimientos* es el proceso de desarrollar una especificación de software. Las especificaciones pretender comunicar las necesidades del cliente a los desarrolladores del sistema. (Sommerville, 2005: 82)
- *Ingeniería de Requerimientos* es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema. (Boehm 1979).

- Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos. (Leite 1987).

2.2. Actividades de la Ingeniería de Requerimientos

Cinco actividades básicas se deben llevar a cabo para completar el proceso de recolección de requerimientos. Realizar las mismas, ayuda a reconocer la importancia que tiene para el desarrollo de un proyecto de software, realizar una especificación y administración adecuada de los requerimientos de los clientes o usuarios. Las actividades son:

- **Obtener requisitos:** es como comúnmente se denomina a las actividades involucradas en el descubrimiento de los requerimientos del sistema.
- **Analizar requisitos:** se enfoca en descubrir problemas con los requerimientos del sistema identificados hasta el momento.
- **Documentar requisitos:** en esta fase se documentan los requerimientos acordados con el cliente, en un nivel apropiado de detalle.
- **Verificar los requisitos:** consiste en comprobar el correcto funcionamiento de un requisito en la aplicación.
- **Validar los requisitos:** en esta fase, el objetivo es ratificar los requerimientos, es decir, verificar todos los requerimientos que aparecen en el documento especificado para asegurar que representan una descripción, por lo menos, aceptable del sistema que se debe implementar.

2.3. Interesados o Stakeholders

Se identificaron a todas aquellas personas o grupo de ellas que se verán afectados por el sistema, ya sea de forma directa o indirecta; Sommerville (2005) denomina a éstos como *stakeholders*. Los stakeholders identificados son:

- **Alumno:** autor del presente proyecto.
- **Director:** quien se encarga de coordinar y gestionar, como así también de guiar al alumno a lo largo de todo el desarrollo del proyecto.
- **Asesores Temáticos:** quienes prestan colaboración para con el alumno, brindando información y compartiendo conocimiento basado en su formación y experiencia.
- **Docentes:** quienes forman el cuerpo docente de la Cátedra de Proyecto Final de Carrera.
- **Usuarios:** quienes van a utilizar el software, pudiendo ser dueños o empleados de la empresa o entidad ganadera.

2.4. Ingeniería de Requerimientos: Técnicas y Herramientas

Serán mencionadas algunas de las técnicas más populares. En la práctica, la técnica más apropiada para cada actividad dependerá del proyecto que se esté desarrollando.

- **Entrevistas:** la entrevista es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades.
- **Observación:** por medio de esta técnica el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades.
- **Estudio de documentación:** varios tipos de documentación, como manuales y reportes, pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones.
- **Tormenta de Ideas (Brainstorming):** consiste en reuniones con pocas personas donde como primer paso sugieren toda clase de ideas sin juzgar su validez, y luego se realiza un análisis detallado de cada propuesta.

Las entrevistas y observaciones fueron claves para recolectar los requerimientos de este trabajo. Esto se debe, principalmente, a la naturaleza del mismo. Las siguientes, son algunas de las preguntas que se formularon y realizaron a los asesores temáticos:

- ¿Cuáles son las razas vacunas que existen en Argentina?
- ¿Cuáles son sus características, qué producto se obtiene, fortalezas, debilidades, ventajas y desventajas de trabajar con estas razas?
- ¿Cómo es el ciclo de vida de un vacuno?
- ¿Cuál es el clima adecuado para trabajar con cada raza y con qué características debe contar el lugar en donde se alojan los animales?
- ¿Cómo debe ser su alimentación?
- ¿Cuáles son los aspectos mas importantes a tener en cuenta en este tipo de negocio?
- ¿Cuáles son los cuidados sanitarios que deben tenerse o respetarse para con los vacunos?
- ¿Cuáles fueron las razones por las que se ha decidido utilizar un sistema informático como herramienta para facilitar las labores?
- ¿De qué manera el nuevo software los beneficiará o ayudará?
- ¿El nuevo software, qué esperan que haga por ustedes?

Luego de haber confeccionado las preguntas, se coordino con las personas a entrevistar un lugar, día y horario para realizar las entrevistas. Los asesores son:

- **Ing. Agro. Jorge Cinquini:** dueño y gerente general de la empresa ganadera que se tomó como caso de estudio y sobre la cual se basó el proyecto. La empresa tiene cerca de 10 años de trayectoria en la cría y producción de ganado y cuenta con mas de 2500 cabezas. Como profesional, dedica su vida a llevar adelante la empresa tomando las decisiones necesarias para lograr el bienestar tanto de empleados como de animales.

La empresa cuenta con 3 campos en los cuales se encuentran distribuidos los bovinos. Esta distribución no es equitativa y encuentra sus fundamentos a raíz de diversas razones, ejemplo de estas suelen ser razones climáticas o estratégicas. Por ejemplo, puede ocurrir que debido a inundaciones sea necesario trasladar vacunos de un campo a otro ó, buscando alimentar con mejores pasturas a aquellos vacunos que hayan sido destinados para engorde y, posteriormente, venta a frigoríficos.

- **Med. Vet. Juan Carlos Garcia:** es el médico veterinario encargado de los campos que posee la empresa. Cuenta con la colaboración de colegas, los cuales también formaron parte de las entrevistas, para controlar y realizar actividades sanitarias como así también de identificación y distribución de bovinos, siendo estas últimas las principales disparadoras de elaborar un sistema preciso para lograr el seguimiento.

Las entrevistas realizadas revelaron la mayoría de los requerimientos encontrados. Mientras que, las observaciones de actividades de campo, permitieron encontrar nuevos requerimientos, como así también aclarar y refinar los ya encontrados.

3. Requerimientos Identificados

De esta manera, siguiendo con las técnicas de recopilación de requerimientos mencionadas anteriormente, se obtuvieron los requerimientos funcionales y no funcionales que definen las características con las que deberá contar el software a ser desarrollado. A continuación, se procede a listar los requerimientos obtenidos junto con la descripción de cada uno de ellos, buscando expresar, de manera breve, la importancia de los mismos en dicho trabajo.

3.1. Requerimientos Funcionales.

Los requerimientos funcionales que se han encontrado, son los siguientes:

- **RF01 - Registro y Autenticación de Usuarios:** el sistema debe permitir el registro y autenticación de usuarios.
- **RF02 - Perfiles de Usuario:** el sistema debe contar con varios perfiles que permitan acceder a distintas características o funcionalidades. Los perfiles con los que contará el software son: Administrador, Encargado en Jefe, Encargado, Veterinario en Jefe, Veterinario y Peón.

- **RF03 - RBAC (Control de Acceso Basado en Roles):** el sistema debe ser capaz de implementar este tipo de control, para salvaguardar la información.
- **RF04 - Validación:** el sistema debe validar que la información ingresada por el usuario sea correcta, respetando los tipos y formatos predefinidos.
- **RF05 - Carga de Información de Campos:** el sistema debe contar con un modulo para configuración de Campos, en los cuales, se tengan animales. Junto con ello, deberá permitir que el usuario defina la manera en que los vacunos serán dispuestos, es decir, geográficamente, en dicho campo. En otras palabras, el sistema debe permitir al usuario dividir sus tierras por lotes, potreros, zonas, corrales, entre otros.
- **RF06 - Carga de Información de Razas:** el sistema debe contar con un módulo para configuración de Razas con las que se desee trabajar. A través de dicho módulo, la empresa pueda ajustar el sistema a sus necesidades, es decir, con qué de razas, categorías y subcategorías llevan adelante sus actividades.
- **RF07 - Registro de Vacunos:** el sistema, a la hora de ingresar un nuevo vacuno, debe permitir asignarle un identificador, como así también ingresar información adicional sobre el bovino como por ejemplo raza, línea genética, premios obtenidos, progenitores, peso, condición corporal, categoría, edad, entre otros.
- **RF08 - Carga de Información de Vacunos:** el sistema debe permitir la carga de las actividades sanitarias, reproductivas y alimenticias realizadas sobre los bovinos, como vacunaciones, modificaciones en la dieta según la edad, servicios, entre otras.
- **RF09 - Stock:** el sistema debe permitir visualizar en tiempo real el stock de animales. El mismo debe estar presentado de manera clara, discriminando por razas, categorías y subcategorías.
- **RF10 - Registro de Operaciones:** El sistema debe contar con un módulo para el registro de operaciones. Entre ellas, se encuentran la Compra, Venta, Cambio de Categoría, Mortandad, referentes al ciclo de vida de los animales y al desempeño de la empresa.
- **RF11 - Creación de Eventos:** el sistema debe permitir la creación de eventos que permitan llevar adelante el seguimiento de los vacunos. Un ejemplo de estos eventos suelen ser, por ejemplo, cuándo dar servicio a las hembras y cuándo realizar tacto sobre las mismas.
- **RF12 - Notificaciones:** el sistema debe notificar al usuario ante el advenimiento de algún evento. El aviso se hará a través del envío de un correo electrónico a los usuarios interesados y, a través de mensajes en ventanas popup dentro del sistema.
- **RF13 - Creacion de Reportes:** el sistema debe permitir la creación de reportes mediante los cuales se pueda obtener, de manera resumida, la información almacenada en el sistema. Entre los reportes mas comunes se encuentran el resumen de stock, el resumen de operaciones realizas en un período específico o la información de un vacuno en particular.
- **RF14 - Log:** el sistema debe poder llevar algún tipo de registro de accesos al sistema, de operaciones o consultas realizadas. En caso de ocurrir algún error, esto será de utilidad para poder identificar el motivo que dió origen al mismo.

3.2. Requerimientos No Funcionales

- **RNF01 - Disponibilidad (Availability):** medida de disponibilidad del sistema para su uso. El sistema debe estar disponible para su uso las 24 horas del día.
- **RNF02 - Seguridad/Confidencialidad (Confidentiality):** ausencia de acceso no autorizado a la información. El sistema debe brindar un servicio de control de acceso basado en roles, con el mismo se espera brindar protección a los datos e información tanto de usuarios como propias del negocio. La aplicación estará habilitada solo para aquellos usuarios registrados. Se deberá validar la autenticación de usuario al momento de interactuar con el sistema.
- **RNF03 - Funcionalidad (Functionality):** habilidad del sistema para realizar el trabajo para el cual fue concebido. El sistema debe comportarse de manera adecuada, según los objetivos para los cuales se lo haya creado.
- **RNF04 - Desempeño (Performance):** grado en el cual un sistema o componente cumple con sus funciones designadas, dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria. Los tiempos de respuesta que serán tolerados deberán estar por debajo de 5 segundos.
- **RNF05 - Confiabilidad (Reliability):** medida de la habilidad de un sistema de mantenerse operativo a lo largo del tiempo. El sistema debe funcionar de igual manera sin importar el tiempo de uso que posea.
- **RNF06 - Seguridad externa (Safety):** medida de ausencia de errores que generan pérdidas de información. El sistema debe estar libre de fallas que provoquen la perdida de información o la inoperabilidad del mismo.
- **RNF07 - Seguridad interna (Security):** medida de la habilidad del sistema para resistir a intentos de uso no autorizados y negación de servicios, mientras se sirve a usuarios legítimos. El sistema debe impedir el acceso a usuarios no autorizados, mientras habilita a los que sí poseen autorización.
- **RNF08 - Adaptabilidad:** el sistema debe funcionar independientemente del sistema operativo y navegador web con el que se cuente. Como así también, deberá adaptarse para ser utilizado en dispositivos móviles como Tablets y SmartPhones.
- **RNF09 - Diseño Escalable y Mantenible:** el sistema debe desarrollarse de forma tal que su mantenimiento, corrección de errores o introducción de modificaciones, pueda hacerse de manera sencilla. Así mismo, en caso de ser necesario agregar nuevas funcionalidades o módulos, el sistema debe ser capaz de permitir el anexo de nuevas características sin afectar las ya existentes. El diseño de la interfaz debe ser simple, consistente y de navegación intuitiva.
- **RNF10 - Acceso a Internet:** el sistema trabaja con conexión a internet, por defecto.
- **RNF11 - Concurrencia:** sin restricciones. Todos los usuarios pueden utilizar la aplicación al mismo tiempo. Se estima que el sistema pueda contar con un máximo de 30 usuarios.

- **RNF12 - Manejo de Errores:** en caso de ocurrir errores, los mismos deben estar expresados a través de mensajes claros, ofreciendo al usuario una descripción de lo ocurrido y, por otro lado, brindar datos de contacto para comunicarse con el administrador.

4. Representación de los requerimientos recolectados

El *Lenguaje Unificado de Modelado*, UML, tiene como objetivo permitir visualizar, especificar, construir y documentar un sistema para generar un modelo que represente la interacción del sistema desde el punto de vista del usuario. Para llevar a cabo el proyecto se optó por utilizar Diagramas de Casos de Uso. Por otra parte, los diagramas tienen la característica de ser útiles como herramientas de validación y verificación en la etapa de pruebas, encontrando las discrepancias existentes entre los requerimientos y las funcionalidades del software.

Según Sommerville (2005), un Caso de Uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede ejecutar y que produce un resultado observable para un actor particular. En otras palabras, es una descripción de los pasos o actividades que deberán realizarse para llevar a cabo un proceso. Por otro lado, Sommerville (2005) afirma que un actor es un componente que representa una entidad externa al sistema y que estimula al caso de uso con eventos de entrada, esperando una respuesta. Por tanto, podríamos decir que un diagrama de casos de uso muestra las relaciones entre los actores y los casos de uso. Dicho de otro modo, es un diagrama que muestra la comunicación y el comportamiento de un sistema mediante su interacción con los actores y/o otros sistemas.

Con la ayuda de un diagrama de casos de uso, es posible analizar y comunicar:

- *Los escenarios en los que el sistema o aplicación interactúa con personas, organizaciones o sistemas externos.*
- *Los objetivos que el sistema o aplicación contribuye a lograr.*
- *El ámbito del sistema.*

En un diagrama de casos de uso no se muestran los casos de uso en detalle; solamente se resumen algunas de las relaciones entre los distintos componentes que forman parte del diagrama. En concreto, en el diagrama no se muestra el orden en que se llevan a cabo los pasos para lograr los objetivos de cada caso de uso. Esos detalles pueden describirse en otros diagramas y documentos, que pueden vincularse a cada caso de uso.

En las descripciones que se proporcionen de los casos de uso, se usarán diversos términos relacionados con el dominio en el que trabaja el sistema:, como *Cambios de Categoría*, *Destetes*, *Raza*, *Campo*, entre otros. Es importante definir de manera clara estos términos y sus relaciones. Para ello, resulta útil contar con un diagrama de clases UML, el mismo será incluido en el capítulo 3.

4.1. Casos de Uso

Los casos de uso solamente se usan para los requisitos funcionales de un sistema. Otros requisitos, como las reglas de negocio, calidad del servicio y las restricciones de implementación, deben representarse por separado. La arquitectura y los detalles internos también deben describirse por separado.

Un diagrama de casos de uso esta compuesto por distintos componentes, los cuales se describen a continuación:

- **Actor:** *un actor es una clase de persona, organización, dispositivo o componente de software externo que interactua con el sistema.*
- **Caso de Uso:** *un caso de uso representa las acciones que uno o varios de los actores realizan a fin de conseguir un objetivo determinado.*
- **Asociaciones:** *en un diagrama de casos de uso, los casos de uso están asociados a los actores que los realizan.*
- **Sistema:** *el sistema es aquello que se está desarrollando. Puede ser un pequeño componente de software cuyos actores simplemente son otros componentes de software; puede ser una aplicación completa; o puede ser un gran conjunto de aplicaciones distribuidas que se implementan en muchos equipos y dispositivos.*

Existen innumerables herramientas que ayudan a generar diagramas UML, como por ejemplo: StarUML, ArgoUML, Enterprise Architect, entre otras. Se optó por utilizar Enterprise Architect debido al conocimiento y experiencia previa con la herramienta.

El modelo de casos de uso, además de su representación gráfica, está compuesto por las especificaciones textuales de cada caso de uso. De forma tal que podemos refinar o especificar con mayor nivel de detalle el objetivo de los casos de uso.

La especificación de cada caso de uso se baso en el siguiente formato:

- **Caso de Uso:** *nombre del caso de uso.*
- **Descripción:** *resumen del objetivo del caso de uso.*
- **Actor(es):** *actor o actores que hacen uso del caso de uso.*
- **Precondición:** *resumen de las condiciones que deben cumplirse para que el caso de uso comience. (Opcional)*
- **Curso Normal:** *describe los pasos que suceden al momento de ejecutar el caso de uso.*
- **Curso Alternativo:** *describen las alternativas que surgen a partir del flujo normal.*
- **Poscondición:** *resumen las condiciones que se deben cumplir luego de la ejecución de un caso de uso. (Opcional)*

- **Inclusiones:** aquellos casos de uso que necesitan ser invocados para lograr cumplir con el objetivo del caso de uso actual.
- **Extensiones:** aquellos casos de uso que no necesariamente deben ser invocados para cumplir con el objetivo del caso de uso actual.

Los diagramas de casos de uso pueden verse en la Sección 1 mientras que las fichas textuales se encuentran en la Sección 2, estando ambas secciones en el Apéndice 7

Cabe destacar que, por cuestiones de practicidad y legibilidad, se decidió dividir al diagrama de casos de uso en cuatro diagramas más pequeños, en los que se pueda apreciar de manera más clara las acciones de cada actor al utilizar el sistema. De esta manera, se obtuvieron los diagramas para el usuario **Administrador** (ver Fig. (7.1)), para el usuario **Encargado** fue necesario, a su vez, generar dos diagramas que permitan visualizar el total de las acciones que el mismo puede realizar (ver Fig. (7.2) y (7.3)) y finalmente, para el usuario **Veterinario** (ver Fig. (7.4)).

En los diagramas podrá observarse que algunos de los casos de uso o actores aparecen repetidos. En caso de estar repetidos se antepone a su nombre un * (asterisco), esto forma parte de la estrategia de diseño que se optó al dividir al diagrama de casos de uso, buscando encontrar legibilidad y claridad, facilitando su comprensión.

5. Análisis y Definición de Framework y Tecnologías Complementarias

Desarrollo web es un término que define la creación de sitios web para Internet. Para conseguirlo se hace uso de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador web a fin de realizar determinadas tareas o mostrar información.

Tradicionalmente un software puede ser desarrollado usando lenguajes compilados (C, C++, Delphi), semicompileados (.NET, Java), o interpretados (Python, PHP) para crear tanto la funcionalidad como la interfaz de usuario, pero hoy en día se ha vuelto popular el desarrollo orientado a web, siendo más homogéneo y multiplataforma, y dependiendo de las tecnologías utilizadas, más rápido y robusto para diseñar, implementar y probar, una vez terminado.

En diseño de software el *front-end* es la parte del software que interactúa con los usuarios y el *back-end* es la parte que procesa la entrada desde el front-end. Esta división es un tipo de abstracción que ayuda a mantener separadas las diferentes partes del sistema. La idea general es que el front-end sea el responsable de recolectar los datos de entrada del usuario, que pueden ser de muchas y variadas formas, y los transforma ajustándolos a las especificaciones que demanda el back-end para poder procesarlos, devolviendo generalmente una respuesta que el front-end recibe y expone al usuario de una forma entendible para este. La conexión del front-end y el back-end es un tipo de interfaz.

5.1. Lenguajes de Programación Back-End

En la actualidad existen diversos lenguajes de programación del lado del servidor entre los cuales podemos mencionar Java, Ruby, PHP y Python. A continuación se realiza una breve descripción de los mismos.

- **Java:** es un lenguaje de programación compilado orientado a objetos que, en la actualidad, se ha convertido en uno de los lenguajes más usados y más demandados por los desarrolladores. Es de tipado fuerte, pues exige declaraciones explícitas para funcionar. Posee una curva de aprendizaje más marcada. La gran ventaja de Java es que puede ser usado para crear aplicaciones independientes de la plataforma. Cualquier ordenador o dispositivo móvil que pueda ejecutar una máquina virtual de Java puede ejecutar una aplicación Java. El inconveniente de ejecutarse dentro de una máquina virtual es que el programa Java se ejecuta más lentamente que los programas desarrollados con otros lenguajes como PHP o Python.
- **Ruby:** es un lenguaje de scripting interpretado cuya sintaxis es concisa. Es similar a Java en que ambos son lenguajes orientados a objetos y están fuertemente tipados. Ruby está tipado dinámicamente mientras que Java está tipado estáticamente, de esta manera, las declaraciones de tipo no se utilizan. Tanto Java como Ruby proporcionan herencia y tienen métodos públicos, privados y protegidos. Ruby posee una curva de aprendizaje mas leve en comparación con Java y más rápida también.
- **PHP (Hypertext Pre-processor o Procesador de Hipertexto):** es un lenguaje de programación interpretado del lado del servidor, especialmente adecuado para el desarrollo web. Es rápido, flexible y pragmático. Con su crecimiento, surgieron proyectos asociados tales como Frameworks, IDE's (Entorno de desarrollo integrado), que le han aportado robustez y consistencia. Otro aspecto a tener en cuenta y que suma confianza a este lenguaje, es el hecho de que muchas de las páginas con mayor número de visitas han sido desarrolladas bajo esta tecnología. Es de tipado débil, lo cual lo hace más flexible y más tendente al sentido común de cómo llevar a cabo una tarea.
- **Python:** es fácil de dominar. Su sintaxis está diseñada para ser intuitiva y su relativa simplicidad permite a los principiantes comenzar rápidamente a escribir código para diversas aplicaciones. Es de tipado débil y para ejecutarlo se necesita un compilador que pueda convertir el código a uno que el sistema operativo pueda entender.

La elección del lenguaje se basó, principalmente, en la curva de aprendizaje. Luego se tuvo en cuenta el tamaño y el grado de actividad de la comunidad, como así también la disponibilidad de documentación y conocimientos previos. Siguiendo los criterios anteriormente mencionados, el lenguaje elegido para el desarrollo de este proyecto es PHP. Sus características, mencionadas anteriormente, hacen de este lenguaje de programación el más conveniente a ser tomado como lenguaje base para el desarrollo del sitio web propuesto.

5.2. Lenguajes de Programación Front-End

HTML (HyperText Markup Language) es el lenguaje en que se construyen los sitios web. CSS (Cascading Style Sheets) es el lenguaje que se utiliza para dar forma a los sitios web. Javascript es un lenguaje de programación que permite que el cliente -o sea, el usuario- pueda hacer cambios en un sitio web sin tener que recargarlo cada vez que quiera modificar algo. La desventaja de este idioma es que cada navegador puede interpretarlo de forma distinta, por lo que no es fácil utilizarlo correctamente.

“HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto que es extremadamente significativo” (Gauchat, 2012, p.1).

Por tanto se ha definido utilizar HTML, CSS y Javascript como los lenguajes de programación del lado del cliente, para el desarrollo del presente trabajo.

5.3. Frameworks Back-End

Un framework es una estructura de software creada para ayudar en el desarrollo. Es una aplicación o conjunto de módulos que permiten, o tienen por objetivo, el desarrollo ágil de aplicaciones mediante la aportación de bibliotecas y/o funcionalidades ya creadas. Las principales razones para utilizar un framework son las siguientes:

- *Evitan escribir código repetitivo.*
- *Permiten centrarse en programar la aplicación.*
- *Utilizan buenas prácticas.*
- *Ofrecen una estructura base común.*
- *Aceleran el desarrollo.*
- *Reducen los costos.*
- *Optimizan el código.*

Por estos motivos es que se ha decidido utilizar un framework para el desarrollo de este proyecto. Los criterios de aceptación para la selección del mismo son los siguientes:

- ***Programación Orientación a Objetos:*** debe permitir y facilitar la utilización de este paradigma.
- ***Patrón de diseño MVC:*** debe permitir la utilización de este patrón de diseño.

- **Bases de datos relacionales:** debe facilitar la utilización de bases de datos relacionales.

Hasta aquí, hemos definido el lenguaje de programación back-end (PHP) y se han mencionado las ventajas de utilizar un framework. Pero aún queda por definir, entre otras cosas, cuál de todos los frameworks PHP existentes se va a utilizar. Pero, ¿Por qué elegir un framework PHP?. Algunos de los beneficios de utilizar un framework PHP son:

- *Un framework PHP hace que el desarrollo sea más rápido. Por ejemplo, no es necesario escribir consultas complejas para recuperar información almacenada en la base de datos debido a que proporcionan operaciones CRUD (Create, Read, Update and Delete o Crear, Leer, Actualizar y Eliminar).*
- *Los framework permiten a los desarrolladores escalar sistemas fácilmente.*
- *El mantenimiento de código es más fácil que con una aplicación desarrollada en PHP puro. El código de la aplicación es conciso y fácil de usar.*
- *El modelo MVC asegura un desarrollo rápido.*
- *Los framework son mejores para proteger la aplicación web de amenazas de seguridad comunes.*
- *El principio DRY (Don't Repeat Yourself o Una vez y sólo una) garantiza que el mínimo código tenga un máximo impacto.*

Los beneficios anteriores son demasiado importantes para ser ignorados. A pesar de que PHP sin formato se puede utilizar para crear cualquier aplicación, los estándares de desarrollo actuales requieren herramientas y habilidades de administración del tiempo, para satisfacer la demanda del mercado. Con lo cual, la utilización de un framework PHP, es la mejor forma de llevar adelante el presente trabajo.

En la actualidad, existe una amplia gama de framework PHP disponibles. Decidir cuál de todos ellos es el mas indicado para el trabajo que se desea realizar, se vuelve una tarea crucial ya que el mismo será utilizado a lo largo de todo el desarrollo y sus características deben ser las adecuadas de manera que asimilar esta herramienta nos beneficie según las propiedades mencionadas anteriormente. De esta manera surge la pregunta ¿Cómo elegir un framework PHP?. Responder las siguientes cuestiones puede ayudar a elegir el framework adecuado:

- *¿Cuáles son las características y la funcionalidad del framework?, ¿Ofrece lo que necesito?.*
- *¿Cuál es la curva de aprendizaje del framework?.*
- *¿Qué tan escalable es el framework?.*
- *¿El equipo de trabajo desarrolla y mantiene activamente el framework?.*
- *¿El framework proporciona soporte a largo plazo (soporte LTS)?.*
- *¿El framework tiene un fuerte apoyo comunitario?.*

Dentro de los frameworks que cumplen con los criterios anteriores se han analizado los siguientes: Symfony, Laravel y Yii. Los tres frameworks son buenas opciones para construir aplicaciones web, pero cada uno tiene un propósito diferente. Se ha realizado una comparativa de sus características, ventajas y desventajas con el fin de elegir el que mejor se ajuste a las necesidades de este proyecto.

Symfony ¹

Es un framework Open Source diseñado para optimizar el desarrollo de las aplicaciones web basándose en el patrón de diseño MVC (Modelo Vista Controlador). Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web generando así, código fácil de leer, permitiendo un sencillo mantenimiento. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de un sistema. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Utiliza programación orientada a objetos y es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas basadas en Unix como en plataformas Windows.

Es sencillo utilizarlo en la mayoría de casos. Sin embargo, es recomendable su empleo en desarrollos de grandes aplicaciones en lugar de pequeños proyectos.

Laravel ²

Es un framework Open Source para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el “código espagueti”. Tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta valerse de lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.

Facilita la interacción con bases de datos con la ayuda de un mecanismo avanzado de creación de consultas. La autenticación de usuarios es mucho más simple ya que contiene el sistema de autenticación incorporado, listo para usar. Su curva de aprendizaje es relativamente baja, reduciendo costos y tiempos en el desarrollo y mantenimiento. Es flexible y adaptable al modelo MVC y posee una buena y completa documentación en su sitio oficial, acompañado de una amplia comunidad y foros.

Se encuentra en constante cambio y actualización, pero esto no implica la necesidad de reescribir código si se desea contar siempre con la última versión. En lugar de esto, sólo se deben actualizar sus dependencias y automáticamente se encuentran disponibles las nuevas características aportadas por la actualización. Siempre es recomendable utilizar una versión LTS.

¹Symfony. <https://symfony.com/what-is-symfony>.

²Laravel. <https://laravel.com/>.

Yii ³

Es un framework Open Source basado en componentes de alto rendimiento para el desarrollo rápido de aplicaciones web modernas. Debido a su arquitectura basada en componentes y su sofisticado soporte de caché, es especialmente adecuado para desarrollar aplicaciones a gran escala como portales, foros, sistemas de gestión de contenido (CMS), proyectos de comercio electrónico, servicios web REST, entre otros. Al igual que Laravel, utiliza el administrador de dependencias Composer, con el que maneja diferentes dependencias e instalaciones.

Admite patrón de diseño MVC y cuenta con soporte de autenticación de usuarios incorporado. Permite el manejo de errores y *logging*. Los errores son manejados y personalizados, y los *log* de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.

Luego de haber mencionado las principales características de los frameworks, Laravel ha resultado ser el framework back-end seleccionado para llevar adelante este proyecto. Se ha elegido por su curva de aprendizaje baja, por su gran comunidad y documentación disponible, por su sencillez, escalabilidad y mantenibilidad.

5.4. Frameworks Front-End

Bootstrap ⁴

Es un kit de herramientas de código abierto para desarrollar con HTML, CSS y Javascript. Es la biblioteca de componentes front-end más popular del mundo. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de Javascript adicionales.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS3, pero es compatible con la mayoría de los navegadores web modernos. Desde la versión 2.0 también soporta diseños web adaptables. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (computadoras, tabletas, teléfonos inteligentes).

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo Lees ⁵ que implementan la variedad de componentes de la herramienta. Utilizarlo permite el uso de variables, funciones y operadores, selectores anidados, así como clases *mixin* ⁶. Por otra parte, los desarrolladores eligen en un formulario los componentes y ajustes deseados, según sus necesidades.

³Yii. <https://www.yiiframework.com/>.

⁴Bootstrap. <https://getbootstrap.com/>.

⁵Lees: lenguaje dinámico de hojas de estilo que puede ser compilado en hojas de estilo en cascada (CSS) y ejecutarse del lado del cliente o del lado del servidor.

⁶Clase Mixin: es una clase que ofrece cierta funcionalidad para ser heredada por una subclase. Heredar de un mixin no es una forma de especialización sino más bien un medio de obtener funcionalidad.

Vue.js ⁷

Es un framework de Javascript progresivo. Es accesible, versatil y de buen rendimiento, con él es posible crear una base de código fácil de mantener y probar.

A diferencia de otros frameworks, está diseñado desde el inicio para ser adoptado incrementalmente. Por otro lado, también es perfectamente capaz de soportar aplicaciones sofisticadas utilizando en combinación con herramientas modernas y bibliotecas compatibles.

Su núcleo está formado por una biblioteca encargada de renderizar vistas en el navegador. Su forma de organizar el código es por medio de pequeños componentes que contienen todo el HTML, CSS y Javascript necesario para funcionar como pieza independiente. Estas piezas se organizan en un árbol jerárquico de componentes hasta formar la aplicación.

Algunas de las características mas importantes son:

- **Framework MVVM:** la ventaja de este tipo de frameworks es la facilidad para construir código bien estructurado, logrando aplicaciones complejas.
- **Solución ligera:** una de las grandes ventajas es el tamaño del núcleo. Este tamaño puede ir aumentando, debido a la flexibilidad y facilidad para extender el framework con variedad de soluciones de terceros que son bien recibidas por la comunidad. Su tamaño compactado reduce los tiempos de carga y velocidad de las aplicaciones.
- **Templates declarativos:** los templates se escriben en HTML fácilmente modificable por cualquier involucrado en el proyecto.
- **DOM virtual:** la implementación del DOM Virtual proporciona un alto performance, ubicándolo como líder en rendimiento de renderizado.
- **Two-way Data Binding:** al igual que otros frameworks, emplea un data-binding bidireccional que sincroniza automáticamente el modelo con el DOM.
- **Curva de aprendizaje baja:** comparado con otros frameworks, es una de las tecnologías Javascript más sencillas para comenzar a desarrollar. Una de las mejores características es que se trata de un framework *friendly developer* (*amigable para el desarrollador*). Además posee una excelente documentación.

⁷Vue. <https://vuejs.org/>.

5.5. Bases de Datos

Laravel tiene soporte para diversos motores de bases de datos tales como:

- MySQL
- PostgreSQL
- SQLite
- SQL Server

En SQLite la base de datos completa consta de un único archivo en el disco, lo que lo hace extremadamente portátil. Es ideal para desarrollar e incluso probar. En contrapartida, no cuenta con soporte para usuarios, es decir, conexiones administradas con privilegios de acceso a los datos. No permite concurrencia de conexiones, esto quiere decir que si un usuario está modificando datos, otro no podrá hacerlo hasta que el anterior termine sus acciones. Por lo tanto se ha decidido descartar a SQLite de la lista anterior. En cuanto a SQL Server, también debe ser descartado, debido a la falta de licencia de software libre.

Entonces, quedan MySQL y PostgreSQL. Para optar por uno de ellos, se ha realizado un análisis en el cual son expuestas las ventajas y desventajas, similitudes y diferencias que existen entre ambos. A continuación se listan los aspectos en los que se ha basado la decisión de utilizar uno de estos motores de bases de datos.

- **Código Abierto:** *el software de código abierto tiene beneficios únicos —costo, flexibilidad, seguridad y responsabilidad— que no pueden ser superados por las soluciones de software privativo. El software de código abierto está disponible gratuitamente y puede ser redistribuido y modificado por cualquiera. El software de código abierto tiene viabilidad a largo plazo y siempre está a la vanguardia de la tecnología.*

PostgreSQL es desarrollado por el Grupo Global de Desarrollo de PostgreSQL, un grupo diverso de múltiples compañías y contribuyentes individuales. Es software libre y de código abierto. PostgreSQL se lanza bajo la licencia PostgreSQL, una licencia de código abierto liberal, similar a las licencias BSD o MIT.

El proyecto de desarrollo de MySQL ha hecho disponible su código fuente bajo los términos de la Licencia Pública General de GNU, así como una variedad de acuerdos privativos. Ahora es propiedad de Oracle Corporation y ofrece varias ediciones pagas para uso privativo.

- **Cumplimiento de ACID:** *ACID (atomicidad, consistencia, aislamiento, durabilidad) es un conjunto de propiedades de las transacciones de bases de datos. El cumplimiento de ACID asegura que ningún dato se pierda o transmita a otro lugar en el sistema en caso de una falla, incluso cuando se hacen varios cambios durante una sola transacción.*

PostgreSQL cumple con ACID por completo y se asegura de que se cumplan todos los requerimientos. MySQL solo cumple con ACID cuando se utiliza alguno de los mecanismos de almacenamiento InnoDB y NDB Cluster.

- **Cumplimiento de SQL:** es un estándar en el cual se establece que una base de datos debe cumplir e implementar los estándares y pautas del lenguaje de consulta estructurado. Cumplir con dicho estándar resulta en que sea muy fácil trasladar información de una base de datos hacia otra (por ejemplo, desde Oracle hacia PostgreSQL o SQL Server).

PostgreSQL cumple con SQL en gran medida. Mientras que MySQL cumple parcialmente en algunas versiones.

- **Replicación:** la replicación de bases de datos es la copia electrónica regular de datos desde una base de datos en un computador o servidor hacia una base de datos en otro, de modo que todos los usuarios comparten el mismo nivel de información. El resultado es una base de datos distribuida en la cual los usuarios pueden acceder a datos relevantes para sus tareas sin interferir con el trabajo de otros.

PostgreSQL soporta replicación maestro-standby e introduce mejoras significativas, lo que resulta en replicación casi en tiempo real y capacidad de espera activa (hot standby) para servidores standby. MySQL, también soporta replicación maestro-standby.

- **Rendimiento:** el rendimiento es un área que sólo se puede medir evaluando las métricas de escenarios potenciales, pues depende netamente de los requerimientos del usuario en específico y en la naturaleza de la aplicación.

PostgreSQL se usa ampliamente en sistemas grandes donde las velocidades de lectura y escritura son cruciales y los datos necesitan ser validados. Además, soporta una variedad de optimizaciones de rendimiento que sólo están disponibles en soluciones comerciales, como soporte para datos geoespaciales o concurrencia sin bloqueo de lecturas. MySQL es una opción popular para proyectos basados en web que necesitan una base de datos simplemente para transacciones de datos directas y sencillas. Sin embargo, es habitual que su rendimiento no sea el adecuado al someterlo a cargas pesadas o al intentar completar consultas complejas.

- **Seguridad:** la seguridad de la base de datos se refiere al conjunto de medidas usadas para proteger y asegurar una base de datos o gestor de base de datos contra el uso ilegítimo, amenazas maliciosas y ataques. Es un término amplio que incluye una variedad de procesos, herramientas y metodologías que garantizan la seguridad dentro de un ambiente de base de datos.

PostgreSQL tiene roles y roles heredados para establecer y mantener permisos. Tiene soporte nativo de SSL en conexiones para cifrar la comunicación cliente/servidor. También tiene seguridad a nivel de registros. Además de esto, viene con una mejora integrada llamada SE-PostgreSQL, la cual provee controles de acceso adicionales basados en las políticas de seguridad de SELinux.

MySQL implementa la seguridad basada en Listas de Control de Acceso (ACLs) para todas las conexiones, consultas y otras operaciones que un usuario pudiera intentar realizar. También posee soporte para conexiones cifradas con SSL entre clientes y servidores MySQL.

- **Hosting:** a medida que más empresas optan por mover sus datos hacia la nube, la capacidad de encontrar proveedores que soporten su base de datos, se vuelve cada vez más importante. El hosting en la nube brinda flexibilidad en los servidores, permitiendo expandir o contraer rápidamente su capacidad. También, permite reducir el tiempo potencial fuera de línea mientras gestiona fácilmente picos de carga de trabajo.

PostgreSQL es soportado por los proveedores más importantes de servicios en la nube, incluyendo Amazon, Google y Microsoft. Por su parte, MySQL también es soportado por estos proveedores.

- **Soporte comunitario:** PostgreSQL tiene una comunidad muy fuerte y activa que constantemente mejora sus características, mientras sus innovadores committers se esfuerzan para asegurar que siga siendo la base de datos más avanzada, con nuevas e innovadoras características y seguridad.

MySQL tiene una gran comunidad de contribuyentes que, particularmente, desde la adquisición por parte de Oracle, se centra en mantener sus características con algunas características nuevas surgiendo ocasionalmente.

- **Soporte para concurrencia:** la concurrencia significa que varios usuarios pueden tener acceso a los datos al mismo tiempo. Es una de las principales funcionalidades consideradas al desarrollar un sistema que requiera que varios suscriptores accedan a los datos al mismo tiempo, pues mejora la capacidad de que muchas personas accedan y utilicen la base de datos en varias ubicaciones, simultáneamente.

PostgreSQL hace frente a la concurrencia de forma eficiente implementando MVCC (control de concurrencia multiversión), la cual logra niveles muy altos de concurrencia. MySQL solo tiene soporte MVCC en InnoDB.

- **Soporte para NoSQL/JSON:** tanto NoSQL como JSON son populares, y las bases de datos NoSQL son cada vez más comunes. JSON es un formato simple de datos que permite a los programadores almacenar y comunicar conjuntos de valores, listas y correlaciones clave-valor entre sistemas.

PostgreSQL soporta JSON y otras características NoSQL como soporte nativo para XML y pares clave-valor con HSTORE. También soporta indexación de datos JSON para acceso más rápido. MySQL soporta JSON, pero no así NoSQL. Tampoco soporta indexación para JSON.

- **Vistas materializadas y tablas temporales:** una vista materializada es un objeto de base de datos que contiene los resultados de una consulta que puede ser actualizada según sea necesario desde la tabla base original. Se la puede ver como un “caché” para bases de datos. Una tabla temporal almacena datos que no se necesite que persistan más allá de lo que dure la sesión que la cree. La principal forma en que difiere de las vistas materializadas es que estas últimas ofrecen la capacidad de actualizar periódicamente los datos, resultando en mejor eficiencia para ese caso de uso.

PostgreSQL soporta vistas materializadas y tablas temporales. MySQL soporta tablas temporales pero no soporta vistas materializadas.

- **Soporte para datos geoespaciales:** los datos geoespaciales son todos los puntos de datos geográficos que una base de datos mantiene y puede proveer para análisis. Es la información sobre un objeto físico que puede ser representada por valores numéricos en un sistema de coordenadas geográfico.

PostgreSQL soporta datos geoespaciales mediante la extensión PostGIS. Hay funciones y tipos dedicados para datos geoespaciales, disponibles directamente a nivel de base de datos, haciendo que el análisis y la codificación sea más fácil para los desarrolladores. En MySQL el soporte para datos geoespaciales está integrado.

- **Soporte para lenguajes de programación:** el soporte para lenguajes de programación ayuda a una gran variedad de desarrolladores a realizar muchas tareas en el lenguaje en el que tengan más experiencia. Los desarrolladores pueden decidir con libertad, a nivel de casos individuales, si van a realizar cierto procedimiento en el servidor o en el cliente, pues el servidor soporta una gran variedad de diferentes lenguajes de programación para funciones de base de datos. Los lenguajes de programación tienden a dar más poder a los desarrolladores.

PostgreSQL soporta una gran variedad de lenguajes de programación incluyendo: C/C++, Java, Javascript, .Net, R, Perl, Python, Ruby, Tcl entre otros; incluso es posible ejecutar código proporcionado por el usuario en un proceso separado (esto es, ejecutándose como trabajadores en segundo plano). MySQL tiene algo de soporte para programación del lado del servidor, en un solo lenguaje no extensible.

- **Sistema de tipos extensible:** una base de datos que soporta un sistema de tipos extensible puede ser extendido por el usuario de varias formas, lo que incluye agregar nuevos tipos de datos, funciones, operadores, funciones de agregado, métodos de indexación y lenguajes de procedimientos.

PostgreSQL tiene varias características dedicadas a la extensibilidad. Es posible agregar nuevos tipos, nuevas funciones o nuevos tipos de índice, por ejemplo. MySQL no tiene soporte para extensibilidad.

En la tabla (2.1) se muestra un cuadro resumen de la comparación realizada, según los puntos listados anteriormente. Aunque hay muchas similitudes y coincidencias entre ambos, también hay diferencias bien marcadas. Sin embargo, es necesario evaluar el escenario particular y determinar qué base de datos es la más adecuada para el caso de uso específico. Por el tipo de trabajo a realizar y por las características buscadas, se elige a PostgreSQL como DBMS.

Características	PostgresSql	MySql
Código abierto	Completamente de código abierto	De código abierto, pero propiedad de Oracle y ofrece versiones comerciales
Cumplimiento de ACID	Cumple completamente con ACID	Algunas versiones cumplen
Cumplimiento de SQL	Cumple casi completamente	Algunas versiones cumplen
Soporte para concurrencia	Implementación de MVCC soporta múltiples peticiones sin bloquear lecturas	Soporte en algunas versiones
Seguridad	Seguro desde las bases con soporte SSL	Soporte SSL en algunas versiones
Soporte para NoSQL/JSON	Varias funcionalidades soportadas	Solo soporte para datos JSON
Métodos de acceso	Soporta todos los estándares	Soporta todos los estándares
Replicación	Múltiples tecnologías de replicación disponibles: - Un solo maestro a un standby - Un solo maestro a varios standbys - Replicación de flujo/hot standby - Replicación bidireccional - Replicación lógica por flujo de logs	Replicación estándar maestro-standby: - Un solo maestro a un standby - Un solo maestro a varios standbys - Un solo maestro a un standby a uno o más standbys - Replicación circular (A hacia B hacia C y de vuelta hacia A) - Maestro a maestro
Vistas materializadas	Soportado	No soportado
Tablas temporales	Soportado	Soportado
Datos geoespaciales	Soportado	Soportado
Lenguajes de programación	Soportado	No soportado
Sistema de tipos extensible	Soportado	No soportado

Tabla. 2.1: Tabla Comparativa de Motores de Base de Datos.

5.6. Frameworks para Testing

- **PHPUnit**⁸: es un entorno para realizar pruebas en el lenguaje de programación PHP. Se creó bajo la idea de que cuanto antes se detecten los errores en el código antes podrán ser corregidos. Este conocido framework para PHP nos permite crear y ejecutar juegos de tests de manera sencilla.
- **Selenium**⁹: es un entorno de pruebas de software para aplicaciones basadas en la web. Provee una herramienta de grabar/reproducir para crear tests sin usar un lenguaje de scripting (Selenium IDE). Incluye también un lenguaje específico de dominio para pruebas (Selanese) que permite crear pruebas para aplicaciones desarrolladas en un amplio número de lenguajes de programación incluyendo Java, C#, Ruby, Groovy, Perl, PHP y Python.

El framework que será utilizado para realizar las pruebas será PHPUnit. Esto se debe a que viene incluido en Laravel, framework back-end seleccionado para el presente desarrollo.

5.7. Manejadores de Dependencias

- **Bower**¹⁰: es un complemento ideal para el desarrollo web. Es un sencillo programa que nos sirve para tener al día las dependencias de un proyecto para la web, en lo que respecta al desarrollo frontend, con Javascript o incluso CSS. Se trata de un programa basado en NodeJS que se ejecuta desde la consola y que tiene un sencillo API de comandos útiles para realizar tareas de mantenimiento y administración de paquetes necesarios para construir un proyecto web, concretamente la parte del lado del cliente. Con Bower es posible descargar y actualizar todo tipo de bibliotecas, frameworks, plugins, entre otros, sin tener que preocuparse por descargarlos e instalarlos.
- **Composer**¹¹: es una herramienta para la gestión de dependencias en PHP. Permite declarar las bibliotecas de las que depende el proyecto y las administrará (instalará/actualizará) por nosotros. Composer no es un gestor de paquetes en el mismo sentido que Yum o Apt. Sí, se trata de “paquetes” o bibliotecas, pero los administra en función de cada proyecto, instalándolos en un directorio (por ejemplo, vendor) dentro del proyecto. Por defecto no instala nada globalmente. Por lo tanto, es un gestor de dependencias.
- **npm**¹²: Node Package Manager o simplemente npm es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript. Utilizado para administrar las dependencias del framework front-end Vue.js.

⁸PHPUnit. <https://phpunit.de/>.

⁹Selenium. <https://www.seleniumhq.org/>.

¹⁰Bower. <https://bower.io/>.

¹¹Composer. <https://getcomposer.org/>.

¹²npm. <https://www.npmjs.com/>.

5.8. Bibliotecas

- **jQuery**¹³: es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Entre otras cosas, se encarga del recorrido y manipulación de documentos HTML, manejo de eventos, animación, y Ajax mucho más simple con una API fácil de usar que funciona en una multitud de navegadores.

5.9. Sistemas de Control de Versiones

- **Git**¹⁴: es un sistema de control de versiones distribuidas de código abierto y gratuito diseñado para manejar todo, desde proyectos pequeños a muy grandes, con velocidad y eficiencia. Es fácil de aprender y tiene una huella pequeña con un rendimiento increíblemente rápido. Supera a las herramientas de SCM como Subversion, CVS, Perforce y ClearCase con funciones como ramificación local barata, áreas de preparación conveniente y flujos de trabajo múltiples.
- **GitHub**¹⁵: es una plataforma de desarrollo inspirada en la forma en que trabajas. Desde el código abierto hasta el negocio, puede alojar y revisar códigos, administrar proyectos y crear software junto a millones de otros desarrolladores.

¹³jQuery. <https://jquery.com/>.

¹⁴Git. <https://git-scm.com/>.

¹⁵GitHub. <https://github.com/>.

Capítulo 3

Diseño del Sistema

1. Etapa 2 - Diseño del Sistema

1.1. Introducción

En el capítulo anterior se realizó un análisis que permitió definir las herramientas tecnológicas necesarias para lograr el objetivo de este proyecto. Por otro lado, se llevaron a cabo las tareas de elicitation de requerimientos, describiendo los procesos utilizados y los resultados obtenidos, siendo estos últimos los que definen qué es lo que el sistema debe hacer o dicho de otra manera, lo que el cliente espera del software.

Por su parte, el presente capítulo busca transformar esos requerimientos recabados en una solución que satisfaga las demandas del cliente.

1.2. Arquitectura del Sistema

Teniendo en cuenta el objetivo general de este trabajo, el de desarrollar un Sitio Web, podemos estimar el tipo de arquitectura que será necesaria utilizar para llevar adelante el mismo. Este tipo de sistemas requiere la implementación del *modelo cliente-servidor*. Este modelo se caracteriza por distribuir las tareas entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente, en este caso un navegador web, realiza peticiones a un servidor, en este caso un programa, buscando obtener una respuesta.

En esta arquitectura, se cuenta con la importante ventaja de tipo organizativa debida a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La interfaz de usuario, es un aspecto importante a tener en cuenta dentro del proceso de desarrollo de sistemas interactivos. Si bien es posible desarrollar una interfaz de usuario partiendo desde cero, la solución más comúnmente usada, por razones de eficiencia y uniformidad, es apoyarse en arquitecturas que incluyen gestores de ventanas y a veces también cajas de herramientas (*tool-box*). Esto implica que la interfaz estará íntimamente relacionada con el sistema de gestión

de ventanas, por lo que se hace aconsejable una separación entre la aplicación y su interfaz, para conseguir los siguientes objetivos:

- **Portabilidad:** permitir que una misma aplicación pueda ser utilizada en diferentes sistemas.
- **Reusabilidad:** la separación incrementa la reusabilidad de los componentes.
- **Múltiples interfaces:** flexibilidad para disponer de diferentes interfaces para la misma funcionalidad.
- **Personalización:** adecuar las características de la interfaz a las preferencias y necesidades del usuario sin tener que modificar la aplicación.

Siguiendo estos principios, se describen a continuación tres modelos de arquitectura cliente-servidor para la construcción de interfaces de usuario: el modelo *Seeheim*, el *Model–View–Controller* (Modelo-Vista-Controlador o MVC), y el PAC.

- **Seeheim:** el modelo plantea la comunicación entre el usuario y la aplicación, de manera estructurada, en tres niveles:
 1. **El nivel de presentación:** es la parte estática y visible de la interfaz que se comunica con el usuario y se construye sobre sistemas de ventanas y cajas de herramientas. Supondría el léxico de la interfaz.
 2. **El nivel de diálogo:** es la parte dinámica que maneja los eventos o mensajes que se producen como consecuencia de las acciones del usuario, sobre la interfaz. Establece la comunicación entre el nivel de presentación y el nivel de aplicación, es decir, establece la relación entre los eventos y su correspondiente respuesta por parte del nivel de aplicación. Se podría equiparar a la sintaxis de la comunicación entre los otros dos niveles.
 3. **Interfaz de aplicación:** es la parte de la aplicación que el usuario controla y que es visible para éste último. Equivaldría a la semántica de la aplicación.

La división en capas facilita el tratamiento de cada una por separado y además promueve la reutilización y la portabilidad, con lo que se posibilita el desarrollo rápido de prototipos, tarea fundamental para un diseñador.

- **MVC (Modelo Vista Controlador):** es un patrón de arquitectura de software que separa, los datos y la lógica de negocio de una aplicación, de la representación y la gestión de los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes: el modelo, la vista y el controlador. Es decir, por un lado define componentes para la representación de la información, y por otro la interacción del usuario.

El modelo es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la vista aquella parte de la información que en cada momento se le solicita sea mostrada. Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador. El controlador responde a eventos, usualmente acciones del usuario, e invoca

peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta el modelo (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos). Por tanto, se podría decir que el controlador hace de intermediario entre la vista y el modelo.

Por último, la vista presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar con el usuario, para lo cual requiere de dicho modelo la información que debe presentar como salida.

- **PAC (Presentación-Abstracción-Control)**: se apoya en tres pilares básicos:

- **Presentación**: gestiona las entradas/salidas.
- **Abstracción**: representa la semántica de la aplicación.
- **Control**: gestiona el diálogo y la correspondencia entre la aplicación y presentación.

Como puede apreciarse, PAC es una arquitectura bastante similar a MVC, aunque existen diferencias entre ellas. PAC agrupa la entrada/salida en la presentación, mientras que en MVC la vista se ocupa de la salida y el controlador de la entrada. Además, PAC introduce el control como mecanismo para mantener la consistencia entre la abstracción y la presentación, lo que en MVC no supone una tarea asignada de antemano. Por último, PAC no ha nacido vinculado a ningún entorno de programación, como le ocurre a MVC, aunque ambos lleven a una aproximación orientada a objetos.

Luego de analizar los patrones de diseño presentados, se llega a la conclusión de que tanto MVC como PAC proveen los medios necesarios para lograr una aplicación modular, escalable y fácil de mantener. En ambos casos se tiene a la interfaz de usuario separada de la lógica de negocio y se propone la organización de la información a través de modelos o abstracciones.

Con lo expuesto anteriormente se cuenta con las herramientas suficientes para definir el patrón de diseño a utilizar para el desarrollo del sistema, el mismo será MVC.

1.3. Descomposición del Sistema en Módulos

El objetivo detrás de la descomposición del sistema en módulos es el de generar subsistemas abstractos lo suficientemente pequeños que permitan independizarlos unos de otros de forma tal que el abordaje de la solución pueda ser abordado de manera mas sencilla.

En ingeniería de software, un *Diagrama de Clases* en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe al sistema mostrando sus clases, atributos, operaciones (o métodos), y las relaciones entre los objetos. En la Fig. (3.1) se puede apreciar el diagrama.

Los diagramas son el nexo entre los requerimientos del sistema y el sistema propiamente dicho. Es decir, que cuanto mas pueda describirse la estructura y lógica del sistema a través de estos modelos UML, mejor aún será el nivel de detalle alcanzado y será mas fácil abordar el problema.

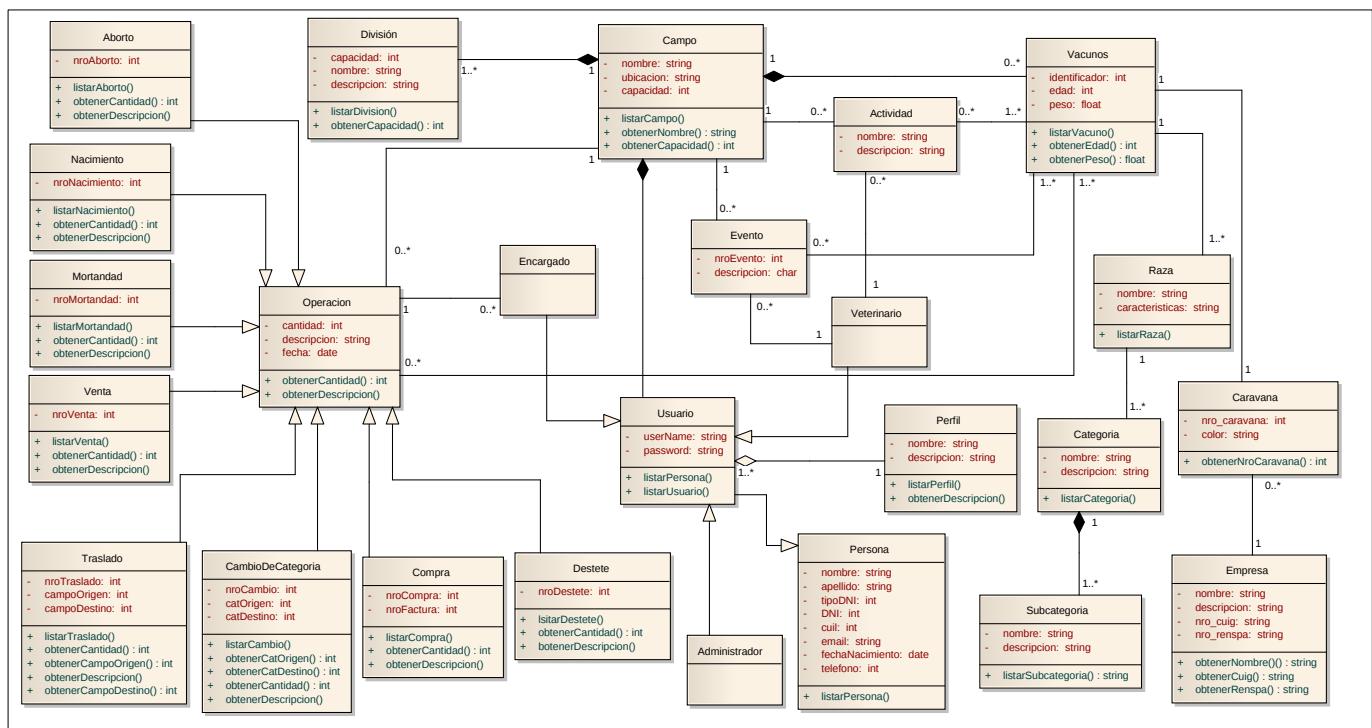


Figura. 3.1: Diseño del Sistema: Diagrama de Clases.

1.4. Diseño físico de la Base de Datos

Tomando referencia del diagrama de clases se diseña la base de datos. El diagrama físico de la base de datos puede apreciarse en la Fig. (3.2). En él, se encuentran los atributos y relaciones entre las distintas tablas. Los elementos utilizados para representar estas relaciones son los denominados Foreign Key (FK) que intentan esbozar la relación de existencia de una instancia de una clase, en otra. Por otro lado, dentro de una misma tabla, la forma de identificar e individualizar cada registro se hace bajo la utilización de lo que se denomina Primary Key (PK). Visualmente, los atributos PK poseen un asterisco (*) a su izquierda mientras que los atributos FK poseen en su nombre dicha propiedad.

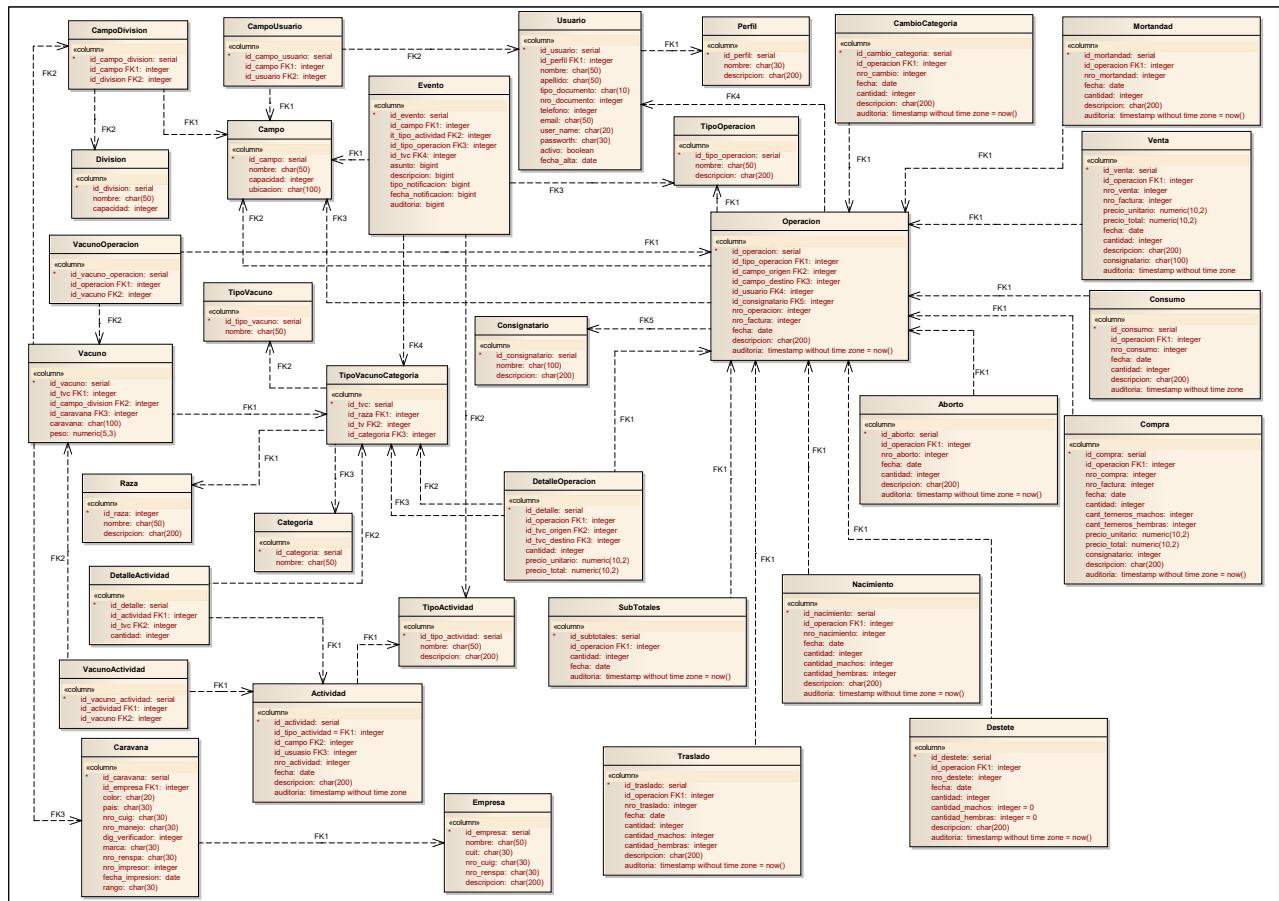


Figura. 3.2: Diseño del Sistema: Diseño Físico de la Base de Datos.

1.5. Diseño de la Interfaz Gráfica de Usuario

En la vida cotidiana interactuamos constantemente con Interfaces Gráficas de Usuario (GUI por su nombre en inglés Graphical User Interface) no sólo al usar una computadora sino también en objetos de uso diario como el celular, el cajero automático, etc. entre muchas otras cosas. La interfaz tiene un papel fundamental para que el producto sea o no competitivo. El producto no será exitoso si el usuario no consigue concretar una acción (por ejemplo: una transacción económica), o no entiende la secuencia de pasos a seguir, o cuando no encuentra con facilidad cómo concretar la acción que necesita (por ejemplo: realizar una compra) o cuando no considera atractivo el diseño de la aplicación que está utilizando.

La Interfaz de Usuario es la parte del software que las personas pueden ver, oír, tocar, hablar; es decir, donde se pueden entender. Contiene, esencialmente, dos componentes: la *entrada* y la *salida*. La entrada es cómo una persona le comunica sus necesidades o deseos al sistema. La salida es la forma en que la computadora transmite los resultados a lo solicitado por el usuario.

En la actualidad la GUI es parte fundamental de cualquier aplicación, y por lo tanto tiene tanta importancia como el desarrollo de la aplicación en sí.

Principios de diseño de interfaz gráfica de usuarios, según Sommerville:

- **Familiaridad del usuario:** significa que la interfaz debe utilizar términos e imágenes conocidos por el usuario; y los objetos que manipula el sistema deben estar relacionados con el ámbito de trabajo.
- **Uniformidad de la Interfaz:** significa que tanto comandos como menús deben tener el mismo formato. Las Interfaces uniformes reducen el tiempo de aprendizaje.
- **Mínima sorpresa:** el comportamiento del sistema no debe mostrar situaciones inesperadas. Ante éste tipo de situaciones el usuario puede mostrar irritabilidad, por lo tanto perder interés en utilizar la aplicación.
- **Recuperación de estados:** éste es uno de los principios más importantes al diseñar una Interfaz. Es inevitable cometer errores, por lo tanto el sistema le debe proporcionar al usuario la manera de subsanarlos o volver a estados anteriores. Éste principio involucra varias acciones como pedir al usuario que confirme acciones destructivas, que el usuario pueda deshacer, etc.
- **Guía de usuarios:** la Interfaz debe proporcionar al usuario asistencia, ayuda. No sólo cuando se cometen errores sino también cuando no se sabe qué hacer o cómo hacer alguna tarea. Esta ayuda debe estar integrada al sistema (algunas además ofrecen ayuda on line) y debe ser clara cuando el usuario la requiera, sin saturar con información.
- **Diversidad de usuarios:** se debe tener en cuenta los diferentes usuarios que pueden utilizar la aplicación. Aquellos casuales, que necesitan que los guíen, y aquellos que podrían usarla constantemente los cuales necesitarán trabajar con métodos abreviados, tan rápido como sea posible. Además se podría incluir recursos para mostrar diferentes tamaños de texto, reemplazar sonido por texto y al revés, modificar tamaño de botones, etc. Esto refleja la noción de Diseño Universal, principio de diseño cuyo objetivo es evitar excluir usuarios.

1.5.1. Aspectos fundamentales a la hora de Diseñar la Interfaz

A continuación se describen aquellos aspectos que han sido considerados a la hora de realizar el diseño de la Interfaz Gráfica de Usuario:

- **Diseño adaptable:** el diseño web adaptable (también conocido como diseño web adaptativo o responsivo), es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visitarlas.

La ventaja de utilizar un diseño adaptable radica en que con una sola versión en **HTML** y **CSS** se pueden cubrir todas las resoluciones de pantalla, con lo que el sitio web estará optimizado para distintos dispositivos y resoluciones.

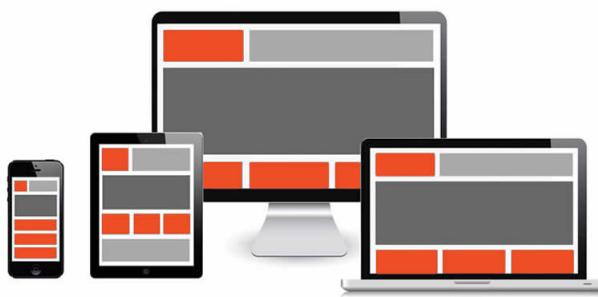


Figura. 3.3: Diseño del Sistema: Diseño Adaptable.

- **Compatibilidad:** el objetivo de la compatibilidad web es lograr que una aplicación se vea de la misma manera en todos los navegadores y que funcione correctamente en cada uno de ellos. Para conseguir la compatibilidad web se utilizaron tablas de compatibilidad (quirksmode.org, w3c.org) donde se pueden realizar consultas acerca de la compatibilidad de cada recurso utilizado con los distintos navegadores. Por otro lado, se validó el código de manera automática con la herramienta de validación de lenguaje de marcado del W3C (World Wide Web Consortium).
- **Usabilidad:** la usabilidad web es el grado de facilidad de uso que tiene una página web para los visitantes que entran e interactúan con ella. Un sitio web con buena usabilidad es aquel que permite a los usuarios una interacción sencilla, intuitiva, agradable y segura.

Para lograr buena usabilidad es necesario combinar con habilidad una serie de factores de diversos tipos: tecnológicos, de diseño, de contenidos, etc. Jakob Nielsen¹ (Enero, 1995) formuló diez principios sobre usabilidad de páginas web:

1. **Visibilidad del estado del sistema:** siempre se debe tener informado al usuario de lo que está pasando en nuestra web y ofrecerle una respuesta en el menor tiempo posible.
2. **Relación entre el sistema y el mundo real:** el sistema tiene que “habla” el lenguaje del usuario con palabras o frases que a éste le sean familiares y que pueda reconocer con facilidad. La información tiene que mostrarse con un orden lógico y las imágenes o iconos usados tienen que ser claros, sin darle la posibilidad al usuario de equivocarse.

¹ Jakob Nielsen. <https://www.nngroup.com/articles/ten-usability-heuristics/>.

3. **Control y libertad del usuario:** a veces, un usuario se equivoca, es normal. Tenemos que darle al usuario la posibilidad de subsanar el error y no sentirse frustrado por no poder realizar algo.
4. **Consistencia y estándares:** se deben tener en cuenta los convenios establecidos para ciertos iconos. Por ejemplo, las líneas horizontales que indican el menú desplegable. En la versión responsive es mejor implementar este ícono como menú desplegable y no utilizar otro, ya que el usuario puede llegar a no entender dicho ícono.
5. **Prevención de errores:** se debe prevenir cualquier error que pueda cometer el usuario. Y dado el caso de que este cometa uno, tenemos que poner a su alcance todas las opciones posibles para poder corregirlo. La opción de autocompletar es un buen ejemplo de este principio de usabilidad web.
6. **Reconocer antes que recordar:** siempre es mejor reconocer que obligar al usuario a memorizar acciones u objetos para que pueda cumplir su objetivo. Ayudar al usuario a no memorizar es importante a la hora de pensar en el diseño de un sitio web.
7. **Flexibilidad y eficiencia de uso:** el sitio web debe estar preparado para todo tipo de usuario, desde los más novatos hasta los más experimentados. Si conseguimos que cualquiera pueda navegar por nuestra web logramos flexibilidad. Y si tenemos opciones para los más experimentados obtenemos eficiencia.
8. **Diseño estético y minimalista:** las páginas web no deben contener información innecesaria, distrae al usuario y puede llegar a molestar en la navegación.
9. **Ayudar a los usuarios a reconocer, diagnosticar y corregir los errores:** todos los errores que puedan ocurrir en el sitio web deben estar expresados en un lenguaje entendible por todos, no por códigos.
10. **Ayuda y documentación:** con estos principios se intenta siempre que el usuario no tenga que usar documentos de ayuda para poder navegar o utilizar una aplicación. Aun así, siempre tenemos que dar al usuario la posibilidad de tener un pequeño manual de funcionamiento. Esta ayuda debe ser fácil de localizar, definir los pasos claramente y no ser muy extensa.

A continuación se muestran a modo de ejemplo unas ilustraciones de páginas que formarán parte del sitio web. Cabe aclarar que las mismas son meramente ilustrativas, ya que al no haber comenzado de manera formal con el desarrollo del sitio, no es posible representar de manera precisa el contenido de las mismas.

En la Fig. (3.4) se muestra la pantalla de *Inicio de Sesión* mientras que en la Fig. (3.5) se muestra el formulario mediante el cual se procede a registrar la *Compra de Vacunos*. Ambas están representadas en dispositivos que poseen diferentes características de resolución de pantalla esbozando, de esta manera, la adaptabilidad del sitio.



Figura. 3.4: Diseño del Sistema: Inicio de Sesión.



Figura. 3.5: Diseño del Sistema: Formulario de Compra.

Capítulo 4

Desarrollo

1. Introducción

Siguiendo el modelo en Cascada propuesto para este proyecto, la etapa 3 corresponde al *Desarrollo del Sistema*. En el presente informe se realiza una breve descripción del desarrollo de las funcionalidades del software, definidas previamente en las etapas de *Especificación de Requerimientos* y *Diseño del Sistema*. Al llevar adelante las actividades propias de la instancia bajo análisis se obtuvo como resultado una primer versión del Sistema de Gestión y Seguimiento de Trazabilidad de Ganado Vacuno.

Para lograr los objetivos de esta etapa ha sido necesario conseguir un grado de entendimiento y dominio de las tecnologías seleccionadas para el proyecto, a la vez que se iniciaron las tareas de programación del sistema.

Por otro lado, conforme fueron avanzando las tareas del proyecto, fue necesario modificar y reestructurar aspectos de diseño para adaptarlos de manera acorde a los requerimientos del software. Al finalizar, se presenta un cronograma con las actividades restantes para culminar con el proyecto.

2. Desarrollo del Sistema

En la etapa previa, *Diseño del Sistema*, se definieron pautas a respetar para lograr una interfaz gráfica de usuario adecuada. Basándose en dichas pautas se confeccionaron las vistas que permiten a los usuarios interactuar con las funcionalidades del software.

A continuación, siguiendo la división modular definida para el sistema, se presentan algunas capturas de pantallas del software junto con una descripción de las funcionalidades contenidas dentro de las mismas. Sin embargo, antes de comenzar con la descripción, es relevante mencionar que no todas las pantallas del sistema son ilustradas debido a que muchas de ellas presentan aspectos similares tanto a nivel estético como funcional.

2.1. Página de Bienvenida e Inicio de Sesión

La página de bienvenida es una pantalla que contiene, como única acción, la de *Iniciar Sesión*, como puede verse en la *Fig. (4.1)*.



Figura. 4.1: Página de Bienvenida.

Cabe destacar que el software cuenta con un módulo completo de autenticación de usuario, tal como puede encontrarse en cualquier otra aplicación. Este módulo contiene las funcionalidades de *Iniciar Sesión*, *Recordar Datos*, *Recuperación de Contraseña* y *Verificación de Correo Electrónico*. En la Fig. (4.2) puede observarse el formulario de inicio de sesión, el cual presenta acceso a las características mencionadas.

The screenshot shows a web browser window for the SGSTGV application. The title bar at the top left says "SGSTGV" and the top right has a link "Iniciar sesión". The main content area is titled "Iniciar sesión". It contains two input fields: "Correo electrónico" with the value "mariorosales941@gmail.com" and "Contraseña" with a masked value. Below these is a checkbox labeled "Recordar datos". At the bottom are two buttons: "Ingresar" (in blue) and "Olvidó su contraseña?" (in blue). A small red square icon is visible in the bottom-left corner of the browser window.

Figura. 4.2: Página de Inicio de Sesión.

2.2. Página Principal

La página principal es el punto de acceso primario a todas las pantallas del sistema. Es decir, en ella se encuentra el menú principal, desde el cual cada usuario puede acceder a las funcionalidades del software.

La aplicación cuenta con un conjunto de perfiles que identifican a cada uno de los usuarios y limitan el acceso a las diferentes funcionalidades implementadas. Los perfiles definidos son los siguientes:

- **Administrador:** usuario que posee acceso a todas las funcionalidades del sistema. Encargado de realizar la puesta a punto del sistema, dar de alta usuarios, entre otras.
- **Encargado en Jefe:** usuario con permisos para ver, registrar, modificar y eliminar operaciones. Responsable de controlar y coordinar las actividades de los encargados.
- **Encargado:** usuario con permisos para ver, registrar, modificar y eliminar operaciones correspondientes al campo que haya sido asignado. Responsable de controlar y coordinar las actividades de los peones.
- **Veterinario en Jefe:** usuario con permisos para ver, registrar, modificar y eliminar operaciones veterinarias. Responsable de controlar y coordinar las actividades de los veterinarios.
- **Veterinario:** usuario con permisos para ver, registrar, modificar y eliminar operaciones veterinarias correspondientes al campo que haya sido asignado.
- **Peón:** usuario con permisos para ver operaciones correspondientes al campo al que haya sido asignado. No podrán registrar, modificar o eliminar operaciones.

La Fig. (4.3) muestra todas las opciones disponibles para el usuario administrador, de esta forma, el menú contendrá diferentes opciones dependiendo del perfil del usuario autenticado.

Por otro lado, se brinda en forma reducida un acceso directo a los *Últimos Movimientos* registrados en el sistema. Los mismos, pueden ser *Compras*, *Cambios de Categoría*, *Traslados*, *Mortandad*, *Actividad Veterinaria* o bien *Ventas*. Por ejemplo, en la Fig. (4.4), puede observarse información breve sobre las compras registradas. Sobre cada una de ellas se muestra su Nro. de Factura, el Campo al que pertenece, el Proveedor, la Fecha y qué Usuario ha registrado la misma. Finalmente, a través del botón *Ver*, el usuario puede acceder al detalle de una compra en particular.

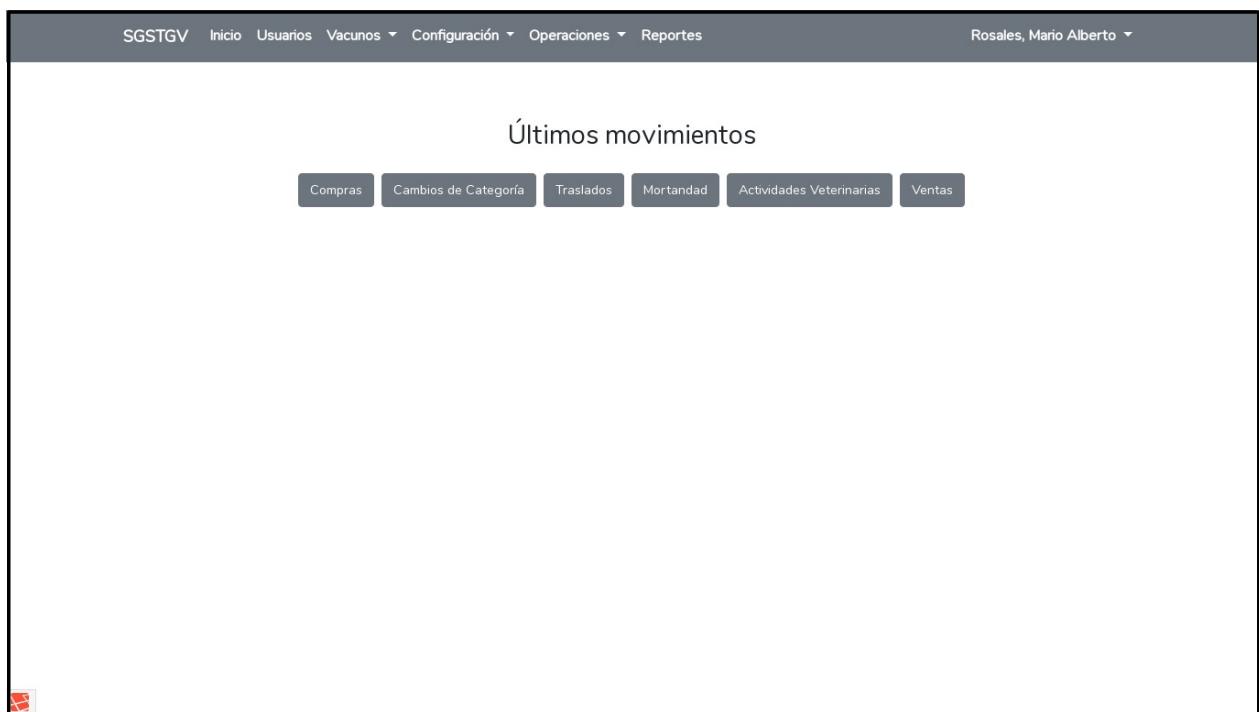


Figura. 4.3: Página Principal.

Nro. Factura	Campo	Proveedor	Fecha	Ingresado por	Acciones
4	Las Tunas	Colombo y Magliano	05/11/2019	Rosales, Mario Alberto	<button>Ver</button>
3	Las Estefania	Haciendas Correntinas	05/11/2019	Rosales, Mario Alberto	<button>Ver</button>
2	La Elsa	Haciendas Correntinas	05/05/2019	Rosales, Mario Alberto	<button>Ver</button>
1	Las Tunas	Colombo y Magliano	05/01/2019	Rosales, Mario Alberto	<button>Ver</button>

Figura. 4.4: Últimos movimientos: Compras.

2.3. Funcionalidades Administrativas y de Configuración

Las funcionalidades administrativas y de configuración, permiten adaptar el software a las necesidades del cliente. Este tipo de configuraciones sólo pueden ser llevadas a cabo por el usuario administrador. No es posible modificar las reglas de negocio preestablecidas para cada caso, en su lugar, pueden configurarse aspectos relacionados a usuarios, campos, divisiones, razas, categorías, sub categorías, operaciones, clientes y proveedores. A cada una de estas características se las puede modificar, eliminar o bien crear nuevas.

Es posible tomar contacto con las características mencionadas anteriormente a través del menú que posee la aplicación. Menú que puede apreciarse en la página principal del sistema (ver *Fig. (4.3)*), y que contiene las siguientes opciones:

- **SGSTGV:** Sistema de Gestión y Seguimiento de Trazabilidad de Ganado Vacuno, abreviado como SGSTGV, es una opción presente en todas las pantallas del sistema sin importar el perfil del usuario autenticado. Esta opción dirige al usuario a la Página de Bienvenida (ver *Fig. (4.1)*).
- **Inicio:** acceso a la Página Principal. También presente en todas las pantallas del sistema, sin importar el perfil del usuario autenticado.
- **Usuarios:** acceso las funcionalidades correspondientes a los usuarios.
- **Vacunos:** acceso a las funcionalidades asociadas a Caravanas y Vacunos.

- **Configuración:** acceso a las características del sistema que el usuario administrador debe configurar para ajustar el sistema a sus necesidades. Dentro de dicho sub menú se encuentran las siguientes opciones:

- Campos
- Divisiones
- Campos / Divisiones
- Razas
- Categorías
- Sub Categorías
- Categorías / Sub Categorías
- Operaciones
- Clientes
- Proveedores
- Actividades Veterinarias

- **Operaciones:** acceso a las distintas operaciones que pueden ser realizadas, entre ellas:

- Compras
- Cambios de Categorías
- Traslados
- Mortandad
- Actividades Veterinarias
- Ventas

- **Reportes:** acceso a resúmenes que el sistema permite emitir.

A continuación se describen con mayor nivel de detalles las configuraciones disponibles.

2.3.1. Administración de Usuarios

La página de usuarios (ver *Fig. (4.5)*), brinda información acerca de los usuarios registrados en el sistema. Desde allí, es posible observar una lista con los datos mas relevantes de cada persona. El administrador puede ver el *Detalle*, *Editar*, *Eliminar* o bien crear un nuevo usuario. Además, se ha implementado la paginación que permite ver a los usuarios de forma simple y clara, como así también es posible realizar filtros a través del campo de búsqueda destinado para tal fin.

Tanto para la creación de un nuevo usuario como para la edición de uno ya existente, es necesario interactuar con un formulario que contiene la información solicitada por el sistema (ver *Fig. (4.6)*).

Usuarios					Acciones		
ID	Apellido, Nombre	Email	Perfil		Detalle	Editar	Eliminar
1	Rosales, Mario Alberto	mariorosales941@gmail.com	Administrador		Detalle	Editar	Eliminar
2	Navarro, Jerónimo	jerojero@gmail.com	Encargado en Jefe		Detalle	Editar	Eliminar
3	Rosales, Cuqui 1	cuquirosales_1@gmail.com	Encargado		Detalle	Editar	Eliminar
4	Rosales, Cuqui 2	cuquirosales_2@gmail.com	Encargado		Detalle	Editar	Eliminar
5	Rosales, Cuqui 3	cuquirosales_3@gmail.com	Encargado		Detalle	Editar	Eliminar

1 2 3 Siguiente

Figura. 4.5: Página principal de usuarios.

Nombres:

Apellidos:

Contraseña:

Documento:

Teléfono:

Correo electrónico:

Perfil:

Campo (*)

(*) Para los roles globales (para cualquier campo) no es necesario elegir un campo en el cual el usuario pueda realizar actividades. Para los perfiles de Encargado y Veterinario, si es necesario que se designe un Campo en el cual el usuario pueda ejercer.

[Cancelar](#) [Crear usuario](#)

Figura. 4.6: Formulario de registro de usuarios.

Para aquellos usuarios que posean un perfil de administrador, encargado en jefe o veterinario en jefe, no será necesario seleccionar un campo. Esto se debe a que un usuario con un perfil similar a los mencionados podrá ejercer sobre cualquiera de los campos registrados en el sistema.

2.3.2. Administración de Campos y Divisiones

La empresa o entidad ganadera, puede contar con mas de un campo en los cuales estén distribuidos sus vacunos. Cada campo, a su vez, puede estar dividido en sectores que permitan aprovechar al máximo los recursos. Las divisiones, en el contexto del presente proyecto, hacen referencia a las secciones en las que un campo es particionado y en las cuales los vacunos son distribuidos. Por tanto, el sistema brinda la posibilidad de crear, actualizar y eliminar campos y divisiones.

Por ejemplo, por razones de organización y control de animales, estos suelen ser organizados y agrupados según razas, categorías y sub categorías en, por ejemplo, *corrales* o *potreros*, según el espacio físico de cada división, o bien, en caso de animales de avanzada edad, en *campo abierto*, donde son liberados para su engorde final y posterior venta.

Luego de haberse creado los campos y divisiones, es necesario indicar cuales divisiones estarán comprendidas en un campo en particular. Por tanto, queda registrada una relación entre un campo y una división que posteriormente podrá ser utilizada en las funcionalidades operativas. De esta manera es posible elegir el lugar en donde serán alojados los vacunos recientemente adquiridos a través de una compra.

En la página de campos (ver *Fig. (4.7)*), puede apreciarse una tabla con los campos registrados en el sistema. Desde allí, el usuario puede acceder a las acciones mencionadas mas arriba. El registro de un nuevo campo puede hacerse a través del formulario correspondiente (ver *Fig. (4.8)*).

#	Nombre	Capacidad	Dirección	Acciones		
1	Las Tunas	1000	Buenos Aires, Capital	Detalle	Editar	Eliminar
2	La Elsa	1500	Chaco, Capital	Detalle	Editar	Eliminar
3	Las Estefania	2000	La Pampa, Capital	Detalle	Editar	Eliminar

1 2 Siguiente

Figura. 4.7: Página principal de campos.

Nombre:

Capacidad:

Dirección:

[Cancelar](#) [Crear campo](#)

Figura. 4.8: Formulario de registro de campos.

Por otro lado, la página de divisiones (ver *Fig. 4.9*), contiene las mismas características que la página de campos pero con la información correspondiente a las mismas. El registro de una nueva división puede hacerse a través del formulario correspondiente (ver *Fig. 4.10*).

En ambos casos, tanto para campos como para divisiones, los formularios de actualización son similares a los de registro. Estos no son ilustrados en el presente informe por ser similares a los ya expuestos.

#	Nombre	Descripción	Capacidad	Acciones		
1	Corral	Animales pequeños	500	<button>Detalle</button>	<button>Editar</button>	<button>Eliminar</button>
2	Potrero	Animales medianos	1000	<button>Detalle</button>	<button>Editar</button>	<button>Eliminar</button>
3	Campo abierto	Animales grandes	1500	<button>Detalle</button>	<button>Editar</button>	<button>Eliminar</button>

Figura. 4.9: Página principal de divisiones.

The screenshot shows a web-based application interface for creating a new division. At the top, there is a navigation bar with the logo 'SGSTGV' and links for 'Inicio', 'Usuarios', 'Vacunos', 'Configuración', 'Operaciones', and 'Reportes'. On the right side of the header, it displays the user's name 'Rosales, Mario Alberto'. Below the header, the main content area has a title 'Crear nueva división'. There are two input fields: 'Nombre:' with placeholder 'Nombre...' and 'Descripción:' with placeholder 'Descripción...'. Below these is a single input field labeled 'Capacidad:' with placeholder 'Capacidad...'. At the bottom right of the form are two buttons: a red 'Cancelar' (Cancel) button and a blue 'Crear división' (Create division) button.

Figura. 4.10: Formulario de registro de divisiones.

Por último, resta generar la relación entre campos y divisiones. Esto puede hacerse desde el botón *Campos y Divisiones* presente tanto en el formulario de registro de campos como en el de divisiones. Esto dirige al usuario al formulario de registro de una nueva relación (ver Fig. 4.12). Cada relación creada puede observarse en la página de relaciones entre campos y divisiones (ver Fig. 4.11).

The screenshot shows the SGSTGV application interface. At the top, there is a navigation bar with links: Inicio, Usuarios, Vacunos, Configuración, Operaciones, Reportes, and a user profile for Rosales, Mario Alberto. Below the navigation bar, the title "Campos y Divisiones" is displayed, followed by a "Nueva relación" button. A list of locations is shown in a hierarchical tree structure:

- Las Tunas
 - Corral
 - Potrero
 - Campo abierto
- La Elsa
 - Corral
 - Campo abierto
- Las Estefania
 - Potrero
 - Campo abierto
- Cuay Chico
- Estancia El Cuqui

Figura. 4.11: Página principal de relaciones entre campos y divisiones.

The screenshot shows the SGSTGV application interface for creating a new relationship. At the top, there is a navigation bar with links: Inicio, Usuarios, Vacunos, Configuración, Operaciones, Reportes, and a user profile for Rosales, Mario Alberto. Below the navigation bar, the title "Crear nueva relación entre Campo y División" is displayed. The form has two dropdown menus: "Seleccione un Campo:" containing "Elija un campo..." and "Seleccione una División:" containing "Elija una división...". At the bottom right of the form are two buttons: "Cancelar" (Cancel) and "Crear relación" (Create relation).

Figura. 4.12: Formulario de registro de relaciones entre campos y divisiones.

2.3.3. Administración de Caravanas

Las caravanas son una parte fundamental del presente proyecto ya que sin ellas no se podría identificar e individualizar a los vacunos.

El *Servicio Nacional de Sanidad y Calidad Agroalimentaria* (Senasa), es un organismo descentralizado dependiente del *Ministerio de Producción y Trabajo de la Nación y de la Secretaría de Agroindustria*. Es el encargado de ejecutar las políticas nacionales en materia de sanidad y calidad animal y vegetal e inocuidad de los alimentos de su competencia, así como de verificar el cumplimiento de la normativa vigente en la materia. En síntesis, el Senasa es responsable de planificar, organizar y ejecutar programas y planes específicos que reglamentan la producción, orientándola hacia la obtención de alimentos inocuos para el consumo humano y animal.

Una de las responsabilidades que posee el Senasa es la de proporcionar caravanas que son solicitadas por los productores o empresas dedicados a la cría y producción de ganado. Por tanto, una vez solicitadas las caravanas, las mismas deben ser registradas en el sistema a través del formulario implementado para tal fin, el cual se ilustra en la Fig. (4.14). El resultado de dicha operación puede observarse en la página principal de caravanas (ver Fig. (4.13)).

Cabe destacar que las Funcionalidades Operativas, que serán descriptas en el Apartado (2.4), dependen en gran medida de las caravanas ya que cada una de ellas las utilizan para representar a aquellos vacunos que formen parte de la operación que se este llevando a cabo.

#	Lote	Nro. Caravana	Asignada	Activa	Acciones
1	1	JC423-A200-8	Si	Si	<button>Detalle</button>
2	1	JC423-A201-8	Si	Si	<button>Detalle</button>
3	1	JC423-A202-8	Si	Si	<button>Detalle</button>
4	1	JC423-A203-8	Si	Si	<button>Detalle</button>
5	1	JC423-A204-8	No	Si	<button>Detalle</button>
6	1	JC423-A205-8	No	Si	<button>Detalle</button>

Figura. 4.13: Página principal de Caravanas.

The screenshot shows the SGSTGV application's 'Crear nuevas caravanas' (Create new caravans) page. At the top, there's a navigation bar with links like Inicio, Usuarios, Vacunos, Configuración, Operaciones, and Reportes, and a user profile for Rosales, Mario Alberto. Below the header, the main title 'Crear nuevas caravanas' is displayed. The form is divided into two main sections: 'Datos de la Empresa' (Company Data) and 'Datos para las Caravanas' (Caravan Data). In the 'Company Data' section, fields include Empresa (LA VACA PEREZ S), CUIT (30-21414377-0), CUIG (JC432), RENSPA (01.008.0.01458), País (Argentina), and Identificador País (AR). Below this, the 'Caravan Data' section has fields for Lote (2), Color (Elija...), Marca (empty), Fecha de Impresión (empty), Nro. de Impresor (empty), and Letra (empty). Below these are dropdowns for 'Rango desde' and 'Rango hasta'. At the bottom right of the form are 'Cancelar' (Cancel) and 'Crear caravanas' (Create caravans) buttons.

Figura. 4.14: Formulario de registro de nuevas caravanas.

2.3.4. Administración de Razas, Categorías y Sub Categorías

En biología, raza se refiere a los grupos en que se subdividen algunas especies basándose en sus rasgos fenotípicos, a partir de una serie de características que se transmiten por herencia genética. El concepto de raza tiene gran importancia en Ganadería, pues la raza es importante para la obtención del óptimo rendimiento de la crianza.

Por tanto, en el contexto del presente trabajo, las razas permiten agrupar a los individuos que comparten características biológicas y, debido a que es posible trabajar con más de una raza al mismo tiempo, el sistema debe contemplar esta posibilidad.

Los vacunos pertenecientes a una raza, a su vez, poseen diferencias que ameritan clasificarlos de forma tal que sea fácil su gestión y seguimiento. Esta clasificación se lleva adelante utilizando los conceptos de categoría y sub categoría, los cuales permiten clasificar a los vacunos de una misma raza principalmente por su género y edad.

De esta manera, las razas, categorías y sub categorías son características del sistema que pueden ser configuradas de forma independiente. Sin embargo, existe una relación tanto lógica como biológica que llevan a la necesidad de relacionarlas unas con otras. El administrador, puede cargar en el sistema las razas, categorías y sub categorías que sean necesarias. Por último, deben registrarse las relaciones entre ellas indicando aquellas sub categorías que pertenezcan a una categoría y aquellas categorías que pertenezcan a una raza. De esta forma, es de esperarse que un vacuno pertenezca a una única raza y, conforme crece el individuo, puede cambiar de sub categoría estando en una categoría dada. Es posible, también, que el mismo individuo pueda cambiar de categoría, alcanzando así las sub categorías de esta última.

Tanto las razas, las categorías como las sub categorías tienen un nombre y una descripción que permiten identificarlas y distinguirlas. Entonces, las funcionalidades implementadas son similares para cualquiera de ellas como así también las pantallas que representan las mismas. A continuación se ilustra la página de categorías (ver *Fig. (4.15)*), la página de registro de nuevas categorías (ver *Fig. (4.16)*), la página que muestra las relaciones entre categorías y sub categorías (ver *Fig. (4.17)*) y la página de registro de nuevas relaciones entre categorías y sub categorías (ver *Fig. (4.18)*).

The screenshot shows the 'SGSTGV' application interface. The top navigation bar includes links for 'Inicio', 'Usuarios', 'Vacunos', 'Configuración', 'Operaciones', and 'Reportes'. On the right, it shows the user 'Rosales, Mario Alberto'. The main content area is titled 'Categorías' and contains a table with two rows of data:

#	Nombre	Descripción	Acciones		
1	Toros	Vacunos machos mayores a 2 años	Detalle	Editar	Eliminar
2	Vacas	Vacunos hembras mayores a 2 años	Detalle	Editar	Eliminar

Below the table are navigation buttons: '1' (highlighted in blue), '2', '3', and 'Siguiente'.

Figura. 4.15: Página principal de categorías.

The screenshot shows a web-based application interface for managing cattle categories. At the top, there is a navigation bar with the logo 'SGSTGV' and links for 'Inicio', 'Usuarios', 'Vacunos', 'Configuración', 'Operaciones', and 'Reportes'. On the right side of the header, it displays the user's name 'Rosales, Mario Alberto'. The main content area has a title 'Crear nueva categoría' (Create new category). Below the title are two input fields: 'Nombre:' (Name) and 'Descripción:' (Description). The 'Nombre:' field contains a placeholder 'Nombre...'. The 'Descripción:' field is a larger text area with a placeholder 'Descripción...'. At the bottom right of the form are two buttons: a red 'Cancelar' (Cancel) button and a blue 'Crear categoría' (Create category) button.

Figura. 4.16: Formulario de registro de nuevas categorías.

The screenshot shows a list of categories and their sub-categories. The categories listed are: Toros, Vacas, Novillos, Vaquillonas, Terneros, and Terneras. Each category has associated sub-categories listed below it. For example, 'Toros' has sub-categories like Recría, Servicio, and Invernada. The page has a dark header with the 'SGSTGV' logo and navigation links. On the right side, there is a button labeled 'Nueva Relación' (New Relationship).

- Toros
 - Recría
 - Servicio
 - Invernada
- Vacas
 - Vacias
 - Prefañadas Otoño VN
 - Prefañadas Primavera VN
 - Con Cría al Pie VN
 - Prefañadas Otoño VV
 - Prefañadas Primavera VV
 - Con Cría al Pie VV
 - Invernada
- Novillos
 - 1-2
 - 2+
- Vaquillonas
 - 1
 - 2
 - 1er Servicio
 - 2do Servicio
- Terneros
 - Ternero al Pie Macho
 - Ternero al Pie Hembra
 - Ternero Destete Macho
 - Ternero Destete Hembra

Figura. 4.17: Página principal de relaciones entre categorías y sub categorías.

The screenshot shows a software interface titled "Crear nueva relación entre Categoría y Sub Categoría". At the top, there is a navigation bar with links: SGSTGV, Inicio, Usuarios, Vacunos, Configuración, Operaciones, and Reportes. On the right side of the header, it says "Rosales, Mario Alberto". Below the title, there are two dropdown menus: "Categoría:" and "Sub Categoría:", both currently showing "Elija...". At the bottom right of the form area are two buttons: "Cancelar" (Cancel) in red and "Crear combinación" (Create combination) in blue.

Figura. 4.18: Formulario de registro de relaciones entre categorías y sub categorías.

2.3.5. Administración de Tipos de Actividades Veterinarias

Las actividades veterinarias son aquellas en las que se da atención a los vacunos buscando el bienestar de los mismos. Hoy en día, se encuentran vigentes ciertas prácticas y estándares definidos por el Senasa y el sistema desarrollado en el presente proyecto busca contemplar las mismas. Para ello, es posible configurar las actividades veterinarias que deban llevarse a cabo para poder cumplir con los estándares.

Dentro de las actividades veterinarias que suelen practicarse se encuentran la monta o servicio, la inseminación artificial y el tacto. La eficiencia en la reproducción es uno de los aspectos más críticos de un rodeo. Las pérdidas económicas que se producen como consecuencia de una reproducción retrasada poseen múltiples facetas:

- La vida de la vaca produciendo leche se reduce debido a que el pico de producción de leche no se produce con tanta frecuencia y los períodos de seca se extienden.
- El número de terneros nacidos por año decrece, dando menos oportunidades para descartar vacas con baja producción de leche, disminuyendo la posible ganancia genética en el valor genético del rodeo.
- El costo directo para el tratamiento de los desordenes reproductivos, servicio y honorarios veterinarios se incrementa.

Ya sea que el productor utilice inseminación artificial o servicio natural, la detección de celo es un componente crítico de un buen manejo reproductivo en la explotación lechera. Cualquiera que sea el caso, el registro de las vacas en celo o fechas de servicio es necesario para predecir celos futuros o fechas de parto y para manejar a las vacas de una manera apropiada.

Por lo expuesto anteriormente es que resulta crucial incluir actividades veterinarias reproductivas al sistema.

2.3.6. Administración de Clientes y Proveedores

Entre las formas mas comunes en las que el número de vacunos puede aumentar o disminuir son la compra y la venta, respectivamente. Las razones por la cual se llega a una compra o venta de bovinos puede variar, pero en la mayoría de los casos se debe a la posibilidad de adquirir nuevos individuos para criarlos, alimentarlos y aumentar sus aptitudes o características para luego venderlos a un precio mayor. Si bien puede que este no sea el principal objetivo, es uno de los mas importantes perseguido por aquellos dedicados a la cría de ganado vacuno.

Es importante, entonces, brindar la posibilidad de administrar tanto clientes como proveedores para que, al momento de realizar una operación de compra o venta de vacunos, sea posible registrar compradores y vendedores. Para ello, al igual que con las demás características de puesta punto y configuración, se cuenta con apartados diseñados específicamente para tal fin.

2.4. Funcionalidades Operativas

Las funcionalidades operativas reflejan los movimientos ocurridos en el día a día de la entidad ganadera. Por citar un ejemplo, si se han comprado animales, a través del módulo de compra se puede registrar dicha operación haciendo uso de las configuraciones expuestas anteriormente (ver *Apartado (2.3)*).

Los módulos que representan a las funcionalidades operativas cuentan con una estructura común. Esta estructura consta de cuatro páginas. La primera de ellas, y la mas importante, es la que contiene un historial en forma de lista de todos los registros asociados a la operación. Dicha pantalla, cuenta con un barra de búsqueda que permite acceder a la información de manera rápida y sencilla. Por otro lado, también es posible filtrar la información según la fecha de creación o por el usuario que haya ingresado la operación.

Las siguientes dos páginas, la página de creación y de actualización, son similares para cada uno de los módulos. La diferencia entre un módulo u otro radica en la información necesaria para crear o actualizar los registros del mismo.

Por último, la cuarta pantalla esta destinada a los detalles de una operación. Por cuestiones de espacio y legibilidad, la pantalla principal para cada operación contendrá información reducida con la intención de presentar al usuario aquellos campos que permiten distinguir una operación, de otra. Es así que, al ingresar a la pantalla de detalles, pueden encontrarse la totalidad de los campos que forman parte de la operación.

A continuación, serán descritos los módulos operacionales del software de Gestión y Seguimiento de Trazabilidad de Ganado Vacuno, pudiendo resaltar sus diferencias y objetivos a cumplir.

2.4.1. Modulo Compras

El módulo de compras es aquel que permite al usuario registrar en el sistema el ingreso de nuevos vacunos. Para lograr almacenar una compra es necesario contar con ciertos datos, estos datos son proporcionados por el usuario y siguen una cierta secuencia. A continuación se describe el proceso de creación de una compra.

En primer lugar el usuario debe seleccionar y completar los siguientes datos (ver Fig. (4.19)):

- **Campo:** en donde los vacunos serán alojados.
- **Número de Factura:** cada compra posee un número de factura.
- **Fecha:** fecha del momento en el cual se haya realizado la compra.
- **Proveedor:** proveedor que haya intervenido en la compra.

Crear nueva Compra

Datos de la compra

Campo:	Nro. de Factura:	Fecha:	Proveedor:
Elija...	Número de Factura...	dd/mm/aaaa	Elija...

Figura. 4.19: Datos de la compra.

Toda compra esta conformada por uno o mas artículos y, la compra de vacunos, no es la excepción. Por tanto, lo siguiente es la carga de vacunos a la compra. Para poder agregar ítems a una compra, se deben seleccionar los siguientes datos (ver Fig. (4.20)):

- **División:** sección del campo en donde estarán situados los vacunos.
- **Raza:** raza a la que pertenecen los vacunos.
- **Categoría:** categoría a la que pertenecen los vacunos.
- **Sub categoría:** sub categoría a la que pertenecen los vacunos.
- **Cantidad:** total de vacunos.
- **Peso promedio:** utilizado para tener información estimativa del peso de cada individuo.
- **Precio unitario:** precio, por kilo, de un individuo.

Agregar vacunos a la compra

División:	Raza:	Categorías:
Elija...	Elija...	Toros
Sub categorías:	Cantidad:	Peso promedio (Kg):
Elija...		0,00
Precio x Kg (\$):	0,00	
<input type="button" value="Agregar vacunos"/>		
Items de la compra		
Aún no se han agregado vacunos a la compra		

Figura. 4.20: Agregar ítem a la compra.

Según lo seleccionado en la *Fig. (4.20)*, se completa automáticamente una lista de ítems, esta puede observarse en la *Fig. (4.21)*. Cada fila de la tabla representa un ítem y cada ítem puede ser modificado o eliminado, utilizando las acciones que se encuentran en la columna de acciones, según corresponda.

Items de la compra

División	Raza	Categoría	Sub Categoría	Cantidad	Peso	Precio	Acciones
Corral	Aberdeen Angus	Vacas	Vacias	2	125	65.8	
Potrero	Hereford	Toros	Servicio	4	450	90.35	

Figura. 4.21: Ítems a la compra.

A continuación, deben seleccionarse las caravanas que serán utilizadas para identificar a los nuevos vacunos. Cabe destacar que las caravanas que pueden seleccionarse deben estar previamente cargadas en el sistema, a través de las funcionalidades descritas en el *Apartado (2.3)*. En la *Fig. (4.22)* se observa una lista con los números de las caravanas disponibles.

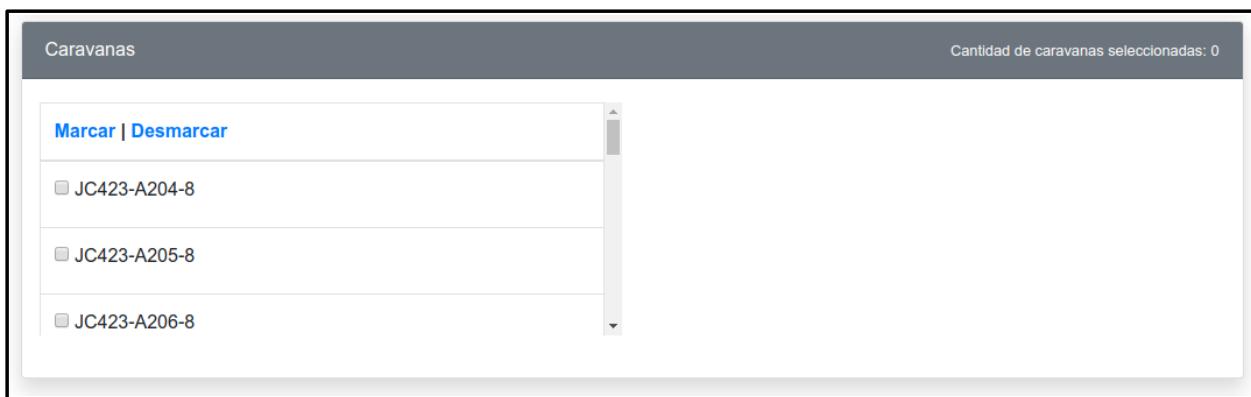


Figura. 4.22: Caravanas.

El último campo es el de las observaciones, donde el usuario puede agregar un comentario referente a la operación (ver *Fig. (4.23)*). Finalmente, el usuario puede cancelar o guardar la compra, para lo cual utiliza los botones implementados para tal fin.

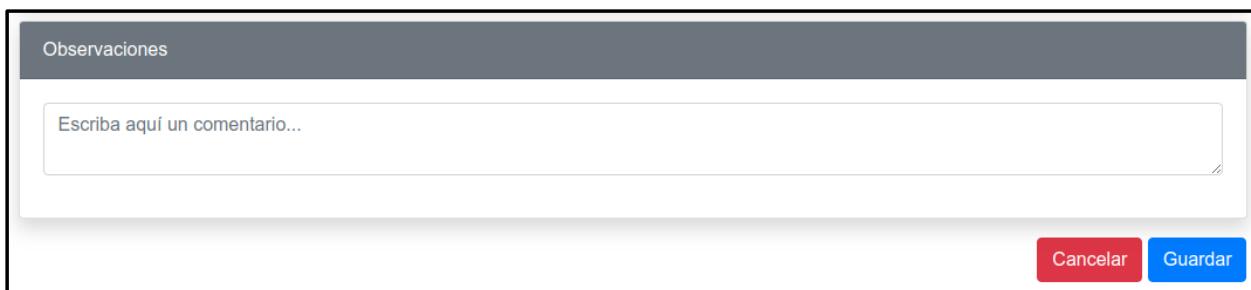


Figura. 4.23: Observaciones.

2.4.2. Módulo Cambios de Categorías

El módulo de cambios de categoría es el que permite al usuario representar, a nivel sistema, la maduración de un vacuno. Es decir, al igual que ocurre con los humanos, los bovinos tienen su propio ciclo de vida y la forma de indicar que un individuo pasa de una etapa del ciclo a otra, es mediante un cambio de categoría. Al momento de registrar el cambio de categoría, es necesario que el usuario proporcione la siguiente información:

- **Campo:** lugar en donde se encuentran los vacunos.
- **División origen:** sección del campo en donde están situados los vacunos.
- **División destino:** sección del campo en donde estarán situados los vacunos al finalizar la operación, pudiendo ser la misma que la de origen.
- **Raza:** raza a la que pertenecen los vacunos.
- **Categoría origen:** categoría a la que pertenecen los vacunos.
- **Sub categoría origen:** sub categoría a la que pertenecen los vacunos.

- **Categoría destino:** categoría a la que pertenecerán los vacunos, pudiendo ser la misma que la de origen.
- **Sub categoría destino:** sub categoría a la que pertenecerán los vacunos, pudiendo ser la misma que la de origen.
- **Fecha:** momento en el cual se haya efectuado el cambio de categoría.
- **Descripción:** el usuario puede agregar un comentario referente a la operación.

Luego de haber ingresado los datos recientemente mencionados, serán listadas las caravanas de aquellos vacunos que cumplen con la configuración ingresada por el usuario. De esta lista de caravanas, el usuario debe seleccionar las correspondientes a los vacunos que forman parte de la operación en cuestión.

Los cambios de categoría tienen la particularidad de que, según la categoría y sub categoría de origen y destino seleccionada, pueden ser catalogados de diferente manera. Un cambio de categoría puede ser:

- **Simple:** cambio de categoría que no implica agregar o modificar vacunos adicionales. Sólo los vacunos seleccionados serán actualizados.
- **Aborto:** vaca preñada que pierde la cría.
- **Nacimiento:** vaca que da a luz un ternero. En este caso es necesario registrar el cambio de categoría para la vaca, pasando de *Vaca Preñada* a *Vaca con Cría al Pie*, y registrar nuevos vacunos, *Terneros al Pie*, en el sistema.
- **Destete:** los terneros son separados de la madre para continuar su crianza en base a otra alimentación. Por tanto, se registra el cambio de categoría de la vaca, pasando de *Vaca con Cría al Pie* a *Vaca Vacía*, como del ternero, pasando de *Ternero al Pie* a *Ternero Destete*.

2.4.3. Módulo Traslados

El traslado, suele ser utilizado en momentos en los cuales es necesario situar a los vacunos en un campo distinto. Por ejemplo, en épocas de condiciones meteorológicas desfavorables o de grandes lluvias, es necesario trasladar los animales hacia tierras mas altas buscando salvaguardar la salud de los mismos y sectores con mejores pasturas. De esta forma, se evita que los animales sufran lesiones, pasen hambre o mueran.

Por tanto, es importante brindar la posibilidad de trasladar los vacunos de forma tal que sea posible continuar con el seguimiento y trazabilidad de los mismos. Para registrar un traslado es necesario que el usuario proporcione los siguientes datos:

- **Campo origen:** lugar en donde se encuentran los vacunos.
- **División origen:** sección del campo en donde están situados los vacunos.
- **Campo destino:** lugar en donde se encontrarán los vacunos. Distinto al de origen.

- **División destino:** sección del campo en donde estarán situados los vacunos, pudiendo ser la misma que la de origen.
- **Raza:** raza a la que pertenecen los vacunos.
- **Categoría:** categoría a la que pertenecen los vacunos.
- **Sub categoría:** sub categoría a la que pertenecen los vacunos.
- **Fecha:** momento en el cual se haya efectuado el traslado.
- **Descripción:** el usuario puede agregar un comentario referente a la operación.

Luego de haber ingresado los datos recientemente mencionados, serán listadas las caravanas de aquellos vacunos que cumplan con la configuración ingresada por el usuario. De esta lista de caravanas, el usuario debe seleccionar las correspondientes a los vacunos que forman parte de la operación en cuestión.

2.4.4. Módulo Mortandad

Debido a causas climáticas, enfermedades o accidentes, los animales pueden morir. Es fundamental llevar un control de las bajas ocurridas ya que, al tener un registro de ellas, es posible hacer un análisis y determinar un plan de acción para evitar próximas muertes. Para llevar adelante el registro de una mortandad es necesario que el usuario proporcione los siguientes datos:

- **Campo:** lugar en donde se encuentran los vacunos.
- **División:** sección del campo en donde están situados los vacunos.
- **Raza:** raza a la que pertenecen los vacunos.
- **Categoría:** categoría a la que pertenecen los vacunos.
- **Sub categoría:** sub categoría a la que pertenecen los vacunos.
- **Fecha:** momento en el cual se haya efectuado la o las muertes.
- **Descripción:** el usuario puede agregar un comentario referente a la operación.

Luego de haber ingresado los datos recientemente mencionados, serán listadas las caravanas de aquellos vacunos que cumplan con la configuración ingresada por el usuario. De esta lista de caravanas, el usuario debe seleccionar las correspondientes a los vacunos que forman parte de la operación en cuestión.

2.4.5. Módulo Actividades Veterinarias

El módulo de actividades veterinarias permite llevar un registro de las diversas acciones llevadas a cabo en lo que respecta a atenciones médicas para con los vacunos. Las mismas se realizan de forma tal de cumplimentar con las normas y estándares definidos por el Senasa, procurando salvaguardar la salud de los bovinos. Para llevar un registro de las diversas actividades sanitarias realizadas es necesario que el usuario proporcione los siguientes datos:

- **Campo:** lugar en donde se encuentran los vacunos.
- **División:** partición del campo en donde están situados los vacunos.
- **Raza:** raza a la que pertenecen los vacunos.
- **Categoría:** categoría a la que pertenecen los vacunos.
- **Sub categoría:** sub categoría a la que pertenecen los vacunos.
- **Tipo de actividad:** existen diversas actividades sanitarias que deben ser realizadas ya sea de forma periódica o paulatina. Por tanto el usuario debe indicar cuál de todas ellas es la que se esta queriendo registrar en el sistema.
- **Fecha:** momento en el cual se haya efectuado la actividad veterinaria.
- **Descripción:** el usuario puede agregar un comentario referente a la operación.

Luego de haber ingresado los datos recientemente mencionados, serán listadas las caravanas de aquellos vacunos que cumplan con la configuración ingresada por el usuario. De esta lista de caravanas, el usuario debe seleccionar las correspondientes a los vacunos que forman parte de la operación en cuestión.

En aquellos casos en los que, el tipo de actividad veterinaria seleccionada este relacionado con aspectos reproductivos, tales como la detección de celo o detección de vacas preñadas, es generado un evento dentro del sistema. Dicho evento, emite notificaciones pertinentes a través de correo electrónico, tanto al momento de su creación como así también al acercarse la fecha de cumplimiento de dicho evento.

2.4.6. Módulo Ventas

Otra forma de disminuir la cantidad de individuos en un campo dado, es a través de una venta. El módulo de ventas permite registrar en el sistema el egreso de vacunos. Para llevar adelante el registro de una venta es necesario que el usuario proporcione los siguientes datos:

- **Cliente:** cliente que haya intervenido en la venta.
- **Número de Factura:** cada venta posee un número de factura.
- **Fecha:** fecha del momento en el cual se haya realizado la venta.

- **Precio unitario:** precio, por kilo, de un individuo.
- **Campo:** lugar en donde se encuentran los vacunos.
- **División:** sección del campo en donde están situados los vacunos.
- **Raza:** raza a la que pertenecen los vacunos.
- **Categoría:** categoría a la que pertenecen los vacunos.
- **Sub categoría:** sub categoría a la que pertenecen los vacunos.
- **Cantidad:** total de vacunos.
- **Descripción:** el usuario puede agregar un comentario referente a la operación.

Luego de haber ingresado los datos recientemente mencionados, serán listadas las caravanas de aquellos vacunos que cumplan con la configuración ingresada por el usuario. De esta lista de caravanas, el usuario debe seleccionar las correspondientes a los vacunos que forman parte de la operación en cuestión.

2.5. Reportes

Los reportes son una visualización de la información existente en la base de datos o información que se va generando a medida que se trabaja con el sistema. Dicha información estará dispuesta en forma de planilla y los datos contenidos en el reporte dependerán de lo que se haya solicitado visualizar.

El procedimiento para generar reportes es cómodo y rápido. Aquellos usuarios con perfil *Administrador*, *Encargado en Jefe* y *Veterinario en Jefe*, podrán acceder a los reportes desde la última opción del menú. Ingresando a la página de reportes, es posible solicitar reportes acerca de *Usuarios*, *Vacunos* y *Operaciones*. En la Fig. (4.24) puede verse la página principal de reportes.

Los reportes generados pueden ser exportados a una planilla de cálculo con formato *.xlsx* o bien a formato portable *.pdf*. A continuación se realiza una breve descripción de los diversos reportes que pueden generarse.

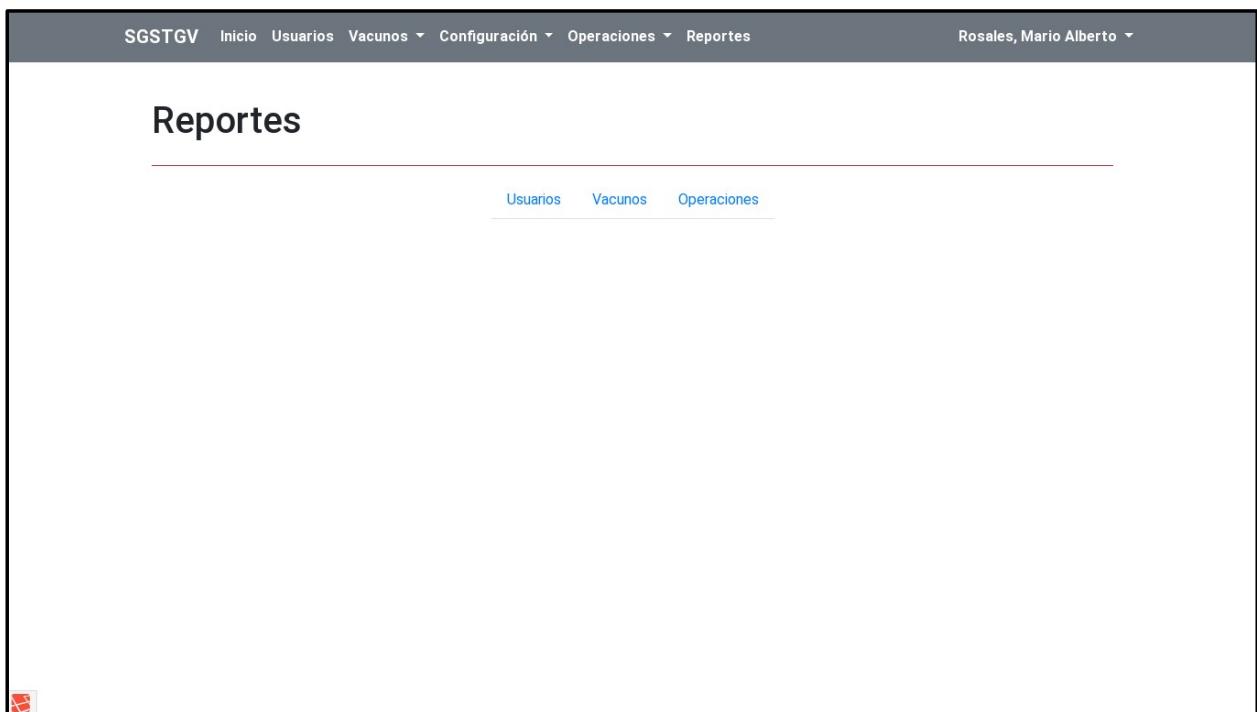


Figura. 4.24: Reportes.

2.5.1. Reporte de Usuarios

El reporte de usuarios puede hacerse por perfil o por campo. En caso de seleccionar el filtro por perfil, es necesario seleccionar el perfil que se desea visualizar. De este modo se puede obtener una lista de usuarios que contengan dicho perfil, como se muestra en la *Fig. (4.25)*.

Por otro lado, el filtro por campo, implica la selección de un campo y el informe resultante contendrá a los usuarios asociados a dicho campo. Este informe puede verse en la *Fig. (4.26)*

The screenshot shows the SGSTGV application's reporting module. At the top, there is a navigation bar with links for Inicio, Usuarios, Vacunos, Configuración, Operaciones, and Reportes. The current user is identified as Rosales, Mario Alberto.

Reportes

Below the navigation bar, there are three tabs: Usuarios (selected), Vacunos, and Operaciones.

Selección de tipo de filtro que desea aplicar:

Filtrar usuarios por Perfil: Administrador

Buscador

Apellido, Nombre	Documento	Teléfono	Email	Campo
Rosales, Mario Alberto	35650881	3425947097	mariorosales941@gmail.com	<ul style="list-style-type: none"> Las Tunas La Elsa Las Estefanía Cuay Chico Estancia El Cuqui

Exportar xls Exportar pdf

Figura. 4.25: Reporte de usuarios por perfil.

The screenshot shows the SGSTGV application's reporting module. At the top, there is a navigation bar with links for Inicio, Usuarios, Vacunos, Configuración, Operaciones, and Reportes. The current user is identified as Rosales, Mario Alberto.

Reportes

Below the navigation bar, there are three tabs: Usuarios (selected), Vacunos, and Operaciones.

Selección de tipo de filtro que desea aplicar:

Filtrar usuarios por Campo: Las Tunas

Buscador

Apellido, Nombre	Documento	Teléfono	Email	Campo
Navarro, Jerónimo	4444444	3425947097	jerojero@gmail.com	Encargado en Jefe
Rosales, Cuqui 1	7777777	3425947097	cuquirosales_1@gmail.com	Encargado
Rosales, Santiago	8888888	3425947097	cabezón@gmail.com	Veterinario en Jefe
Fenoglio, Angelo 1	9999999	3425947097	angelo_1@gmail.com	Veterinario
Rosales, Mario Alberto	35650881	3425947097	mariorosales941@gmail.com	Administrador

Exportar xls Exportar pdf

Figura. 4.26: Reporte de usuarios por campo.

2.5.2. Reporte de Vacunos

El reporte de vacunos es un informe que contiene todos los vacunos que cumplen con los filtros seleccionados por el usuario. Por tanto, es necesario que el usuario seleccione un campo, división, raza, categoría y sub categoría. En la *Fig. (4.27)* puede observarse un ejemplo de reporte de vacunos.

Caravana	Campo	División	Raza	Categoría	Sub Categoría	Activo
JC423-A200-8	Las Tunas	Corral	Aberdeen Angus	Toros	Recria	Si
JC423-A201-8	Las Tunas	Corral	Aberdeen Angus	Toros	Recria	Si

Figura. 4.27: Reporte de usuarios por campo.

2.5.3. Reporte de Operaciones

El reporte de operaciones contiene todos los registros almacenados en la base de datos. Para solicitar un reporte de operación es necesario indicar el tipo de operación, campo, quien haya registrado la operación, fecha desde y fecha hasta.

El único parámetro obligatorio es el tipo de operación, de esta manera el resto de los parámetros pueden no tenerse en cuenta y la información solicitada no contemplará los mismos. Los parámetros fecha desde y fecha hasta pueden filtrar la información de forma tal que la fecha de carga de cada operación esté comprendida en dicho rango de fechas. En la *Fig. (4.28)* se ilustra un ejemplo de reporte de operaciones.

Figura. 4.28: Reporte de usuarios por campo.

3. Resultados

Luego de haber realizado las actividades correspondientes al desarrollo del presente Proyecto Final de Carrera, se pueden realizar algunas observaciones respecto a los objetivos y requerimientos propuestos, como así también sobre modificaciones que han sido necesarias realizar para lograr los resultados obtenidos.

3.1. Objetivos y Alcance

En base a los objetivos y alcance definidos en la *Propuesta de Proyecto Final de Carrera*, se pueden realizar las siguientes observaciones:

- Se diseñó y desarrolló un sistema web para la gestión y seguimiento de trazabilidad de ganado vacuno. El software es una herramienta pensada para facilitar el trabajo de la cría y producción de ganado vacuno, brindando funcionalidades y características que permiten representar las actividades de campo de una manera fácil y sencilla. Permitiéndole al productor asegurar el bienestar y calidad de los vacunos. Para tal fin ha sido necesario tomar contacto con el dominio en cuestión para aprender la lógica de negocio asociada y poder diseñar un sistema capaz de reflejar el trabajo diario con los animales.
- Se implementaron funcionalidades administrativas y de configuración mediante las cuales es posible adaptar el sistema a las necesidades del cliente. De esta manera, el sistema está preparado para solventar cambios que deban ser establecidos para poder asegurar el cumplimiento del objetivo para el cual ha sido pensado.

- Se implementaron las funcionalidades operativas, aquellas que permiten llevar un registro de todas las actividades que intervienen en la cría y producción del ganado.
- La información generada y almacenada en la base de datos del sistema tiene un gran valor. Por tanto, se implementaron funcionalidades que permiten generar reportes con la información deseada, los mismos pueden ser exportados a archivos para un posterior análisis.

3.2. Requerimientos

El desarrollo del sistema se llevo a cabo respetando los requerimientos definidos en la primer etapa del proyecto. Sin embargo, ha sido necesario considerar algunas modificaciones sobre algunos de ellos. A saber:

- **RF02 - Perfiles de Usuario:** en una primera instancia se plantearon 3 perfiles de usuario (administrador, veterinario y peón). Durante el desarrollo del sistema ha surgido la necesidad de contar con perfiles un poco mas definidos y específicos, dando lugar a los perfiles mencionados en la Administración de Usuarios (ver Apartado (2.3)).
- **RF07 - Registro de Vacunos:** el ingreso de vacunos al sistema se realiza, principalmente, a través de una operación de compra. A partir de la misma, es posible asignar una caravana a cada individuo (permitiendo identificarlo y distinguirlo del resto), como así también definir su raza, categoría, sub categoría y peso promedio. En caso de ser necesario, información adicional puede ser cargada accediendo a los detalles de cada individuo.
- **RNF06 - Seguridad externa (Safety):** a continuación se listan algunas de las funciones proporcionadas por Laravel, que han sido utilizadas para dar seguridad a la aplicación:
 - Sistema de autenticación: el sistema de autenticación utiliza “providers” y “guards” para facilitar la tarea. Con Guards se puede controlar cómo se autenticarán los usuarios para cada solicitud realizada mientras que los Providers permiten recuperar usuarios de la base de datos. Por tanto, solo resta configurar la base de datos, los controladores y los modelos relacionados con el usuario para completar la autenticación.
 - Protección contra Inyección SQL: el ORM Eloquent en Laravel utiliza el enlace de parámetros PDO para luchar contra la inyección SQL. Este tipo de enlace de parámetros garantiza que los datos pasados por parte del usuario no se utilicen en forma directa en las consultas SQL. Evitando así que la consulta de un usuario malintencionado pueda provocar el robo de datos sensibles y otras consecuencias graves.
 - Protección contra CSRF (Cross Site Request Forgery) o falsificación de solicitud entre sitios: cuando un usuario ya autenticado visita otro sitio que posee un enlace malicioso que envía una solicitud a la ruta de su aplicación, su back-end solo sabe que se trata de una solicitud de un usuario autenticado. Sin embargo, el atacante controlará los datos enviados junto con la solicitud. Laravel, usa tokens CSRF para restringir que terceros generen dichas solicitudes falsificadas. Esto se hace utilizando un token válido que debe agregarse en cada solicitud. Luego, Laravel compara este token automáticamente con el valor que ha guardado. En caso de que el token no coincida con el almacenado, se considera que esa solicitud en particular no es válida, de lo contrario, esa solicitud es válida.

- *Protección contra XSS (Cross Site Scripting): los ataques XSS tienen lugar al momento en que un usuario usa campos de entrada para agregar código JavaScript. De esta forma, cada vez que los usuarios abran esa página, también se ejecutará este código JavaScript específico, que puede ser malicioso. Para solventar esto, Laravel realiza un escape automático al guardar contenido en la base de datos y también al imprimir contenido en HTML.*
- **RNF09 - Diseño Escalable y Mantenible:** gracias a la modularización del sistema es posible escalar y mantener el mismo. Las funcionalidades administrativas y de configuración (ver apartado (2.3)) permiten adaptar el sistema a nuevos cambios. Por otro lado, si lo que se necesita es adicionar una nueva funcionalidad operativa, la misma puede añadirse al sistema sin dañar las ya existentes.

3.3. Interfaz Gráfica de Usuario

En lo referente a la interfaz gráfica de usuario, se han respetado los principios de diseño definidos en la segunda etapa del proyecto. Sin embargo, cabe destacar que se ha utilizado una herramienta adicional que previamente no había sido seleccionada. Esta herramienta es **VueJS**, un framework JavaScript progresivo para construir interfaces de usuario. A diferencia de otros frameworks, Vue está diseñado desde el inicio para ser adoptado incrementalmente. Por otro lado, Vue también es perfectamente capaz de soportar aplicaciones sofisticadas utilizando en combinación con herramientas modernas y librerías compatibles.

El núcleo de VueJS está formado por una librería encargada de renderizar vistas en el navegador. Su forma de organizar el código es por medio de pequeños componentes que contienen todo el HTML, CSS y JavaScript necesario para funcionar como pieza independiente. Estas piezas se van componiendo en un árbol jerárquico de componentes hasta formar la aplicación.

Pero, ¿por qué elegir Vue.js?

- **Framework MVVM:** la ventaja de este tipo de frameworks es la facilidad para construir código bien estructurado, permitiendo construir aplicaciones complejas.
- **Solución ligera:** una de las grandes ventajas de Vue es el tamaño del núcleo. Este tamaño puede ir aumentando, debido a la flexibilidad y facilidad para extender el framework con variedad de soluciones de terceros que son bien recibidas por la comunidad. Su tamaño compactado reduce los tiempos de carga y velocidad de las aplicaciones.
- **Templates declarativos:** los templates en Vue se escriben en HTML fácilmente modificable por cualquier involucrado en el proyecto.
- **DOM virtual:** la implementación del DOM Virtual proporciona un alto performance que ponen a Vue como líder en rendimiento de renderizado.
- **Two-way Data Binding:** al igual que otros frameworks, emplea un data-binding bidireccional que sincroniza automáticamente el modelo con el DOM.

- **Curva de aprendizaje baja:** comparado con otros frameworks, Vue es una de las tecnologías JavaScript más sencillas para comenzar a desarrollar. Una de las mejores características es que se trata de un framework “*friendly developer*”. Además posee una excelente documentación.

De esta manera, al emplear los beneficios de Vue en conjunto con *Bootstrap*, herramienta previamente seleccionada, se ha logrado una interfaz gráfica de usuario limpia, sencilla y de gran rendimiento, haciendo posible una experiencia de usuario satisfactoria.

Capítulo 5

Pruebas

aca van las pruebas

Capítulo 6

Conclusiones y Trabajos Futuros

Capítulo 7

Apéndice

1. Diagramas de Casos de Uso

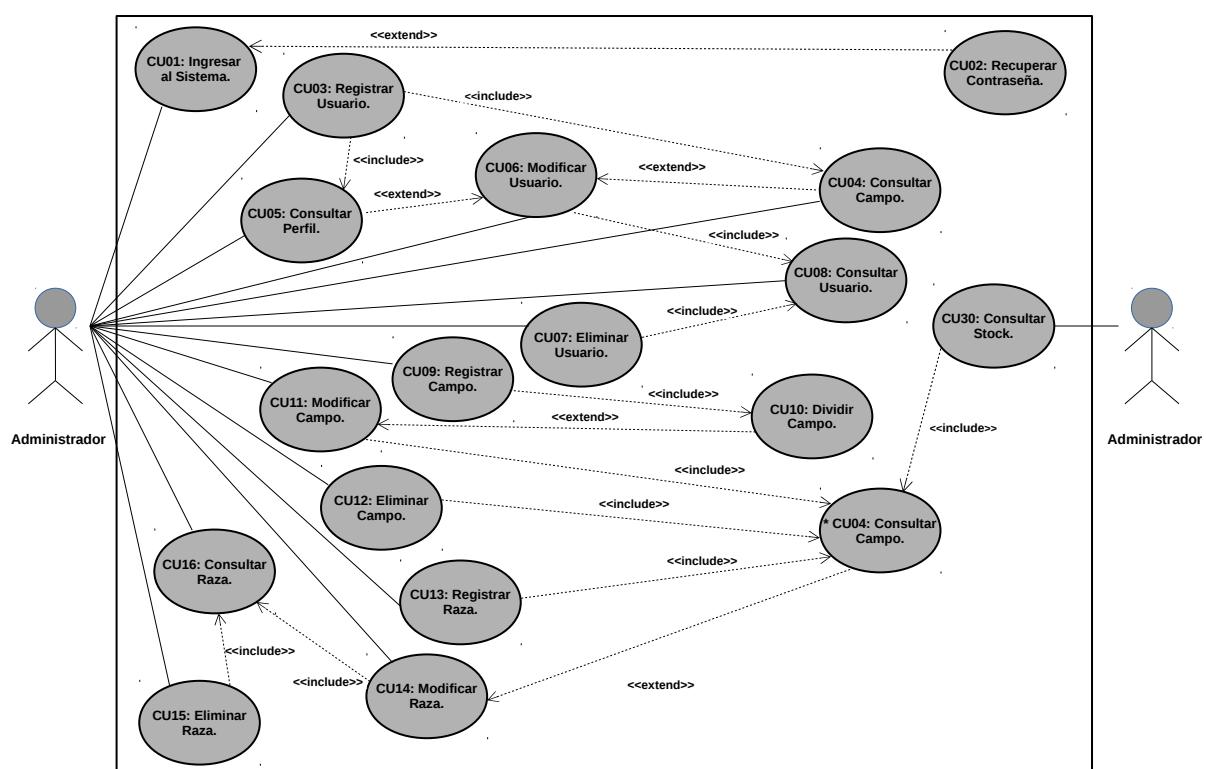


Figura. 7.1: Diagrama de Casos de Uso del Actor **Administrador**.

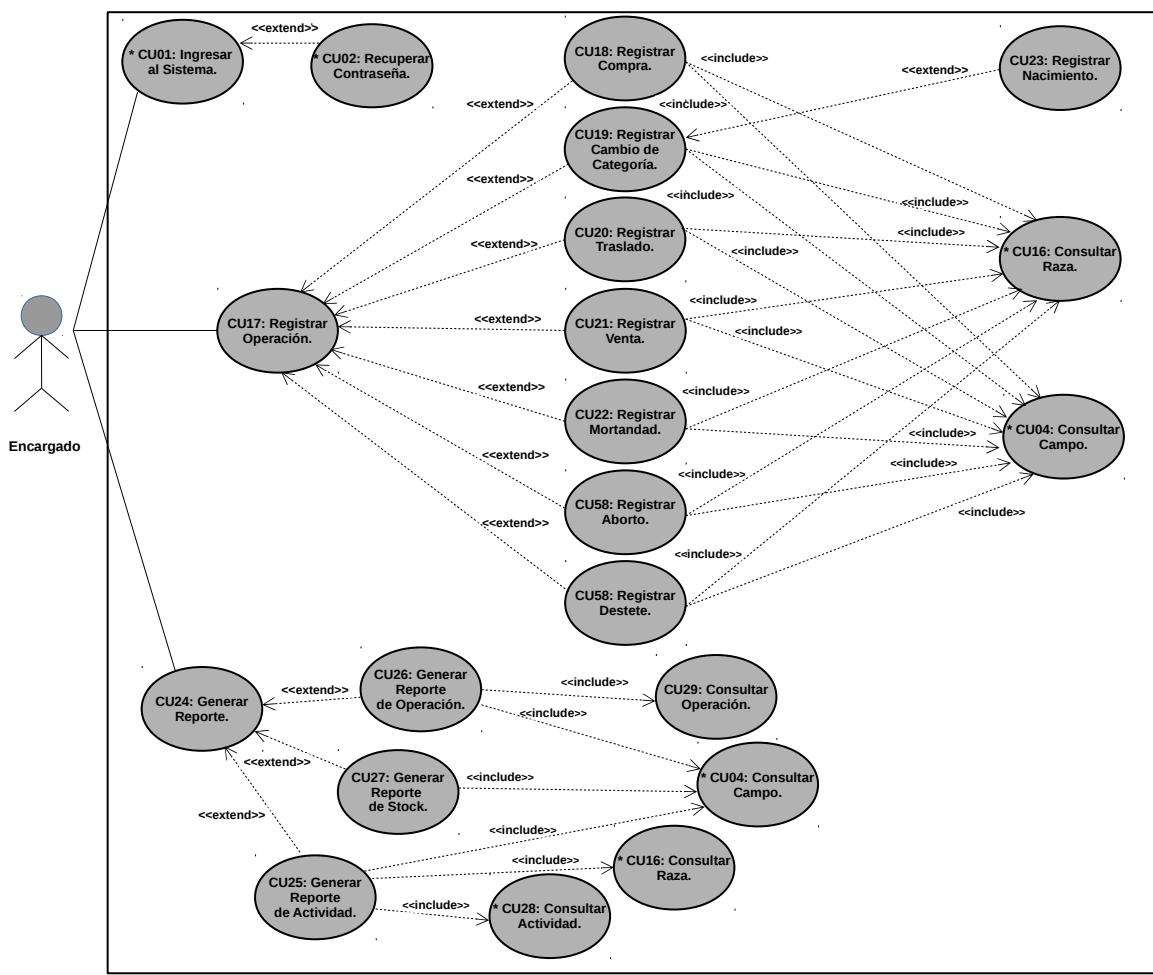


Figura. 7.2: Diagrama de Casos de Uso del Actor **Encargado** - Parte 1.

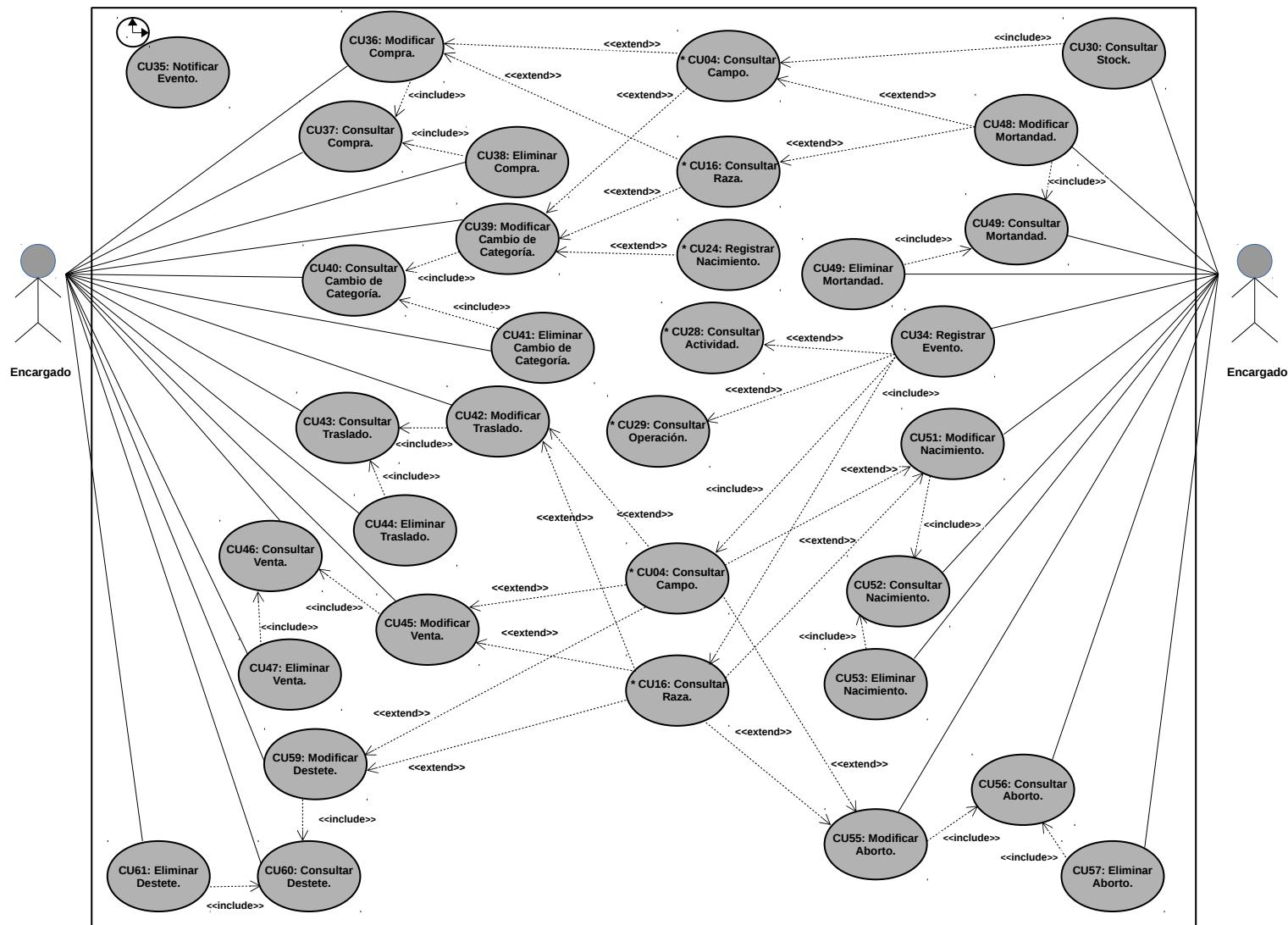


Figura. 7.3: Diagrama de Casos de Uso del Actor **Encargado** - Parte 2.

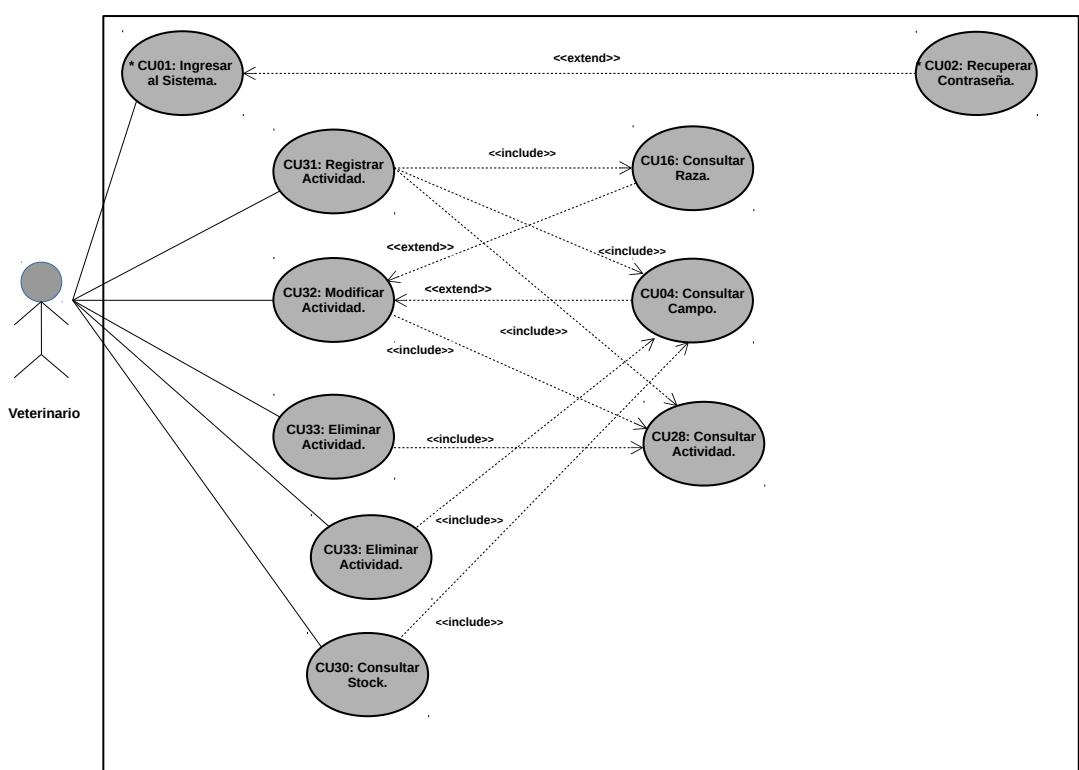


Figura. 7.4: Diagrama de Casos de Uso del Actor **Veterinario**.

2. Fichas Textuales de Casos de Uso

CU01: Ingresar al Sistema.	
Descripción:	El sistema autoriza el ingreso al actor.
Actor(es):	* Administrador * Encargado * Veterinario
Precondición:	Se debe contar con conexión a Internet. También es necesario contar con un navegador web.
	Curso Normal
1. El caso de uso comienza cuando el actor ingresa la dirección del sitio web en su navegador.	
2. El sistema solicita al actor ingresar los siguientes datos para lograr la autenticación: * Usuario: campo obligatorio de tipo texto. * Contraseña: Campo obligatorio de tipo contraseña (Por cada carácter ingresado se muestra un asterisco "*").	
3. El actor ingresa los datos solicitados y presiona el botón "Continuar".	3.a. Si el usuario presiona el botón "Recuperar Contraseña": 3.a.1. El sistema invoca al caso de uso CU02: Recuperar Contraseña . 3.a.2. El sistema vuelve al paso 2 del caso de uso.
4. El sistema coteja los datos ingresados por el actor y los compara con los almacenados en la base de datos.	4.a. En caso de que los datos no sean válidos: 4.a.1. El sistema muestra en pantalla un mensaje de error. 4.a.2. El sistema vuelve al paso 2 del caso de uso.
5. El sistema muestra en pantalla un mensaje de bienvenida y se sitúa en la pantalla principal del sitio.	
6. El caso de uso finaliza aquí.	
Poscondición:	El sistema mostrará al actor el menú principal con las opciones y funcionalidades correspondientes al perfil del actor.
Inclusiones:	
Extensiones:	* CU02: Recuperar Contraseña.

Figura. 7.5: CU01: Ingresar al Sistema.

CU02: Recuperar Contraseña.	
Descripción:	En caso de no recordar su contraseña de acceso, el actor puede solicitar una nueva al Administrador .
Actor(es):	* Administrador * Encargado * Veterinario
Precondición:	El actor quiere ingresar al sistema pero no recuerda la contraseña de acceso al mismo.
	Curso Normal
1. El caso de uso comienza al ser invocado por el caso de uso CU01: Ingresar al Sistema.	
2. El sistema solicita al actor que ingrese los siguientes datos: * Correo electrónico: campo obligatorio de tipo alfanumérico con formato de correo electrónico. * Fecha de Nacimiento: campo obligatorio de tipo fecha.	
3. El actor ingresa los datos solicitados y presiona el botón "Obtener Nueva Contraseña".	3.a. Si el actor presiona el botón Cancelar: 3.a.1. El caso de uso finaliza aquí.
4. El sistema coteja los datos ingresados por el usuario y los compara con los almacenados en la base de datos.	4.a. En caso de que los datos no sean válidos: 4.a.1. El sistema muestra en pantalla un mensaje de error. 4.a.2. El sistema vuelve al paso 2 del caso de uso.
5. El sistema envía al actor correo electrónico comunicando que se ha solicitado una nueva contraseña proporcionándole una aleatoria de manera provisoria.	
6. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con una nueva contraseña de manera provisoria que le permite acceder al sistema.
Inclusiones:	
Extensiones:	

Figura. 7.6: CU02: Recuperar Contraseña.

CU03: Registrar Usuario.	
Descripción:	El actor puede dar de alta nuevos usuarios con el perfil necesario.
Actor(es):	* Administrador
Precondición:	Haber ingresado al sistema.
Curso Normal	Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que ingrese los datos referentes al nuevo usuario: * Nombre : campo obligatorio de tipo texto. * Apellido : campo obligatorio de tipo texto. * Tipo y N° de Documento : campo obligatorio de tipo alfanumérico. * Fecha de Nacimiento : campo obligatorio de tipo fecha. * Domicilio : campo de tipo alfanumérico. * Correo Electrónico : campo obligatorio de tipo alfanumérico con formato de correo electrónico. * Nombre de Usuario : campo obligatorio alfanumérico.	
3. El actor ingresa la datos solicitada y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema evalúa que los datos ingresados sean del tipo y formato correctos.	4.a. En caso de encontrarse algún error: 4.a.1. El sistema muestra en pantalla un mensaje de error. 4.a.2. El sistema vuelve al paso 2 del caso de uso.
5. El sistema solicita al actor que asigne al nuevo usuario un Campo en el cual desempeñará sus actividades. Para lo cual el sistema invoca al CU04: Consultar Campo.	
6. El actor selecciona el Campo y presiona el botón "Continuar".	6.a. Si el actor presiona el botón "Cancelar": 6.a.1. El sistema vuelve al paso 2 del caso de uso.
7. El sistema solicita al actor indicar el Perfil del nuevo usuario a registrar. Para lo cual el sistema invoca al CU05: Consultar Perfil.	
8. El actor selecciona el Perfil y presiona el botón "Guardar".	8.a. Si el actor presiona el botón "Cancelar": 8.a.1. El sistema vuelve al paso 2 del caso de uso.
9. El sistema procede a almacenar en la base de datos la información del nuevo usuario.	9.a. En caso de ocurrir un error: 9.a.1. El sistema muestra en pantalla un mensaje de error. 9.a.2. El sistema vuelve al paso 2 del caso de uso.
10. El sistema envía un correo electrónico al actor y al nuevo usuario brindando un mensaje de bienvenida junto con una contraseña de acceso provisoria.	
11. El caso de uso finaliza aquí.	
Poscondición:	El nuevo usuario cuenta con los datos de acceso: Nombre de Usuario y Contraseña . Esto le permitirá acceder al sistema.
Inclusiones:	* CU04: Consultar Campo. * CU05: Consultar Perfil.
Extensiones:	

Figura. 7.7: CU03: Registrar Usuario.

CU04: Consultar Campo.		
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Campo , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.	
Actor(es):	* Administrador	
Precondición:	Una vez ingresado al sistema, el actor solicita consultar o seleccionar un Campo .	
	Curso Normal	Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor o por alguno de los siguientes casos de uso:		
* CU03: Registrar Usuario. * CU06: Modificar Usuario. * CU10: Dividir Campo. * CU11: Modificar Campo. * CU12: Eliminar Campo. * CU13: Registrar Raza. * CU14: Modificar Raza. * CU18: Registrar Compra. * CU19: Registrar Cambio de Categoría. * CU20: Registrar Traslado. * CU21: Registrar Venta. * CU22: Registrar Mortandad. * CU25: Reporte de Actividad. * CU26: Reporte de Operación. * CU27: Reporte de Stock. * CU30: Consultar Stock. * CU31: Registrar Actividad. * CU32: Modificar Actividad. * CU34: Registrar Evento. * CU36: Modificar Compra. * CU39: Modificar Cambio de Categoría. * CU42: Modificar Traslado. * CU45: Modificar Venta. * CU48: Modificar Mortandad. * CU51: Modificar Nacimiento. * CU14: Consultar Stock.		
2. El sistema solicita al actor que seleccione el Campo que desea consultar.		
3. El actor selecciona al Campo y presiona el botón "Aceptar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.	
4. El sistema muestra en pantalla toda la información almacenada en la base de dato sobre el Campo seleccionado.		
5. El caso de uso finaliza aquí.		
Poscondición:	El actor cuenta con la información del Campo seleccionado.	
Inclusiones:		
Extensiones:		

Figura. 7.8: CU04: Consultar Campo.

CU05: Consultar Perfil.		
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Perfil , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.	
Actor(es):	* Administrador	
Precondición:	Una vez ingresado al sistema el actor solicita consultar o seleccionar un Perfil .	
	Curso Normal	Curso Alternativo
1. El caso de uso comienza cuando es invocado por el actor o por los casos de uso:		
* CU03: Registrar Usuario. * CU06: Modificar Usuario.		
2. El sistema muestra en pantalla los Perfiles almacenados en la base de datos.		
3. El actor selecciona un Perfil y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.	
4. El sistema muestra en pantalla el detalle del perfil seleccionado.		
5. El caso de uso finaliza aquí.		
Poscondición:	El actor cuenta con la información detallada del Perfil seleccionado.	
Inclusiones:		
Extensiones:		

Figura. 7.9: CU05: Consultar Perfil.

CU06: Modificar Usuario.	
Descripción:	El sistema muestra en pantalla los datos de un Usuario determinado para actualizarlos.
Actor(es):	* Administrador
Precondición:	Debe existir al menos un Usuario almacenado en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Usuario que desea modificar. Para lo cual el sistema invoca al caso de uso CU08: Consultar Usuario .	
3. El actor selecciona el Usuario que desea actualizar y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla los datos asociados al Usuario seleccionado.	
5. El sistema solicita al actor que modifique los siguientes datos: * Modificar Campo . * Modificar Perfil . * Nombre : campo obligatorio de tipo texto. * Apellido : campo obligatorio de tipo texto. * Tipo y N° de Documento : campo obligatorio de tipo alfanumérico. * Fecha de Nacimiento : campo obligatorio de tipo fecha. * Domicilio : campo opcional de tipo alfanumérico. * Correo Electrónico : campo obligatorio de tipo alfanumérico con formato de correo electrónico. * Nombre de Usuario : campo obligatorio alfanumérico.	5.a. Si el actor presiona el botón "Modificar Campo": 5.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 5.a.2. El sistema vuelve al paso 5 del caso de uso. 5.b. Si el actor presiona el botón "Modificar Perfil": 5.b.1. El sistema invoca al caso de uso CU05: Consultar Perfil . 5.b.2. El sistema vuelve al paso 5 del caso de uso.
6. El actor modifica los datos solicitados y presiona el botón "Guardar".	6.a. Si el actor presiona el botón "Cancelar": 6.a.1. El sistema vuelve al paso 2 del caso de uso.
7. El sistema procede a guardar las modificaciones realizadas.	7.a. Si ocurre un error al guardar: 7.a.1. El sistema muestra un mensaje de error. 7.a.2. El sistema vuelve al paso 5 del caso de uso.
8. El sistema muestra en pantalla el mensaje " Usuario actualizado con éxito!".	
10. El caso de uso finaliza aquí.	
Poscondición:	Han sido actualizados los datos asociados al Usuario seleccionado.
Inclusiones:	* CU08: Consultar Usuario.
Extensiones:	* CU04: Consultar Campo. * CU05: Consultar Perfil.

Figura. 7.10: CU06: Modificar Usuario.

CU07: Eliminar Usuario.	
Descripción:	El sistema elimina de la base de datos a un Usuario determinado.
Actor(es):	* Administrador
Precondición:	Debe existir al menos un Usuario almacenado en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Usuario que desea eliminar. Para lo cual el sistema invoca al caso de uso CU08: Consultar Usuario .	
3. El actor selecciona el Usuario y presiona el botón "Continuar".	3.a. El actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos del Usuario seleccionado.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema muestra en pantalla el mensaje: "Esta seguro de eliminar el Usuario ?".	
7. El actor presiona "Aceptar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema procede a eliminar al Usuario de la base de datos.	8.a. Si ocurre un error al eliminar: 8.a.1. El sistema muestra un mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema muestra en pantalla el mensaje " Usuario eliminado correctamente."	
10. El caso de uso finaliza aquí.	
Poscondición:	Se ha eliminado el Usuario seleccionado de la base de datos.
Inclusiones:	* CU08: Consultar Usuario.
Extensiones:	

Figura. 7.11: CU07: Eliminar Usuario.

CU08: Consultar Usuario.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Usuario , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Administrador
Precondición:	Una vez ingresado al sistema, el actor solicita consultar o seleccionar un Usuario .
	Curso Normal Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor o por los siguientes casos de uso: * CU06: Modificar Usuario * CU07: Eliminar Usuario.	
2. El sistema solicita al actor que seleccione el Usuario que desea consultar.	
3. El actor selecciona el Usuario y presiona el botón "Aceptar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla toda la información almacenada en la base de datos acerca del Usuario seleccionado.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información del Usuario seleccionado.
Inclusiones:	
Extensiones:	

Figura. 7.12: CU08: Consultar Usuario.

CU09: Registrar Campo.	
Descripción:	El actor puede dar de alta nuevos Campos en los que se puedan criar Vacunos , pudiendo llevar adelante la gestión y seguimiento de los mismos.
Actor(es):	* Administrador
Precondición:	Haber ingresado al sistema y seleccionar la opción de registrar un nuevo Campo .
	Curso Normal Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que ingrese los siguientes datos: * Nombre del Campo : campo obligatorio de tipo texto. * Ubicación : campo obligatorio del tipo alfanumérico. * Capacidad : campo obligatorio de tipo numérico. * Información Adicional : campo opcional de tipo alfanumérico.	
3. El actor ingresa la información solicitada y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema evalúa que los datos ingresados sean del tipo y formato correctos.	4.a. En caso de encontrarse algún error: 4.a.1. El sistema muestra en pantalla un mensaje de error. 4.a.2. El sistema vuelve al paso 2 del caso de uso.
5. El sistema solicita al actor que indique las divisiones en las cuales el nuevo Campo va a tener distribuidos los Vacunos . Para lo cual el sistema invoca al caso de uso CU11: Dividir Campo.	
6. El actor indica el número de divisiones que el nuevo Campo tendrá, indicando también la capacidad de cada sector. Luego presiona el botón "Continuar".	6.a. Si el actor presiona el botón "Cancelar": 6.a.1. El sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje "Desea registrar el nuevo Campo ?"	
8. El actor presiona el botón "Aceptar"	8.a. El actor presiona el botón "Cancelar": 8.a.1. El sistema vuelve al paso 2 del caso de uso.
9. El sistema procede a guardar los datos del nuevo Campo .	9.a. En caso de ocurrir un error: 9.a.1. El sistema muestra en pantalla un mensaje de error. 9.a.2. El sistema vuelve al paso 2 del caso de uso.
10. El sistema muestra en pantalla el mensaje "Campo registrado correctamente".	
11. El caso de uso finaliza aquí.	
Poscondición:	El sistema cuenta con un nuevo Campo en el cual se realizarán las actividades de gestión y seguimiento de Vacunos .
Inclusiones:	* CU11: Dividir Campo.
Extensiones:	

Figura. 7.13: CU09: Registrar Campo.

CU10: Dividir Campo.	
Descripción:	Las divisiones, dentro de un Campo , se utilizan para organizar y distribuir a los Vacunos según su raza, sexo, edad, peso, alimentación, entre otras. Esto facilita la tarea de gestión y seguimiento de los mismos.
Actor(es):	* Administrador
Precondición:	Haber ingresado al sistema.
Curso Normal	
1. El caso de uso comienza al ser invocado por los casos de uso: * CU09: Registrar Campo * CU11: Modificar Campo	
2. El sistema solicita al actor que indique cuáles y cuántos serán los sectores en los que el Campo será dividido. Para lo cual el sistema muestra al actor una lista de posibles divisiones y cantidades.	
3. El actor ingresa la información solicitada y presiona el botón "Guardar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El sistema vuelve al paso 2 del caso de uso.
4. El sistema procede a guardar la información proporcionada por el usuario.	4.a. En caso de ocurrir un error. 4.a.1. El sistema muestra en pantalla el mensaje de error. 4.a.2. El sistema vuelve al paso 2 del caso de uso.
5. El sistema muestra en pantalla el mensaje "Datos guardados correctamente".	
6. El caso de uso finaliza aquí.	
Poscondición:	El campo seleccionado por el actor cuenta con una nueva División para alojar Vacunos .
Inclusiones:	
Extensiones:	

Figura. 7.14: CU10: Dividir Campo.

CU11: Modificar Campo.	
Descripción:	El actor puede modificar los atributos y propiedades de un Campo .
Actor(es):	* Administrador
Precondición:	Debe existir al menos un Campo almacenado en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Campo que desea modificar. Para lo cual invoca al caso de uso CU04: Consultar Campo.	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que modifique los datos del Campo Seleccionado. Entre ellos se encuentran: * Nombre del Campo : campo obligatorio de tipo texto. * Ubicación : campo obligatorio del tipo alfanumérico. * Capacidad : campo obligatorio de tipo numérico. * Información Adicional : campo opcional de tipo alfanumérico. * Modificar División .	4.a. El actor selecciona la opción "Modificar División": 4.a.1. El sistema invoca al caso de uso CU11: Dividir Campo. 4.a.2. El sistema avanza al paso 4 del caso de uso.
5. El actor modifica los datos y presiona el botón "Continuar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El actor presiona el botón "Guardar".	
7. El sistema procede a guardar los cambios realizados.	7.a. En caso de ocurrir un error. 7.a.1. El sistema muestra un mensaje de error en pantalla. 7.a.2. El sistema vuelve al paso 2 del caso de uso.
8. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada al Campo seleccionado ha sido actualizada.
Inclusiones:	* CU04: Consultar Campo
Extensiones:	* CU11: Dividir Campo

Figura. 7.15: CU11: Modificar Campo.

CU12: Eliminar Campo.	
Descripción:	El actor puede eliminar un Campo de la base de datos.
Actor(es):	* Administrador
Precondición:	Haber almacenado al menos un Campo en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Campo que desea eliminar. Para ello el sistema invoca al caso de uso CU04: Consultar Campo .	
3. El actor selecciona un Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
3. El sistema muestra los datos del Campo a eliminar.	
4. El actor presiona el botón "Eliminar".	4.a. El actor presiona el botón "Cancelar": 4.a.1. El sistema vuelve al paso 2 del caso de uso.
5. El sistema procede a eliminar el Campo de la base de datos.	5.a. Si ocurre un error al eliminar: 5.a.1. El sistema muestra un mensaje de error. 5.a.2. El sistema vuelve al paso 2 del caso de uso.
6. El sistema muestra en pantalla el mensaje: " Campo eliminado correctamente!".	
7. El caso de uso finaliza aquí.	
Poscondición:	El Campo seleccionado ha sido eliminado de la base de datos.
Inclusiones:	* CU04: Consultar Campo .
Extensiones:	

Figura. 7.16: CU12: Eliminar Campo.

CU13: Registrar Raza.	
Descripción:	El actor puede registrar las Razas de Vacunos con las que se va a trabajar.
Actor(es):	* Administrador
Precondición:	Haber almacenado al menos un Campo en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que ingrese los siguientes datos: * Nombre : campo obligatorio de tipo texto. * Descripción : campo opcional de tipo alfanumérico. * Categorías : campo obligatorio de tipo alfanumérico. * Subcategorías : campo obligatorio de tipo alfanumérico.	
3. El actor ingresa los datos solicitados y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione el Campo en el que dicha Raza estará alojada. Para lo cual invoca al caso de uso CU04: Consultar Campo .	
5. El actor selecciona el Campo y presiona el botón "Aceptar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a guardar los cambios ingresados por el actor.	6.a. En caso de ocurrir un Error: 6.a.1. El sistema muestra en pantalla un mensaje de Error. 6.a.2. El sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje "Datos guardados correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	La nueva Raza se ha almacenado en la base de datos.
Inclusiones:	* CU04: Consultar Campo .
Extensiones:	

Figura. 7.17: CU13: Registrar Raza.

CU14: Modificar Raza.	
Descripción:	El actor puede modificar los atributos y propiedades de una Raza .
Actor(es):	* Administrador
Precondición:	Debe haber al menos una Raza almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza cuando es invocado por el actor.	
2. El sistema solicita al actor que seleccione la Raza que desea modificar. Para lo cual invoca al caso de uso CU16: Consultar Raza .	
3. El actor selecciona la Raza y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que modifique los datos de la Raza seleccionada. Entre ellos se encuentran: * Nombre : campo obligatorio de tipo texto. * Descripción : campo opcional de tipo texto. * Categorías : campo obligatorio de tipo texto. * Subcategorías : campo obligatorio de tipo texto. * Modificar Campo .	4.a. Si el actor selecciona la opción "Modificar el Campo": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema avanza al paso 4 del caso de uso.
5. El actor modifica los datos y presiona el botón "Guardar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a guardar los cambios ingresados por el actor.	6.a. En caso de ocurrir un Error: 6.a.1. El sistema muestra en pantalla un mensaje de Error. 6.a.2. El sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje "Datos guardados correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada a la Raza seleccionada se encuentra actualizada.
Inclusiones:	* CU16: Consultar Raza .
Extensiones:	* CU04: Consultar Campo .

Figura. 7.18: CU14: Modificar Raza.

CU15: Eliminar Raza.	
Descripción:	El actor puede eliminar una Raza de la base de datos.
Actor(es):	* Administrador
Precondición:	Debe haber al menos una Raza almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Raza que desea eliminar. Para lo cual invoca al caso de uso CU16: Consultar Raza .	
3. El actor selecciona una Raza y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos de la Raza seleccionada.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar la Raza .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Raza eliminada correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	La Raza ha sido eliminada de la base de datos.
Inclusiones:	* CU16: Consultar Raza .
Extensiones:	

Figura. 7.19: CU15: Eliminar Raza.

CU16: Consultar Raza.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Raza , la cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Administrador
Precondición:	Debe haber al menos una Raza almacenada en la base de datos.
	Curso Normal Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU14: Modificar Raza * CU15: Eliminar Raza.	
2. El sistema solicita al actor que seleccione la Raza que desea consultar.	
3. El actor selecciona la Raza y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla toda la información almacenada sobre la Raza seleccionada, indicando Categoría y Subcategoría correspondientes.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada de la Raza seleccionada.
Inclusiones:	
Extensiones:	

Figura. 7.20: CU16: Consultar Raza.

CU17: Registrar Operación.	
Descripción:	El actor puede registrar las operaciones que modifiquen la Cantidad y Estados de los Vacunos .
Actor(es):	* Encargado
Precondición:	Debe haber almacenada al menos una operación en la base de datos.
	Curso Normal Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Tipo de Operación que desea registrar. El sistema muestra en pantalla el siguiente listado de operaciones: * Registrar Compra. * Registrar Cambio de Categoría. * Registrar Traslado. * Registrar Venta. * Registrar Mortandad.	2.a. Si el actor selecciona la opción "Registrar Compra": 2.a.1. El sistema invoca al caso de uso CU18: Registrar Compra . 2.a.2. El sistema vuelve al paso 2 del caso de uso. 2.b. Si el actor selecciona la opción "Registrar Cambio de Categoría": 2.b.1. El sistema invoca al caso de uso CU19: Registrar Cambio de Categoría . 2.b.2. El sistema vuelve al paso 2 del caso de uso. 2.c. Si el actor selecciona la opción "Registrar Traslado": 2.c.1. El sistema invoca al caso de uso CU20: Registrar Traslado . 2.c.2. El sistema vuelve al paso 2 del caso de uso. 2.d. Si el actor selecciona la opción "Registrar Venta": 2.d.1. El sistema invoca al caso de uso CU21: Registrar Venta . 2.d.2. El sistema vuelve al paso 2 del caso de uso. 2.e. Si el actor selecciona la opción "Registrar Mortandad": 2.e.1. El sistema vuelve al caso de uso CU22: Registrar Mortandad . 2.e.2. El sistema vuelve al paso 2 del caso de uso.
3. El actor presiona el botón "Salir".	
4. El caso de uso finaliza aquí.	
Poscondición:	La Operación ha sido almacenado en la base de datos.
Inclusiones:	
Extensiones:	* CU18: Registrar Compra. * CU19: Registrar Cambio de Categoría. * CU20: Registrar Traslado. * CU21: Registrar Venta. * CU22: Registrar Mortandad.

Figura. 7.21: CU17: Registrar Operación.

CU18: Registrar Compra.	
Descripción:	El actor puede aumentar la cantidad de Vacunos alojados en un Campo registrando en la base de datos una Compra .
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Campo almacenado en la base de datos y éste, debe tener espacio disponible para albergar a los nuevos Vacunos .
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU17: Registrar Operación .	
2. El sistema solicita al actor que seleccione el Campo en cual los nuevos Vacunos serán alojados. Para lo cual invoca al CU04: Consultar Campo .	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza de los nuevos Vacunos . Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza .	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
7. El actor ingresa los datos solicitados y presiona el botón "Guardar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema procede a guardar los datos ingresados por el actor.	8.a. El caso de ocurrir un error: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema muestra en pantalla el mensaje: " Compra registrada correctamente!".	
10. El caso de uso finaliza aquí.	
Poscondición:	La Compra de Vacunos se encuentra almacenada en la base de datos.
Inclusiones:	* CU04: Consultar Campo . * Cu16: Consultar Raza .
Extensiones:	

Figura. 7.22: CU18: Registrar Compra.

CU19: Registrar Cambio de Categoría.	
Descripción:	El actor puede modificar la Categoría o Subcategoría de un Vacuno . De esta manera se puede almacenar en base de datos el ciclo de vida de un Bovino .
Actor(es):	* Encargado
Precondición:	Debe haber almacenado en la base de datos la cantidad suficiente de Vacunos con la Categoría y Subcategoría solicitada por el actor.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU17: Registrar Operación .	
2. El sistema solicita al actor que seleccione el Campo en el que desea registrar el Cambio de Categoría . Para lo cual invoca al caso de uso CU04: Consultar Campo .	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Categoría y Subcategoría de los Vacunos a ser afectados. Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza .	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que ingrese la Cantidad de Vacunos a ser afectados.	
7. El actor ingresa la Cantidad y presiona el botón "Continuar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema evalúa si existen suficientes Vacunos con la Raza seleccionada.	8.a. Si no existen Vacunos suficientes: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 6 del caso de uso.
9. El sistema solicita al actor que seleccione la Raza que los Vacunos afectados tendrán. Para lo cual invoca al caso de uso CU16: Consultar Raza .	
10. El actor selecciona la Raza y presiona el botón "Continuar".	10.a. Si el actor presiona el botón "Cancelar": 10.a.1. El sistema vuelve al paso 2 del caso de uso.
11. El sistema evalúa si la Raza Origen y la Raza Destino definen un Parto .	11.a. Si la Raza Origen y la Raza Destino definen un Parto : 11.a.1. El sistema invoca al caso de uso CU24: Registrar Nacimiento . 11.a.2. El sistema avanza al paso 12 del caso de uso. 11.b. Si la Raza Origen y la Raza Destino no definen un Parto : 11.b.1. El sistema avanza al paso 12 del caso de uso.
12. El sistema solicita al actor que ingrese los siguientes datos: * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
13. El actor ingresa los datos solicitados y presiona el botón "Guardar".	13.a. Si el actor presiona el botón "Cancelar": 13.a.1. El sistema vuelve al paso 2 del caso de uso.
14. El sistema procede a guardar los datos ingresados por el actor.	14.a. El caso de ocurrir un error: 14.a.1. El sistema muestra en pantalla el mensaje de error. 14.a.2. El sistema vuelve al paso 2 del caso de uso.
15. El sistema muestra en pantalla el mensaje: " Cambio de Categoría registrado correctamente!".	
16. El caso de uso finaliza aquí.	
Poscondición:	El Cambio de Categoría ha sido almacenado en la base de datos.
Inclusiones:	* CU04: Consultar Campo . * Cu16: Consultar Raza .
Extensiones:	* CU24: Registrar Nacimiento .

Figura. 7.23: CU19: Registrar Cambio de Categoría.

CU20: Registrar Traslado.	
Descripción:	El actor puede retirar Vacunos de un Campo y llevarlos a otro. De esta manera se modifica el Stock de Vacunos disminuyendo o aumentando en el Campo Origen y Destino , respectivamente
Actor(es):	* Encargado.
Precondición:	Debe haber almacenados al menos dos Campo en la base de datos. El Campo Destino debe contar con espacio disponible para alojar a los Vacunos trasladados.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU17: Registrar Operación.	
2. El sistema solicita al actor que seleccione el Campo Origen y Destino . Para lo cual invoca al caso de uso CU04: Consultar Campo .	
3. El actor selecciona ambos Campos y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza que desea trasladar. Para lo cual invoca al caso de uso CU16: Consultar Raza .	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que ingrese la Cantidad de Vacunos a trasladar.	
7. El actor ingresa la Cantidad y presiona el botón "Continuar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema evalúa, según la Raza seleccionada y la Cantidad , si existen Vacunos suficientes en el Campo Origen para ser trasladados y si el Campo Destino posee espacio disponible.	8.a. En caso de ser posible el traslado: 8.a.1. El sistema avanza al paso 9 del caso de uso. 8.b. En caso de no ser posible el traslado: 8.b.1. El sistema muestra en pantalla el mensaje de error. 8.b.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema solicita al actor que ingrese los siguientes datos: * Fecha: campo obligatorio de tipo fecha. * Descripción: campo opcional de tipo alfanumérico.	
10. El actor ingresa los datos solicitados y presiona "Guardar".	10.a. Si el actor presiona el botón "Cancelar": 10.a.1. El sistema vuelve al paso 2 del caso de uso.
11. El sistema procede a almacenar en base de datos el Traslado .	11.a. El caso de ocurrir un error: 11.a.1. El sistema muestra en pantalla el mensaje de error. 11.a.2. El sistema vuelve al paso 2 del caso de uso.
12. El sistema muestra en pantalla el mensaje: "Traslado registrado correctamente!".	
13. El caso de uso finaliza aquí.	
Poscondición:	El Traslado ha sido almacenado en la base de datos.
Inclusiones:	* CU04: Consultar Campo. * Cu16: Consultar Raza.
Extensiones:	

Figura. 7.24: CU20: Registrar Traslado.

CU21: Registrar Venta.	
Descripción:	El actor puede disminuir la cantidad de Vacunos en un Campo registrando una Venta en la base de datos.
Actor(es):	* Encargado.
Precondición:	Debe haber almacenado en la base de datos al menos un Campo con la suficiente cantidad de Vacunos a vender.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU17: Registrar Operación.	
2. El sistema solicita al actor que seleccione el Campo en el cual será registrada la Venta . Para lo cual invoca al CU04: Consultar Campo.	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza . Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza.	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El caso de uso finaliza aquí.
6. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
7. El usuario ingresa los datos solicitados y presiona el botón "Guardar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema evalúa que existan Vacunos suficientes en el Campo seleccionado, según la Cantidad ingresada.	8.a. En caso de no existir Vacunos suficientes: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema procede a almacenar los datos ingresados por el actor.	9.a. En caso de ocurrir un error: 9.a.1. El sistema muestra en pantalla el mensaje de error. 9.a.2. El sistema vuelve al paso 2 del caso de uso.
10. El sistema muestra en pantalla el mensaje: "Venta registrada correctamente!".	
11. El caso de uso finaliza aquí.	
Poscondición:	Se ha almacenado en la base de datos la Venta .
Inclusiones:	* CU04: Consultar Campo. * Cu16: Consultar Raza.
Extensiones:	

Figura. 7.25: CU21: Registrar Venta.

CU22: Registrar Mortandad.	
Descripción:	El actor puede disminuir la cantidad de Vacunos en un Campo registrando en la base de datos el fallecimiento de Bovinos .
Actor(es):	* Encargado.
Precondición:	Debe haber al menos un Campo con Vacunos suficientes para registrar la Mortandad almacenados en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU17: Registrar Operación.	
2. El sistema solicita al actor que seleccione el Campo en el cual han fallecido los Vacunos . Para lo cual invoca al caso de uso CU04: Consultar Campo.	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza . Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza.	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
7. El usuario ingresa los datos solicitados y presiona el botón "Guardar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema evalúa que existan Vacunos suficientes en el Campo seleccionado, según la Cantidad ingresada.	8.a. En caso de no existir Vacunos suficientes: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema procede a almacenar los datos ingresados por el actor.	9.a. En caso de ocurrir un error: 9.a.1. El sistema muestra en pantalla el mensaje de error. 9.a.2. El sistema vuelve al paso 2 del caso de uso.
10. El sistema muestra en pantalla el mensaje: "Mortandad registrada correctamente!".	
11. El caso de uso finaliza aquí.	
Poscondición:	La Mortandad ha sido almacenada en la base de datos.
Inclusiones:	* CU04: Consultar Campo. * Cu16: Consultar Raza.
Extensiones:	

Figura. 7.26: CU22: Registrar Mortandad.

CU23: Registrar Nacimiento.	
Descripción:	El actor puede aumentar la cantidad de Vacunos alojados en un Campo registrando en la base de datos un Nacimiento . Esto implica que se registran nuevos Bovinos con Categoría Terneros pudiendo ser tanto Machos como Hembras.
Actor(es):	* Encargado .
Precondición:	Debe haber almacenado en la base de datos al menos un Campo y éste debe tener capacidad para albergar a los Terneros.
	Curso Normal
1. El caso de uso comienza al ser invocado por el caso de uso CU19: registrar Cambio de Categoría .	
2. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad de Terneros Machos : campo obligatorio de tipo numérico. * Cantidad de Terneros Hembras : campo obligatorio de tipo numérico.	
3. El actor ingresa las Cantidades y presiona el botón "Aceptar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El caso de uso finaliza aquí.	
Poscondición:	Se tiene la Cantidad de Terneros Machos y Hembras a ser almacenados en la base de datos.
Inclusiones:	
Extensiones:	

Figura. 7.27: CU23: Registrar Nacimiento.

CU24: Generar Reporte.	
Descripción:	El actor puede emitir un Reporte que permita organizar la información almacenada en la base de datos de una manera clara y concreta, para luego ser analizada por los Usuarios.
Actor(es):	* Encargo .
Precondición:	Haber ingresado al sistema.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Tipo de Reporte que desea generar. El sistema muestra en pantalla el siguiente listado de reportes: * Reporte de Actividad . * Reporte de Operación . * Reporte de Stock .	2.a. Si el actor selecciona la opción "Reporte de Actividad": 2.a.1. El sistema invoca al caso de uso CU25: Reporte de Actividad . 2.a.2. El sistema vuelve al paso 2 del caso de uso. 2.b. Si el actor selecciona la opción "Reporte de Operación": 2.b.1. El sistema invoca al caso de uso CU26: Reporte de Operación . 2.b.2. El sistema vuelve al paso 2 del caso de uso. 2.c. Si el actor selecciona la opción "Reporte de Stock": 2.c.1. El sistema invoca al caso de uso CU27: Reporte de Stock . 2.c.2. El sistema vuelve al paso 2 del caso de uso.
3. El actor presiona el botón "Salir".	
4. El caso de uso finaliza aquí.	
Poscondición:	El reporte ha sido generado.
Inclusiones:	
Extensiones:	* CU25: Reporte de Actividad . * CU26: Reporte de Operación . * CU27: Reporte de Stock .

Figura. 7.28: CU24: Generar Reporte.

CU25: Generar Reporte de Actividad.	
Descripción:	El actor puede emitir un reporte en el cual estén representadas todas las actividades referidas al cuidado de los Vacunos.
Actor(es):	* Encargado.
Precondición:	Haber ingresado al sistema.
	Curso Normal
1. El caso de uso comienza al ser invocado por el caso de uso CU24: Generar Reporte.	
2. El sistema solicita al actor que seleccione el Campo sobre el cual desea generar el reporte. Para lo cual invoca al caso de uso CU04: Consultar Campo.	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza de los nuevos Vacunos. Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza.	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que seleccione la Actividad sobre la cual quiere generar el reporte. Para lo cual invoca al caso de uso CU28: Consultar Actividad.	
7. El actor selecciona la Actividad y presiona el botón "Continuar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema solicita al actor que elija qué hacer con el Reporte de Actividad . El sistema muestra en pantalla el siguiente listado de opciones: * Descargar. * Enviar por Correo.	8.a. El actor selecciona la opción "Descargar": 8.a.1. El sistema solicita al actor que seleccione el directorio en donde el Reporte será almacenado. 8.a.2. El actor selecciona el directorio y presiona el botón "Guardar". 8.a.2.1. El sistema procede a guardar el archivo. 8.a.2.1.1. Si ocurre un error al guardar: 8.a.2.1.1.1. El sistema muestra en pantalla el mensaje de error. 8.a.2.1.1.2. El sistema vuelve al paso 8 del caso de uso. 8.a.2.2. El sistema muestra en pantalla el mensaje: "Reporte guardado correctamente". 8.a.3. El actor presiona el botón "Cancelar": 8.a.3.1. El sistema vuelve al paso 8 del caso de uso. 8.b. El actor selecciona la opción "Enviar por Correo": 8.b.1. El sistema solicita al actor que ingrese las direcciones de correo electrónico de los destinatarios. 8.b.2. El actor ingresa las direcciones y presiona el botón "Enviar". 8.b.2.1. El sistema procede a enviar el correo: 8.b.2.1.1. Si ocurre un error al enviar: 8.b.2.1.1.1. El sistema muestra en pantalla el mensaje de error. 8.b.2.1.1.2. El sistema vuelve al paso 8 del caso de uso. 8.b.3. El actor presiona el botón "Cancelar": 8.b.3.1. El sistema vuelve al paso 8 del caso de uso.
9. El actor presiona el botón "Salir".	
10. El caso de uso finaliza aquí.	
Poscondición:	El Reporte ha sido generado.
Inclusiones:	* CU04: Consultar Campo. * CU16: Consultar Raza. * CU28: Consultar Actividad.
Extensiones:	

Figura. 7.29: CU25: Reporte de Actividad.

CU26: Generar Reporte de Operación.	
Descripción:	El actor puede emitir un reporte en el cual estén representadas todas las Operaciones registradas.
Actor(es):	* Encargado.
Precondición:	Haber ingresado al sistema.
Curso Normal	Curso Alternativo
1. El caso de uso comienza al ser invocado por el caso de uso CU24: Generar Reporte.	
2. El sistema solicita al actor que seleccione el Campo sobre el cual desea generar el reporte. Para lo cual invoca al CU04: Consultar Campo.	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Operación . Para lo cual el sistema invoca al caso de uso CU29: Consultar Operación.	
5. El actor selecciona la Operación y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema muestra en pantalla todos los registros almacenados en la base de datos para el Campo y la Operación seleccionados.	
7. El sistema solicita al actor que elija qué hacer con el Reporte de Operación . El sistema muestra en pantalla el siguiente listado de opciones: * Descargar. * Enviar por Correo.	7.a. El actor selecciona la opción "Descargar": 7.a.1. El sistema solicita al actor que seleccione el directorio en donde el Reporte será almacenado. 7.a.2. El actor selecciona el directorio y presiona el botón "Guardar". 7.a.2.1. El sistema procede a guardar el archivo. 7.a.2.1.1. Si ocurre un error al guardar: 7.a.2.1.1.1. El sistema muestra en pantalla el mensaje de error. 7.a.2.1.1.2. El sistema vuelve al paso 7 del caso de uso. 7.a.2.2. El sistema muestra en pantalla el mensaje: " Reporte guardado correctamente". 7.a.3. El actor presiona el botón "Cancelar". 7.a.3.1. El sistema vuelve al paso 7 del caso de uso. 7.b. El actor selecciona la opción "Enviar por Correo": 7.b.1. El sistema solicita al actor que ingrese las direcciones de correo electrónico de los destinatarios. 7.b.2. El actor ingresa las direcciones y presiona el botón "Enviar". 7.b.2.1. El sistema procede a enviar el correo: 7.b.2.1.1. Si ocurre un error al enviar: 7.b.2.1.1.1. El sistema muestra en pantalla el mensaje de error. 7.b.2.1.1.2. El sistema vuelve al paso 7 del caso de uso. 7.b.2.2. El sistema muestra en pantalla el mensaje: " Reporte enviado correctamente". 7.b.3. El actor presiona el botón "Cancelar". 7.b.3.1. El sistema vuelve al paso 7 del caso de uso.
9. El actor presiona el botón "Salir".	
10. El caso de uso finaliza aquí.	
Poscondición:	El Reporte ha sido generado.
Inclusiones:	* CU04: Consultar Campo * CU29: Consultar Operación.
Extensiones:	

Figura. 7.30: CU26: Reporte de Operación.

CU27: Generar Reporte de Stock.	
Descripción:	El actor puede emitir un reporte en el cual esté representado el Stock de Vacunos de un Campo .
Actor(es):	* Encargado .
Precondición:	Haber ingresado al sistema.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU24: Generar Reporte .	
2. El sistema solicita al actor que seleccione el Campo sobre el cual desea generar el reporte. Para lo cual invoca al CU04: Consultar Campo .	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla el Stock de Vacunos para el Campo seleccionado.	
5. El sistema solicita al actor que elija qué hacer con el Reporte de Sotck . El sistema muestra en pantalla el siguiente listado de opciones: * Descargar . * Enviar por Correo .	5.a. El actor selecciona la opción "Descargar": 5.a.1. El sistema solicita al actor que seleccione el directorio en donde el Reporte será almacenado. 5.a.2. El actor selecciona el directorio y presiona el botón "Guardar". 5.a.2.1. El sistema procede a guardar el archivo. 5.a.2.1.1. Si ocurre un error al guardar: 5.a.2.1.1.1. El sistema muestra en pantalla el mensaje de error. 5.a.2.1.1.2. El sistema vuelve al paso 5 del caso de uso. 5.a.2.2. El sistema muestra en pantalla el mensaje: " Reporte guardado correctamente". 5.a.3. El actor presiona el botón "Cancelar": 5.a.3.1. El sistema vuelve al paso 5 del caso de uso. 5.b. El actor selecciona la opción "Enviar por Correo": 5.b.1. El sistema solicita al actor que ingrese las direcciones de correo electrónico de los destinatarios. 5.b.2. El actor ingresa las direcciones y presiona el botón "Enviar". 5.b.2.1. El sistema procede a enviar el correo: 5.b.2.1.1. Si ocurre un error al enviar: 5.b.2.1.1.1. El sistema muestra en pantalla el mensaje de error. 5.b.2.1.1.2. El sistema vuelve al paso 5 del caso de uso. 5.b.2.2. El sistema muestra en pantalla el mensaje: " Reporte enviado correctamente". 5.b.3. El actor presiona el botón "Cancelar": 5.b.3.1. El sistema vuelve al paso 5 del caso de uso.
6. El actor presiona el botón "Salir".	
7. El caso de uso finaliza aquí.	
Poscondición:	El Reporte ha sido generado.
Inclusiones:	* CU04: Consultar Campo .
Extensiones:	

Figura. 7.31: CU27: Reporte de Stock.

CU28: Consultar Actividad.	
Descripción:	El actor puede consultar todas las Actividades que se hayan registrado en el sistema.
Actor(es):	* Encargado .
Precondición:	Debe haber almacenado al menos una Actividad en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU25: Reporte de Actividad .	
2. El sistema solicita al actor que seleccione la Actividad que desea consultar.	
3. El actor selecciona la Actividad y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada acerca de la Actividad seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada de la Actividad seleccionada.
Inclusiones:	
Extensiones:	

Figura. 7.32: CU28: Consultar Actividad.

CU29: Consultar Operación.	
Descripción:	El actor puede consultar todas las Operaciones que se hayan registrado en el sistema.
Actor(es):	* Encargado.
Precondición:	Debe haber almacenado al menos una Operación en la base de datos.
Curso Normal	Curso Alternativo
1. El caso de uso comienza al ser invocado por el caso de uso CU26: Reporte de Operación.	
2. El sistema solicita al actor que seleccione la Operación que desea consultar.	
3. El actor selecciona la Operación y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada acerca de la Operación seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada de la Operación seleccionada.
Inclusiones:	
Extensiones:	

Figura. 7.33: CU29: Consultar Operación.

CU30: Consultar Stock.	
Descripción:	El actor consultar la Cantidad de Vacunos que se encuentren en un Campo .
Actor(es):	* Administrador. * Encargado. * Veterinario.
Precondición:	El actor debe ingresar al sistema y debe haber almacenado en la base de datos al menos un Campo con Vacunos alojados en él.
Curso Normal	Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione un Campo . Para lo cual invoca al caso de uso CU04: Consultar Campo.	
3. El actor selecciona un Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que ingrese los siguientes datos: * Año : campo obligatorio de tipo numérico de 4 dígitos. * Mes : campo obligatorio de tipo numérico de 2 dígitos, con valores entre 01 y 12.	
5. El actor ingresa los datos solicitados y presiona el botón "Continuar".	5.a. Si el actor no ingresa los datos solicitados y presiona el botón "Continuar": 5.a.1. El sistema avanza al paso 6 del caso de uso asignando a los campos Año y Mes los valores correspondientes a la fecha actual. 5.b. Si el actor presiona el botón "Cancelar": 5.b.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema muestra en pantalla una planilla en la cual se encuentra, de manera detallada, los Vacunos alojados en el Campos seleccionado. Los mismos se encuentran organizados por Raza , Categoría y Subcategoría .	
7. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información suficiente para tener conocimiento del Stock de Vacunos .
Inclusiones:	* CU04: Consultar Campo.
Extensiones:	

Figura. 7.34: CU30: Consultar Stock.

CU31: Registrar Actividad.	
Descripción:	El actor puede almacenar en la base de datos la Actividad realizada sobre los Vacunos alojados en un Campo .
Actor(es):	* Veterinario .
Precondición:	Debe ingresar al sistema y debe haber almacenado en la base de datos al menos un Campo con Vacunos alojados en él. También es necesario que haya almacenada al menos una Actividad .
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione un Campo . Para lo cual invoca al caso de uso CU04: Consultar Campo .	
3. El actor selecciona un Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza . Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza .	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que seleccione la Actividad . Para lo cual invoca al caso de uso CU28: Consultar Actividad .	
7. El actor selecciona la Actividad y presiona el botón "Continuar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
9. El actor ingresa los datos solicitados y presiona el botón "Guardar".	9.a. Si el actor presiona el botón "Cancelar": 9.a.1. El sistema vuelve al paso 2 del caso de uso.
10. El sistema procede a guardar los datos ingresados por el actor.	10.a. El caso de ocurrir un error: 10.a.1. El sistema muestra en pantalla el mensaje de error. 10.a.2. El sistema vuelve al paso 2 del caso de uso.
11. El sistema muestra en pantalla el mensaje: "Actividad registrada correctamente!".	
12. El caso de uso finaliza aquí.	
Poscondición:	La Actividad ha sido almacenada.
Inclusiones:	* CU04: Consultar Campo . * CU16: Consultar Raza . * CU28: Consultar Actividad .
Extensiones:	

Figura. 7.35: CU31: Registrar Actividad.

CU32: Modificar Actividad.	
Descripción:	El actor puede modificar los atributos y propiedades de una Actividad .
Actor(es):	* Veterinario .
Precondición:	Debe haber al menos una Actividad almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Actividad que desea modificar. Para lo cual invoca al caso de uso CU28: Consultar Actividad .	
3. El actor selecciona la Actividad y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que modifique los datos de la Actividad seleccionado. Entre ellos se encuentran: * Modificar Campo . * Modificar Raza . * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	4.a. Si el actor selecciona la opción "Modificar Campo": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza . 4.b.2. El sistema vuelve al paso 4 del caso de uso. 4.c. Si el actor modifica la Cantidad , Fecha , Descripción y presiona el botón "Continuar": 4.c.1. El sistema avanza al paso 5 del caso de uso. 4.d. Si el actor presiona el botón "Cancelar": 4.d.1. El sistema vuelve al paso 2 del caso de uso.
5. El actor presiona el botón "Guardar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a guardar los cambios ingresados por el actor.	6.a. En caso de ocurrir un Error: 6.a.1. El sistema muestra en pantalla un mensaje de Error. 6.a.2. El caso de uso finaliza aquí.
8. El sistema muestra en pantalla el mensaje "Datos guardados correctamente!".	
9. El caso de uso finaliza aquí.	
Poscondición:	La información relaciona a la Actividad seleccionada ha sido actualizada.
Inclusiones:	* CU28: Consultar Actividad .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .

Figura. 7.36: CU32: Modificar Actividad.

CU33: Eliminar Actividad.	
Descripción:	El actor puede eliminar una Actividad de la base de datos.
Actor(es):	* Veterinario.
Precondición:	Debe haber al menos una Actividad almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Actividad que desea eliminar. Para lo cual invoca al caso de uso CU28: Consultar Actividad .	
3. El actor selecciona la Actividad y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos de la Actividad seleccionada.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar la Actividad .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Raza eliminada correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	La Actividad ha sido eliminada de la base de datos.
Inclusiones:	* CU28: Consultar Actividad .
Extensiones:	

Figura. 7.37: CU33: Eliminar Actividad.

CU34: Registrar Evento.	
Descripción:	El actor puede registrar un Evento . Un Evento es considerado como un "Recordatorio" referido a alguna tarea que debe ser atendida en el futuro de manera obligatoria. Un ejemplo de este tipo de tareas puede ser Realizar Tacto en aquellas Vacas a las cuales se les dio Servicio o se las Inceminó de manera Artificial.
Actor(es):	* Encargado.
Precondición:	Debe haber ingresado al sistema.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Campo en cual es necesario Registrar el Evento . Para lo cual el sistema invoca al caso de uso CU04: Consultar Campo .	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza de los Vacunos que intervienen en el Evento . Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza .	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que seleccione el Tipo de Evento que desea registrar, para lo cual el sistema muestra en pantalla las siguientes opciones: * Seleccionar Actividad. * Seleccionar Operación.	6.a. Si el actor selecciona la opción "Seleccionar Actividad": 6.a.1. El sistema invoca al caso de uso CU28: Consultar Actividad . 6.a.2. El sistema vuelve al paso 6 del caso de uso. 6.b. Si el actor selecciona la opción "Seleccionar Operación": 6.b.1. El sistema invoca al caso de uso CU29: Consultar Operación . 6.b.2. El sistema vuelve al paso 6 del caso de uso.
7. El sistema solicita al actor que ingrese los siguientes datos: * Fecha: campo obligatorio de tipo fecha. * Descripción: campo opcional de tipo alfanumérico.	
8. El actor ingresa los datos solicitados y presiona el botón "Continuar".	8.a. Si el actor presiona el botón "Cancelar": 8.a.1. El sistema vuelve al paso 2 del caso de uso.
9. El sistema solicita al actor que seleccione el modo en que el Evento será notificado a los actores interesados. El sistema muestra en pantalla las siguientes opciones: * Aviso a través del sistema. * Enviar correo electrónico.	
10. El actor selecciona el modo de notificación y presiona "Guardar".	10.a. Si el actor presiona el botón "Cancelar": 10.a.1. El sistema vuelve al paso 2 del caso de uso.
11. El sistema procede a guardar los datos ingresados por el actor.	11.a. En caso de ocurrir un error: 11.a.1. El sistema muestra en pantalla el mensaje de error. 11.a.2. El sistema vuelve al paso 2 del caso de uso.
12. El sistema muestra en pantalla el mensaje: " Evento registrado correctamente!".	
13. El caso de uso finaliza aquí.	
Poscondición:	El Evento ha sido almacenado en la base de datos.
Inclusiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .
Extensiones:	* CU28: Consultar Actividad . * CU29: Consultar Operación .

Figura. 7.38: CU34: Registrar Evento.

CU35: Notificar Evento.	
Descripción:	Al cumplirse la Fecha asignada al Evento el sistema debe Notificar a los actores, según el modo de notificación que se haya elegido al crear el Evento .
Actor(es):	* Sistema .
Precondición:	
	Curso Normal
1. El caso de uso comienza al cumplirse la Fecha configurada en el Evento .	
2. El sistema procede a comunicar a los actores interesados que se ha llegado a la Fecha de configuración.	<p>3.a. Si el modo de notificación configurado es "Aviso a Tráves del Sistema":</p> <ul style="list-style-type: none"> 3.a.1. El sistema muestra en pantalla un mensaje informando que se cumplió la Fecha configurada para el Evento. 3.a.2. El sistema avanza al paso 3 del caso de uso. <p>3.b. Si el modo de notificación configurado es "Enviar Correo Electrónico":</p> <ul style="list-style-type: none"> 3.b.1. El sistema procede a enviar un correo notificando a los actores interesados que se cumplió la Fecha configurada para el Evento. 3.b.2. El sistema avanza al paso 3 del caso de uso.
3. El caso de uso finaliza aquí.	
Poscondición:	El Evento ha sido notificado.
Inclusiones:	
Extensiones:	

Figura. 7.39: CU35: Notificar Evento.

CU36: Modificar Compra.	
Descripción:	El actor puede modificar los atributos o propiedades de una Compra .
Actor(es):	* Encargado
Precondición:	Debe haber al menos una Compra almacenada en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Compra que desea modificar. Para lo cual el sistema invoca al caso de uso CU37: Consultar Compra .	
3. El actor selecciona la Compra y presiona el botón "Continuar".	<p>3.a. Si el actor presiona el botón "Cancelar":</p> <ul style="list-style-type: none"> 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla las siguientes opciones: * Modificar Campo . * Modificar Raza .	<p>4.a. Si el actor selecciona la opción "Modificar Campo":</p> <ul style="list-style-type: none"> 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo. 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza. 4.b.2. El sistema vuelve al paso 2 del caso de uso.
5. El sistema solicita al actor que modifique los siguientes campos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
6. El actor ingresa los datos solicitados y presiona el botón "Guardar".	<p>6.a. Si el actor presiona el botón "Cancelar":</p> <ul style="list-style-type: none"> 6.a.1. El sistema vuelve al paso 2 del caso de uso.
7. El sistema procede a guardar los cambios ingresados por el actor.	<p>7.a. En caso de ocurrir un Error:</p> <ul style="list-style-type: none"> 7.a.1. El sistema muestra en pantalla un mensaje de Error. 7.a.2. El caso de uso finaliza aquí.
8. El sistema muestra en pantalla el mensaje "Datos guardados correctamente".	
9. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada a la Compra seleccionada por el actor se encuentra actualizada.
Inclusiones:	* CU37: Consultar Compra .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .

Figura. 7.40: CU36: Modificar Compra.

CU37: Consultar Compra.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Compra , la cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos una Compra almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU36: Modificar Compra * CU38: Eliminar Compra.	
2. El sistema solicita al actor que seleccione la Compra que desea consultar.	
3. El actor selecciona la Compra y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre la Compra seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada de la Compra seleccionada.
Inclusiones:	
Extensiones:	

Figura. 7.41: CU37: Consultar Compra.

CU38: Eliminar Compra.	
Descripción:	El actor puede eliminar una Compra de la base de datos.
Actor(es):	* Encargado.
Precondición:	Debe haber al menos una Compra almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Compra que desea eliminar. Para lo cual invoca al caso de uso CU37: Consultar Compra.	
3. El actor selecciona una Compra y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos de la Compra a eliminar.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar la Compra .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Compra eliminada correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	La Compra ha sido eliminada de la base de datos.
Inclusiones:	* CU37: Consultar Compra.
Extensiones:	

Figura. 7.42: CU38: Eliminar Compra.

CU39: Modificar Cambio de Categoría.	
Descripción:	El actor puede modificar los atributos o propiedades de un Cambio de Categoría .
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Cambio de Categoría almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Cambio de Categoría que desea modificar. Para lo cual el sistema invoca al caso de uso CU40: Consultar Cambio de Categoría .	
3. El actor selecciona el Cambio de Categoría y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla las siguientes opciones: * Modificar Campo . * Modificar Raza Origen . * Continuar .	4.a. Si el actor selecciona la opción "Modificar Campo": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza Origen": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza . 4.b.2. El sistema vuelve al paso 4 del caso de uso. 4.c. Si el actor selecciona la opción "Continuar": 4.c.1. El sistema avanza al paso 5 del caso de uso.
5. El sistema solicita al actor que ingrese la Cantidad de Vacunos a ser afectados.	
6. El actor ingresa la Cantidad y presiona el botón "Continuar".	6.a. Si el actor presiona el botón "Cancelar": 6.a.1. El sistema vuelve al paso 2 del caso de uso.
7. El sistema evalúa si existen suficientes Vacunos con la Raza Origen seleccionada.	7.a. Si no existen Vacunos suficientes: 7.a.1. El sistema muestra en pantalla el mensaje de error. 7.a.2. El sistema vuelve al paso 6 del caso de uso.
8. El sistema muestra en pantalla las siguientes opciones: * Modificar Raza Destino . * Continuar	8.a. Si el actor selecciona la opción "Modificar Raza Destino": 8.a.1. El sistema invoca al caso de uso CU16: Consultar Raza . 8.a.2. El sistema vuelve al paso 9 del caso de uso. 8.b. Si el actor selecciona la opción "Continuar": 8.b.1. El sistema avanza al paso 9 del caso de uso.
9. El sistema evalúa si la Raza Origen y la Raza Destino definen un Parto .	9.a. Si la Raza Origen y la Raza Destino definen un Parto : 9.a.1. El sistema invoca al caso de uso CU24: Registrar Nacimiento . 9.a.2. El sistema avanza al paso 10 del caso de uso. 9.b. Si la Raza Origen y la Raza Destino no definen un Parto : 9.b.1. El sistema avanza al paso 10 del caso de uso.
11. El sistema solicita al actor que ingrese los siguientes datos: * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
12. El actor ingresa los datos solicitados y presiona el botón "Guardar".	12.a. Si el actor presiona el botón "Cancelar": 12.a.1. El sistema vuelve al paso 2 del caso de uso.
13. El sistema procede a guardar los datos ingresados por el actor.	13.a. El caso de ocurrir un error: 13.a.1. El sistema muestra en pantalla el mensaje de error. 13.a.2. El sistema vuelve al paso 2 del caso de uso.
14. El sistema muestra en pantalla el mensaje: "Cambio de Categoría registrado correctamente".	
15. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada al Cambio de Categoría seleccionado se encuentra actualizada.
Inclusiones:	* CU40: Consultar Cambio de Categoría .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza . * CU24: Registrar Nacimiento .

Figura. 7.43: CU39: Modificar Cambio de Categoría.

CU40: Consultar Cambio de Categoría.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Cambio de Categoría , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Cambio de Categoría almacenado en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU39: Modificar Cambio de Categoría . * CU41: Eliminar Cambio de Categoría .	
2. El sistema solicita al actor que seleccione el Cambio de Categoría que desea consultar.	
3. El actor selecciona el Cambio de Categoría y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre la Cambio de Categoría seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada del Cambio de Categoría seleccionado.
Inclusiones:	
Extensiones:	

Figura. 7.44: CU40: Consultar Cambio de Categoría.

CU41: Eliminar Cambio de Categoría.	
Descripción:	El actor puede eliminar un Cambio de Categoría de la base de datos.
Actor(es):	* Encargado .
Precondición:	Debe haber al menos un Cambio de Categoría almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Cambio de Categoría que desea eliminar. Para lo cual invoca al caso de uso CU40: Consultar Cambio de Categoría .	
3. El actor selecciona un Cambio de Categoría y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos del Cambio de Categoría seleccionado.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar el Cambio de Categoría .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Cambio de Categoría eliminado correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	El Cambio de Categoría ha sido eliminado de la base de datos.
Inclusiones:	* CU40: Consultar Cambio de Categoría .
Extensiones:	

Figura. 7.45: CU41: Eliminar Cambio de Categoría.

CU42: Modificar Traslado.	
Descripción:	El actor puede modificar los atributos o propiedades de un Traslado .
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Traslado almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Traslado que desea modificar. Para lo cual el sistema invoca al caso de uso CU43: Consultar Traslado .	
3. El actor selecciona el Traslado y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla las siguientes opciones: * Modificar Campo Origen . * Modificar Raza . * Continuar .	4.a. Si el actor selecciona la opción "Modificar Campo Origen": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza . 4.b.2. El sistema vuelve al paso 4 del caso de uso. 4.c. Si el actor selecciona la opción "Continuar": 4.c.1. El sistema avanza al paso 5 del caso de uso.
5. El sistema solicita al actor que ingrese la Cantidad de Vacunos a trasladar.	
6. El actor ingresa la Cantidad y presiona el botón "Continuar".	6.a. Si el actor presiona el botón "Cancelar": 6.a.1. El sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla la siguiente opción: * Modificar Campo Destino . * Continuar .	7.a. Si el actor selecciona la opción "Modificar Campo Destino": 7.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 7.a.2. El sistema vuelve al paso 8 del caso de uso. 7.c. Si el actor selecciona la opción "Continuar": 7.c.1. El sistema avanza al paso 8 del caso de uso.
8. El sistema evalúa, según la Raza seleccionada y la Cantidad , si existen Vacunos suficientes en el Campo Origen para ser trasladados y si el Campo Destino posee espacio disponible.	8.a. En caso de ser posible el traslado: 8.a.1. El sistema avanza al paso 9 del caso de uso. 8.b. En caso de no ser posible el traslado: 8.b.1. El sistema muestra en pantalla el mensaje de error. 8.b.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema solicita al actor que ingrese los siguientes datos: * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
10. El actor ingresa los datos solicitados y presiona "Guardar".	10.a. Si el actor presiona el botón "Cancelar": 10.a.1. El sistema vuelve al paso 2 del caso de uso.
11. El sistema procede a guardar los datos ingresados por el actor.	11.a. En caso de ocurrir un error: 11.a.1. El sistema muestra en pantalla el mensaje de error. 11.a.2. El sistema vuelve al paso 2 del caso de uso.
12. El sistema muestra en pantalla el mensaje: " Traslado registrado correctamente!".	
13. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada al Traslado seleccionado por el actor se encuentra actualizada.
Inclusiones:	* CU43: Consultar Traslado .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .

Figura. 7.46: CU42: Modificar Traslado.

CU43: Consultar Traslado.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Traslado , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Traslado almacenado en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU42: Modificar Traslado. * CU44: Eliminar Traslado.	
2. El sistema solicita al actor que seleccione el Traslado que desea consultar.	
3. El actor selecciona el Traslado y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre la Traslado seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada del Traslado seleccionado.
Inclusiones:	
Extensiones:	

Figura. 7.47: CU43: Consultar Traslado.

CU44: Eliminar Traslado.	
Descripción:	El actor puede eliminar un Traslado de la base de datos.
Actor(es):	* Encargado .
Precondición:	Debe haber al menos un Traslado almacenada en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Traslado que desea eliminar. Para lo cual el sistema invoca al caso de uso CU43: Consultar Traslado .	
3. El actor selecciona el Traslado y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos del Traslado seleccionado.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar el Traslado .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Traslado eliminado correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	El Traslado ha sido eliminado de la base de datos.
Inclusiones:	CU43: Consultar Traslado.
Extensiones:	

Figura. 7.48: CU44: Eliminar Traslado.

CU45: Modificar Venta.	
Descripción:	El actor puede modificar los atributos o propiedades de una Venta .
Actor(es):	* Encargado
Precondición:	Debe haber al menos una Venta almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Venta que desea modificar. Para lo cual el sistema invoca al caso de uso CU46: Consultar Venta .	
3. El actor selecciona la Venta y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla las siguientes opciones: * Modificar Campo . * Modificar Raza . * Continuar .	4.a. Si el actor selecciona la opción "Modificar Campo": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza . 4.b.2. El sistema vuelve al paso 4 del caso de uso. 4.c. Si el actor selecciona la opción "Continuar": 4.c.1. El sistema avanza al paso 5 del caso de uso.
5. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
6. El usuario ingresa los datos solicitados y presiona el botón "Guardar".	
7. El sistema evalúa que existan Vacunos suficientes en el Campo seleccionado, según la Cantidad ingresada.	7.a. En caso de no existir Vacunos suficientes: 7.a.1. El sistema muestra en pantalla el mensaje de error. 7.a.2. El sistema vuelve al paso 2 del caso de uso.
8. El sistema procede a almacenar los datos ingresados por el actor.	8.a. El caso de ocurrir un error: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso
10. El sistema muestra en pantalla el mensaje: " Venta registrada correctamente!".	
11. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada a la Venta seleccionado por el actor se encuentra actualizada.
Inclusiones:	* CU46: Consultar Venta .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .

Figura. 7.49: CU45: Modificar Venta.

CU46: Consultar Venta.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Venta , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos una Venta almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU45: Modificar Venta . * CU47: Eliminar Venta .	
2. El sistema solicita al actor que seleccione la Venta que desea consultar.	
3. El actor selecciona la Venta y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre la Venta seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada de la Venta seleccionada.
Inclusiones:	
Extensiones:	

Figura. 7.50: CU46: Consultar Venta.

CU47: Eliminar Venta.	
Descripción:	El actor puede eliminar una Venta de la base de datos.
Actor(es):	* Encargado.
Precondición:	Debe haber al menos una Venta almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Venta que desea eliminar. Para lo cual invoca al caso de uso CU46: Consultar Venta .	
3. El actor selecciona la Venta y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos de la Venta seleccionada.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar la Venta .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: "Venta eliminada correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	La Venta ha sido eliminado de la base de datos.
Inclusiones:	* CU46: Consultar Venta .
Extensiones:	

Figura. 7.51: CU47: Eliminar Venta.

CU48: Modificar Mortandad.	
Descripción:	El actor puede modificar los atributos o propiedades de una Mortandad .
Actor(es):	* Encargado
Precondición:	Debe haber al menos una Mortandad almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Mortandad que desea modificar. Para lo cual el sistema invoca al caso de uso CU49: Consultar Mortandad .	
3. El actor selecciona la Venta y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla las siguientes opciones: * Modificar Campo . * Modificar Raza . * Continuar .	4.a. Si el actor selecciona la opción "Modificar Campo": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza . 4.b.2. El sistema vuelve al paso 4 del caso de uso. 4.c. Si el actor selecciona la opción "Continuar": 4.c.1. El sistema avanza al paso 5 del caso de uso.
5. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
6. El usuario ingresa los datos solicitados y presiona el botón "Guardar".	
7. El sistema evalúa que existan Vacunos suficientes en el Campo seleccionado, según la Cantidad ingresada.	7.a. En caso de no existir Vacunos suficientes: 7.a.1. El sistema muestra en pantalla el mensaje de error. 7.a.2. El sistema vuelve al paso 2 del caso de uso.
8. El sistema procede a almacenar los datos ingresados por el actor.	8.a. En caso de ocurrir un error: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema muestra en pantalla el mensaje: " Mortandad registrada correctamente!".	
11. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada a la Mortandad seleccionado por el actor se encuentra actualizada.
Inclusiones:	* CU49: Consultar Mortandad .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .

Figura. 7.52: CU48: Modificar Mortandad.

CU49: Consultar Mortandad.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Mortandad , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos una Mortandad almacenada en la base de datos.
Curso Normal	Curso Alternativo
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU45: Modificar Mortandad. * CU47: Eliminar Mortandad.	
2. El sistema solicita al actor que seleccione la Mortandad que desea consultar.	
3. El actor selecciona la Mortandad y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre la Mortandad seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada de la Mortandad seleccionada.
Inclusiones:	
Extensiones:	

Figura. 7.53: CU49: Consultar Mortandad.

CU50: Eliminar Mortandad.	
Descripción:	El actor puede eliminar una Mortandad de la base de datos.
Actor(es):	* Encargado .
Precondición:	Debe haber al menos una Mortandad almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione la Mortandad que desea eliminar. Para lo cual invoca al caso de uso CU49: Consultar Mortandad .	
3. El actor selecciona la Mortandad y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos de la Mortandad seleccionada.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar la Mortandad .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Mortandad eliminada correctamente."	
8. El caso de uso finaliza aquí.	
Poscondición:	La Mortandad ha sido eliminado de la base de datos.
Inclusiones:	* CU49: Consultar Mortandad .
Extensiones:	

Figura. 7.54: CU50: Eliminar Mortandad.

CU51: Modificar Nacimiento.	
Descripción:	El actor puede modificar los atributos o propiedades de un Nacimiento .
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Nacimiento almacenada en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Nacimiento que desea modificar. Para lo cual el sistema invoca al caso de uso CU52: Consultar Nacimiento .	
3. El actor selecciona el Nacimiento y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad de Terneros Machos : campo obligatorio de tipo numérico. * Cantidad de Terneros Hembras : campo obligatorio de tipo numérico.	
5. El actor ingresa las Cantidades y presiona el botón "Guardar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
6. El sistema procede a almacenar los datos ingresados por el actor.	4.a. En caso de ocurrir un error: 4.a.1. El sistema muestra en pantalla el mensaje de error. 4.a.2. El sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Nacimiento registrada correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada al Nacimiento seleccionado por el actor se encuentra actualizada.
Inclusiones:	* CU52: Consultar Nacimiento .
Extensiones:	

Figura. 7.55: CU51: Modificar Nacimiento.

CU52: Consultar Nacimiento.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Nacimiento , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Nacimiento almacenada en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU51: Modificar Nacimiento. * CU53: Eliminar Nacimiento.	
2. El sistema solicita al actor que seleccione el Nacimiento que desea consultar.	
3. El actor selecciona el Nacimiento y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre el Nacimiento seleccionada.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada del Nacimiento seleccionado.
Inclusiones:	
Extensiones:	

Figura. 7.56: CU52: Consultar Nacimiento.

CU53: Eliminar Nacimiento.	
Descripción:	El actor puede eliminar un Nacimiento de la base de datos.
Actor(es):	* Encargado .
Precondición:	Debe haber al menos un Nacimiento almacenada en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Nacimiento que desea eliminar. Para lo cual invoca al caso de uso CU52: Consultar Nacimiento.	
3. El actor selecciona el Nacimiento y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos del Nacimiento seleccionado.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar el Nacimiento .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: "Nacimiento eliminado correctamente!".	
8. El caso de uso finaliza aquí.	
Poscondición:	El Nacimiento ha sido eliminado de la base de datos.
Inclusiones:	* CU52: Consultar Nacimiento.
Extensiones:	

Figura. 7.57: CU53: Eliminar Nacimiento.

CU54: Registrar Aborto.	
Descripción:	El actor puede registrar un Aborto en la base de datos.
Actor(es):	* Encargado
Precondición:	Debe haber Vacas que por algún motivo hayan perdido la Cría que estaban gestando.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU17: Registrar Operación .	
2. El sistema solicita al actor que seleccione el Campo en cual ha tenido lugar el Aborto . Para lo cual invoca al caso de uso CU04: Consultar Campo .	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza de las Vacas que hayan Abortado . Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza .	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanuméricico.	
7. El actor ingresa los datos solicitados y presiona el botón "Guardar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema procede a guardar los datos ingresados por el actor.	8.a. El caso de ocurrir un error: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema muestra en pantalla el mensaje: " Aborto registrada correctamente!".	
10. El caso de uso finaliza aquí.	
Poscondición:	El Aborto se encuentra almacenado en la base de datos.
Inclusiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .
Extensiones:	

Figura. 7.58: CU54: Registrar Aborto.

CU55: Modificar Aborto.	
Descripción:	El actor puede modificar los atributos o propiedades de un Aborto .
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Aborto almacenado en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Aborto que desea modificar. Para lo cual el sistema invoca al caso de uso CU56: Consultar Aborto .	
3. El actor selecciona el Aborto y presiona el botón "Continuar".	3.a Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla las siguientes opciones: * Modificar Campo . * Modificar Raza . * Continuar .	4.a. Si el actor selecciona la opción "Modificar Campo": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza . 4.c. Si el actor selecciona la opción "Continuar": 4.c.1. El sistema avanza al paso 5 del caso de uso.
5. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanuméricico.	
6. El usuario ingresa los datos solicitados y presiona el botón "Guardar".	
7. El sistema procede a almacenar los datos ingresados por el actor.	7.a. El caso de ocurrir un error: 7.a.1. El sistema muestra en pantalla el mensaje de error. 7.a.2. El sistema vuelve al paso 2 del caso de uso.
8. El sistema muestra en pantalla el mensaje: " Aborto registrada correctamente!".	
9. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada al Aborto seleccionado por el actor se encuentra actualizada.
Inclusiones:	* CU56: Consultar Aborto .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .

Figura. 7.59: CU55: Modificar Aborto.

CU56: Consultar Aborto.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Aborto , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Aborto almacenada en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU55: Modificar Aborto. * CU57: Eliminar Aborto.	
2. El sistema solicita al actor que seleccione el Aborto que desea consultar.	
3. El actor selecciona el Aborto y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre el Aborto seleccionado.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada del Aborto seleccionado.
Inclusiones:	
Extensiones:	

Figura. 7.60: CU56: Consultar Aborto.

CU57: Eliminar Aborto.	
Descripción:	El actor puede eliminar un Aborto de la base de datos.
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Aborto almacenado en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Aborto que desea eliminar. Para lo cual invoca al caso de uso CU56: Consultar Aborto.	
3. El actor selecciona el Aborto y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos del Aborto seleccionado.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar el Aborto .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: " Aborto eliminada correctamente."	
8. El caso de uso finaliza aquí.	
Poscondición:	El Aborto ha sido eliminado de la base de datos.
Inclusiones:	* CU56: Consultar Aborto.
Extensiones:	

Figura. 7.61: CU57: Eliminar Aborto.

CU58: Registrar Destete.	
Descripción:	El actor puede registrar un Cambio de Categoría de Terneros Machos o Hembras alojados en un Campo en la base de datos.
Actor(es):	* Encargado
Precondición:	Los Terneros que van a ser destetados deben contar con las aptitudes físicas adecuadas para poder alimentarse por si solos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el caso de uso CU17: Registrar Operación .	
2. El sistema solicita al actor que seleccione el Campo en cual ha tenido lugar el Aborto . Para lo cual invoca al CU04: Consultar Campo .	
3. El actor selecciona el Campo y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema solicita al actor que seleccione la Raza de las Terneros que estén en condiciones de ser separados de las madres. Para lo cual el sistema invoca al caso de uso CU16: Consultar Raza .	
5. El actor selecciona la Raza y presiona el botón "Continuar".	5.a. Si el actor presiona el botón "Cancelar": 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
7. El actor ingresa los datos solicitados y presiona el botón "Guardar".	7.a. Si el actor presiona el botón "Cancelar": 7.a.1. El sistema vuelve al paso 2 del caso de uso.
8. El sistema procede a guardar los datos ingresados por el actor.	8.a. El caso de ocurrir un error: 8.a.1. El sistema muestra en pantalla el mensaje de error. 8.a.2. El sistema vuelve al paso 2 del caso de uso.
9. El sistema muestra en pantalla el mensaje: "Aborto registrada correctamente!".	
10. El caso de uso finaliza aquí.	
Poscondición:	El Destete se encuentra almacenado en la base de datos.
Inclusiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .
Extensiones:	

Figura. 7.62: CU58: Registrar Destete.

CU59: Modificar Destete.	
Descripción:	El actor puede modificar los atributos o propiedades de un Destete .
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Destete almacenado en la base de datos.
Curso Normal	
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Destete que desea modificar. Para lo cual el sistema invoca al caso de uso CU60: Consultar Destete .	
3. El actor selecciona el Destete y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra en pantalla las siguientes opciones: * Modificar Campo . * Modificar Raza . * Continuar .	4.a. Si el actor selecciona la opción "Modificar Campo": 4.a.1. El sistema invoca al caso de uso CU04: Consultar Campo . 4.a.2. El sistema vuelve al paso 4 del caso de uso. 4.b. Si el actor selecciona la opción "Modificar Raza": 4.b.1. El sistema invoca al caso de uso CU16: Consultar Raza . 4.c. Si el actor selecciona la opción "Continuar": 4.c.1. El sistema avanza al paso 5 del caso de uso.
5. El sistema solicita al actor que ingrese los siguientes datos: * Cantidad : campo obligatorio de tipo numérico. * Fecha : campo obligatorio de tipo fecha. * Descripción : campo opcional de tipo alfanumérico.	
6. El usuario ingresa los datos solicitados y presiona el botón "Guardar".	
7. El sistema procede a almacenar los datos ingresados por el actor.	7.a. El caso de ocurrir un error: 7.a.1. El sistema muestra en pantalla el mensaje de error. 7.a.2. El sistema vuelve al paso 2 del caso de uso.
8. El sistema muestra en pantalla el mensaje: " Destete registrada correctamente!".	
9. El caso de uso finaliza aquí.	
Poscondición:	La información relacionada al Destete seleccionado por el actor se encuentra actualizada.
Inclusiones:	* CU60: Consultar Destete .
Extensiones:	* CU04: Consultar Campo . * CU16: Consultar Raza .

Figura. 7.63: CU59: Modificar Destete.

CU60: Consultar Destete.	
Descripción:	El actor podrá consultar y/o acceder a la información almacenada de cada Destete , el cual formará parte de una de las posibles acciones que dicho actor puede realizar.
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Destete almacenada en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor o por los casos de uso: * CU59: Modificar Destete. * CU61: Eliminar Destete.	
2. El sistema solicita al actor que seleccione el Destete que desea consultar.	
3. El actor selecciona el Destete y presiona el botón "Continuar".	
4. El sistema muestra en pantalla toda la información almacenada sobre el Destete seleccionado.	
5. El caso de uso finaliza aquí.	
Poscondición:	El actor cuenta con la información detallada del Destete seleccionado.
Inclusiones:	
Extensiones:	

Figura. 7.64: CU60: Consultar Destete.

CU61: Eliminar Destete.	
Descripción:	El actor puede eliminar un Destete de la base de datos.
Actor(es):	* Encargado
Precondición:	Debe haber al menos un Destete almacenado en la base de datos.
	Curso Normal
1. El caso de uso comienza al ser invocado por el actor.	
2. El sistema solicita al actor que seleccione el Destete que desea eliminar. Para lo cual invoca al caso de uso CU60: Consultar Destete.	
3. El actor selecciona el Destete y presiona el botón "Continuar".	3.a. Si el actor presiona el botón "Cancelar": 3.a.1. El caso de uso finaliza aquí.
4. El sistema muestra los datos del Destete seleccionado.	
5. El actor presiona el botón "Eliminar".	5.a. El actor presiona el botón "Cancelar". 5.a.1. El sistema vuelve al paso 2 del caso de uso.
6. El sistema procede a eliminar el Destete .	6.a. En caso de ocurrir un error: 6.a.1. El sistema muestra un mensaje de error. 6.a.2. El caso sistema vuelve al paso 2 del caso de uso.
7. El sistema muestra en pantalla el mensaje: "Destete eliminada correctamente."	
8. El caso de uso finaliza aquí.	
Poscondición:	El Destete ha sido eliminado de la base de datos.
Inclusiones:	* CU60: Consultar Destete.
Extensiones:	

Figura. 7.65: CU61: Eliminar Destete.