



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DESARROLLO DEL PROTOTIPO DE UN SISTEMA DE ADMINISTRACIÓN DEL GANADO  
BOVINO DE LA GRANJA EXPERIMENTAL DE LA UNIVERSIDAD DE LAS AMÉRICAS

Trabajo de Titulación presentado en conformidad con los requisitos establecidos  
para optar por el título de Ingeniera en Sistemas de Computación e Informática

Profesor Guía  
Msc. Juan José León Guerrero

Autora  
Sindy Viveka Vera Cedeño

Año  
2016

## **DECLARACIÓN DEL PROFESOR GUÍA**

“Declaro haber dirigido este trabajo a través de reuniones periódicas con la estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

.....  
Juan José León Guerrero  
MASTER SPECIALISE EN INGENIERIE DES MEDIAS NUMERIQUES  
CI.: 1707506760

### **DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE**

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

.....  
Sindy Viveka Vera Cedeño  
CI.: 1310478514

## **AGRADECIMIENTOS**

Agradezco infinitamente a mi Dios por ser mi guía y mi ejemplo. A mi esposo y mi hijo por ser mi fuerza y mi razón de vivir. A mis padres por darme los recursos para enfrentar retos. A los maestros y a Juan José por transmitirme sus conocimientos. Gracias por apoyarme con sus palabras de motivación, tiempo y amor.

## **DEDICATORIA**

Este proyecto está dedicado a mi Dios, por ser mi guía. A Ángel, Roberto y mi Familia, por ser mi fuerza y apoyo incondicional, a ustedes va este logro.

“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa.”

(Mahatma Gandhi)

## RESUMEN

El Departamento de Administración de la Granja de la Universidad de las Américas tiene la necesidad de automatizar todos sus registros de animales, debido a esto se desarrolló un prototipo de sistema denominado **Farmware**, bajo la plataforma .NET y se utilizaron tecnologías como HTML5.

**Farmware** es una aplicación web que permite administrar y automatizar eficientemente la información que posee la granja experimental de la UDLA, mejorando el tiempo de ingreso de datos e integridad de los mismos.

Este prototipo será utilizado en el departamento de Administración de la Granja. El fin de esta aplicación es garantizar que los datos se encuentren disponibles en cualquier momento; permitiendo la toma rápida de decisiones al momento de analizar los resultados de sus reportes.

## **ABSTRACT**

The Department of Farm Management at the University of the Americas is the need to automate all your animal records, because this prototype system called Farmware developed under the .NET platform and used technologies like HTML5.

Farmware is a web application that manages and efficiently automates the information at the experimental farm of the UDLA, improving data entry time and integrity of the same.

This prototype will be used in the department of farm management. The purpose of this application is to ensure that data is available at any time; allowing quick decisions when analyzing the results of their reports.

.

# ÍNDICE

1. Introducción.....	1
1.1 Antecedentes.....	1
1.2 Marco Referencial .....	2
1.3 Alcance.....	2
1.4 Justificación .....	3
1.5 Objetivo General.....	3
1.6 Objetivos específicos.....	4
1.7 Metodología a utilizar.....	4
2. Metodologías.....	5
2.1 Metodologías de Desarrollo de Software .....	5
2.1.1 Metodologías ágiles vs metodologías tradicionales.....	6
2.2 Metodología Scrum .....	7
2.2.1 Roles de Scrum .....	8
2.2.2 Roles Auxiliares .....	8
2.2.3 Reuniones de Scrum .....	8
2.2.4 Documentación de Scrum .....	10
2.2.5 Herramientas de Scrum.....	11
2.3 Metodología Extreme Programming.....	13
2.3.1 Historias de usuarios .....	14
2.3.2 Roles de XP.....	14
2.3.3 Procesos de XP.....	15
2.3.4 Prácticas en XP .....	17
2.3.5 Cuadro comparativo entre Scrum y XP .....	18
2.3.6 Metodología Ágil seleccionada para el Proyecto de titulación. ....	20
2.4 Diagrama Entidad-Relación .....	21
2.4.1 Entidad .....	21
2.4.2 Atributo .....	22
2.4.3 Clave primaria.....	22



2.4.4	Clave foránea .....	22
2.4.5	Relaciones de cardinalidad.....	22
2.4.6	Clave primaria.....	24
2.4.7	Clave foránea .....	25
2.5	Diagramas UML.....	25
2.5.1	Tipos de diagramas UML:.....	25
2.6	Diagrama de clase .....	26
2.6.1	Clases.....	27
2.6.2	Asociaciones.....	27
2.6.3	Generalizaciones .....	28
2.7	Diagramas de casos de uso.....	29
2.7.1	Actores .....	29
2.7.2	Caso de uso .....	30
2.7.3	Relaciones.....	30
2.8	Diagrama de secuencia .....	31
2.8.1	Línea de Vida.....	32
2.8.2	Mensaje .....	33
2.8.3	Ocurrencia en ejecución .....	34
2.8.4	Mensaje self.....	34
2.8.5	Mensajes perdidos y encontrados .....	35
2.8.6	Inicio y final de línea de vida.....	36
2.9	Diagrama de actividades.....	37
2.9.1	Actividades .....	38
2.9.2	Acciones .....	38
2.9.3	Restricciones de Acción .....	38
2.9.4	Flujo de control .....	39
2.9.5	Nodo inicial .....	39
2.9.6	Nodo final .....	40
2.9.7	Flujos de objetos y objeto .....	41
2.9.8	Nodos de decisión y combinación .....	42
2.9.9	Partición.....	42

2.10	Diagrama de despliegue.....	43
2.10.1	Nodo.....	43
2.10.2	Artefacto.....	44
2.10.3	Asociaciones.....	45
3.	Tecnologías.....	46
3.1	Visual Studio 2012.....	46
3.2	Sql Server 2008.....	46
3.3	Modelo Vista Controlador.....	47
3.4	Versiones de Microsoft MVC.....	50
3.5	MVC4.....	52
3.6	Diseño Web Sensible.....	52
3.7	Hyper Text Markup Language 5.....	55
3.8	Hojas de estil en cascada nivel 3.....	55
3.9	Lenguaje de definición de informe del lado del cliente.....	56
3.10	Diseño Web.....	56
3.10.1	Ventajas.....	57
3.10.2	Desventajas.....	58
3.10.3	Tipos.....	58
3.10.4	Elementos.....	60
3.11	Lenguaje Javascript.....	63
4.	Análisis.....	65
4.1	Requerimientos del sistema.....	65
4.2	Planificación de Sprints y sus historias de usuarios.....	66
4.3	Diseño de la aplicación.....	77
4.3.1	Diagrama de clases.....	77
4.3.2	Diagrama Entidad - Relación.....	79
4.3.3	Diagrama de secuencia.....	80
4.3.4	Diagrama de actividades.....	82
5.	Conclusiones y recomendaciones.....	84

5.1. Conclusiones.....	84
5.2 Recomendaciones .....	85
REFERENCIAS .....	86
ANEXOS.....	88

## ÍNDICE DE TABLAS

Tabla 1. Comparación entre la Metodología de desarrollo ágil Scrum y XP. ...	19
Tabla 2. Comparación de las versiones de ASP.NET MVC. ....	51
Tabla 3. Cuadro comparativo entre el DWA y el DWS. ....	54
Tabla 4. Instalación del entorno de desarrollo.....	68
Tabla 5. Modelado del diagrama Entidad – Relación.....	68
Tabla 6. Modelado del diagrama Físico.....	68
Tabla 7. Funciones CRUD.....	69
Tabla 8. Módulo de gestión alimenticia.....	69
Tabla 9. Formulario crear nuevos pesajes.....	70
Tabla 10. Reporte animales pesados.....	70
Tabla 11. Módulo de gestión de manejo.....	70
Tabla 12. Formulario crear animal.....	71
Tabla 13. Reporte animales creados.....	71
Tabla 14. Módulo de gestión de reproducción.....	72
Tabla 15. Formulario registro de parto.....	72
Tabla 16. Formulario registro de nuevo servicio.....	72
Tabla 17. Reporte de servicios y partos creados.....	73
Tabla 18. Módulo gestión de producción.....	73
Tabla 19. Formulario crear control lechero.....	73
Tabla 20. Formulario crear control lechero por animal.....	74
Tabla 21. Reporte detalle ordeñada.....	74
Tabla 22. Módulo de gestión de ingresos y egresos.....	75
Tabla 23. Formulario registro de ingreso por actividad.....	75
Tabla 24. Formulario registro de egreso por actividad.....	75
Tabla 25. Módulo de resumen.....	76
Tabla 26. Reporte de producción.....	76
Tabla 27. Reporte número de partos por animal.....	76
Tabla 28. Reporte promedio histórico de producción por rodeo.....	77

## ÍNDICE DE FIGURAS

Figura 1. Cuadro explicativo del ciclo de vida de la Metodología Scrum.....	10
Figura 2. Análisis del seguimiento de un proyecto utilizando Burn-up. ....	11
Figura 3. Trabajo pendiente del producto o proyecto. ....	12
Figura 4. Análisis de las horas pendiente en cada iteración con Burn-down. ..	13
Figura 5. Ejemplo de cardinalidad uno a uno. ....	23
Figura 6. Ejemplo de relación de cardinalidad uno a muchos. ....	24
Figura 7. Ejemplo de relación de muchos a muchos. ....	24
Figura 8. Representación gráfica del diagrama de clases.....	26
Figura 9. Representación de la clase rectángulo. ....	27
Figura 10. Representación de asociación. ....	28
Figura 11. Representación de generalizaciones. ....	28
Figura 12. Ejemplo gráfico de un Diagrama de casos de uso. ....	29
Figura 13. Representación de un actor. ....	30
Figura 14. Representación de un caso de uso. ....	30
Figura 15. Representación de la relación asociación. ....	30
Figura 16. Representación de la relación de dependencia. ....	31
Figura 17. Representación de la relación de generalización. ....	31
Figura 18. Representación de la línea de vida. ....	32
Figura 19. Representación de los elementos de los casos de uso.....	33
Figura 20. Representación de un mensaje.....	34
Figura 21. Representación de mensaje self. ....	35
Figura 22. Representación de mensajes perdidos y encontrados.....	36
Figura 23. Representación del inicio y final de línea de vida.....	36
Figura 24. Representación del diagrama de actividad. ....	37
Figura 25. Representación de una actividad. ....	38
Figura 26. Representación de acción.....	38
Figura 27. Representación de restricciones de acción.....	39
Figura 28. Representación de flujo de control. ....	39
Figura 29. Representación de nodo inicial. ....	40
Figura 30. Representación de nodo final de actividad.....	40
Figura 31. Representación de nodo final de flujo. ....	41

Figura 32. Representación de objeto. ....	41
Figura 33. Representación de un flujo de objeto. ....	41
Figura 34. Representación de nodo de decisión y combinación. ....	42
Figura 35. Representación de partición.....	43
Figura 36. Representación de nodo. ....	44
Figura 37. Representación de artefacto. ....	44
Figura 38. Representación de asociación. ....	45
Figura 39. Representación del patrón MVC. ....	47
Figura 40. Representación del flujo de control del patrón MVC. ....	50
Figura 41. Representación de diseño web sensible. ....	53
Figura 42. Representación de css3.....	56
Figura 43. Representación de un wireframe.....	57
Figura 44. Representación de wireframe de baja fidelidad. ....	59
Figura 45. Representación de wireframe de alta fidelidad. ....	60
Figura 46. Representación del diseño de información. ....	61
Figura 47. Representación de diseño de navegación. ....	62
Figura 48. Representación de diseño de interfaz. ....	63
Figura 49. Representación diagrama de clases de Farmware. ....	78
Figura 50. Representación del modelo Entidad - Relación de la aplicación. ....	79
Figura 51. Representación del diagrama de secuencia login administrador. ...	80
Figura 52. Representación de la administración del módulo de producción. ....	81
Figura 53. Representación del diagrama del login del administrador.....	82
Figura 54. Representación del diagrama de actividad listar un animal pesado.	83

## CAPÍTULO I

### Introducción

#### 1.1 Antecedentes

La granja de la Universidad de las Américas fue adquirida en septiembre de 2012, su ubicación geográfica es en la parroquia de Nono, a 2.6 km al noroccidente de pichincha y su extensión es de 46 hectáreas.

La actividad principal es la producción de leche de vaca, la que se vende a la Empresa Nono Lácteos Cia. Ltda., actualmente tiene con 41 reses.

Uno de los beneficios que brinda la granja a los estudiantes de agroindustria es que ellos trabajan con la leche, procesándola y convirtiéndola en productos elaborados como: queso, manjar, mantequilla. Por otra parte los alumnos de veterinaria realizan los estudios y controles médicos de cada animal.

Además, cuenta con un proyecto de vinculación con la comunidad de Nono, su objetivo es recolectar información sobre la cosecha que realiza su gente, y dentro de la granja crear líneas de producción para beneficiar a este sector poblacional.

La granja presenta mayor desarrollo en: registro del ganado, producción diaria de leche, reproducción, peso, entre otros.

Este crecimiento ha saturado el registro de la información, verbigracia: datos de los animales desde el momento que este nace, la fecha que fue llevado a la granja adicionalmente cuando una hembra está preñada, se lleva un control del estado de gestación del animal, medio que permite obtener información precisa para controlar al animal.

En la actualidad no se cuenta con un sistema de gestión que permita administrar la Información del ganado bovino que existe en esta; los encargados de la administración llevan registros manuales de los animales.

## 1.2 Marco Referencial

En la actualidad la Universidad maneja en su totalidad el proceso de gestión de la granja mediante elementos físicos, tales como carpetas y hojas electrónicas, asistido por el administrador de la misma. Con la implementación de estos módulos se brindará un medio por la cual se dará seguimiento a la información, proporcionando datos reales y rapidez en la obtención de estos.

## 1.3 Alcance

El Alcance del proyecto de titulación es el desarrollo de un prototipo de sistema de administración del ganado bovino de la granja experimental de la Universidad de las Américas que permita llevar una hoja de vida de cada animal existente con toda su trazabilidad o ciclo de vida. El prototipo de sistema incluirá los módulos:

### ✓ **Producción de leche**

Este módulo realizara el registro del Control lechero y el control por ordeñada de cada animal; de esta manera se conocerá cuantos litros el animal produce.

### ✓ **Manejo de ganado**

En el módulo se hará el registro de datos de un nuevo animal: como su nombre, fecha de nacimiento, los comentarios acerca del animal, el nombre de la madre y del progenitor; tendremos una vista donde se mostrarán los animales creados con sus características.

### ✓ **Gestión de Reproducción**

En este módulo se maneja todo lo referente a preñez, número de partos, fecha de parto, sexo de la cría y además permite agregar un servicio.

### ✓ **Gestión Alimenticia**

En este módulo se registrara peso, ganancia de cada animal joven y la fecha del pesaje.



### ✓ **Ingresos y Egresos**

En este módulo se registra los gastos e ingresos de la venta de producción de leche.

### ✓ **Resumen**

En el módulo se podrán obtener a manera de resumen los siguientes reportes: Número de partos por animal, litros de leche ordeñados por animal y promedio histórico de producción por rodeo. El sistema mostrará un listado de la información.

Con la finalidad de mejorar la producción de los ganados, se permitirá realizar registros de los animales, obteniendo un listado con información de cada animal y generación de informes con resúmenes de índices productivos, entre otros.

El proyecto de titulación será desarrollado sobre la plataforma de .NET bajo el framework MVC4.

## **1.4 Justificación**

En la actualidad muchas granjas han decidido incorporar sistemas de administración para tener una mayor gestión y control de la misma, la propuesta es desarrollar un prototipo de sistema de administración del ganado bovino que permita que todo el proceso manual sea automatizado y se gestione de manera sencilla y dinámica toda la operatoria del ganado de la granja; brindando al usuario información al instante que facilite el control y la toma oportuna de decisiones.

## **1.5 Objetivo General**

Realizar un prototipo de un sistema de administración de ganado bovino para la granja experimental de la Universidad de las Américas, que permita la gestión de su ganadería.

## 1.6 Objetivos específicos

- ✓ Tener control total del ganado vacuno de la granja.
- ✓ Desarrollar una aplicación en un entorno amigable de tal manera que cualquier usuario pueda utilizar el sistema sin ninguna complicación.
- ✓ Crear un sistema que permita obtener una descripción del estado en el que se encuentra el ganado bovino.
- ✓ Administrar eficientemente sus registros y actividades de reses permitiendo ahorrar tiempo.

## 1.7 Metodología a utilizar

La metodología a utilizar para la construcción de un prototipo de sistema de administración del ganado bovino de la granja de la Universidad de las Américas es la experimental.

Los procedimientos y variables identificados para la elaboración del trabajo de titulación son:

- ✓ Levantamiento de información, proporcionada por el departamento de administración de la granja.
- ✓ Análisis de Requerimientos del sistema.
- ✓ Elaboración de diagramas de caso de uso.
- ✓ Desarrollo de la aplicación.

La experimentación hará factible la identificación y comprensión de las variables que se van evaluar en la construcción del software y estas variables identificadas son: calidad e integridad.

## CAPÍTULO II

### Metodologías

#### 2.1 Metodologías de Desarrollo de Software

En la Ingeniería de Software las metodologías de desarrollo son herramientas, métodos y modelos utilizados para planificar, controlar y analizar los procesos de desarrollo de sistemas.

El objetivo de estas metodologías es construir y desarrollar sistemas de calidad.

Existen varios enfoques de cada una de las metodologías para el desarrollo de sistemas:

##### ✓ **Modelo en cascada**

En el modelo en cascada el proyecto está dividido en fases y su enfoque semeja a una cascada de agua ya que los pasos son visualizados hacia abajo. Este modelo se basa en el análisis, diseño, implementación y pruebas; mantiene un control a través de la documentación escrita de cada una de las fases.

##### ✓ **Incremental**

Este modelo se va desarrollando a medida; este sistema es evolutivo y se enfoca en controlar los riesgos que se presentan en cada incremento.

##### ✓ **Prototipado**

Este sistema está orientado a desarrollos de prototipos.

##### ✓ **Espiral**

Permite evaluar riesgos durante todo el ciclo del proyecto; este sistema se divide en fragmentos pequeños para garantizar la facilidad de cambio durante el desarrollo del proceso.

Existen dos modelos de metodologías de desarrollo:

- ✓ Metodologías tradicionales o formales.
- ✓ Metodologías ágiles o informales.

### **2.1.1 Metodologías ágiles vs metodologías tradicionales**

Una de las principales ventajas de la metodología ágil es que el equipo está sujeto a los cambios que surgen durante el periodo de vida del sistema; a diferencia de una metodología tradicional la cual no es flexible y es muy resistente a cambios.

En una metodología informal el cliente forma parte primordial del grupo de desarrolladores; en las formales o tradicionales el cliente puede intervenir o involucrarse con el equipo de trabajo solo en el momento de las reuniones.

Las metodologías ágiles son aplicadas en proyectos en los cuales se tiene mucha inseguridad e incertidumbre ya que el ambiente es variable porque no se conoce con seguridad que requisitos exactos son los que se necesitan; en cambio en las tradicionales es indispensable que el cliente tome decisiones al momento que empieza el proyecto.

El equipo de trabajo en las metodologías ágiles es pequeño por lo general todos trabajan en el mismo lugar a diferencia de una metodología tradicional que posee un grupo de trabajo grande y los mismos se encuentran distribuidos en diferentes plazas. Existen menos roles de trabajo en una metodología informal que en una metodología formal.

Se tomaron en cuenta todas las consideraciones expuestas en la comparación. Se realizó una evaluación entre las metodologías descritas y se decidió optar por la Metodología ágil por los siguientes puntos descritos a continuación:

- ✓ Los requerimientos del sistema están sujetos a cambios en lo que dure el ciclo de vida del proyecto.
- ✓ Se estima concluir con el sistema en un periodo corto de tiempo.
- ✓ El equipo de trabajo es pequeño.

En este proyecto se escogerán dos metodologías Ágiles para ser analizadas:

- ✓ Metodología SCRUM.
- ✓ Metodología XP.

Se presenta una descripción de las metodologías para evaluar cuál es la más conveniente para la realización del prototipo de sistema.

## 2.2 Metodología Scrum

Es la metodología ágil basada en la gestión de técnicas iterativas e incrementales; esta metodología posee ciertos aspectos de los modelos o procesos en cascada, prototipados y espiral.

Surge de un artículo en el que se mencionan las mejores prácticas utilizadas por empresas japonesas y también toma referencias del juego de rugby por su concepto de trabajo en equipo. En el año de 1995 se formalizó el proceso denominado Scrum.

Scrum es una de las metodologías Ágiles más utilizada en la actualidad para el proceso de desarrollo de software, una de las características de esta metodología es la flexibilidad, calidad en cada entregable y satisfacción en el grupo de desarrollo, ya que posee un equipo multidisciplinario. La principal ventaja es que se adapta fácilmente a los cambios que surgen en durante el ciclo de vida del proyecto.

Al iniciar un proyecto se debe definir la pila del producto (*product backlog*) el cual contiene los requerimientos funcionales y no funcionales que el sistema necesita para su construcción. Los accionistas (*stakeholders*) se encargaran de concretar el product backlog durante las reuniones de planificación y a partir de esto se definirán las iteraciones denominadas sprint; el tiempo de duración de cada Sprint es de cuatro semanas.

El Facilitador o líder (*Scrum Master*) es el encargado dentro de cada Sprint de convocar diariamente a una reunión (*daily scrum*) la cual no dura más de

quince minutos; el propósito es mostrar una retroalimentación de todas las dificultades que se presentaron al realizar cada actividad.

Al concluir el Sprint se realiza una revisión (*sprint review*) donde se planea el siguiente Sprint.

### 2.2.1 Roles de Scrum

Se deben identificar claramente las responsabilidades que intervienen en esta metodología con la finalidad que el equipo de trabajo se pueda guiar en el desarrollo del producto.

Los roles principales roles que se manejan son:

**Scrum Master:** Es el encargado de hacer que los procesos de la metodología Scrum se realicen correctamente. Él ejecuta su trabajo a la par del Product Owner.

**Product Owner:** Es el que representa a los Stakeholders, él escribe las historias de usuario, las categoriza y las registra dentro del Product Backlog.

**Team:** Es el equipo de trabajo; son los responsables de entregar el producto. El equipo tiene carácter multidisciplinario se encargan del Análisis, diseño, desarrollo, pruebas y documentación para el desarrollo del sistema.

### 2.2.2 Roles Auxiliares

**Stakeholders:** Son los clientes, los proveedores y los vendedores; son los encargados de observar y asesorar; solo participan en las revisiones del Sprint.

**Users:** Son las personas que van a manejar el sistema.

### 2.2.3 Reuniones de Scrum

Diariamente se realiza una reunión de cada Sprint sobre el estado en el que se encuentra el proyecto.

**Daily Scrum:** Denominadas reunión diaria, los participantes son: El Scrum Manager y el equipo de trabajo estas reuniones no duran más de quince

minutos y deben realizarse siempre en el mismo lugar y a una hora determinada en donde cada integrante responde las siguientes preguntas:

- ✓ ¿Qué has empezado a hacer desde ayer?
- ✓ ¿Qué es lo que estás planeando realizar hoy?
- ✓ ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

Al finalizar la Daily Scrum se actualiza el Sprint Backlog.

**Sprint Planning Meeting:** En esta reunión se realiza la planificación del Sprint, los participantes son: El Product Owner, el Scrum Master y el equipo de trabajo; el tiempo de duración es de una jornada. El Product Owner se encarga de resolver las inquietudes que posea el equipo y se fija una fecha para la realización del Sprint.

**Sprint Review:** En esta reunión participa el equipo, el Scrum Master, el Product Owner y las demás personas involucradas en el proyecto. El tiempo de duración de este sprint review es de 4 horas. El principal objetivo es presentar al dueño del producto y al resto de personas implicadas en el sistema los nuevos cambios funcionales que se implementaron.

**Sprint Retrospective:** Los integrantes de esta reunión son el Team y el Scrum Master; en esta reunión se tiene que responder las siguientes preguntas:

- ✓ ¿Qué procesos funcionaron correctamente en el último Sprint?
- ✓ ¿Qué procesos se pueden mejorar?

El Scrum Master se encarga de anotar todas las respuestas referentes a estas dos preguntas y las mejoras son registradas en el siguiente Sprint del Product Backlog.

**Sprint:** Dentro de esta metodología se definen reuniones entre un periodo de una a cuatro semanas estas son denominadas sprint para ajustar los tiempo de trabajo del equipo; al final se debe mostrar un adelanto del producto para ser un entregable al cliente.

## 2.2.4 Documentación de Scrum

En esta metodología después de las reuniones se genera una lista de requisitos que se transforman en tareas y actividades para ser desarrolladas por los integrantes del grupo, estas listas son documentos dinámicos que pueden variar según las exigencias de cada uno de los clientes.

**Product Backlog:** Son los requisitos o necesidades del sistema desde la perspectiva del cliente. Es la lista de usuario que contiene todas las tareas a ser resueltas, el responsable de esta lista se denomina Dueño del Producto o Product Owner.

**Sprint Backlog:** En este documento se detalla como el equipo de trabajo va a implementar los requisitos del sprint. Estas tareas son tomadas por cualquiera de los integrantes del equipo y no pueden durar más de 16 horas.

En la figura 1 se mostrara de manera gráfica como intervienen todos los elementos de Scrum durante el ciclo de vida del proyecto.

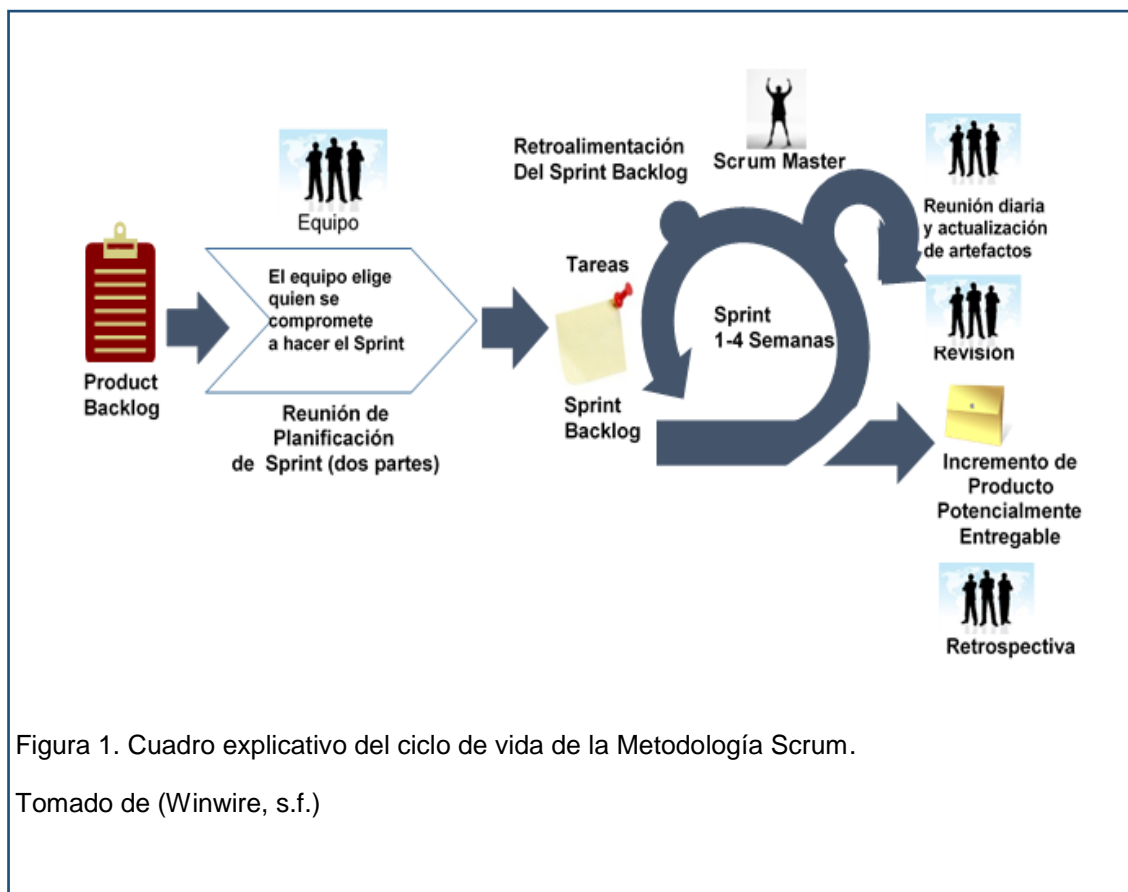


Figura 1. Cuadro explicativo del ciclo de vida de la Metodología Scrum.

Tomado de (Winwire, s.f.)



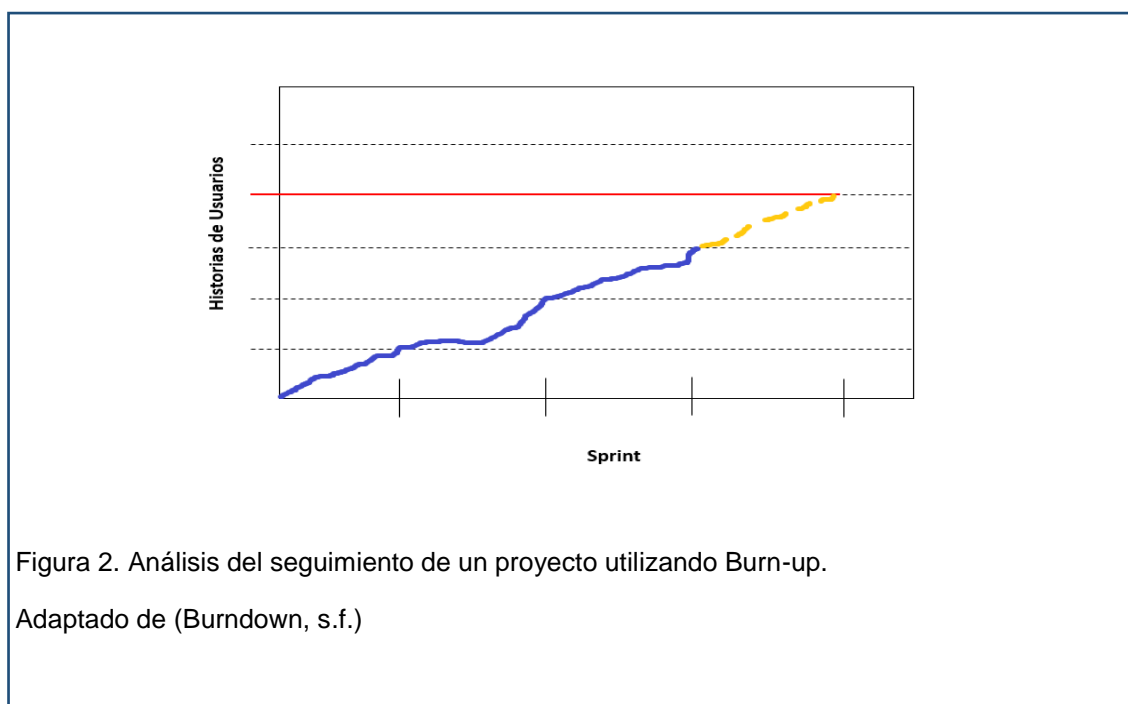
### 2.2.5 Herramientas de Scrum.

Los gráficos Burn-up y Burn-down son herramientas claves para dar seguimiento y ver el progreso del proyecto, permitiendo hacer un cálculo del trabajo completado en el tiempo.

En el eje vertical se cuantifica las historias de usuarios y en el eje horizontal se miden los sprints.

**Burn-up chart:** Esta herramienta es utilizada por el dueño del proyecto para dar seguimiento y ver cómo avanza el proyecto. La línea parte desde el cero y se va marcando la cantidad de trabajo completado de manera ascendente.

En la figura 2 se muestra un gráfico de burn-up, donde el eje Y representa las historias de usuarios y el eje X el tiempo. La línea azul indica que se ha completado el sprint 3 y la línea roja indica el objetivo.



**Burn-Down chart:** Esta herramienta representa el trabajo pendiente del proyecto a lo largo del tiempo. Permite mostrar la rapidez en la que se están completando los requisitos del proyecto y concluir si el equipo podrá culminar el trabajo en el tiempo definido.

Se representa dibujando una línea, que conecta los puntos de todos los Sprints que han sido completados; de esta manera se determina el avance del mismo. La línea deberá ser descendente en el caso de que todo fluya bien y al llegar al eje horizontal el proyecto habrá finalizado. Si existen nuevos requisitos la pendiente deberá ascender hasta que se concluyan.

Existen dos tipos de burndown chart:

- ✓ **Días pendientes:** Permite completar los requisitos del proyecto, se lo realiza de la lista de requisitos del Product Backlog.

Este gráfico muestra el avance y culminación de los requisitos de cada sprint, la línea va descendiendo y se ha completado el sprint 4.

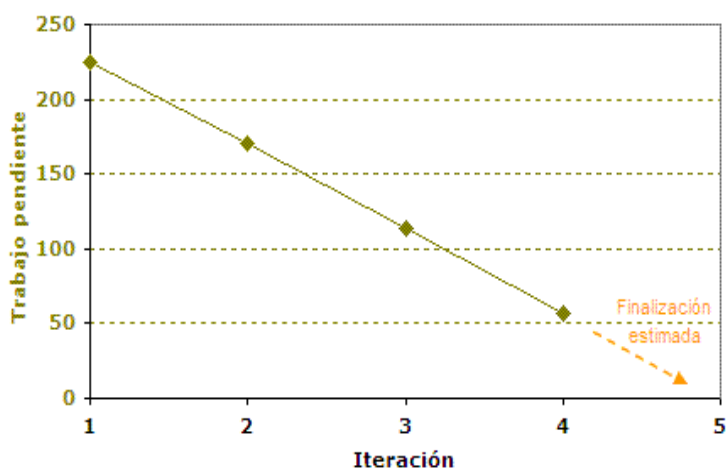
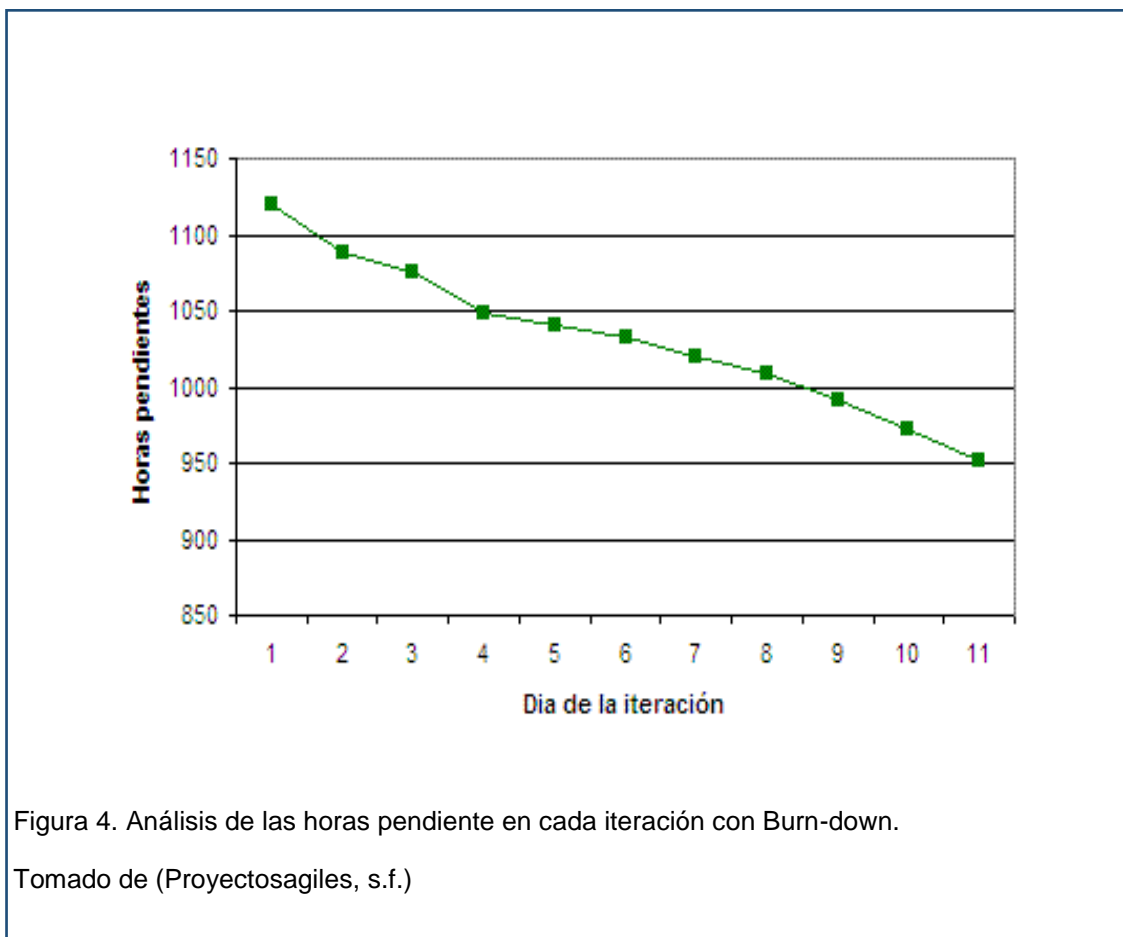


Figura 3. Trabajo pendiente del producto o proyecto.

Tomado de (Proyectosagiles, s.f.)

- ✓ **Horas pendientes:** Permite completar las tareas del sprint, se lo realiza de la lista de tareas del Sprint.

El gráfico muestra en el eje vertical las horas pendientes de la iteración y en el eje horizontal tenemos los días del sprint, además se observa que han sido completadas las horas de los días de iteración.



### 2.3 Metodología Extreme Programming

La metodología ágil XP creada por Kent Beck, es una de las primeras metodologías ágiles existentes y de las más destacadas ya que posee un nivel alto de disciplina entre las personas involucradas en el proyecto.

Esta metodología está orientada a promover el trabajo en equipo, también se interesa en el constante aprendizaje del grupo de desarrollo. Una de las principales características de XP es que se basa en la retroalimentación constante entre los desarrolladores y los clientes permitiendo una comunicación íntegra con el grupo de trabajo.

XP proporciona mayor énfasis a la adaptabilidad en el diseño del software. Esta metodología supone que los cambios presentados en los requisitos son normales y necesarios ya que todo proyecto debe ser flexible al cambio.

En esta metodología sus integrantes se encuentran dispuestos a enfrentar los cambios que surjan durante todo el camino, garantizando un buen ambiente laboral y el éxito en el desarrollo del sistema.

Las características principales de XP son:

- ✓ Historias de usuarios
- ✓ Roles
- ✓ Procesos
- ✓ Prácticas

### **2.3.1 Historias de usuarios**

Son utilizadas para especificar los requerimientos del software. En esta lista el cliente registra los requerimientos funcionales o no funcionales que el sistema debe tener.

La historia de usuario debe poseer los requisitos claros y esto va a tener un tiempo para que los desarrolladores puedan realizarlas. Según Beck cada historia de usuario debe registrar la fecha, la descripción de la actividad, el número de historia de usuario, la prioridad, el estado de las actividades y los comentarios.

Las historias de usuarios pueden ser plantificadas de una a cuatro semanas y este tiempo no puede ser superado, cada tarea es asignada a los desarrolladores para que esta se resuelva durante cada iteración.

### **2.3.2 Roles de XP**

Los roles que intervienen en XP son:

**Programador:** Es el encargado de producir el código del sistema, también realiza las pruebas unitarias.

**Cliente:** Es el que se encarga de escribir las historias de usuario y de darles prioridad a cada actividad para sea realizada en cada iteración; también realiza las pruebas de funcionamiento para validar que se implemente el sistema.

**Encargado de pruebas:** También conocido como tester, es el encargado de ejecutar las pruebas y de entregar los resultados al equipo.

**Entrenador:** También conocido como coach, es un experto en la metodología XP y el responsable del proceso, y de proveer las guías a los miembros del equipo de tal manera que se apliquen las practicas XP correctamente.

**Consultor:** Es una persona externa al equipo; es especialista en algún tema necesario para el proyecto y se encarga de ayudar al grupo de trabajo cuando surgen problemas.

**Gestor:** También conocido como big boss, es el enlace entre los clientes y el equipo de desarrolladores; se encarga de coordinar que el grupo trabaje eficientemente.

### 2.3.3 Procesos de XP

El ciclo de desarrollo del proyecto está definido por los siguientes pasos:

1. Se define el proyecto a implementar por parte del cliente.
2. Se estima el tiempo y esfuerzo para el desarrollo del proyecto por parte del programador.
3. El cliente especifica las prioridades y se calculan los tiempos.
4. El desarrollador construye el proyecto.
5. Se vuelve al primer paso después de cada iteración.

En cada iteración del ciclo de vida del proyecto el cliente y el desarrollador están en constante aprendizaje. Para asegurar la calidad del software y los plazos establecidos, no se debe cortar los tiempos programados ni aumentar el número de tareas asignadas a los programadores.

El ciclo de vida ideal de la metodología XP consiste de las siguientes seis fases:

## **1. Exploración**

En esta primera fase, los clientes plantean las historias de usuario que son importantes para la entrega del primer producto, por otro lado los programadores se familiarizan con las tecnologías, herramientas y las prácticas que se serán utilizadas en el proyecto.

## **2. Planificación de la Entrega**

En esta fase el cliente define la prioridad de cada historia de usuario y los desarrolladores hacen una estimación del esfuerzo que se requiere en cada una de ellas. Adicional se define el contenido de la entrega del producto, una entrega no puede tardar más de tres meses.

## **3. Iteraciones**

Esta fase contiene varias iteraciones del proyecto antes de ser entregado al cliente. El plan de entrega está compuesto por varias iteraciones que no deben durar más de tres semanas, en la primera iteración se forma la arquitectura del sistema que va a ser utilizada durante el tiempo restante del mismo y al finalizar la última iteración el sistema estará listo para entrar a la etapa de producción.

## **4. Producción**

En esta fase se requiere de pruebas extras y revisiones del rendimiento del producto antes de que el sistema sea entregado al ambiente del cliente. En la fase de producción pueden surgir cambios y aparecer nuevas características para la versión actual del proyecto; todas estas sugerencias son documentadas para implementarlas posteriormente.

## **5. Mantenimiento**

El proyecto XP debe seguir en funcionamiento mientras la primera versión del sistema se encuentra en producción. En esta fase se puede solicitar la presencia de más personas dentro del equipo para ayudar en los cambios y realizar tareas de soporte.

## 6. Muerte del Proyecto

En esta fase se realiza la documentación final del proyecto ya que no existen más historias de usuario para ser incluidas dentro del sistema. La muerte del proyecto también se genera cuando no existe el presupuesto para mantenerlo.

### 2.3.4 Prácticas en XP

La metodología XP se basa en un conjunto de doce importantes prácticas, las cuales se deben seguir responsablemente para el desarrollo evolutivo del proyecto; estas son:

**Juego de planificación:** Se realizan las historias de usuario y se estima el tiempo requerido para su implementación. Los clientes definen el tiempo de cada entregable y el periodo estimado de cada iteración.

**Diseño Simple:** El código del sistema debe ser sencillo, al momento que se encuentre una complejidad en el mismo será descartada.

**Pequeñas Versiones:** Realizar versiones sencillas que sean utilizables y que no posean toda la funcionalidad del sistema. Estas mini versiones se deben desarrollar en periodos cortos de tiempo y luego colocarlas en producción.

**Pruebas:** Se deben realizar pruebas unitarias, las cuales no pueden contener errores y son ejecutadas cada vez que se realice una modificación en el sistema.

**Refactorización:** Se realiza una reestructuración constante del código con la finalidad de eliminar duplicación de código, mejorarlo y hacerlo más flexible para los cambios posteriores, sin afectar el comportamiento del sistema ya programado.

**Programación por parejas:** El desarrollo del sistema se lo debe realizar en pares en un mismo ordenador, con la finalidad de reducir los errores y de efectuar un mejor diseño del sistema y mejorando su calidad.

**Propiedad colectiva del código:** El desarrollador realizará cambios dentro del código cuando lo crea necesario.

**Integración continua:** Cada vez que se tenga listo un nuevo bloque de código se ira integrando continuamente al sistema.

**Cliente en el lugar de desarrollo:** El cliente debe estar disponible y presente para el grupo de desarrollo; con el cliente presente se pueden responder todas las dudas de los programadores de manera inmediata.

**Estándares de codificación:** Se mantienen los principios de programación por parte del equipo de desarrollo, de tal forma que sea fácil la comprensión del código y permita realizar los cambios.

**Metáfora:** Se deben escoger nombres o frases que definan el funcionamiento de las diferentes partes del programa, de manera que con el nombre se pueda hacer una idea de que es lo que realiza cada parte del sistema. Las metáforas facilitan a cualquier usuario comprender el objeto del programa.

**Cuarenta horas por semana:** Es recomendable que se trabaje un máximo de cuarenta horas por semana. No se aconseja que los programadores trabajen horas fuera de tiempo; si se detecta esto, está sucediendo un problema en el equipo y debe solucionarse realizando el juego de la planificación para cambiar la fecha de entrega del proyecto.

### **2.3.5 Cuadro comparativo entre Scrum y XP**

A continuación se muestra un cuadro comparativo entre las dos metodologías ágiles seleccionadas, se detallan las semejanzas y diferencias que poseen cada una de ellas.



Tabla 1. Comparación entre la Metodología de desarrollo ágil Scrum y XP.

<b>SEMEJANZAS</b>	
Las dos metodologías ágiles utilizan historias de usuarios.	
En periodos cortos de tiempo se realizan continuamente entregables al cliente.	
Se realizan reuniones cortas e informales entre los miembros del equipo.	
<b>DIFERENCIAS</b>	
<b>SCRUM</b>	<b>XP</b>
Las iteraciones de entrega son de 2 a 4 semanas y se denominan Sprint.	Las iteraciones de entrega son más rápidas de 1 a 3 semanas.
Scrum es una metodología informal basada en la administración de proyectos.	XP es una metodología de desarrollo ágil centrada en la creación del sistema o programación.
Cada integrante del Scrum Team trabaja individualmente.	Los miembros de un proyecto de XP realizan la programación en parejas.
Al culminar un sprint, las tareas que se han terminado del Sprint Backlog y que ya han sido aprobadas por el Product Owner no se vuelven a revisar.	Las tareas que se van culminando en las entregas al cliente son vulnerables a cambios durante el tiempo de vida del proyecto, inclusive después de que el sistema funcione bien.
El equipo de trabajo de Scrum trata de cumplir con el orden de prioridad del Product Owner en el Sprint Backlog, pero si identifican que es necesario modificar el orden de prioridad para el desarrollo de las tareas del proyecto, lo pueden hacer.	El equipo de desarrollo debe seguir de manera estricta el orden de prioridad de las tareas definidas por el cliente, ellos son los que disponen en el proyecto.

Tomado de (Proyectalis, s.f.)

Nota: Similitudes y diferencias entre la Metodología ágil Scrum y XP.

### **2.3.6 Metodología Ágil seleccionada para el Proyecto de titulación**

Para el desarrollo del prototipo de Sistema de administración del ganado bovino de la UDLA se ha escogido la metodología ágil SCRUM, debido a que su modelo se acopla de mejor manera al alcance y objetivos del proyecto.

#### **Razones por las que se ha seleccionado SCRUM:**

- ✓ Utilizada para equipos pequeños de desarrollo.
- ✓ Flexibilidad frente a un entorno de requisitos cambiantes a lo largo del proyecto.
- ✓ Es posible ajustar la funcionalidad del proyecto en base a la necesidad del cliente.
- ✓ Permite entregar un módulo funcional al finalizar cada Sprint.
- ✓ Fomenta el trabajo en equipo, concentrando sus esfuerzos en alcanzar un objetivo común.
- ✓ Promueve la comunicación entre los distintos miembros del equipo de trabajo.
- ✓ Calidad del software, al tener una versión funcional después de cada iteración ayuda a mejorar el sistema.
- ✓ Reducción de riesgos, el ordenar por prioridad y conocer la velocidad con la que el equipo avanza permite despejar riesgos
- ✓ Predicción de tiempos, con los sprints se puede estimar el tiempo en el que se dispondrá una determinada funcionalidad del sistema.

#### **Razones por las no se escogió XP:**

- ✓ Es aconsejable utilizarla en proyectos cortos.
- ✓ Representa altas comisiones en caso de fallos.
- ✓ Imposible predecir costos y tiempo de desarrollo.
- ✓ Si se utilizan diagramas uml, debido al constante cambio del código estos tienden a estar desactualizados.
- ✓ El equipo programa en pareja el proyecto.
- ✓ El equipo sigue de manera estricta el orden de prioridad de las tareas definidas por el cliente.

## **Definición de SCRUM en el Proyecto:**

Como primer paso se identifican los roles de la metodología:

- ✓ **Cliente (Product Owner).** Administrador de la Granja.
- ✓ **Facilitador (Scrum Master).** Profesor Guía.
- ✓ **Equipo (Team).** Alumno

En el segundo paso definiremos los artefactos:

Se definen las historias de usuario (Product Backlog) con el cliente, las cuales se escriben en un lenguaje no técnico y no son tan detalladas. Después se especificara nuestro Sprint Backlog, documento que contiene la lista de tareas y las horas de trabajo que toma el realizar cada una de ellas.

Se detalla el orden de prioridad de las tareas del Sprint Backlog en nuestro primer Sprint, este ciclo se repetirá hasta que todos los elementos de nuestro Backlog hayan sido entregados al cliente y el avance del desarrollo del prototipo será controlado a través del Burn Down chart.

## **2.4 Diagrama Entidad-Relación**

También conocido por sus siglas E-R que en ingles se traduce “Entity relationship”. Es una herramienta utilizada para el modelamiento de datos, que representa las diferentes entidades de un sistema y como se relacionan esos objetos entre sí. Para realizar un modelado de base de datos es necesario:

### **2.4.1 Entidad**

Representa un objeto o cosa existente en el mundo real y que puede diferenciarse de los demás debido a sus elementos o características.

#### **Ejemplos:**

- ✓ Empresa (Se diferencia por su nombre, dirección)
- ✓ Libro (Se diferencia por su título, autor, genero)
- ✓ Usuario (Se diferencia por su identificación, nombre)
- ✓ Estudiante (Se deferencia por su matrícula, años)

### 2.4.2 Atributo

Son características o propiedades que definen una entidad. El diseñador es el encargado de escoger las características más relevantes ya que estas pueden ser muchas. Cada entidad posee valores determinados para cada uno de sus atributos, de tal forma que permite su identificación única.

#### Ejemplos:

Entidad: Alumno

Atributos:

- (1, Sindy, 27 años, 108075)
- (2, Roberto, 18 años, 108825)
- (3, Ángel, 25 años, 119652)

### 2.4.3 Clave primaria

Permite identificar de manera única un solo atributo, esta no permite que se repita el atributo en la misma entidad.

#### Ejemplo:

El número de matrícula de un carro o el número de chasis; en la entidad carro no pueden existir dos veces el mismo.

### 2.4.4 Clave foránea

Es la clave primaria de otra entidad con la cual se relaciona hacia otra tabla.

### 2.4.5 Relaciones de cardinalidad

Es la asociación entre sí de muchas entidades. Las relaciones tienen nombres. Son representadas en el diagrama de entidad relación por rombos y fechas. Cada entidad interviene en una relación con una cardinalidad. La cardinalidad es el número de elementos de una entidad, estas pueden asociarse a un elemento de la otra entidad relacionada.

**Ejemplo:**

Relación entre estudiante y materia.

**Tipos de relaciones de cardinalidad****✓ Relación uno a uno. (1:1)**

Cada registro de la tabla X solo tiene un registro coincidente en la tabla Y y viceversa.

En la figura 5 se muestra un ejemplo de relación de cardinalidad uno a uno, representado por una línea y en cada extremo se encuentra el número 1.

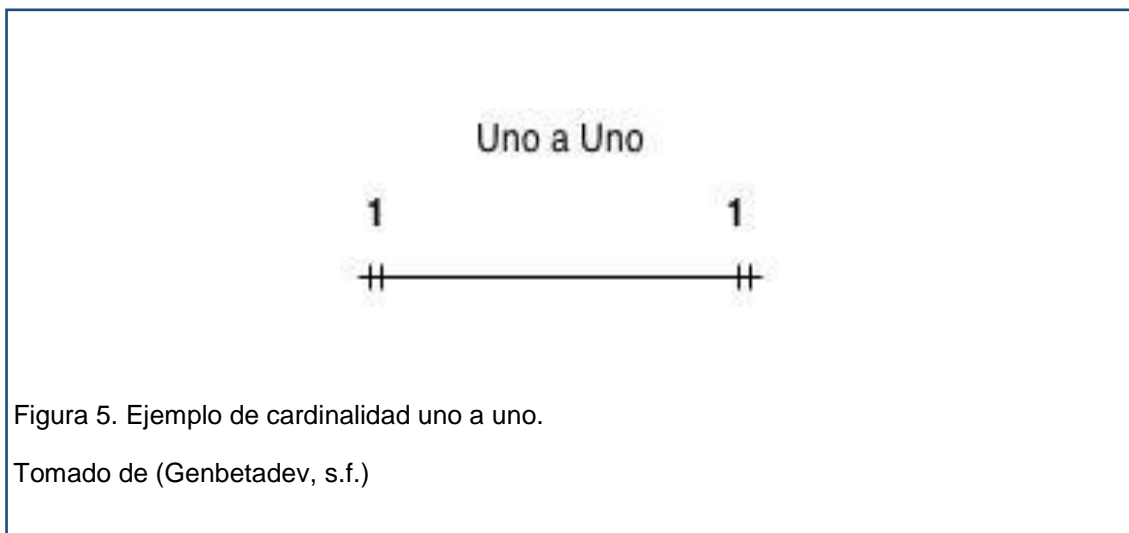


Figura 5. Ejemplo de cardinalidad uno a uno.

Tomado de (Genbetadev, s.f.)

**✓ Relación uno a muchos. (1:N)**

Un registro de la tabla X puede tener varios registros que coinciden en la Tabla Y, pero en la tabla Y solo un registro coincide con la tabla X.

**Ejemplo:**

Un ejemplo es la relación entre la tabla cliente y la tabla pedidos.

En la figura 6 se muestra la relación de uno a muchos, representada por una línea recta, en el extremo izquierdo de la relación uno se escribe el número 1 y en el extremo derecho la relación muchos se denotan con la letra N.

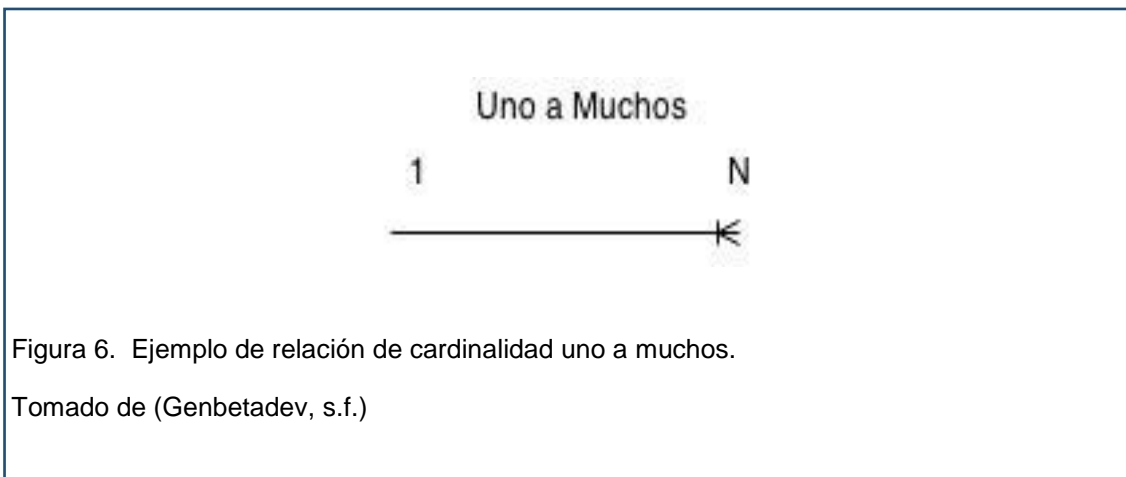


Figura 6. Ejemplo de relación de cardinalidad uno a muchos.

Tomado de (Genbetadev, s.f.)

#### ✓ Relación muchos a muchos. (N:N)

Un registro de la tabla X poder coincidir con muchos registros de la tabla Y y viceversa.

En la figura 7 se muestra la relación de cardinalidad de muchos a muchos, representada por una línea y en sus dos extremos la letra N.

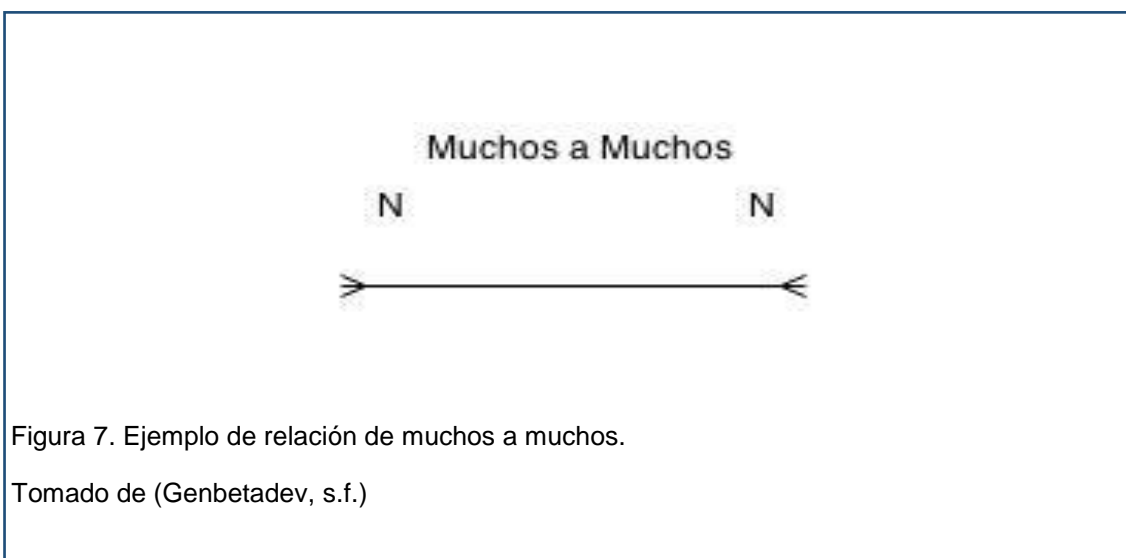


Figura 7. Ejemplo de relación de muchos a muchos.

Tomado de (Genbetadev, s.f.)

### 2.4.6 Clave primaria

Es el atributo considerado como clave para poder identificar otros atributos que representan a una entidad.

### 2.4.7 Clave foránea

Es cuando existe una relación entre dos entidades, depende mucho del tipo de relación. En este caso la clave primaria de una de ellas es colocada en la segunda con el nombre de clave externa.

## 2.5 Diagramas UML

UML, sus siglas significan Lenguaje Unificado de Modelado. Es el lenguaje de modelado de software más utilizado y conocido actualmente; nos ayuda a construir, visualizar, especificar y documentar un sistema de software orientado a objetos.

Su finalidad es mostrar diferentes perspectivas de un sistema, conocidas como modelo. El modelo es la representación resumida del software; el modelo describe gráficamente lo que hará el sistema, pero no explica cómo se lo debe implementar.

### 2.5.1 Tipos de diagramas UML:

**Diagramas de comportamiento:** Nos permite identificar el comportamiento de un sistema o de los procesos una organización. Estos diagramas son empleados para construir, especificar, visualizar y documentar aspectos como el flujo de mensajes en el tiempo y el movimiento de los componentes dentro de una red, por ejemplo:

- ✓ Diagrama de actividad.
- ✓ Diagrama de casos de uso.

**Diagramas de interacción:** Nos ayudan a identificar las interacciones entre un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Representa la forma como se comunican entre sí un actor y una clase en petición a un evento. Los diagramas de interacción muestran la vista dinámica de un sistema de software, tal como:

- ✓ Diagrama de secuencia.

**Diagramas de estructura:** Es un diagrama que nos permite establecer las jerarquías entre los módulos de un programa y las estructuras de programación llevadas a cabo para que funcionen estos módulos, por ejemplo:

- ✓ Diagrama de despliegue.

## 2.6 Diagrama de clase

El diagrama de clase describe la vista estática del modelo de un sistema orientado a objetos, definiendo sus atributos y métodos.

Los diagramas de clases sirven para ilustrar las relaciones entre interfaces y clases de un sistema.

En la figura 8 se representa el diagrama de clases y la relación que existe entre cada una de ellas, en él constan cinco clases: La clase persona, empleado, directivo y cliente, en cada una de sus clases se muestran los atributos y los métodos que posee cada clase; por ejemplo en la clase empleado vemos los atributos privados nombre y edad y el método público mostrar().

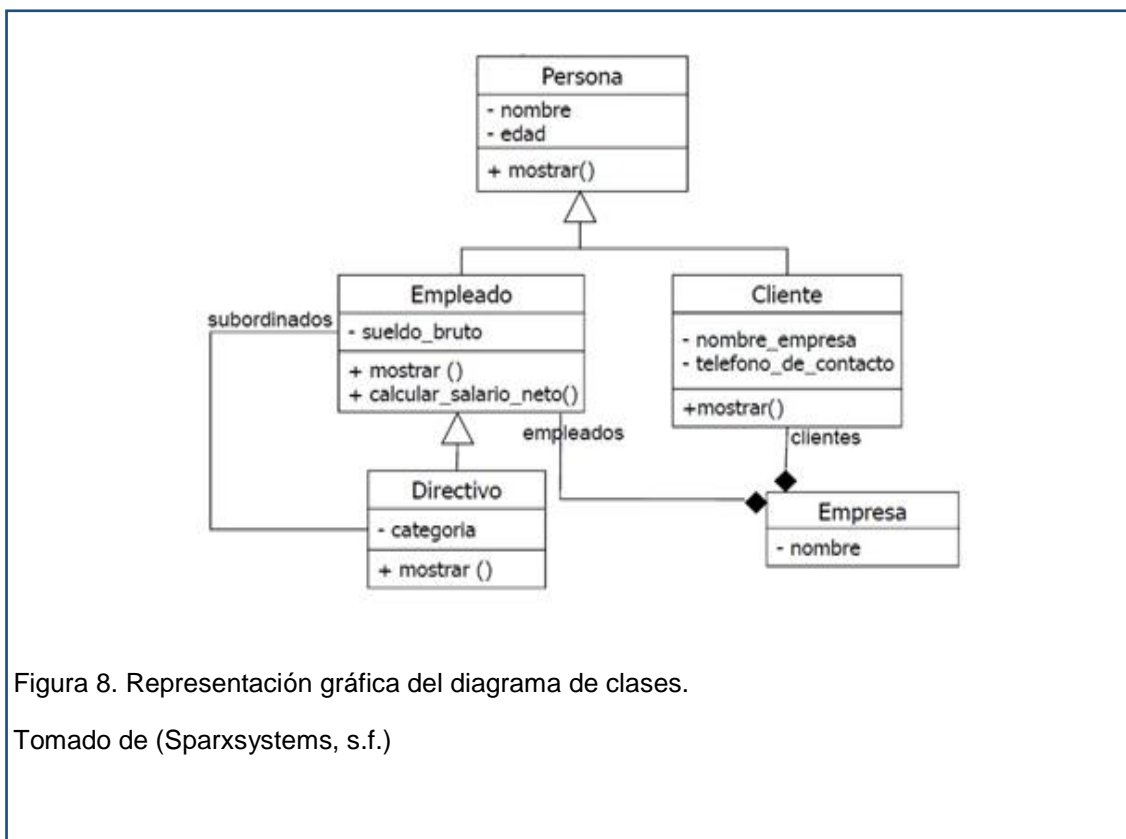


Figura 8. Representación gráfica del diagrama de clases.

Tomado de (Sparxsystems, s.f.)

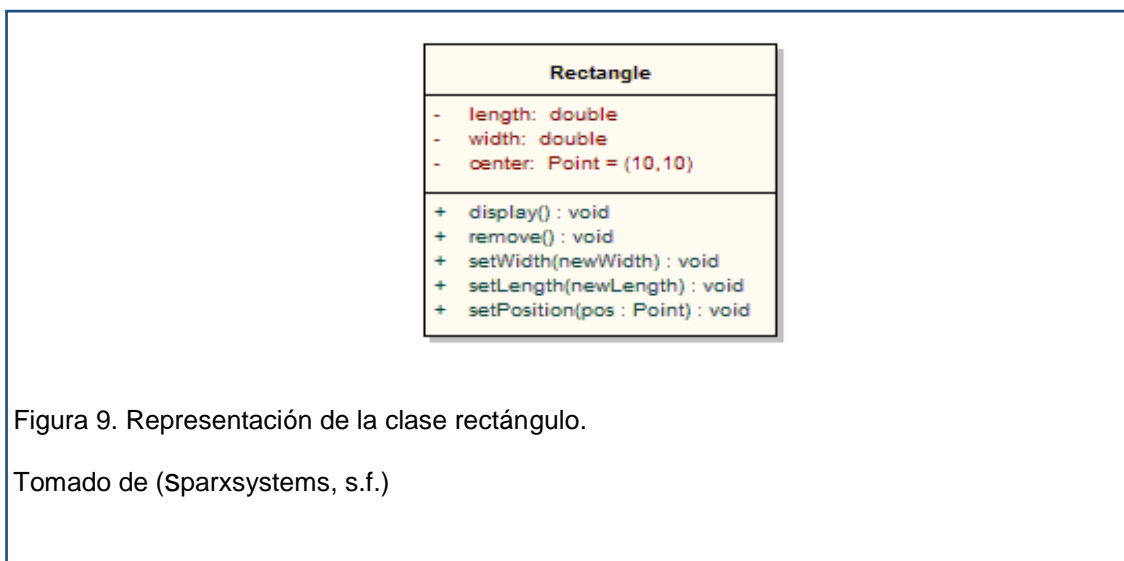


### 2.6.1 Clases

La clase es un elemento que especifica los atributos y métodos que un objeto puede generar.

La notación de la clase es a través de un rectángulo, esta se divide en tres secciones: en la primera parte se define el nombre de la clase, en la segunda sección se encuentran los atributos y en la tercera parte se especifican las funciones o métodos de la clase.

En la figura 9 vemos un ejemplo de la clase rectángulo, en ella se detallan los atributos length y width con tipo de dato double y el atributo center el cual ya muestra valores iniciales. En la última sección se muestran las operaciones setWidth, setLength y setPosition con sus parámetros correspondientes.

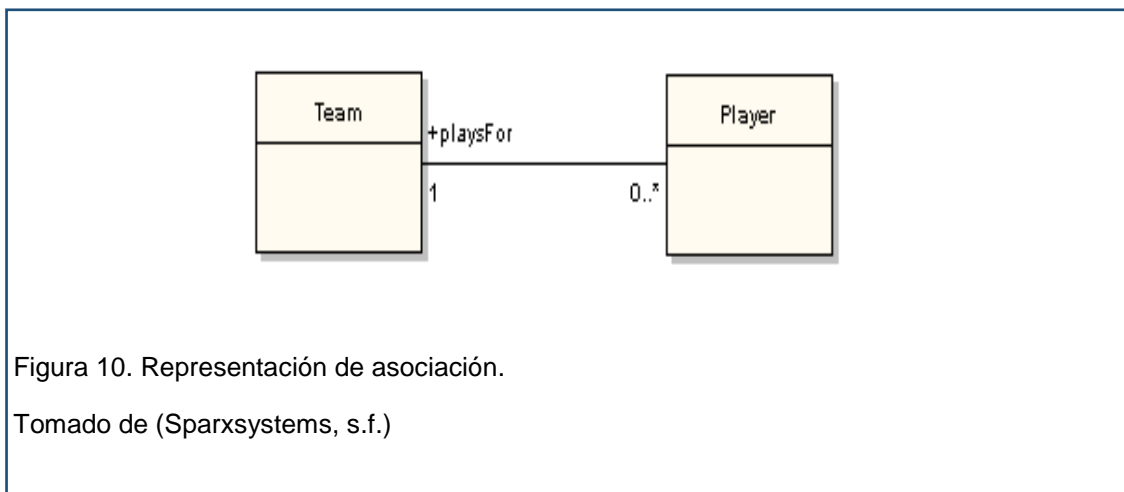


La notación que precede el atributo o método muestra la visibilidad del elemento, si usa (+) tiene visibilidad pública, si tiene el símbolo (-) el atributo o método es privado y el (#) indica que es atributo o método es protegido.

### 2.6.2 Asociaciones

Una asociación define que dos elementos del modelo tienen una relación. Este conector incluye en cada extremo la dirección y cardinalidad.

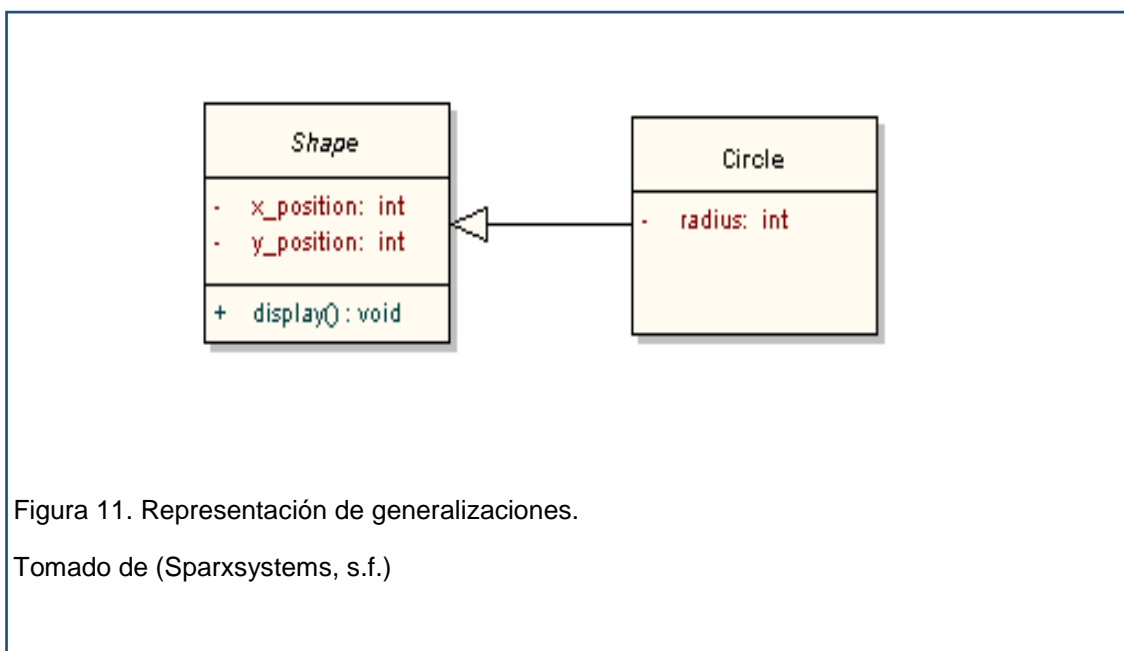
En la figura 9 podemos observar una representación de asociación entre los elementos Team y Player, describiendo que un player pertenece a un team.



### 2.6.3 Generalizaciones

Una generalización es un elemento usado para mostrar herencia. La definición de una generalización es que el origen hereda las características del destino.

En el ejemplo de la figura 11 tenemos la representación de una generalización, un objeto instanciado de la clase Circulo tendrá los atributos x\_position, y\_position, radius y un método display().



## 2.7 Diagramas de casos de uso

El diagrama de casos de uso es similar a un diagrama de lenguaje de modelado unificado (UML). Es una descripción de los pasos que deberán realizarse a iniciar un proceso.

Una de las principales ventajas de estos diagramas es la facilidad para ser interpretados por los clientes, es por eso que son muy utilizados para establecer comunicación con ellos. Además determinan las funciones y características necesarias que la aplicación puede realizar; estos diagramas no muestran el diseño de la aplicación y tampoco describen los elementos internos del sistema.

En la figura 12 se muestra un ejemplo de caso de uso, en el lado izquierdo del diagrama encontramos al actor el cual contiene tres casos de usos.

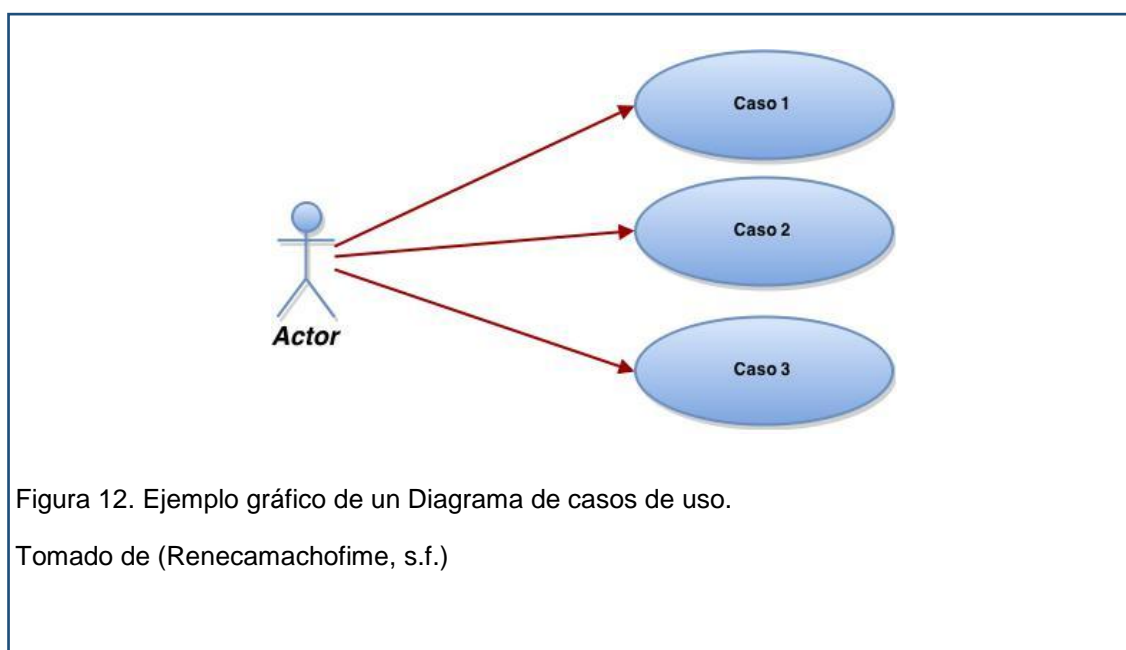


Figura 12. Ejemplo gráfico de un Diagrama de casos de uso.

Tomado de (Renecamachofime, s.f.)

Los elementos básicos son:

### 2.7.1 Actores

Representa un tipo de usuario; es decir; alguien quien interactúa con el sistema. Un rol es representado por un actor, en este diagrama pueden existir diferentes personas.

Los actores no representan a sistemas o personas, estos personifican al rol que desempeñan. El actor se denota con el dibujo de una persona como se muestra en la figura 13.



Figura 13. Representación de un actor.

### 2.7.2 Caso de uso

Son tareas específicas que se desarrollan en el sistema y son representadas mediante un ovalo como se muestra en la Figura 14.



Figura 14. Representación de un caso de uso.

### 2.7.3 Relaciones

**Asociación:** Es la relación o asociación entre un actor y un caso de uso.

En la figura 15 se muestra la relación asociación, la cual se representa con una flecha simple.



Figura 15. Representación de la relación asociación.

**Dependencia:** Es la relación entre clases, en la cual una clase se crea o depende de otra.

En la figura 16 se muestra la relación de dependencia representada por una flecha punteada.



Figura 16. Representación de la relación de dependencia.

**Generalización:** Esta relación es una de las más utilizadas, cumple una función doble dependiendo de su modelo, estos pueden ser: de Herencia (extends) o de Uso (*uses*).

La generalización está orientada solo para los casos de uso, no para los actores.

- ✓ **Extends:** Este estereotipo se utiliza cuando un caso de uso es semejante a otro.
- ✓ **Uses:** Debe ser utilizada en un conjunto de características similares que exista en más de un modelo de caso de uso.

En la figura 17 se muestra como se representa la relación de generalización, dicha relación se denota con una flecha.

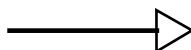


Figura 17. Representación de la relación de generalización.

## 2.8 Diagrama de secuencia

Los diagramas de secuencias son usados como modelos explicativos para los diferentes escenarios de los casos de uso. Estos diagramas muestran como los objetos participan en cada interacción y la secuencia de sus mensajes.

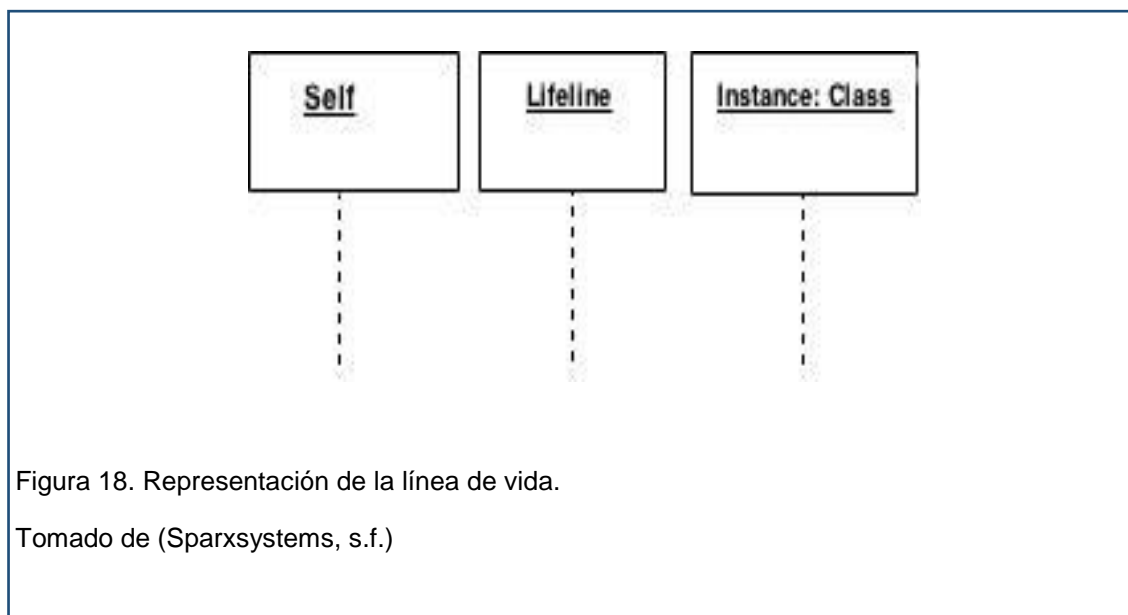
Los diagramas de secuencia diagrama se diseñan para cada caso de uso, permitiendo mostrar la forma de cómo se comunican los objetos entre sí, mientras avanza el tiempo.

Estos diagramas no son utilizados para diseñar lógicas de procedimientos más complejos.

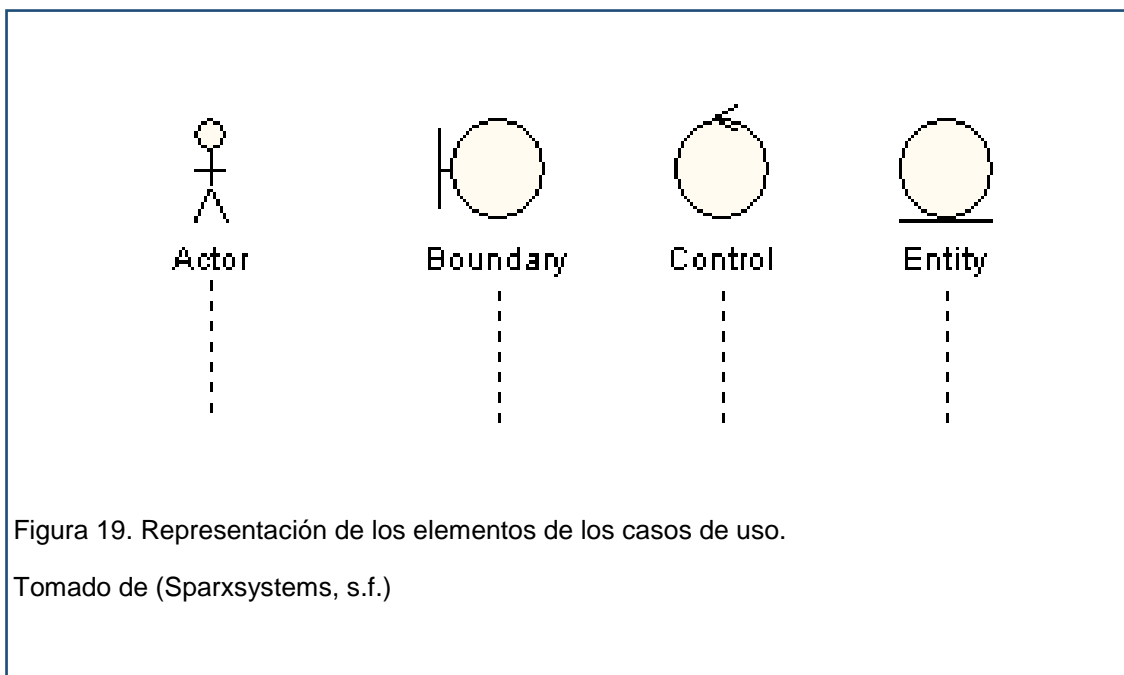
### 2.8.1 Línea de Vida

Se denota con una línea vertical que representa la secuencia de eventos que se producen en un participante (instancia de una clase o actor), durante una interacción, mientras avanza el tiempo a lo largo de la línea. Una línea de vida comúnmente tiene un rectángulo con el nombre del objeto.

En la figura 18 se muestra un ejemplo de la línea de vida, si el nombre del objeto es self entonces nos indica que la línea de vida representa el clasificador que posee el diagrama.



En algunas ocasiones la línea de vida del diagrama de secuencia tendrá un símbolo del objeto actor en su parte superior, esto es cuando un diagrama de secuencia es comprendido por un caso de uso; existen elementos como entidad, límite y control que también pueden contener líneas de vida como se muestra en la Figura 19.



### 2.8.2 Mensaje

Los mensajes son representados con flechas; estos mensajes pueden ser encontrados o perdidos, asíncronos o síncronos y completos.

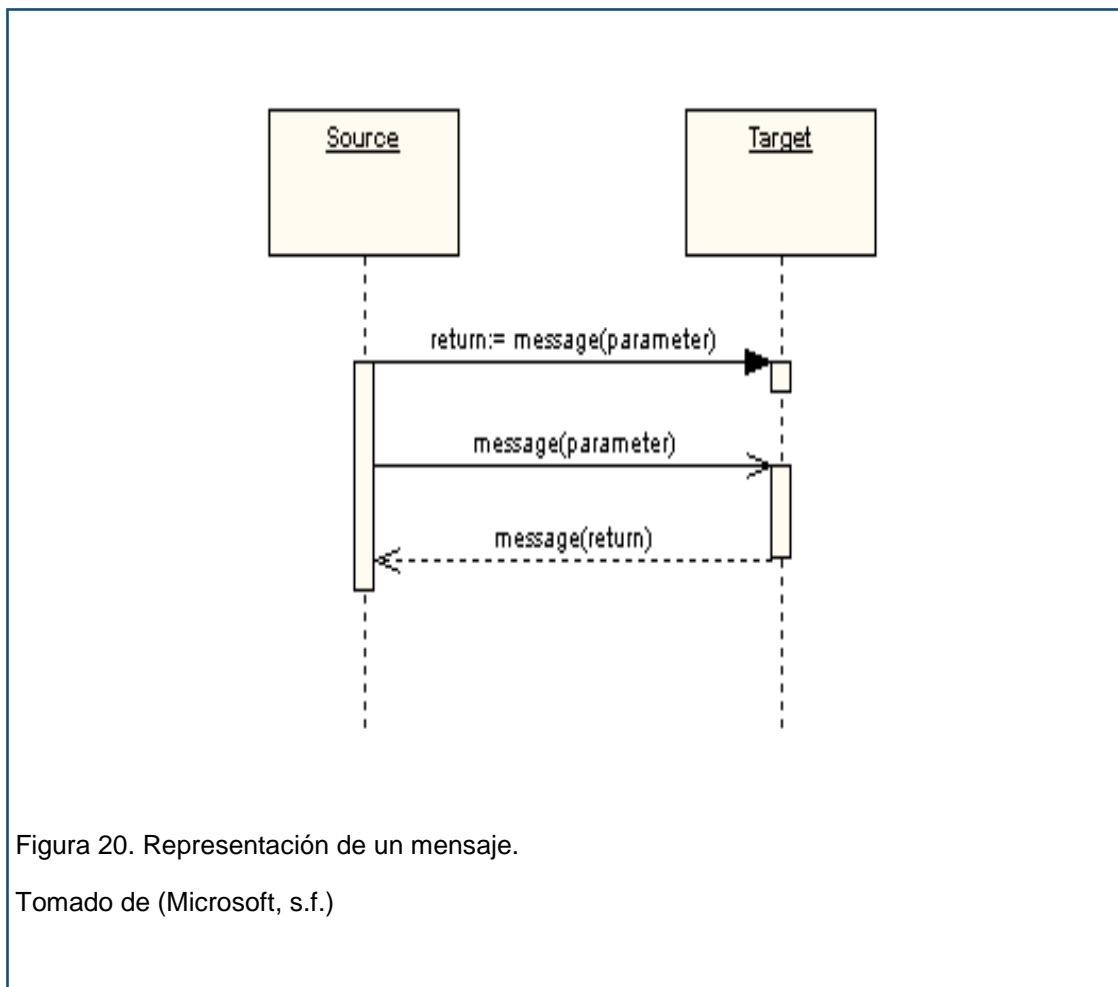
#### Tipos de mensajes:

**Mensaje síncrono:** Se representa con una flecha con punta oscura.

**Mensaje asíncrono:** Se representa con una flecha con punta de línea.

**Mensaje de retorno asíncrono:** Representado por una línea segmentada.

En la figura 20 se muestra la representación gráfica de un mensaje. En el ejemplo, el primer mensaje del diagrama es un mensaje síncrono, completo con un mensaje de retorno, su segundo mensaje es asíncrono y el tercer mensaje es de retorno asíncrono.



### 2.8.3 Ocurrencia en ejecución

La ocurrencia de ejecución se denota por un rectángulo fino a lo largo de la línea de vida, en la figura anterior podemos ver tres ocurrencias de ejecución.

### 2.8.4 Mensaje self

Representa una llamada recursiva de una operación o un método invocando a otro método que pertenece al mismo objeto.

En la figura 21 se muestra cuando se crea una ocurrencia de ejecución anidado en la en la ocurrencia de ejecución de la línea de vida.



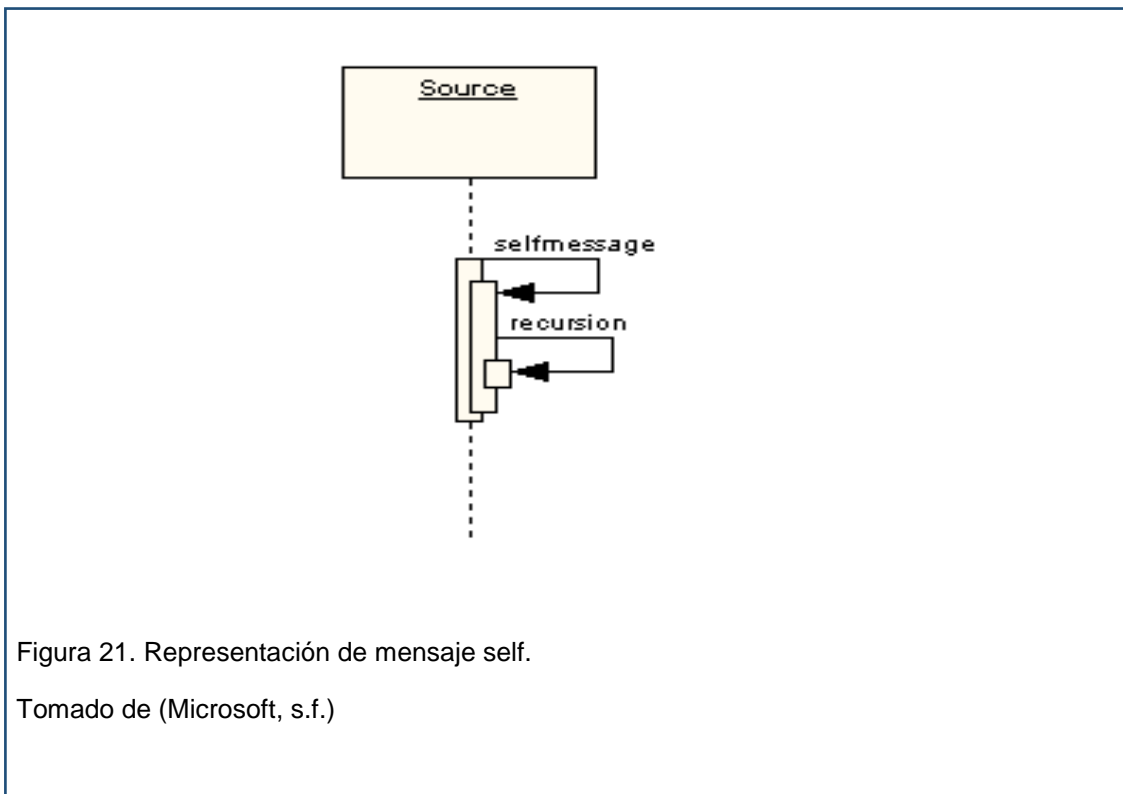


Figura 21. Representación de mensaje self.

Tomado de (Microsoft, s.f.)

### 2.8.5 Mensajes perdidos y encontrados

Existen dos tipos de mensajes.

- ✓ **Mensajes perdidos:** Son los mensajes que fueron enviados pero aun no llegan al destino, como se muestra en la figura 22.
- ✓ **Mensajes encontrados:** Son aquellos mensajes que llegan de un remitente desconocido, ver ejemplo de la figura 22.

Estos mensajes se representan llegando o yendo desde un elemento de punto final.

En la siguiente figura se muestra un diagrama de mensajes perdidos y encontrados, se reconocen porque la flecha del mensaje perdido se encuentra yendo desde un punto final y la flecha del mensaje encontrado está viniendo desde el punto final.

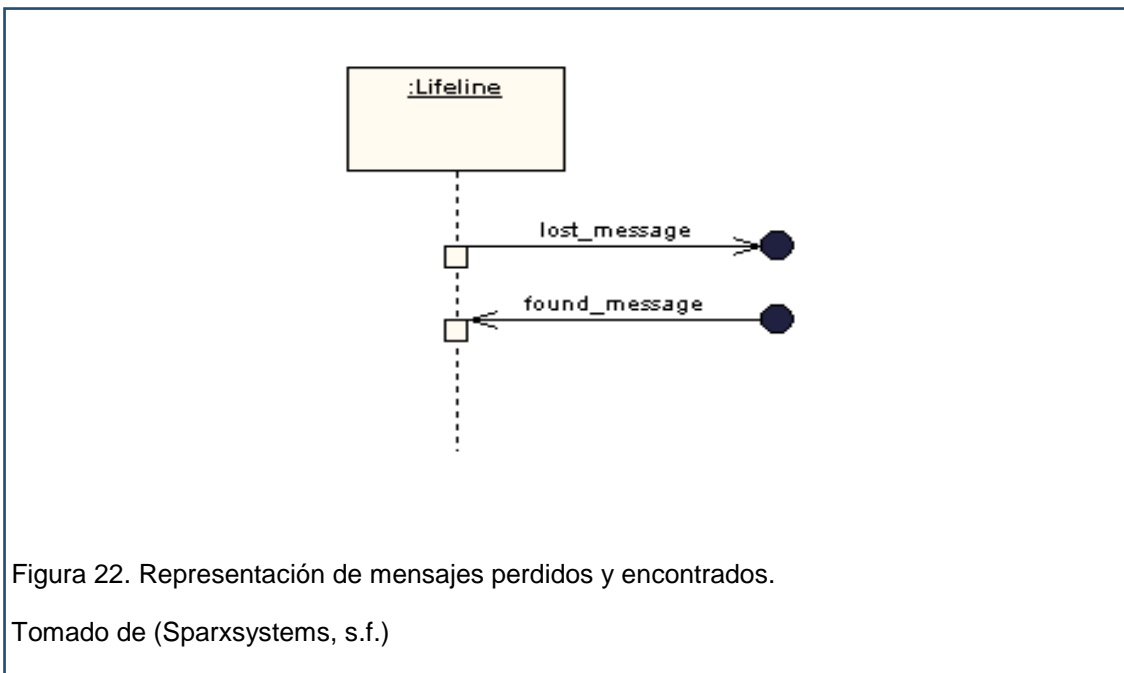


Figura 22. Representación de mensajes perdidos y encontrados.

Tomado de (Sparxsystems, s.f.)

### 2.8.6 Inicio y final de línea de vida

Una línea de vida puede crearse o destruirse durante el tiempo representado en un diagrama de secuencia. La línea de vida se finaliza con un símbolo de interrupción, el mismo que se representa con la cruz.

En el ejemplo de la figura 23 vemos un diagrama de secuencia en donde se muestra un objeto que ha sido creado y luego destruido en el tiempo.

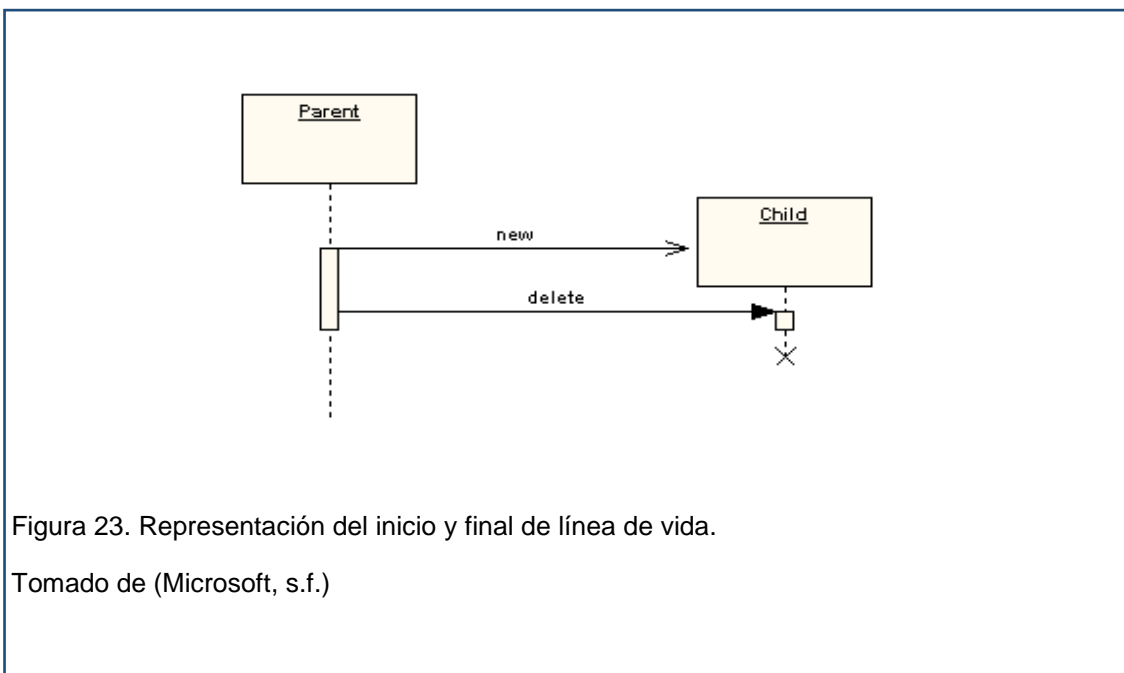


Figura 23. Representación del inicio y final de línea de vida.

Tomado de (Microsoft, s.f.)

## 2.9 Diagrama de actividades

El diagrama de actividades permite mostrar la secuencia de cada actividad. Estos expresan el flujo de trabajo desde su punto de inicio hasta su punto final, especificando las rutas de decisiones que existen en cada evento que contiene la actividad.

En siguiente figura se muestra un diagrama de actividad, el cual modela el proceso de entrega de tareas por parte del profesor al estudiante; iniciando con la emisión de tareas del profesor, seguidamente el estudiante realiza la actividad de recibir la tarea y continúa el proceso hasta finalizar con la actividad de calificar la tarea.

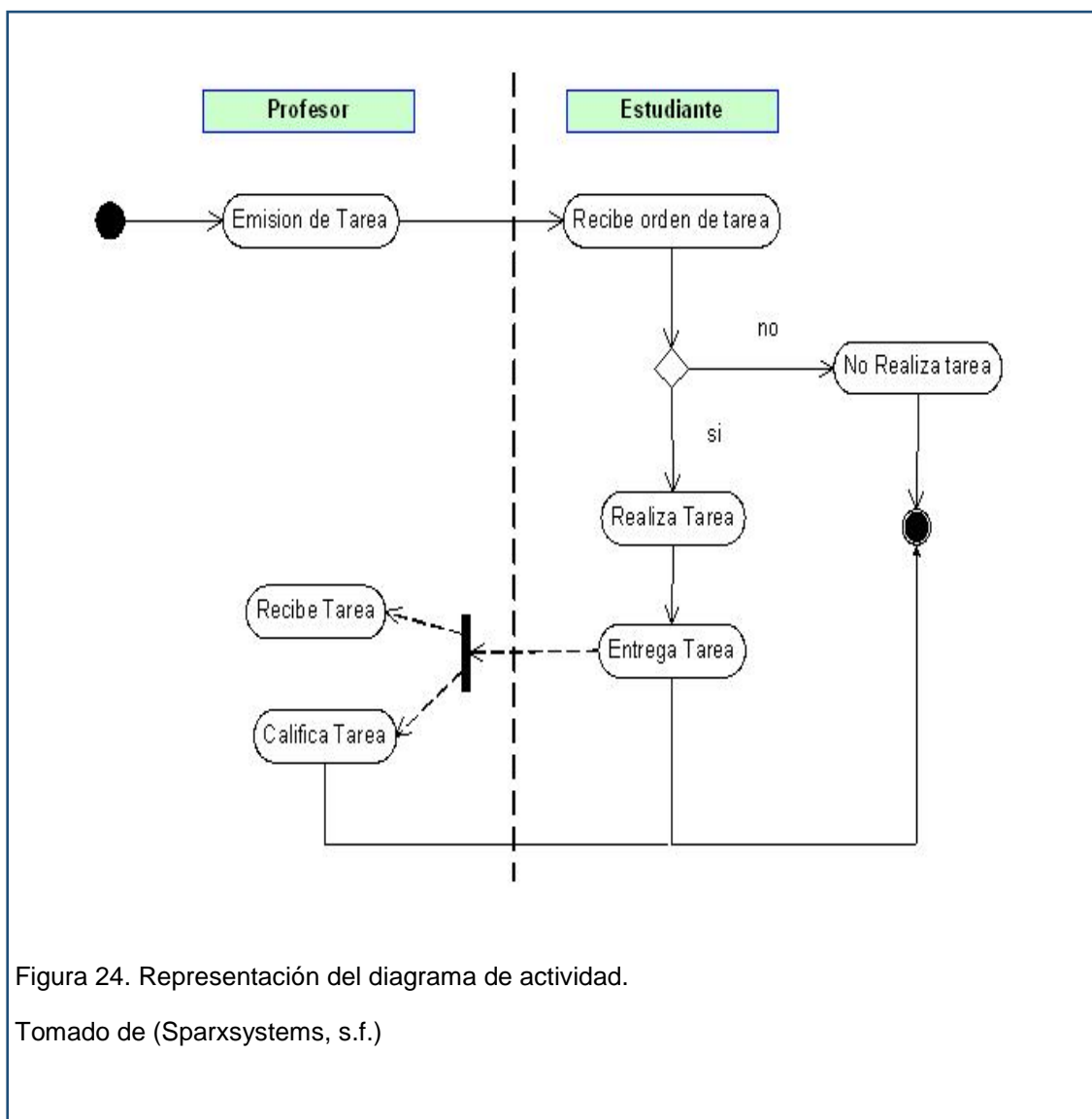


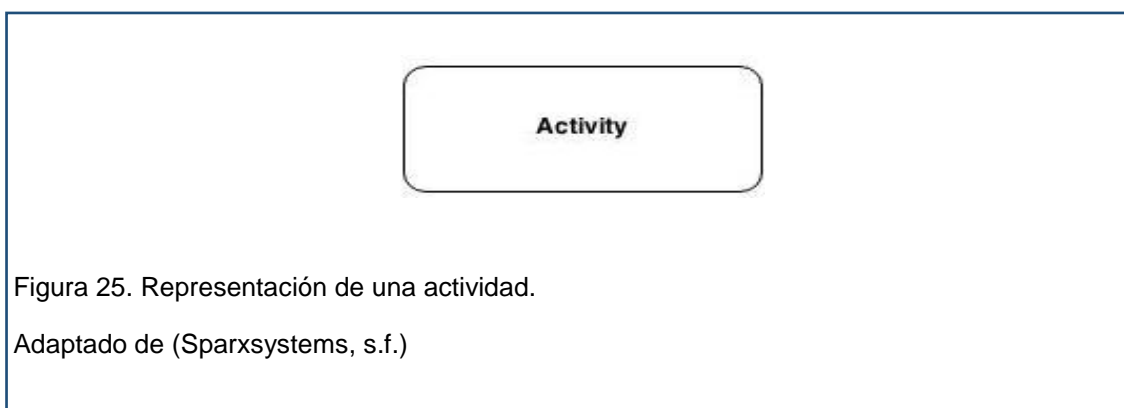
Figura 24. Representación del diagrama de actividad.

Tomado de (Sparxsystems, s.f.)

Los elementos que forman parte de un diagrama de actividades son:

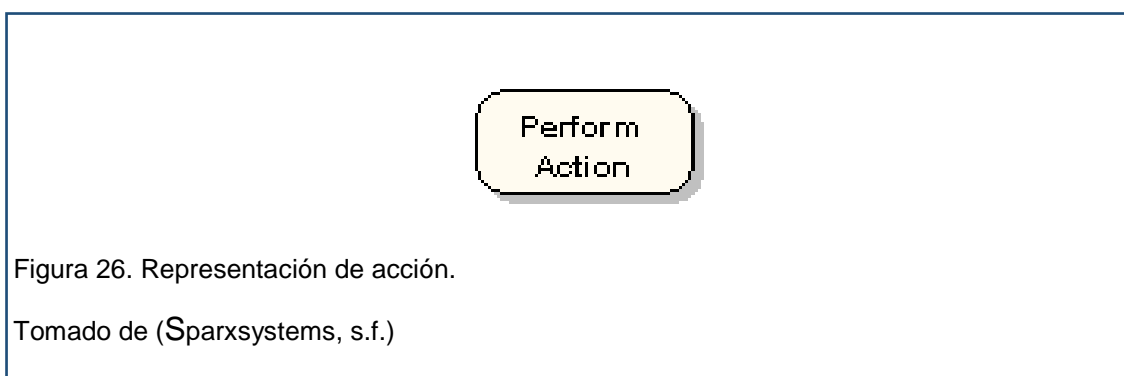
### 2.9.1 Actividades

Una actividad es la descripción de una secuencia de comportamiento o acción y se representa por un rectángulo con puntas redondas. En la figura 25 se muestra la representación de una actividad.



### 2.9.2 Acciones

La acción representa un solo paso en una actividad y se gráfica con rectángulos de puntas redondas como se muestra en la figura 26.



### 2.9.3 Restricciones de Acción

La restricción de acción puede adjuntarse a una acción.

El diagrama de la figura 27 muestra una acción con pre condiciones locales y post condiciones locales.

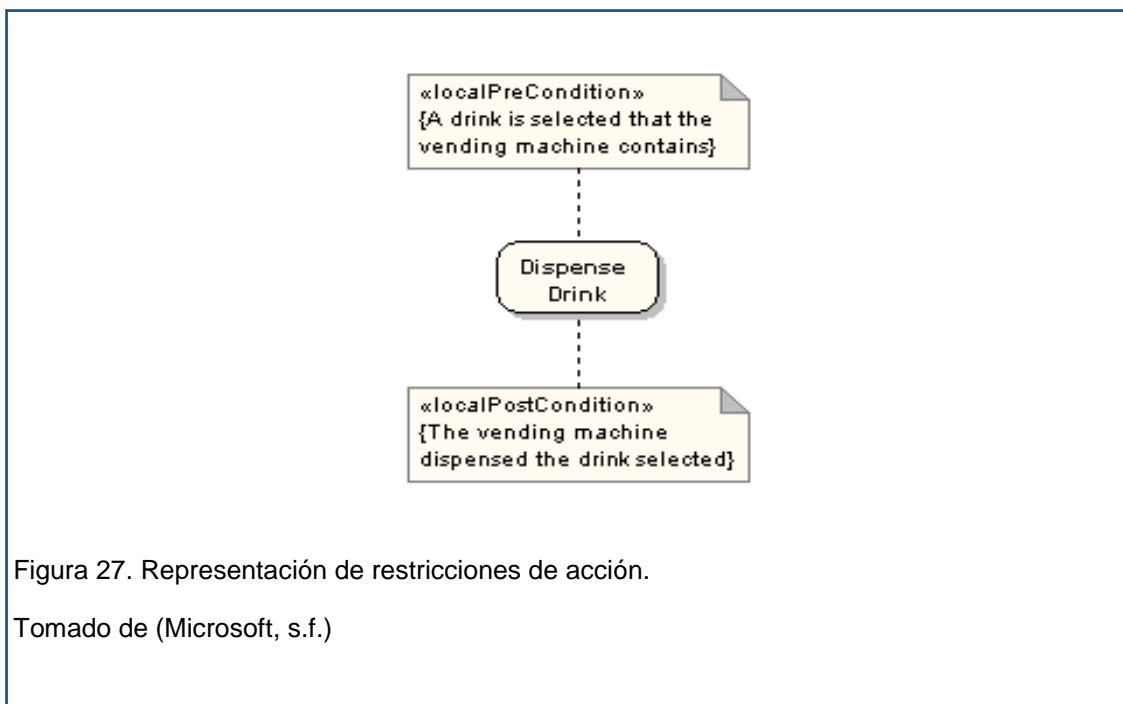


Figura 27. Representación de restricciones de acción.

Tomado de (Microsoft, s.f.)

#### 2.9.4 Flujo de control

El flujo de control se representa con una línea con punta de flecha y muestra el flujo control de una a otra acción como se muestra en la figura 28.

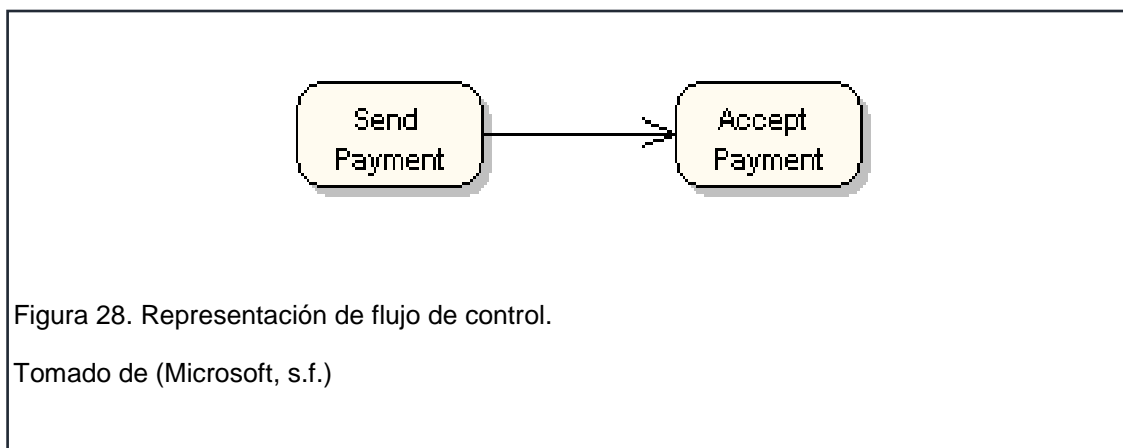


Figura 28. Representación de flujo de control.

Tomado de (Microsoft, s.f.)

#### 2.9.5 Nodo inicial

El nodo de comienzo o inicial se representa por un punto negro grande.

En la figura 29 se muestra el nodo inicial al comienzo de la acción.

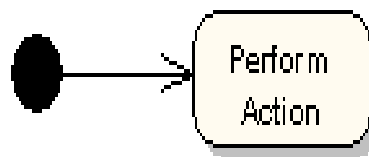


Figura 29. Representación de nodo inicial.

Tomado de (Ingenieriadestistemas, s.f.)

### 2.9.6 Nodo final

Existen dos tipos de nodos finales:

- ✓ **Nodo final de actividad:** Representa el final de un solo flujo de control, se denota con un círculo y un punto negro dentro del círculo, como se muestra en la figura 30.

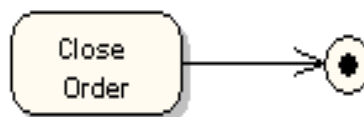


Figura 30. Representación de nodo final de actividad.

Tomado de (Ingenieriadestistemas, s.f.)

- ✓ **Nodo final de flujo:** Representa el final de todos los flujos finales dentro de la actividad, se denota con un círculo y con una cruz dentro del círculo. En la figura 31 se observa el final de la actividad Orden.

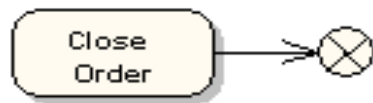


Figura 31. Representación de nodo final de flujo.

Tomado de (Ingenieriadiagramas, s.f.)

### 2.9.7 Flujos de objetos y objeto

El flujo de objeto es la ruta donde se pasan los datos u objetos. El objeto se representa con un rectángulo, como se muestra en la figura 32.

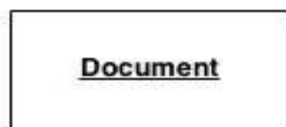


Figura 32. Representación de objeto.

El flujo de objeto se denota como una punta de flecha indicando la dirección donde está pasando el objeto.

En el ejemplo de la figura 33 se observa cómo va pasando el flujo de objeto de la actividad enviar la factura a el objeto factura y luego pasa a la dirección hacer el pago.

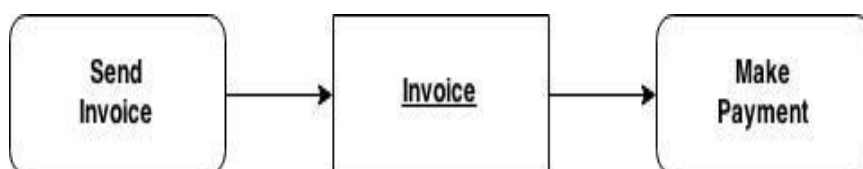


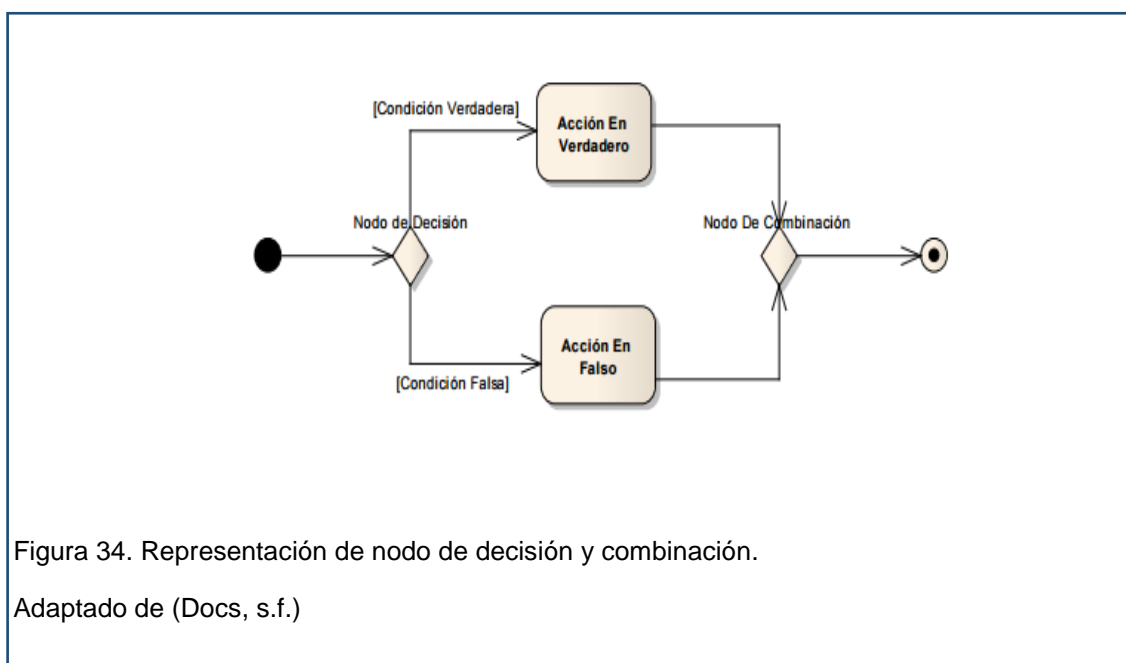
Figura 33. Representación de un flujo de objeto.

Tomado de (Ingenieriadestemas, s.f.)

### 2.9.8 Nodos de decisión y combinación

Los nodos de decisión y combinación se representan con forma de diamante, los dos pueden ser nombrados. Los flujos de control que vienen de un nodo de decisión poseen condiciones que permiten el control para seguir si la condición se realiza.

En el diagrama de la figura 34 podemos observar el uso de un nodo de decisión donde el flujo de objeto le direcciona hacia una acción verdadera o una acción falsa y después de escoger cualquiera de estas acciones podemos llegar al nodo de combinación, el mismo que nos conduce al nodo final.



### 2.9.9 Partición

La partición se utiliza para separar las acciones de una actividad y se representan como calles en sentido horizontal o vertical.

En la figura 35 se observa que la partición es utilizada para separar la acción envió de factura realizada por el departamento de Contabilidad, de la acción realizada por el cliente en este caso el realizar el pago de la factura.



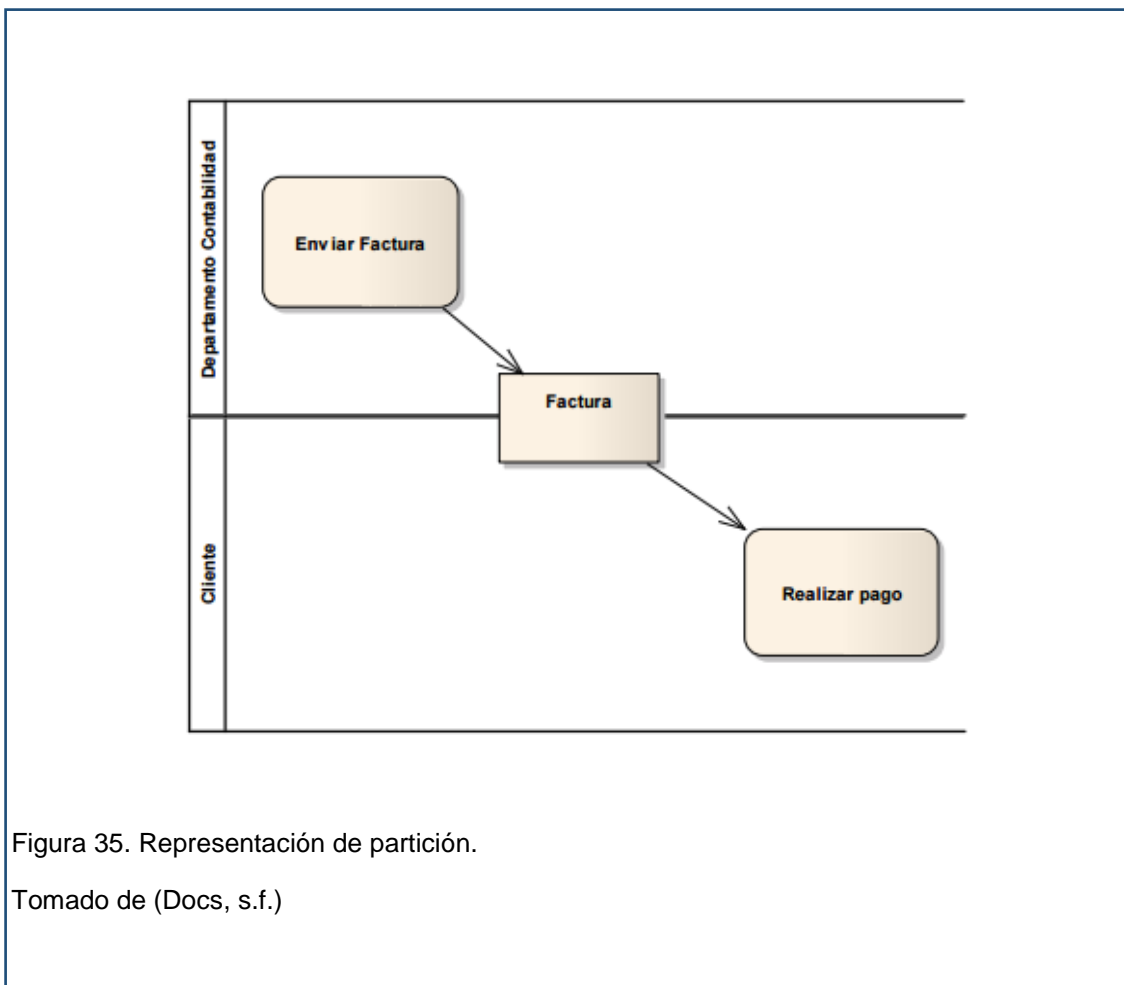


Figura 35. Representación de partición.

Tomado de (Docs, s.f.)

## 2.10 Diagrama de despliegue

Se encarga de modelar la arquitectura de Hardware en el tiempo de ejecución de una aplicación. Estos diagramas muestran la configuración de los elementos de nodos, y como estos elementos y artefactos del sistema se trazan en los nodos.

### 2.10.1 Nodo

El nodo es un elemento ya sea de software o hardware y se representa con la figura de una caja en tres dimensiones.

En la figura 36 se muestra el nodo Server.

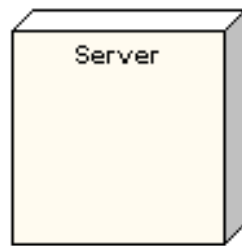


Figura 36. Representación de nodo.

Tomado de (UmdlDiagramadespliegue, s.f.)

### 2.10.2 Artefacto

El artefacto es un producto en el proceso de desarrollo de un sistema, estos pueden incluir: modelos de casos de uso, archivos fuentes, modelos de proceso, documentos, ejecutables, manuales de usuario, entre otros.

El artefacto se representa con un rectángulo, dentro de él se muestra el estereotipo «artifact», un icono de documento y el nombre del artefacto.

En la figura 37 podemos observar el artefacto de un archivo fuente main.c

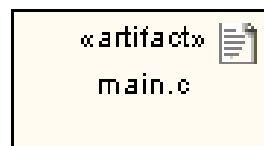


Figura 37. Representación de artefacto.

Adaptado de (UmdlDiagramadespliegue, s.f.)

### 2.10.3 Asociaciones

La asociación representa la ruta de comunicación que existe entre nodos, los nodos envían mensajes o intercambian objetos a través de esta ruta. El tipo de comunicación indica la red o el protocolo de comunicación.

En la figura 38 se observa un diagrama de despliegue para una red, donde se visualizan los protocolos de red tcp ip.

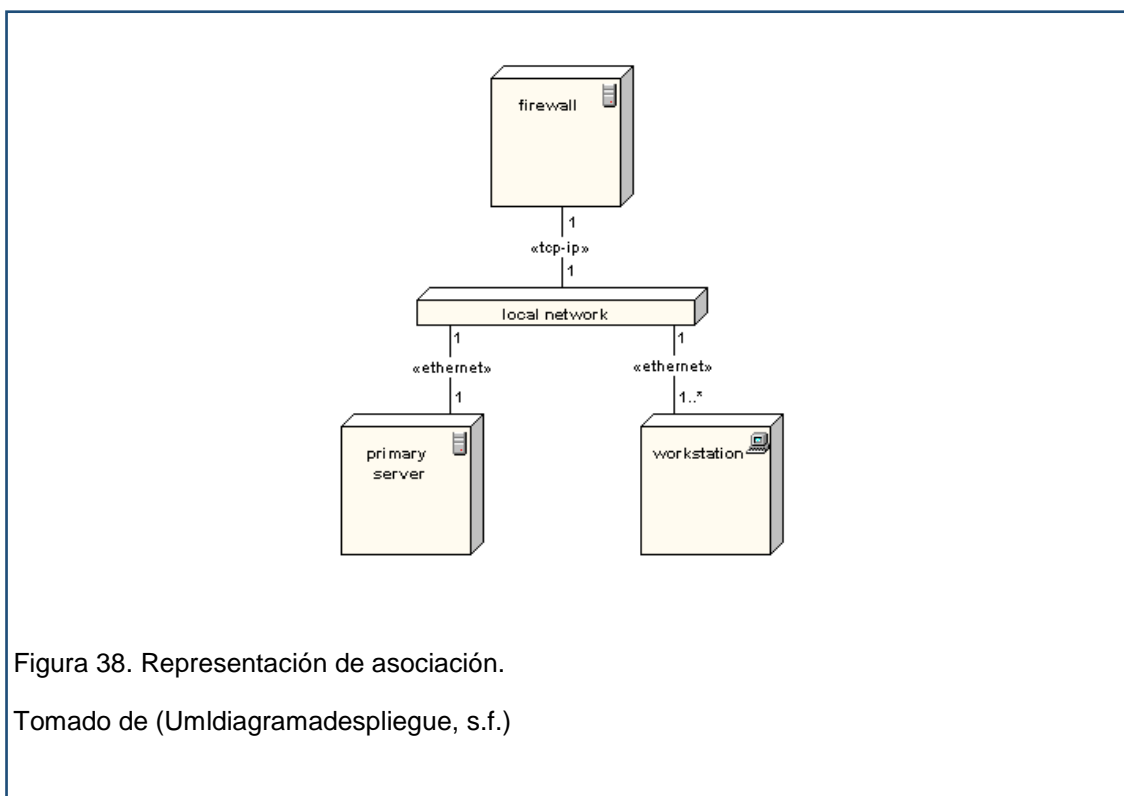


Figura 38. Representación de asociación.

Tomado de (UmdlDiagramadespliegue, s.f.)

## CAPÍTULO III

### Tecnologías

#### 3.1 Visual Studio 2012

Es una de las nuevas herramientas de desarrollo de Microsoft. Visual Studio 2012 ha tenido una importante renovación en su interfaz gráfica, facilitando al usuario el acceso rápido a todas sus herramientas. Ofrece nuevas plantillas de diseño y herramientas de depuración para la creación de aplicaciones en el menor tiempo posible.

Entre las principales novedades tenemos:

- ✓ Permite el acceso a la nube, ya que maneja servidores virtuales, lo que facilita llevar las diferentes aplicaciones a Windows Azure.
- ✓ Últimos avances en ASP.NET que incluye mejoras y nuevas plantillas web basadas en HTML5 y ccs3.
- ✓ Maneja procesos ágiles, ya que incorpora plantillas que nos ayudan a utilizar de mejor manera metodologías ágiles como lo son: Scrum y Kanban.
- ✓ Nos permite compilar y desarrollar las diferentes aplicaciones que realicemos en la tienda de Windows.

#### 3.2 Sql Server 2008

Es un gestor de base de datos muy utilizado en todo el mundo. Permite mayor escalabilidad de exploración de objetos de Bases de Datos y la integración de secuencias de comandos en la base de datos.

Entre sus principales características tenemos:

- ✓ Manejo de alta disponibilidad, lo cual garantiza que las aplicaciones siempre estén utilizables.
- ✓ Mejoras en la velocidad de memoria, lo cual facilita el almacenamiento y el análisis de los datos.

- ✓ Presentación y visualización de datos interactivos para el usuario basada en web.
- ✓ Mejor integración con las herramientas de Excel y desarrollo en la nube.

### 3.3 Modelo Vista Controlador

También conocido por las siglas MVC, es un patrón de diseño de arquitectura de desarrollo que separa la interfaz del usuario, los datos del sistema y la lógica del negocio en tres distintos componentes. El patrón de diseño apareció a fines de los años 70 y fue introducido por Trygve Reenskaug en una visita a los laboratorios de Xerox.

El mvc sirve para implementar sistemas donde se requiere el uso de interfaces de usuario, permitiendo crear aplicaciones de mayor calidad, más robustas, y potenciando su facilidad de mantenimiento y la reutilización de código. Fue diseñado para reducir esfuerzos al momento de la programación, permitiendo una separación del código fuente en tres diferentes capas llamadas Modelos, Vistas y Controladores.

La figura 39 representa el patrón mvc en una aplicación web; donde la vista es la página html, el código que proporciona los datos a la página es el modelo (Base de datos) y la lógica del negocio es el controlador, responsable de recibir los eventos desde la entrada de la vista.

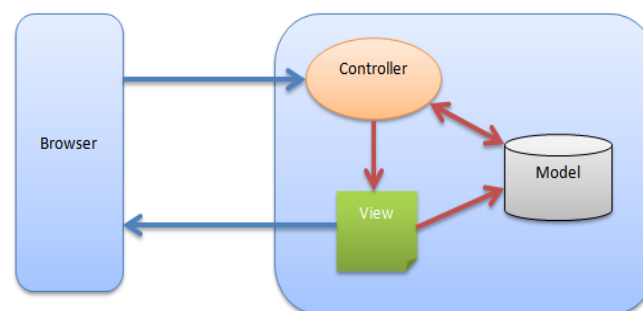


Figura 39. Representación del patrón MVC.

Tomado de (Codeproject, s.f.)

Los elementos de este patrón de diseño son:

✓ **Modelo**

Es la capa que contiene los datos y la funcionalidad del sistema. El modelo se encarga de administrar la lógica del sistema, es decir cómo se presenta la información que utiliza sistema.

Tiene acceso a la Base de Datos, permitiendo gestionar los accesos y privilegios que se hayan especificado en la aplicación.

**Ejemplo:**

En un juego de ajedrez, son las reglas del ajedrez.

✓ **Vista**

Es la que gestiona como se presentan los datos. Es la representación gráfica del modelo, reciben los datos del modelo y los muestra al usuario. La vista es la encargada de representar los componentes visuales en la pantalla del sistema.

**Ejemplo:**

En un juego de ajedrez, es la ventana que dibuja el tablero.

✓ **Controlador**

Es el que escucha los eventos que genera el usuario, permite que el usuario interactúe con el sistema, interpretando los eventos a través del ratón o teclado.

El controlador define que modificaciones hay que realizar en el modelo cuando se comunica con la vista.

**Ejemplo:**

En un juego de ajedrez: El algoritmo para pensar las jugadas.

**Flujo de control en una implementación de MVC**

El flujo de control que se utiliza para su implementación es el siguiente:

1. El usuario genera una petición al sistema, el propósito de esta petición es acceder a una página de nuestro sistema o realizar algún tipo de operación.
2. La petición es recibida por el controlador, este se encarga de gestionar el evento que recibe con un gestor de eventos.
3. El controlador se encarga de acceder al modelo, lo modifica y actualiza de acuerdo al evento solicitado por el usuario.
4. El controlador autoriza a los objetos de la vista desplegar la interfaz al usuario. El modelo proporciona los datos a la vista para poder construir la interfaz.
5. La interfaz espera más peticiones del usuario, y se inicia el ciclo del flujo de control.

En la figura 40 podemos observar cómo se comporta el flujo de control del patrón mvc en aplicaciones.

El usuario genera un evento en la vista, esta sería nuestra página HTML; la vista envía el evento a nuestro controlador, en este caso es el código que obtiene datos dinámicos y genera el contenido HTML, y el Modelo es la información almacenada en nuestra base de datos y el que se encarga de enviar los datos a la vista, la misma que muestra el estado al usuario.

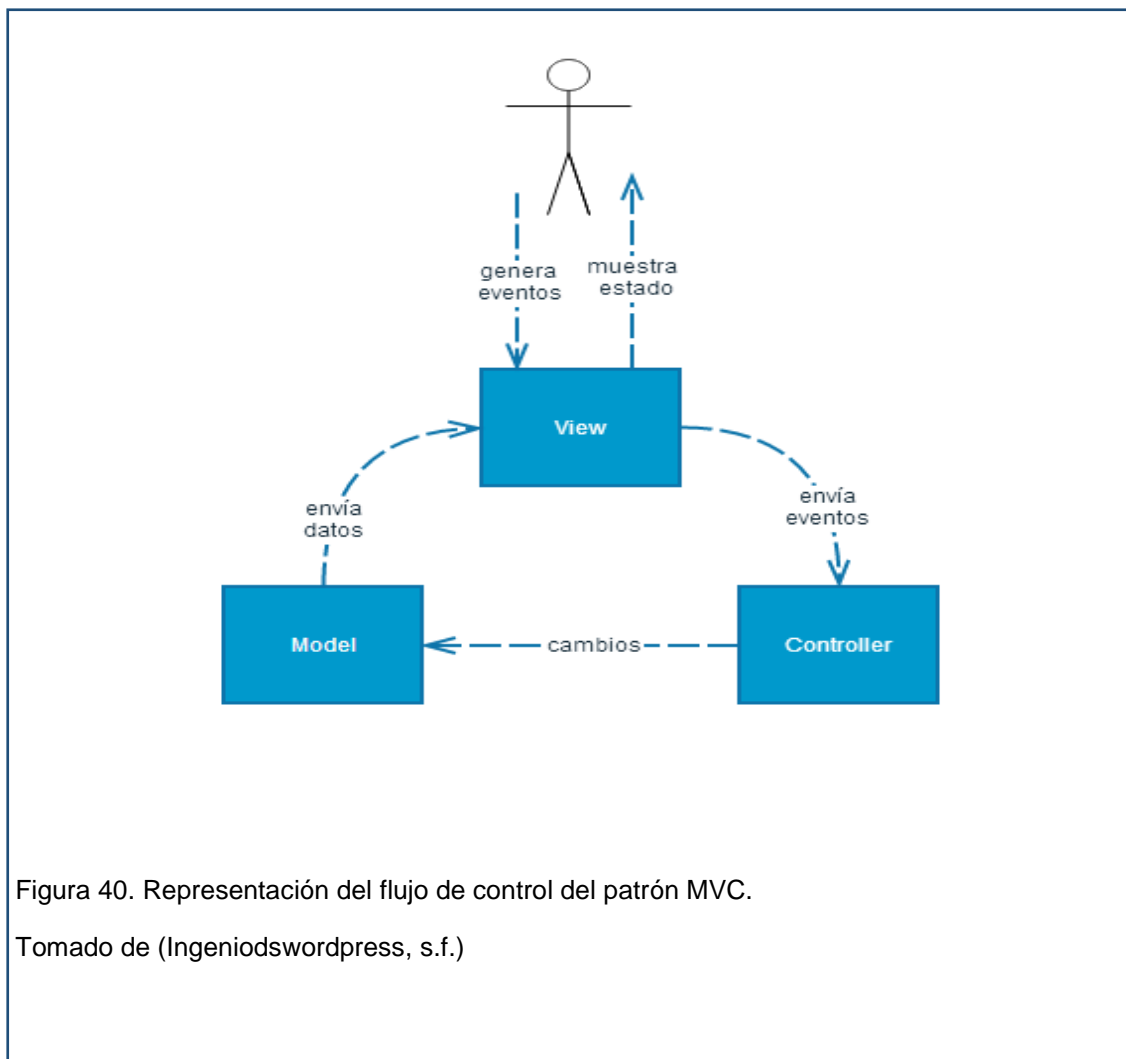


Figura 40. Representación del flujo de control del patrón MVC.

Tomado de (Ingeniodswordpress, s.f.)

### 3.4 Versiones de Microsoft MVC

Microsoft ha agregado características interesantes en cada nueva versión de MVC, las mismas que facilitan a los desarrolladores el crear aplicaciones web.

Entre principales versiones de Microsoft ASP.NET MVC tenemos desde mvc3, mvc4 y mvc5.

A continuación se muestra un cuadro comparativo de las principales características de las versiones mvc3, mvc4 y mv5 de Microsoft ASP.NET.



Tabla 2. Comparación de las versiones de ASP.NET MVC.

VERSIONES DE ASP.NET MVC		
MVC3	MVC4	MVC5
Nuevas plantillas que tienen soporte para HTML5 y CSS3.	ASP.NET Web API	Filtros de autenticación para autenticación de usuario.
Razor View Motor presentó un paquete que incluyen nuevas características.	Mejora de los templates predeterminados de proyecto.	Filtro de anulación, ahora se pueden reemplazar filtros en un método o controlador.
Mejora de la validación del modelo.	Templates de proyectos móviles.	Bootstrap sustituye la plantilla predeterminada MVC.
Cuenta con el apoyo de múltiples View Engines como i.e, web forms, razor.	Modos de visualización.	ASP.NET identidad, para autenticación de identidad, maneja la autorización para sitios de redes sociales.
	Mejora para Razor.	

Adaptado de (Webdevelopmenthelp, s.f.)

Nota: Cuadro comparativo de las principales características de las versiones mvc3, mvc4 y mvc5 de Microsoft ASP.NET

Para el desarrollo del Prototipo de sistema de administración del ganado bovino de la Universidad de las Américas se utilizó la versión ASP.NET MVC4 porque es la más estable a pesar de no ser la más actual ya que la versión mvc5 se encuentra en la etapa beta, además mvc4 se integra perfectamente en Visual Studio 2012.

### 3.5 MVC4

Es el patrón o modelo de desarrollo de software y consumo de servicios de Microsoft. Nos permite construir aplicaciones web dinámicas, facilitándonos el tener separado los datos, la interfaz y la lógica de negocio de la aplicación.

Entre sus principales características de la versión de Microsoft tenemos:

- ✓ ASP.NET Web API, este nuevo framework deja desarrollar y consumir servicios HTTP, permitiendo acceder desde navegadores web hasta dispositivos móviles.
- ✓ Mejora de los templates predeterminados de proyecto, para obtener sitios web con vistas modernas y vistas para dispositivos móviles con rendering adaptativo.
- ✓ Templates de proyectos móviles, esta plantilla basada en jQuery Mobile está diseñada para que un proyecto corra solamente en navegadores de tablets y dispositivos móviles.
- ✓ Modos de visualización, el nuevo modo permite a mvc seleccionar el tipo de vista más beneficioso dependiendo del navegador que realice la petición.
- ✓ Mejoras para Razor, en ASP.NET mvc4 se incluye la última versión de view engine de Razor, permitiendo crear plantillas más claras y concisas, incluyendo un mejor soporte para resolver referencias URL.

### 3.6 Diseño Web Sensible

También conocido por su término en inglés Responsive Web Design (RWD). Es cuando un sitio web está al tanto de las proporciones del navegador, el usuario puede maximizar y minimizar, y la resolución de la pantalla se ajustará al cambio de tamaño de la ventana.

Responsive Web Design permite al usuario mostrar de manera clara y ordenada el contenido del sitio cualquier dispositivos, tales como: tablets, teléfonos inteligentes, etc.

En la figura 41 se muestra un ejemplo de un sitio web en la pantalla de un escritorio, el diseño se presenta en tres columnas de contenido, pero al cargarse en una tablet este se ajusta a dos columnas y en un teléfono inteligente se presenta en una sola columna, este diseño varía automáticamente de acuerdo a la resolución de la pantalla del usuario.



Figura 41. Representación de diseño web sensible.

Tomado de (Estudio, s.f.)

Actualmente los usuarios de los sitios web acceden a los mismos desde distintos medios electrónicos, el diseño web debe estar adaptado para que este funcione; es importante que el usuario tenga la visita al sitio web más satisfactoria sin importar que dispositivo este usando para revisarla. Es por esto que los sitios web están empezando a utilizar los diseños Sensibles y Adaptativos para brindar una mejor experiencia al usuario.

Para la realización del prototipo de sistema de administración del ganado bovino de la granja experimental de la UDLA, se utilizó el diseño web sensible Adaptive Rendering compatible con ASP.NET MVC 4. Antes de escoger la tecnología se realizó una evaluación entre los dos diseños web mencionados, como se muestra en el cuadro comparativo de la Tabla 3.

Tabla 3. Cuadro comparativo entre el DWA y el DWS.

<b>DIFERENCIAS</b>	
<b>ADAPTATIVO</b>	<b>SENSIBLE</b>
Plantillas distintas, optimizadas para cada dispositivo.	Plantilla única para todos los dispositivos.
Se puede crear diferentes diseños para cualquier dispositivo.	El diseño se ajusta automáticamente en función de la resolución de la pantalla del usuario.
Enfoque realizado en capas (Capa de contenido, código html; Capa de presentación, css; Capa de scripting, JavaScript o jQuery).	Utiliza secuencias de comandos en CSS3 para adaptarse a diversos dispositivos y tamaños de la pantalla
<b>La página se carga más rápido.</b>	La página se carga más lento.
<b>VENTAJAS</b>	
Brinda mejor experiencia al usuario, el visitante podrá ver un sitio optimizado para su dispositivo.	Asegura la uniformidad de diseño a través de varios dispositivos y resoluciones de pantallas.
Mejora el rendimiento del sitio web en dispositivos móviles, sus sitios web muestran versiones simplificadas; permitiendo una navegación más simple.	Se mantiene una sola versión del sitio web, ahorrando tiempo y esfuerzos.
<b>DESVENTAJAS</b>	
Se debe configurar las redirecciones del sitio dependiendo del dispositivo donde se encuentre el usuario haciendo clic, para que el enlace redirija a la versión móvil de ese vínculo.	Tener en cuenta los anuncios en los sitios web, no todos encajan en la resolución de cada pantalla, al variar la anchura de las columnas el contenido se muestra diferente.
Se debe realizar un diseño por cada dispositivo móvil.	La resolución de la pantalla de dispositivos antiguos no se adapta correctamente a los diseños web.

Adaptado de (Webdevelopmenthelp, s.f.)

### 3.7 Hyper Text Markup Language 5

Conocido por sus siglas en inglés HTML5 (Lenguaje de marcado de hipertexto versión 5), es el nuevo estándar de html; contiene nuevas etiquetas, atributos, elementos y muchas más opciones de diseño.

Es usado para estructurar y presentar el contenido para la web. Con HTML5, los navegadores como Chrome, i.e., Firefox y demás pueden saber cómo mostrar una determinada página web, conocer donde se encuentran los elementos, ubicar el texto y colocar las imágenes.

HTML5 nos permite una mejor interacción entre las páginas web y el contenido multimedia (audio y video), así como una mayor facilidad a la hora de programar el diseño básico de una página.

Entre las novedades de HTML5 tenemos:

- ✓ Elemento Canvas, permite el renderizado de gráficos e imágenes en 3D.
- ✓ Incorpora etiquetas de audio y video, con codecs para mostrar contenidos multimedia.
- ✓ Hojas de estilo en cascada 3 (css3), ofrece una gran variedad de opciones para realizar diseños más elaborados.
- ✓ API de geolocalización para dispositivos que lo soporten.
- ✓ Etiquetas para manejar grandes conjuntos de datos como: Datagrid, Menu, command, que permiten generar tablas más dinámicas que puedan ordenar, filtrar y ocultar contenido.

### 3.8 Hojas de estil en cascada nivel 3

También conocida por sus siglas en inglés CSS3 (Cascading Style Sheets level 3), su objetivo principal es separar el contenido de la forma de una página web.

CSS3 es utilizado para definir el diseño y los estilos de las páginas, ofreciendo una gran variedad de opciones importantes tales como: sombreado, redondeado y funciones avanzadas de transformación y movimiento, todo esto para satisfacer las necesidades del diseño web.

En la figura 42 se muestra un ejemplo de código css3, donde la identificación id="rcorners1", posee las propiedades para realizar esquinas redondas de un cuadrado verde.

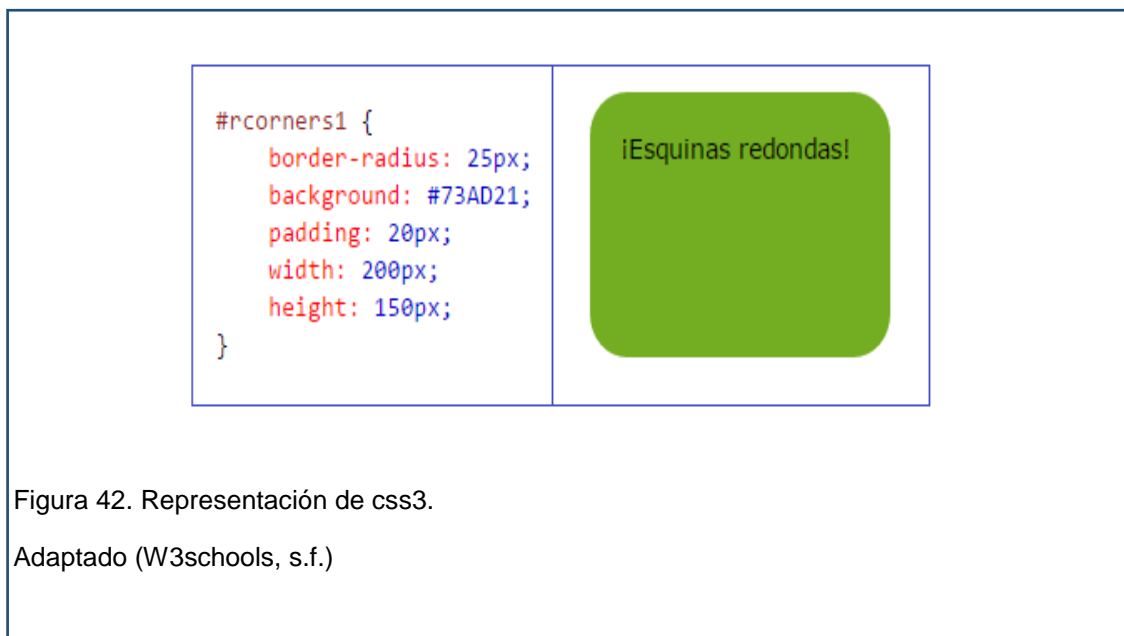


Figura 42. Representación de css3.

Adaptado (W3schools, s.f.)

### 3.9 Lenguaje de definición de informe del lado del cliente

También conocido por sus siglas en inglés RDCL (Report Definition Language Client-side), son archivos creados por Visual Studio 2012 en la opción de diseño de informes. Esta herramienta nos permite crear, implementar y administrar reportes de manera interactiva con gráficos y tabulaciones pueden ser ejecutados directamente por el control ReportViewer del lado del cliente, siendo mucho más fácil su implementación ya que no necesita tener una instancia de Reporting Services.

Uno de los Beneficios del reporte RDLC es que permite tener mejor control sobre los datos; es decir facilita el validar y manipular los datos antes de enviarlos al informe.

### 3.10 Diseño Web

También conocido en inglés como Wireframes, es una herramienta de diseño para la presentación de las funciones, estructura y el contenido de un sitio web.

Los wireframes sirven como herramienta de discusión y comunicación entre diseñadores, programadores y clientes. Los desarrolladores lo utilizan para mostrar la funcionalidad que tiene cada interfaz; estos deben ser utilizados al momento que empiece el proyecto para obtener la aprobación del cliente en el diseño y navegación del sitio.

El objetivo principal de los wireframes, es el de representar de forma gráfica el diseño o las funcionalidades de una página web; este esquema no posee color, ni dibujos, ni estilos tipográficos. La información que debe contener es: contenido de la página, elementos como: enlaces, imágenes, listas, vínculos, títulos y notas que indiquen el comportamiento de ciertos elementos.

En la figura 43 se observa la distribución del contenido de un sitio web: del lado derecho se encuentra el logo y debajo el menú de navegación, en el centro de la pantalla va un slide show con las imágenes del sitio, se muestran secciones de información y accesos a redes sociales en el pie de página.

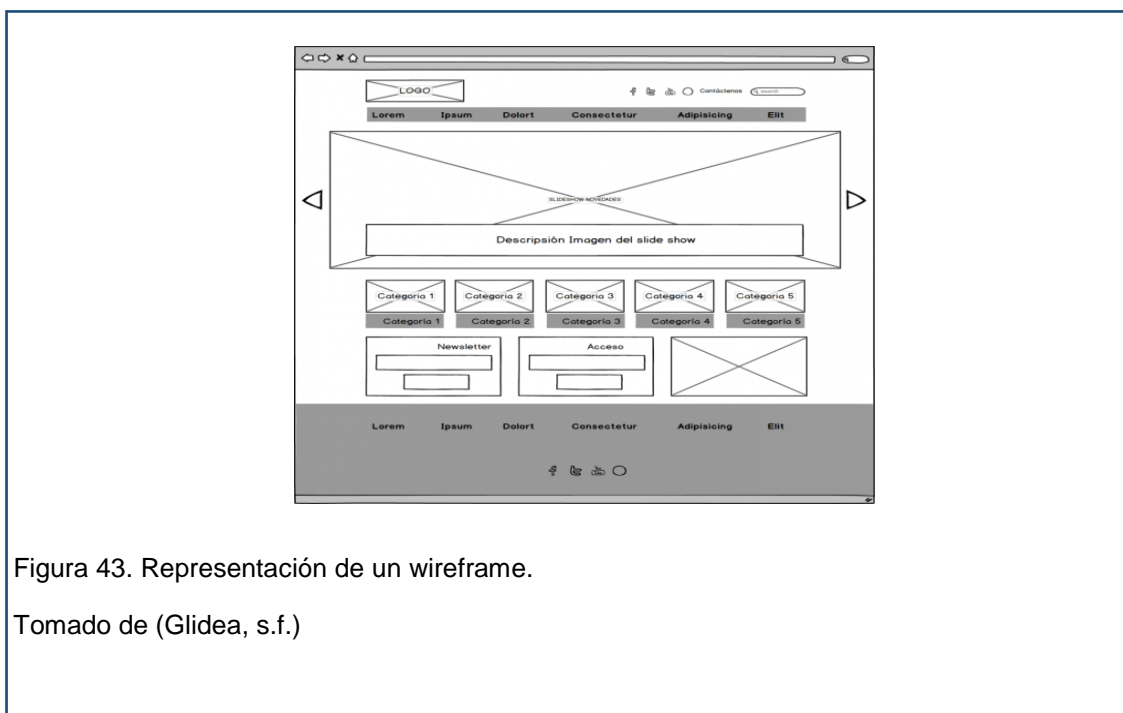


Figura 43. Representación de un wireframe.

Tomado de (Glidea, s.f.)

### 3.10.1 Ventajas

- ✓ La elaboración de estas pantallas es muy sencilla y no toma mucho tiempo.

- ✓ Estos dibujos son realizados en conjunto con el cliente; gráficamente se puede representar lo que el usuario realmente necesita y como desea que la aplicación funcione.
- ✓ Al momento de diseñar un Wireframe no hay que especificar el tipo de letra, el color de cada uno de sus páginas y gráficos, se trata de representar su funcionamiento.

### 3.10.2 Desventajas

- ✓ Los wireframes no incluyen el diseño gráfico de la página web, para el cliente no es tan fácil comprender el concepto de elementos en blanco y negro.
- ✓ Cuando se agrega el contenido al sitio, la información siempre puede ser más de lo que se definió en el diseño del wireframe, por lo que el diseñador deberá trabajar en hacer el ajuste.

### 3.10.3 Tipos

Los Wireframes se clasifican en dos categorías:

**Baja Fidelidad:** Es un dibujo rápido o boceto en bruto, los wireframes de baja fidelidad no poseen muchos detalles y se elaboran con mayor rapidez. Se realizan utilizando símbolos abstractos como etiquetas y rectángulos para la representación del contenido, permitiendo al equipo de proyecto trabajar efectivamente.

En la figura 44 podemos ver un diseño de wireframe de baja fidelidad, ya que está estructurado por cuadrados, rectángulos y leyendas que indican la distribución del contenido del sitio web.





Figura 44. Representación de wireframe de baja fidelidad.

Tomado de (Maquetando, s.f.)

**Alta Fidelidad:** Su diseño se asemeja al del producto final del sitio web, los wireframe de alta fidelidad son utilizados para documentar ya que incorporan más detalles para su elaboración, pero toman mayor tiempo para su creación.

Esta técnica puede ser utilizada en la etapa inicial del proceso de desarrollo cuando aún no se dispone de la interfaz real, cuanto sea mejor sea el diseño del prototipo mucho mejor serán los resultados obtenidos para el sitio web.

En la figura 45 vemos un ejemplo de wireframe de alta fidelidad donde se observa cómo quedará el producto final. Este diseño tiene varios detalles como gráficos, logo y enlaces, los cuales muestran claramente como está distribuido el contenido y la navegación de la página web.



Figura 45. Representación de wireframe de alta fidelidad.

Tomado de (Maquetando, s.f.)

### 3.10.4 Elementos

El cuerpo de una página web se puede dividir en tres elementos:

#### ✓ **Diseño de información:**

Es la forma de presentar y ubicar la información de tal manera que facilite la comprensión del diseño de la página web. Para los sitios es necesario que los elementos informativos estén ordenados, estos deben reflejar los objetivos y las tareas que debe realizar el usuario.

En la figura 46 se observa un ejemplo de diseño de información, donde vemos la información ubicada en secciones como: cultura, energía, sostenibilidad, las noticias destacadas y consejos; haciendo fácil la comprensión del sitio web.

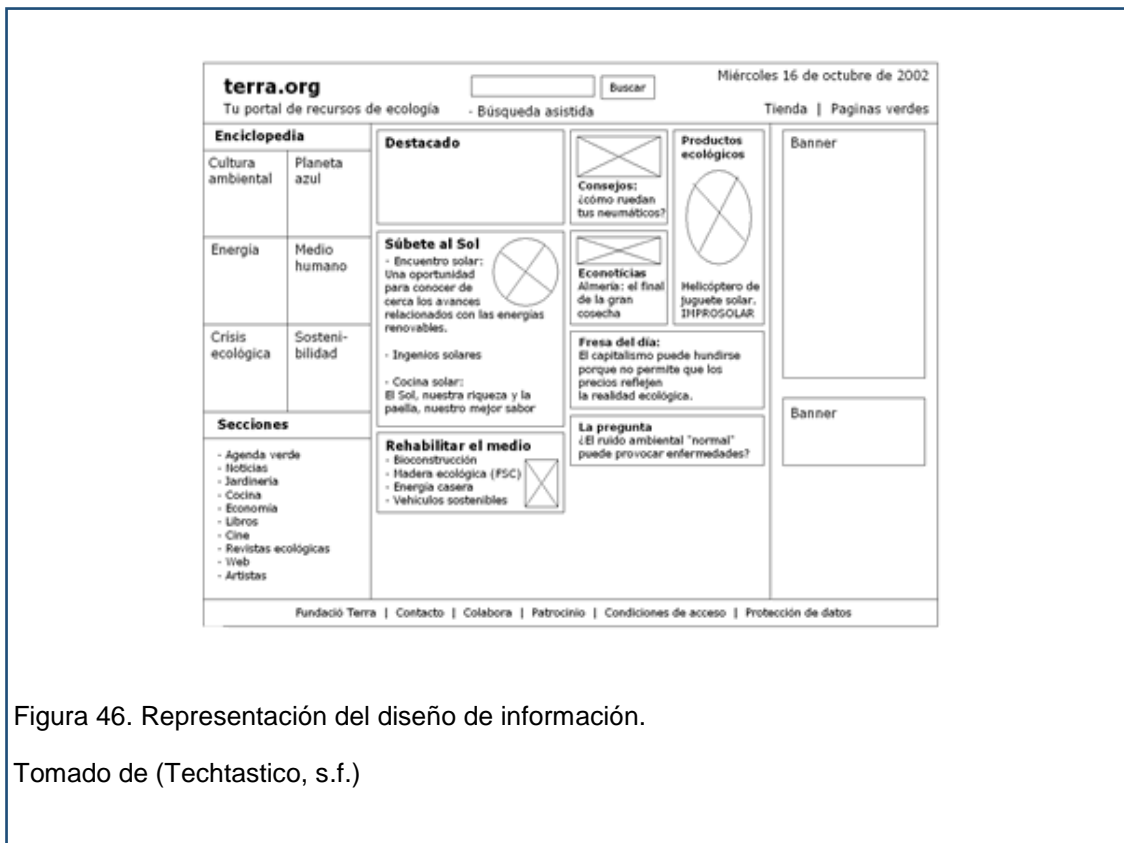


Figura 46. Representación del diseño de información.

Tomado de (Techastico, s.f.)

### ✓ Diseño de navegación

Este diseño permite al usuario interactuar entre páginas dentro del mismo sitio web, ya que posee un conjunto de pantallas.

El diseño de navegación comunica la relación que existe en los enlaces con la finalidad que el usuario conozca la manera de navegar dentro de la aplicación.

En la figura 47 se muestra el diseño de navegación y como permite dar una vista de manera general del desplazamiento y ubicación a través de las diferentes páginas que tiene el sitio web.

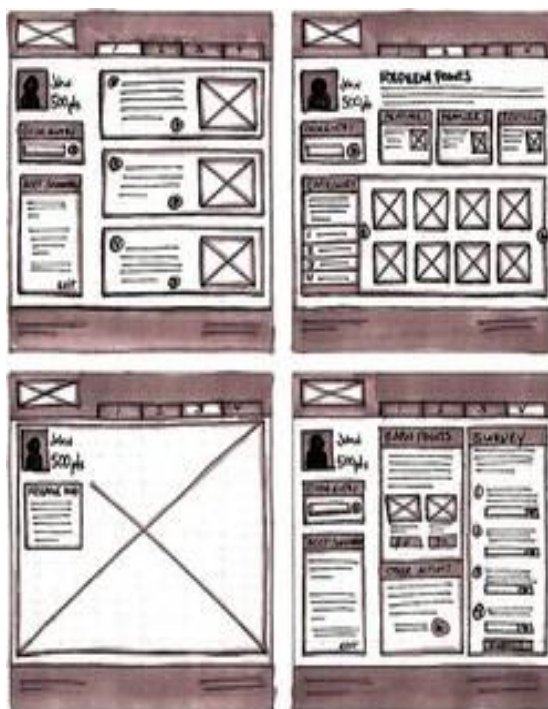


Figura 47. Representación de diseño de navegación.

Tomado de (Techastico, s.f.)

### ✓ **Diseño de interfaz**

Este diseño ordena y organiza los elementos de la interfaz de tal manera que permita al usuario interactuar y manipular todas las funcionalidades que el posee el sistema.

El objetivo del diseño de interfaz es facilitar el uso y eficiencia en el sitio web. Los elementos utilizados en este diseño son los botones, campos de texto y menús desplegables.

En la figura 48 observamos el diseño de interfaz de una página web, donde se pueden apreciar los botones y menús desplegables con la finalidad que el usuario pueda manipular las funcionalidades que posee el sitio.



Figura 48. Representación de diseño de interfaz.

Tomado de (Masternewmedia, s.f.)

### 3.11 Lenguaje Javascript

Javascript es un lenguaje de programación que permite construir sitios web más interactivos, este lenguaje es interpretado en el cliente por el navegador al momento de cargarse el sitio.

Su código se incluye de manera directa en el código HTML, facilitando a los desarrolladores web realizar un contenido más dinámico.

El lenguaje Javascript es OpenSource gracias a esto puede ser utilizado por cualquier persona sin necesidad de comprar una licencia para su uso. Este lenguaje es soportado por la mayoría de los navegadores como son: Internet Explorer, Google Chrome, Mozilla Firefox, Opera, etc.

#### Características

- ✓ Facilidad para crear interfaces de usuario.
- ✓ Soporta html5 y utilizando sus etiquetas podemos ocupar javascript de una manera muy fácil.

- ✓ Es soportado por la mayoría de dispositivos móviles.
- ✓ Viene como themes predeterminados y permite la creación de themes personalizados.

### **Ventajas**

- ✓ Incluye barras de herramientas, botones, entre otros objetos optimizados para ser utilizados en dispositivos móviles.
- ✓ Permite desarrollar páginas webs para diferentes dispositivos móviles.
- ✓ Incluye soporte para ajax lo cual nos ayuda con la creación de páginas dinámicas.
- ✓ No requiere el uso de programas especiales para su utilización.

### **Desventajas**

- ✓ Ofrece muchas funciones pero algunas son difíciles para ser personalizadas.
- ✓ En algunas ocasiones la carga de las páginas es lenta por invocación de librerías.
- ✓ No existen muchas plantillas pre diseñadas.

## CAPÍTULO IV

### Análisis

El presente capítulo explica el desarrollo del prototipo de Sistema de administración del ganado bovino de la granja experimental de la Universidad de las Américas, bautizado con el nombre de FARMWARE, para el desarrollo del aplicativo se utiliza la metodología de desarrollo ágil Scrum.

#### 4.1 Requerimientos del sistema

Se analizan los requerimientos funcionales y no funcionales de la aplicación y se evalúan las condiciones sobre las cuales se desarrollara el sistema; con el fin de cumplir la mayoría de necesidades de la granja experimental de la Universidad de las Américas.

#### Requerimientos funcionales

Los requerimientos funcionales son aquellos que el sistema va a realizar. Las funciones son:

- ✓ **Módulo de producción de leche.**

En el módulo se realiza el registro del Control lechero y el ingreso del control de leche por animal, de esta manera se conocerá cuantos litros el animal produce por ordeñada.

- ✓ **Módulo de manejo de ganado**

En el módulo se realiza el ingreso de datos de un nuevo animal, como su nombre, fecha de nacimiento, nombre de la madre y del progenitor; se tendrá inventariado todas las vacas con sus características.

- ✓ **Módulo de gestión de reproducción**

En este módulo se maneja todo lo referente a preñez, número de partos, fecha de parto, sexo de la cría y además permite agregar un servicio.

- ✓ **Módulo de gestión alimenticia**

En este módulo se registra peso y ganancia de cada animal joven.

✓ **Módulo de ingresos y egresos**

En el módulo se registra los gastos e ingresos, ventas de la producción de leche.

✓ **Módulo de resumen**

En el módulo de resumen se generan los históricos de ciertos módulos, estos son representados de manera gráfica.

### **Requerimientos No Funcionales**

Son los requerimientos que son necesarios para que el sistema funcione de manera correcta, pero no son parte de las funciones que la aplicación realiza.

Los requerimientos no funcionales se detallan a continuación:

- ✓ Para acceder al sistema el usuario debe estar en la red de la Universidad, el aplicativo va a ser utilizado por el departamento de Administración de la Granja.
- ✓ Para ingresar al sistema se deberá acceder a través de un explorador tales como: Chrome, Firefox, Internet Explorer.

### **4.2 Planificación de Sprints y sus historias de usuarios.**

En la etapa de planificación de sprints se realiza la toma de requisitos al cliente, por cada requisito se detalla una historia de usuario.

**Historias de usuarios:** Indican la funcionalidad que debe tener el sistema, su implementación aporta valor al cliente.

#### **Elementos de la historia de usuario**

El formato a llevarse a cabo en las historias de usuarios es el siguiente:

- ✓ **Número de Historia de usuario:** es un identificador único de cada historia de usuario está conformado por el número de sprint (dos



dígitos), las siglas del módulo (dos caracteres), y el número de historia de usuario (dos dígitos) al que corresponde.

- ✓ **Rol:** es el rol que desempeña el usuario que utiliza la funcionalidad que se está describiendo en la historia.
- ✓ **Fecha:** es la fecha en la que se especificó la historia de usuario.
- ✓ **Nombre de Historia de usuario:** es una descripción corta de lo que realiza el requerimiento.
- ✓ **Prioridad en negocio:** muestra el grado de importancia que tiene la historia de usuario en el desarrollo del proyecto. La prioridad más alta tendrá el valor de 1 y la prioridad más baja tendrá el valor de 5.
- ✓ **Sprint asignado:** es el número de sprint al que corresponde la historia de usuario.
- ✓ **Programador responsable:** es el nombre del integrante del equipo que está desarrollando la historia de usuario.
- ✓ **Descripción:** representa la función que el rol necesita realizar en el sistema que se desarrolla.

Una vez creadas las historias de usuario, el cliente las ordena en el product backlog según la prioridad de entrega y se definen los sprints.

### **SPRINT 1:**

Para nuestro primer sprint el equipo de trabajo toma un grupo de historias de usuarios las cuales se detallan a continuación (ver anexo 2).

Tabla 4. Instalación del entorno de desarrollo

<b>Número:</b> 01PL01	<b>Rol:</b> Programador	<b>Fecha:</b> 05/01/2015
<b>Nombre de historia:</b> Instalación de las herramientas de desarrollo.		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 5	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> Se instalará Visual Studio 2012, SQL Server 2008, Power Designer.		
<b>Validación:</b> Las herramientas se encuentran instaladas.		

Tabla 5. Modelado del diagrama Entidad – Relación

Historia de Usuario		
<b>Número:</b> 01PL02	<b>Rol:</b> Programador	<b>Fecha:</b> 08/01/2015
<b>Nombre de historia:</b> Diseñar el diagrama Entidad - Relación		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 9	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> Se modelará el diagrama entidad relación en la herramienta Power Designer.		
<b>Validación:</b> Se modeló el diagrama entidad relación.		

Tabla 6. Modelado del diagrama Físico

Historia de Usuario		
<b>Número:</b> 01PL03	<b>Rol:</b> Programador	<b>Fecha:</b> 11/01/2015
<b>Nombre de historia:</b> Elaboración del Diagrama físico		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 1	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> Elaborar el diagrama físico para el modelado en la base de datos.		
<b>Validación:</b> El diagrama físico se modeló para la construcción de las tablas.		

Tabla 7. Funciones CRUD

Historia de Usuario		
<b>Número:</b> 01PL04	<b>Rol:</b> Programador	<b>Fecha:</b> 18/01/2015
<b>Nombre de historia:</b> Realizar CRUD		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 5	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> Elaborar CRUD funciones create, read, update, delete.		
<b>Validación:</b> Las funciones básicas se encuentran creadas en la capa de persistencia de la aplicación.		

**SPRINT 2:**

El segundo sprint contiene en su grupo de historias de usuario la realización de los siguientes módulos para el aplicativo web: módulo de gestión alimenticia y de gestión de manejo de animales (ver anexo 3).

Tabla 8. Módulo de gestión alimenticia

Historia de Usuario		
<b>Número:</b> 02PL01	<b>Rol:</b> Administrador	<b>Fecha:</b> 25/01/2015
<b>Nombre de historia:</b> Módulo de gestión de alimenticia		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 9	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá llevar el registro de alimentación de los animales de la granja.		
<b>Validación:</b> El administrador puede gestionar la alimentación de los animales.		

Tabla 9. Formulario crear nuevos pesajes

Historia de Usuario		
<b>Número:</b> 02PL02	<b>Rol:</b> Administrador	<b>Fecha:</b> 30/01/2015
<b>Nombre de historia:</b> Formulario para la crear nuevos pesajes		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 5	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá registrar un nuevo pesaje en el módulo de gestión alimenticia.		
<b>Validación:</b> El administrador puede registrar un nuevo pesaje.		

Tabla 10. Reporte animales pesados

Historia de Usuario		
<b>Número:</b> 02PL03	<b>Rol:</b> Administrador	<b>Fecha:</b> 01/02/2015
<b>Nombre de historia:</b> Reporte los animales pesados.		
<b>Prioridad en negocio:</b> 2	<b>Estimación:</b> 4	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá ver el detalle de pesaje de animales, el mismo contiene los campos de Nombre de animal, fecha de peso, peso y comentarios.		
<b>Validación:</b> El administrador puede ver el reporte de pesaje en el sistema.		

Tabla 11. Módulo de gestión de manejo

Historia de Usuario		
<b>Número:</b> 02PL04	<b>Rol:</b> Administrador	<b>Fecha:</b> 03/02/2015
<b>Nombre de historia:</b> Modulo de gestión de manejo.		
<b>Prioridad en negocio:</b> 2	<b>Estimación:</b> 6	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá gestionar el manejo de animales desde el módulo de manejo de ganado.		
<b>Validación:</b> El administrador puede registrar nuevos animales en el sistema.		

Tabla 12. Formulario crear animal

Historia de Usuario		
<b>Número:</b> 02PL05	<b>Rol:</b> Administrador	<b>Fecha:</b> 05/02/2015
<b>Nombre de historia:</b> Formulario para crear animal.		
<b>Prioridad en negocio:</b> 2	<b>Estimación:</b> 8	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá crear animales en el sistema, los campos obligatorios para la creación son nombre del ganado, especie, sexo, nombre del progenitor, nombre de la progenitora, rodeo y comentarios.		
<b>Validación:</b> El administrador puede crear animal en el sistema.		

Tabla 13. Reporte animales creados

Historia de Usuario		
<b>Número:</b> 02PL06	<b>Rol:</b> Administrador	<b>Fecha:</b> 15/02/2015
<b>Nombre de historia:</b> Reporte de animales creados.		
<b>Prioridad en negocio:</b> 2	<b>Estimación:</b> 4	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá generar un reporte de los animales que han sido creados en la granja.		
<b>Validación:</b> El administrador puede crear animales en el sistema.		

**SPRINT 3:**

El tercer sprint contiene en su grupo de historias de usuarios el desarrollo por parte del equipo de trabajo de los siguientes módulos: Módulo de producción y Módulo de reproducción (ver anexo 4).

Tabla 14. Módulo de gestión de reproducción

Historia de Usuario		
<b>Número:</b> 03PL01	<b>Rol:</b> Administrador	<b>Fecha:</b> 20/02/2015
<b>Nombre de historia:</b> Modulo de gestión de reproducción.		
<b>Prioridad en negocio:</b> 3	<b>Estimación:</b> 9	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá gestionar lo referente a la reproducción de animales en la granja.		
<b>Validación:</b> El administrador puede ingresar a la vista del módulo de reproducción.		

Tabla 15. Formulario registro de parto

Historia de Usuario		
<b>Número:</b> 03PL02	<b>Rol:</b> Administrador	<b>Fecha:</b> 25/02/2015
<b>Nombre de historia:</b> Formulario de registro de parto.		
<b>Prioridad en negocio:</b> 3	<b>Estimación:</b> 6	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá crear un nuevo parto, los campos obligatorios para el formulario son: fecha de parto, sexo de la cría, fecha de servicio.		
<b>Validación:</b> El administrador puede crear un nuevo parto en el módulo.		

Tabla 16. Formulario registro de nuevo servicio

Historia de Usuario		
<b>Número:</b> 03PL03	<b>Rol:</b> Administrador	<b>Fecha:</b> 05/03/2015
<b>Nombre de historia:</b> Formulario de registro de un nuevo servicio.		
<b>Prioridad en negocio:</b> 3	<b>Estimación:</b> 5	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá crear un nuevo servicio, los campos obligatorios para el formulario son: fecha de servicio, y método de reproducción.		
<b>Validación:</b> El administrador puede crear un nuevo parto en el módulo de reproducción.		

Tabla 17. Reporte de servicios y partos creados

Historia de Usuario		
<b>Número:</b> 03PL04	<b>Rol:</b> Administrador	<b>Fecha:</b> 15/03/2015
<b>Nombre de historia:</b> Reporte de los servicios y partos creados.		
<b>Prioridad en negocio:</b> 3	<b>Estimación:</b> 6	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá generar un reporte de los partos y servicios ingresados en la aplicación.		
<b>Validación:</b> El administrador puede ver los resultados que arroja el reporte en el módulo de reproducción.		

Tabla 18. Módulo gestión de producción

Historia de Usuario		
<b>Número:</b> 03PL05	<b>Rol:</b> Administrador	<b>Fecha:</b> 19/03/2015
<b>Nombre de historia:</b> Modulo de gestión de producción.		
<b>Prioridad en negocio:</b> 3	<b>Estimación:</b> 6	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá ingresar dentro de la aplicación al módulo de producción, y realizar el registro de la producción de leche de la granja.		
<b>Validación:</b> El administrador puede acceder a la vista del módulo de producción.		

Tabla 19. Formulario crear control lechero

Historia de Usuario		
<b>Número:</b> 03PL06	<b>Rol:</b> Administrador	<b>Fecha:</b> 21/03/2015
<b>Nombre de historia:</b> Formulario para crear control lechero.		
<b>Prioridad en negocio:</b> 3	<b>Estimación:</b> 5	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá crear el nuevo control lechero, los campos obligatorios para este formulario son: fecha, comentarios y responsable.		
<b>Validación:</b> El administrador puede ingresar datos en el formulario dentro del módulo de producción.		

Tabla 20. Formulario crear control lechero por animal

Historia de Usuario		
<b>Número:</b> 03PL07	<b>Rol:</b> Administrador	<b>Fecha:</b> 30/03/2015
<b>Nombre de historia:</b> Formulario para crear control lechero por animal.		
<b>Prioridad en negocio:</b> 3	<b>Estimación:</b> 6	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá crear el nuevo control lechero de cada animal que ha sido ordeñado, los campos obligatorios para este formulario son: nombre del animal, ordeñada 1, ordeñada 2 y ordeñada 3.		
<b>Validación:</b> El administrador puede ingresar los datos y agregar un nuevo control lechero de cada animal en el módulo.		

**SPRINT 4:**

El cuarto sprint contiene en su grupo las historias de usuarios detalladas a continuación (ver anexo 5).

Tabla 21. Reporte detalle ordeñada

Historia de Usuario		
<b>Número:</b> 04PL01	<b>Rol:</b> Administrador	<b>Fecha:</b> 01/04/2015
<b>Nombre de historia:</b> Reporte del detalle de ordeñada.		
<b>Prioridad en negocio:</b> 4	<b>Estimación:</b> 4	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá generar el reporte del detalle de las ordeñadas del animal.		
<b>Validación:</b> El administrador puede generar el reporte de detalle de ordeñada en el módulo de producción.		



Tabla 22. Módulo de gestión de ingresos y egresos

Historia de Usuario		
<b>Número:</b> 04PL02	<b>Rol:</b> Administrador	<b>Fecha:</b> 05/04/2015
<b>Nombre de historia:</b> Modulo de gestión de ingresos y egresos.		
<b>Prioridad en negocio:</b> 4	<b>Estimación:</b> 4	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá ingresar a la vista del módulo de ingresos y egresos.		
<b>Validación:</b> El administrador puede acceder la vista de ingresos y egresos en la aplicación.		

Tabla 23. Formulario registro de ingreso por actividad

Historia de Usuario		
<b>Número:</b> 04PL03	<b>Rol:</b> Administrador	<b>Fecha:</b> 09/04/2015
<b>Nombre de historia:</b> Formulario de registro de ingreso por actividad.		
<b>Prioridad en negocio:</b> 4	<b>Estimación:</b> 4	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá ingresar los datos del ingreso por actividad, en este caso por venta de la leche producida en la granja.		
<b>Validación:</b> El administrador puede generar el formulario de registro de ingresos por actividad.		

Tabla 24. Formulario registro de egreso por actividad

✓ Historia de Usuario		
✓ <b>Número:</b> 04PL05	✓ <b>Rol:</b> Administrador	✓ <b>Fecha:</b> 15/04/2015
✓ <b>Nombre de historia:</b> Formulario de registro de egreso por actividad.		
✓ <b>Prioridad en negocio:</b> 4	✓ <b>Estimación:</b> 5	
✓ <b>Programador responsable:</b> Sindy Vera		
✓ <b>Descripción:</b> El administrador podrá ingresar los datos del egreso por actividad.		
✓ <b>Validación:</b> El administrador puede generar el formulario de registro de egreso por actividad.		

Tabla 25. Módulo de resumen

Historia de Usuario		
<b>Número:</b> 04PL06	<b>Rol:</b> Administrador	<b>Fecha:</b> 25/04/2015
<b>Nombre de historia:</b> Modulo de resumen		
<b>Prioridad en negocio:</b> 4	<b>Estimación:</b> 5	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá acceder a la vista del módulo de resumen dentro de la aplicación, la misma que contiene varios reportes.		
<b>Validación:</b> El administrador puede ingresar a la vista del módulo de resumen.		

Tabla 26. Reporte de producción

Historia de Usuario		
<b>Número:</b> 04PL07	<b>Rol:</b> Administrador	<b>Fecha:</b> 15/05/2015
<b>Nombre de historia:</b> Reporte de Producción		
<b>Prioridad en negocio:</b> 4	<b>Estimación:</b> 4	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá generar el reporte de los litros por ordeñada de cada animal.		
<b>Validación:</b> El administrador puede generar el reporte de producción.		

Tabla 27. Reporte número de partos por animal

Historia de Usuario		
<b>Número:</b> 04PL08	<b>Rol:</b> Administrador	<b>Fecha:</b> 25/07/2015
<b>Nombre de historia:</b> Reporte Número de partos por animal.		
<b>Prioridad en negocio:</b> 4	<b>Estimación:</b> 5	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá generar el reporte de número de partos por animal dentro de la granja.		
<b>Validación:</b> El administrador puede generar el reporte de número de partos dentro del módulo de resumen.		

Tabla 28. Reporte promedio histórico de producción por rodeo

Historia de Usuario		
<b>Número:</b> 04PL09	<b>Rol:</b> Administrador	<b>Fecha:</b> 25/08/2015
<b>Nombre de historia:</b> Reporte Promedio histórico de Producción por rodeo.		
<b>Prioridad en negocio:</b> 4	<b>Estimación:</b> 4	
<b>Programador responsable:</b> Sindy Vera		
<b>Descripción:</b> El administrador podrá generar el reporte de promedio histórico de producción por rodeo de la granja.		
<b>Validación:</b> El administrador puede generar el reporte dentro del módulo de resumen.		

### 4.3 Diseño de la aplicación

Para el diseño del aplicativo web se realizaron los siguientes diagramas expuestos a continuación:

- ✓ Diagrama de clase.
- ✓ Diagrama Entidad – Relación.
- ✓ Diagrama de actividades.
- ✓ Diagrama de secuencia.
- ✓ Diccionario de datos (ver anexo 10)

#### 4.3.1 Diagrama de clases

Diagrama de clases del prototipo de sistema de administración del ganado bovino de la Universidad de las Américas.

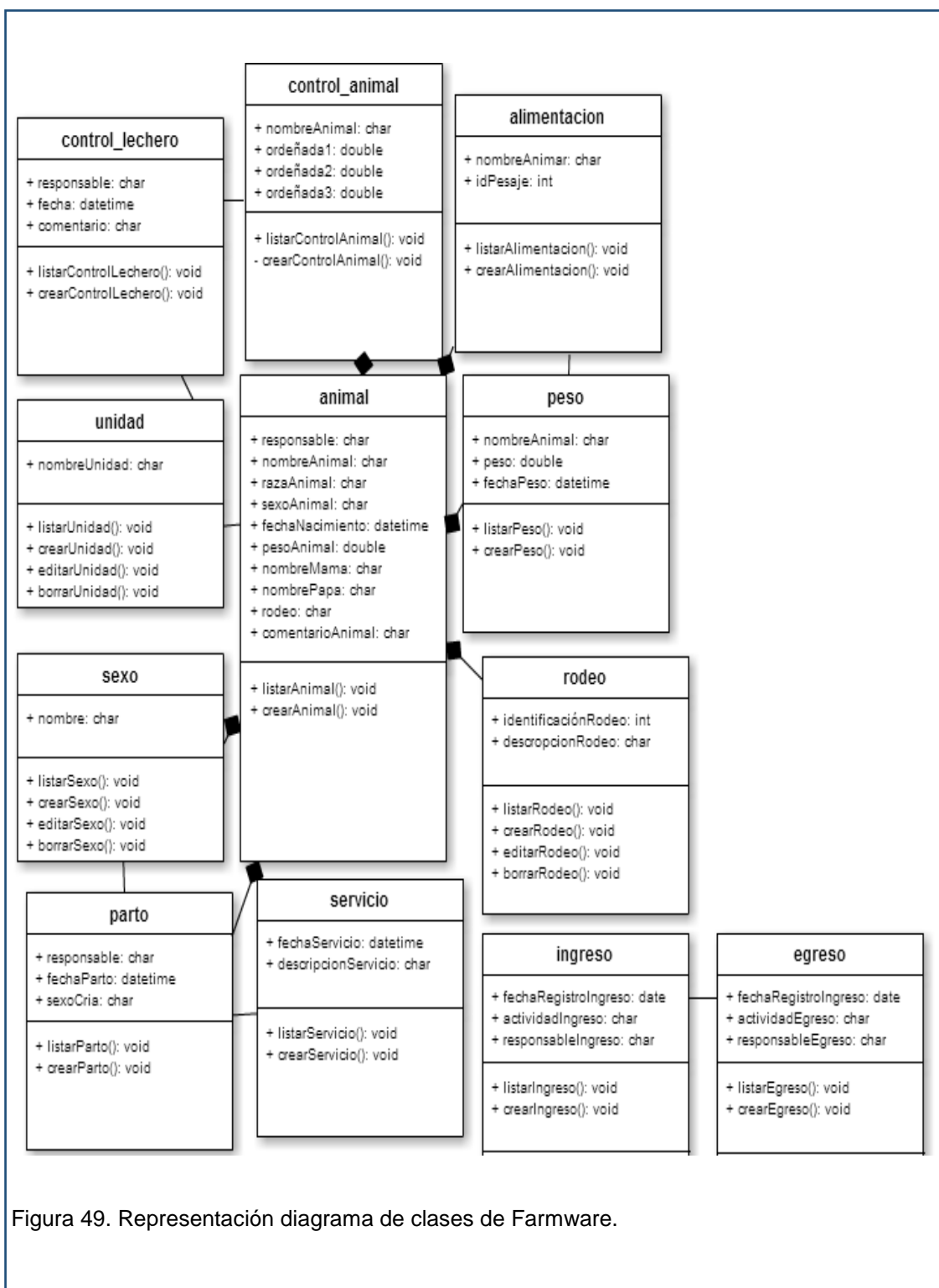


Figura 49. Representación diagrama de clases de Farmware.

### 4.3.2 Diagrama Entidad - Relación

Modelado de base de datos del diagrama Entidad – Relación del aplicativo web Farmware.

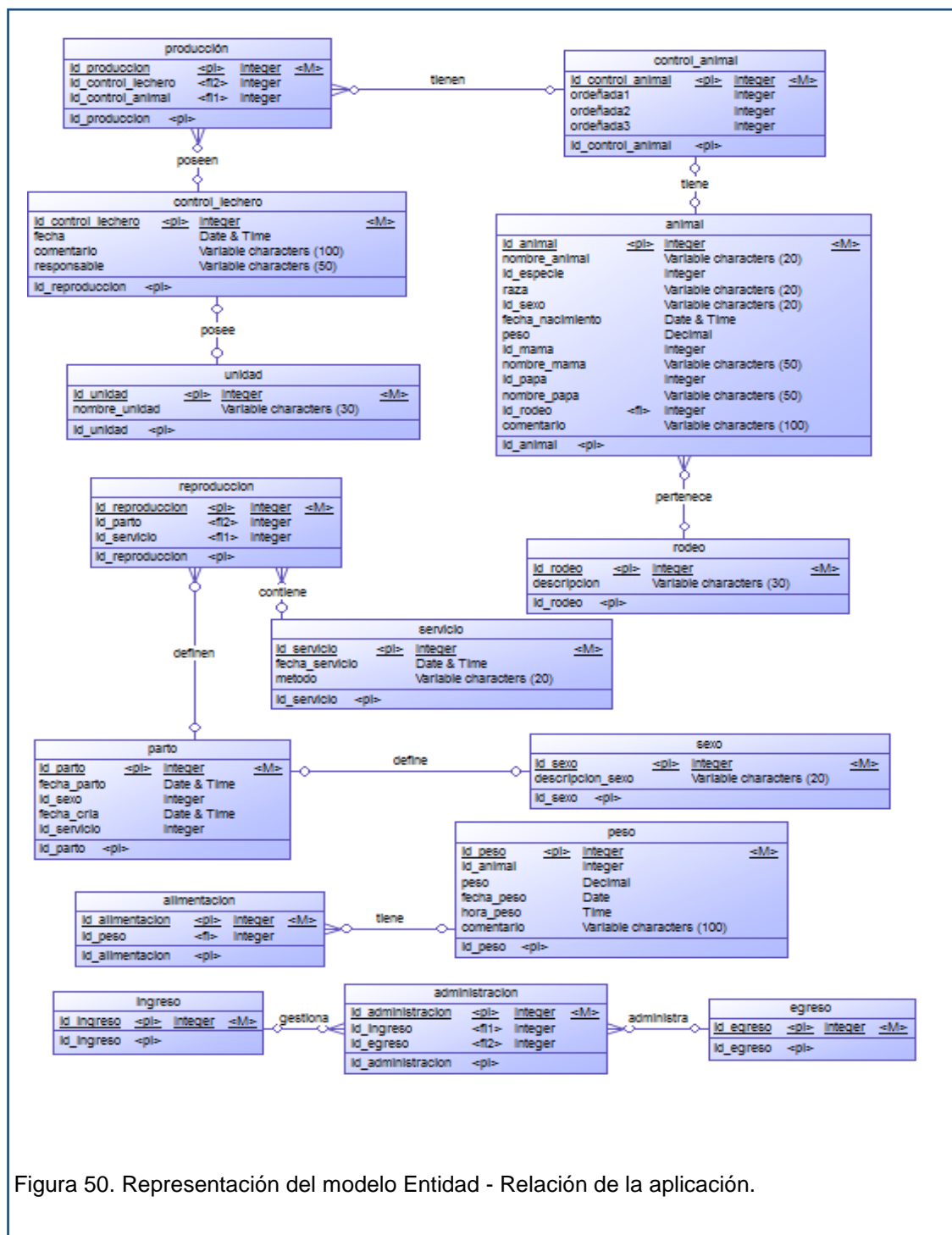
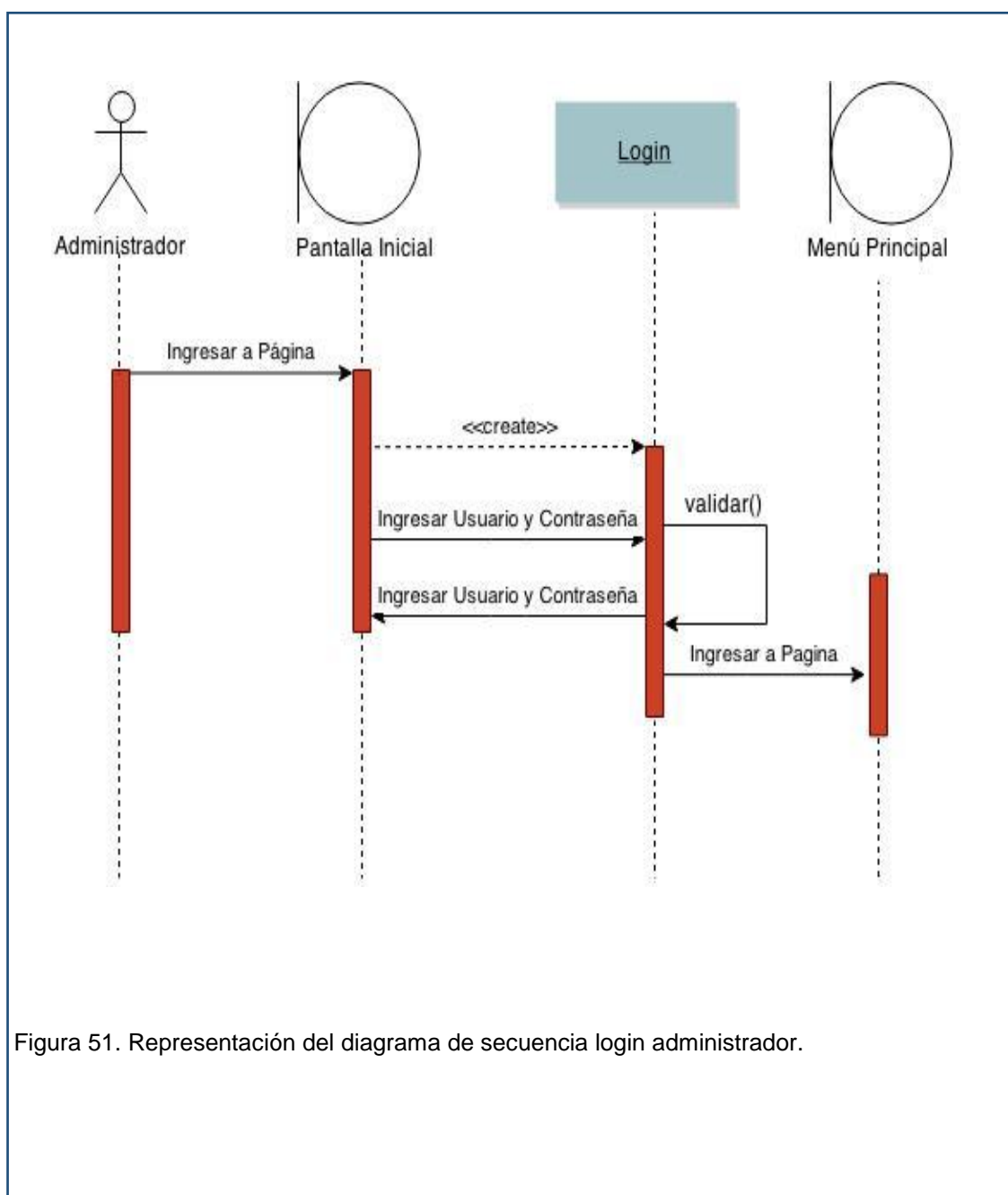


Figura 50. Representación del modelo Entidad - Relación de la aplicación.

### 4.3.3 Diagrama de secuencia

Se representaran los diagramas de secuencia de la aplicación.

#### Diagrama de secuencia Login del usuario administrador



### Diagrama de secuencia administrar módulo de producción de leche.

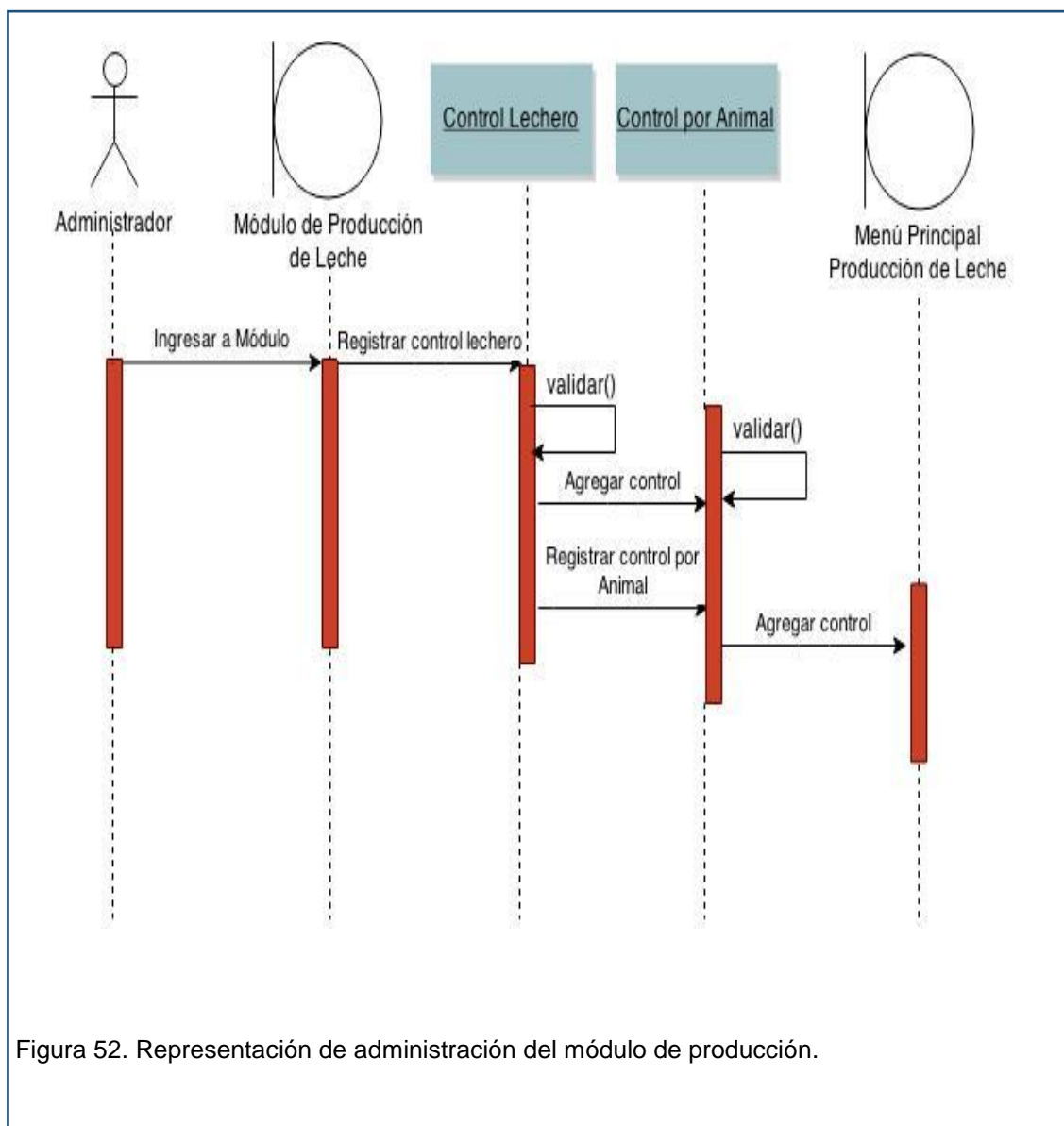


Figura 52. Representación de administración del módulo de producción.

En la sección de anexo 7 del proyecto de titulación se encuentran los diagramas de secuencia restantes; realizados para la elaboración del aplicativo Farmware.

#### 4.3.4 Diagrama de actividades

Se representaran los siguientes diagramas de actividades.

##### Diagrama de actividad de login de administrador

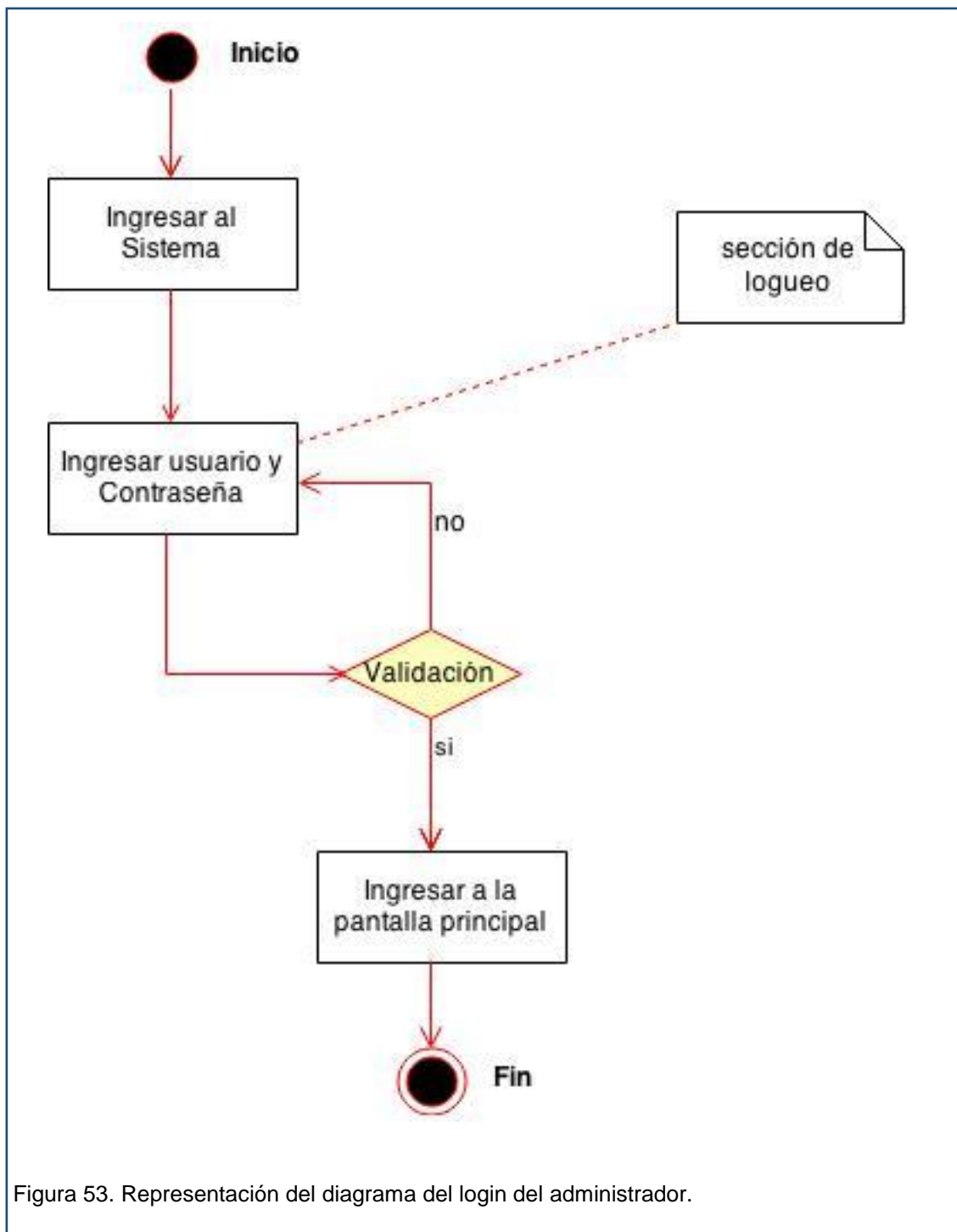


Figura 53. Representación del diagrama del login del administrador.



### Diagrama de actividad listar un animal pesado

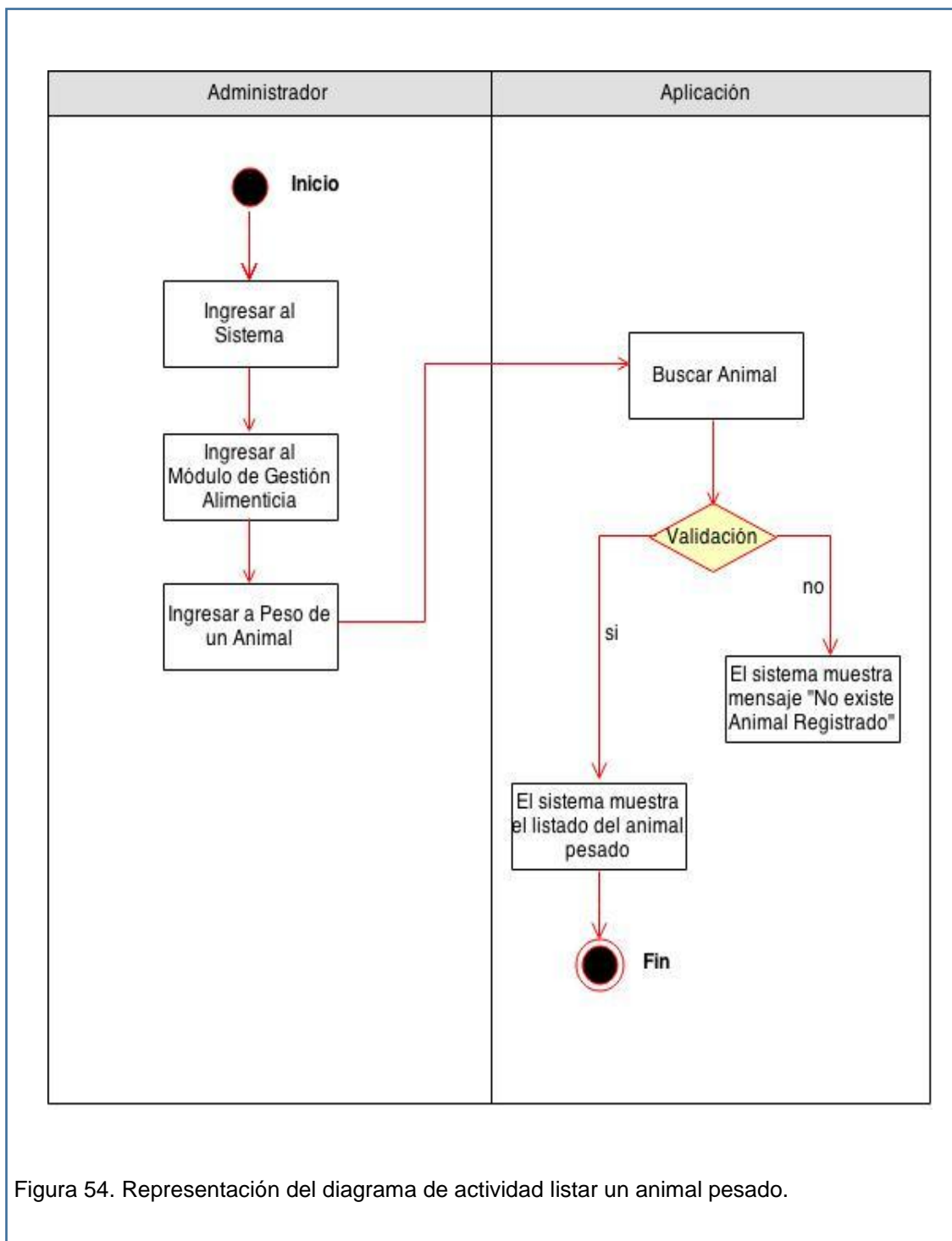


Figura 54. Representación del diagrama de actividad listar un animal pesado.

En anexo 8 se encuentran los diagramas de actividades faltantes del aplicativo Farmware.

## CAPÍTULO V

### Conclusiones y recomendaciones

#### 5.1. Conclusiones

- ✓ La utilización de Wireframes permite al usuario tener una idea clara y precisa de cómo se va a distribuir la información del proyecto de tesis.
- ✓ El diseñar los diagramas de caso de uso del sistema proporciona al usuario la facilidad de interpretar de manera sencilla la funcionalidad de cada proceso, logrando una comunicación más óptima con el cliente.
- ✓ El utilizar la metodología ágil SCRUM para la elaboración de este trabajo de titulación, permite entregar un prototipo funcional en la culminación de cada fase.
- ✓ El automatizar el proceso de registro de animales mejora de manera eficiente los tiempos de respuesta al tomar decisiones basadas en los resultados proporcionados por el sistema.
- ✓ El trabajar con pantallas adaptativas garantiza que el aplicativo pueda ser visualizado desde cualquier dispositivo móvil.

## 5.2 Recomendaciones

- ✓ El usuario debe tener claro los requerimientos del sistema antes de empezar con el desarrollo del software, para evitar la pérdida de tiempo.
- ✓ Antes de escoger una herramienta de desarrollo el estudiante debe cerciorarse que exista toda la documentación necesaria que permita la elaboración del proyecto de titulación.
- ✓ Al finalizar cada fase del proyecto es necesario que el cliente revise el producto entregable por si aparecen nuevos requerimientos es necesario que se los registre en ese momento.
- ✓ Debido a que la aplicación está desarrollada como prototipo, en el caso de que llegue a colocarse en producción es necesario definir políticas de seguridad de intranet, ya que esto no fue considerado en el alcance.

## REFERENCIAS

- Aguilar, M. (2010). *Edición elegante de fechas con jQuery*. Recuperado el 18 de Julio de 2015 de <http://www.variablenotfound.com/2010/06/edicion-elegante-de-fechas-con-jquery.html>
- Aroca, G. (2015). *DropDownList Razor in ASP.NET MVC*. Recuperado el 20 de Agosto de 2015 de <http://germanaroca.wordpress.com/>.
- Crispin, L. y Gregory, J. (2009). *Agile Testing A Practical Guide for Testers and Agile Teams*. United States: Pearson Education.
- Datepicker, (2014). *Paso a paso MVC*. Recuperado el 23 de Agosto de 2015 de <http://www.variablenotfound.com/2010/06/edicion-elegantedefechascon>
- Información Tecnológica, (2012). *Extensión del Diagrama de Secuencias UML*. Recuperado el 27 de Junio de 2015 de <http://www.scielo.cl/pdf/infotec/v23n6/art07.pdf>.
- Microsoft, (2013). *Acerca De Visual Studio*. Recuperado el 13 de Septiembre de 2015 de <http://msdn.microsoft.com/es-es/>
- Palacio, F. (2007). *Flexibilidad con Scrum*. Estados Unidos: Safe Creative.
- Respyitco, (2014). *Lenguaje Unificado de Modelado UML*. Recuperado el 13 de Agosto de 2015 de <http://es.scribd.com/doc/35020783/Lenguaje-Unificado-de-Modelado-UML/>.
- Salinas Caro, P. (s.f.). *Universidad de Chile Casos de Uso*. Recuperado el 10 de septiembre de 2015 de <http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>
- Schools. (s.f). *Que es HTML5*. Recuperado el 23 de marzo de 2015 de [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)
- Schwaber, K. y Sutherland J. (2010). *Scrum Guide*. United States: C4Media.
- Scott, M. (2010). *Displaying a Grid of Data in ASP.NET MVC*. Recuperado el 11 de Agosto de 2015 de <http://www.4guysfromrolla.com/articles/121510-1.aspx/>.
- Softeng, (2014). *Metodología Scrum para desarrollo de software*. Recuperado el 14 de Septiembre de 2015 de <http://www.softeng.es/es-es/empresa/metodologias-detrabajo/metodologia-scrum.html>.

- Sookman, J. (2012). *Startup Life*. Recuperado el 11 de septiembre de 2015 de <http://www.startuplifeblog.com/tag/product-backlog/>
- Storti, G., Ríos, G., y Campodónico, G. (2007). *Colegio Manuel Belgrano Estudios*. Recuperado el 7 de marzo de 2014 de [http://www.belgrano.esc.edu.ar/matestudio/carpeta\\_de\\_access\\_introduccion.pdf](http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf)
- Storti, G., Ríos, G., y Campodónico, G. (2007). *Colegio Manuel Belgrano Estudios Actuales*. Recuperado el 7 de marzo de 2013 de [http://www.belgrano.esc.edu.ar/matestudio/carpeta\\_de\\_access\\_introduccion.pdf](http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf)

## **ANEXOS**

## GLOSARIO

**Wireframes:** es una guía visual para el usuario ya que representa el esqueleto de un programa.

**Html5: (HyperText Markup Language, versión 5)** es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

**Ide de desarrollo: (Integrated Development Environment),** es un programa informático integrado por un conjunto de herramientas de programación, un ide es un editor de código, un compilador, un depurador y un constructor de una interfaz gráfica.

**C#:** (C Sharp) es un lenguaje de programación orientado a objetos de desarrollo se encuentra estandarizado por Microsoft dentro de su plataforma .net

**Framework:** infraestructura digital, los frameworks usan patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento.

**.NET:** La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows.

**Interfaz:** En informática se utiliza para nombrar a la conexión física y funcional entre dos sistemas dando una comunicación entre distintos niveles.

**Interfaz de usuario:** La interfaz de usuario es el medio con el que el usuario puede comunicarse con una máquina o un equipo, y comprende todos los puntos de contacto entre el usuario y el computador.

**Scrum:** Es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

**Servicio:** es el acto de reproducción, el mismo que puede ser por monta natural del animal o mediante inseminación artificial. Es el comienzo del ciclo de reproducción que termina al producirse el parto de la hembra.

**Control lechero:** El control lechero ofrece al productor del establecimiento la posibilidad de conocer la calidad y cantidad estimada de producción en la lactancia de cada una de sus hembras.



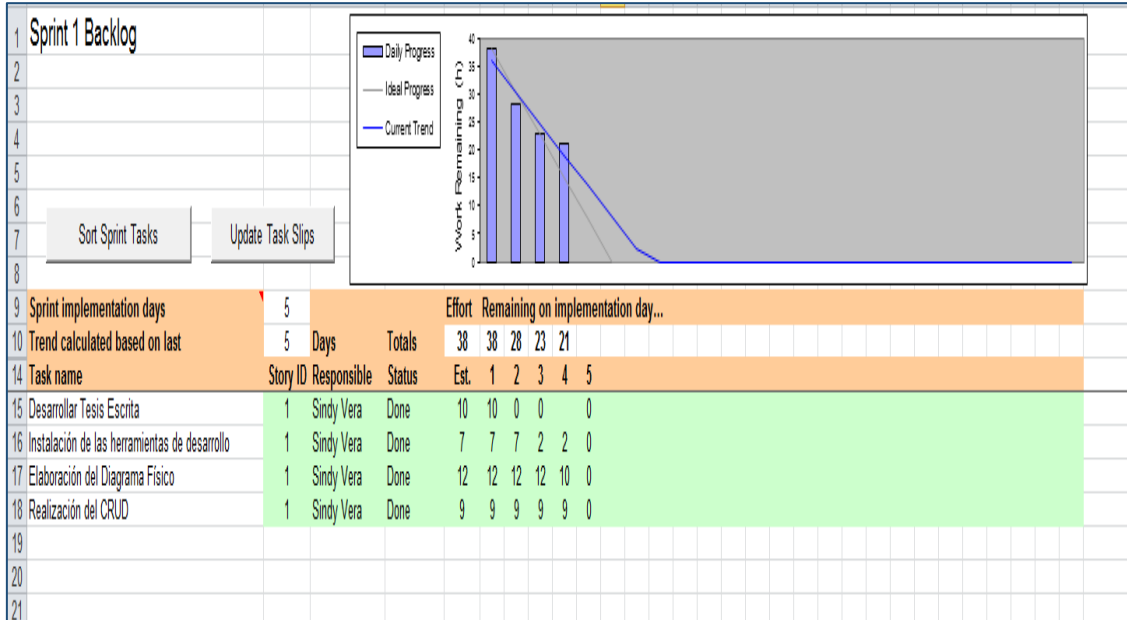
# ANEXO 1

## PRODUCT BACKLOG PROYECTO FARMWARE

1	Product Backlog						
2							Sort Product Backlog
3							
4	Story ID	Story name	Status	Size	Sprint	Priority	Comments
5	1	Desarrollar Tesis Escrita	Done	3	1		Sin inconvenientes
6	2	Instalación de las herramientas de desarroll	Done	4	1		
7	3	Diseñar el Diagrama Entidad Relación	Done	3	1		
8	4	Elaboración del Diagrama Físico	Done	2	1		
9	5	Realización del CRUD	Done	3	2		Sin inconvenientes
10	6	Desarrollo del Módulo de Gestión	Done	4	2		
11	7	Desarrollo de Formulario Crear Pesaje	Done	3	2		
12	8	Desarrollo vista Detalle Pesaje	Done	2	2		
13	9	Desarrollo del Reporte de Gestión	Done	3	2		
14	10	Pruebas Funcionales del Módulo de Alimentación	Done	4	2		Sin inconvenientes
15	11	Diseñar el Módulo de Gestión de Manejo	Done	3	2		Sin inconvenientes
16	12	Crear Formulario para Crear Animales	Done	2	2		
17	13	Desarrollo de la Vista para la Creación de Animales	Done	3	2		
18	14	Desarrollo del Reporte de los Animales creados	Done	4	2		
19	15	Pruebas Funcionales del Módulo de Manejo	Done	3	3		
20	16	Diseño del Módulo de Gestión de Reproducción	Done	2	3		
21	17	Desarrollo del Formulario para el registro de Parto	Done	3	3		Sin inconvenientes
22	18	Desarrollo del Formulario para el registro de servicio	Done	4	3		Sin inconvenientes
23	19	Desarrollo de la Vista para el Módulo de Reproducción	Done	3	3		
24	20	Crear el Reporte para el Módulo de Reproducción	Done	2	3		
25	21	Pruebas Funcionales del Módulo de Gestión de Reproducción	Done	3	3		Sin inconvenientes
26	22	Diseño y Desarrollo del Módulo de Gestión de Producción	Done	4	3		
27	23	Desarrollo del Formulario de ordeñada	Done	3	3		
28	24	Desarrollo del Formulario para el registro de ordeñada por animal	Done	2	3		
29	25	Creación del Reporte de detalle de Ordeñadas	Done	3	4		Sin inconvenientes
30	26	Pruebas Funcionales del Módulo de Gestión de Producción	Done	4	4		Sin inconvenientes
31	27	Diseño y Desarrollo del Módulo de Ingresos y Egresos	Done	3	4		Sin inconvenientes
32	28	Formulario para la Creación de registro de Ingreso por Actividad	Done	2	4		
33	29	Formulario Para la Creación de registro de Egreso por Actividad	Done	3	4		
34	30	Pruebas Funcionales del Módulo de Ingresos y Egresos	Done	4	4		
35	31	Desarrollo y Diseño del Módulo de Producción	Done	3	4		
36	32	Desarrollo del Reporte de Producción	Done	2	4		
37	33	Crear Reporte Promedio de Historico de Producción por Rodeo	Done	3	4		
38	34	Crear Reporte Número de partos por Animal	Done	4	4		
39	35	Pruebas Funcionales del Módulo de Resumen	Done	3	4		Sin inconvenientes
40							
41							
42							
43							
44							

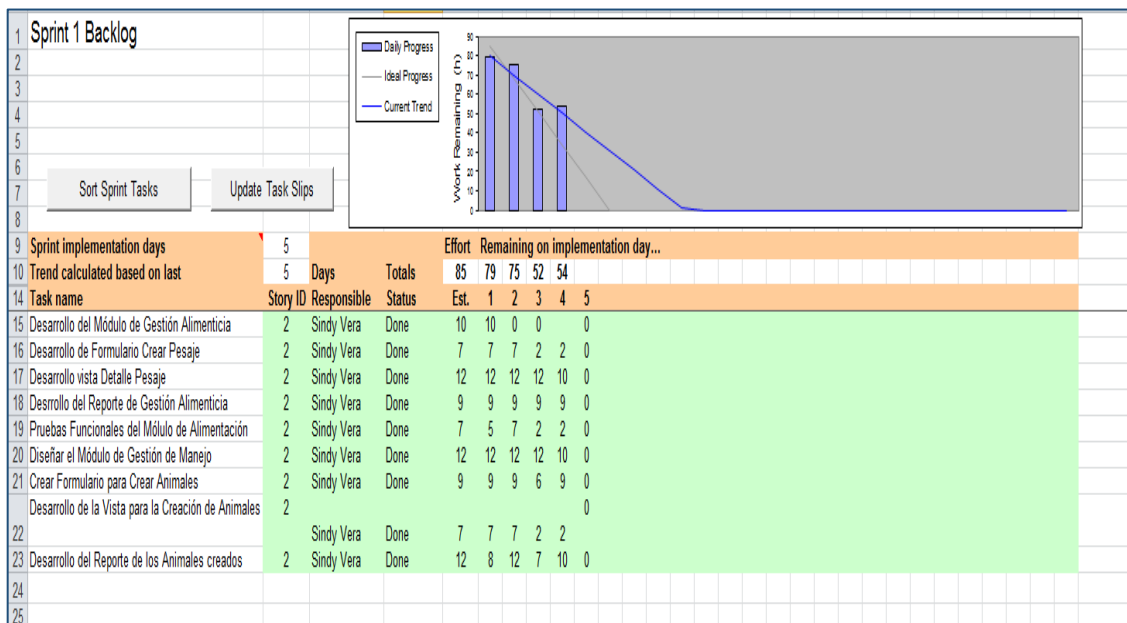
## ANEXO 2

### PRIMER SPRINT PROYECTO FARMWARE



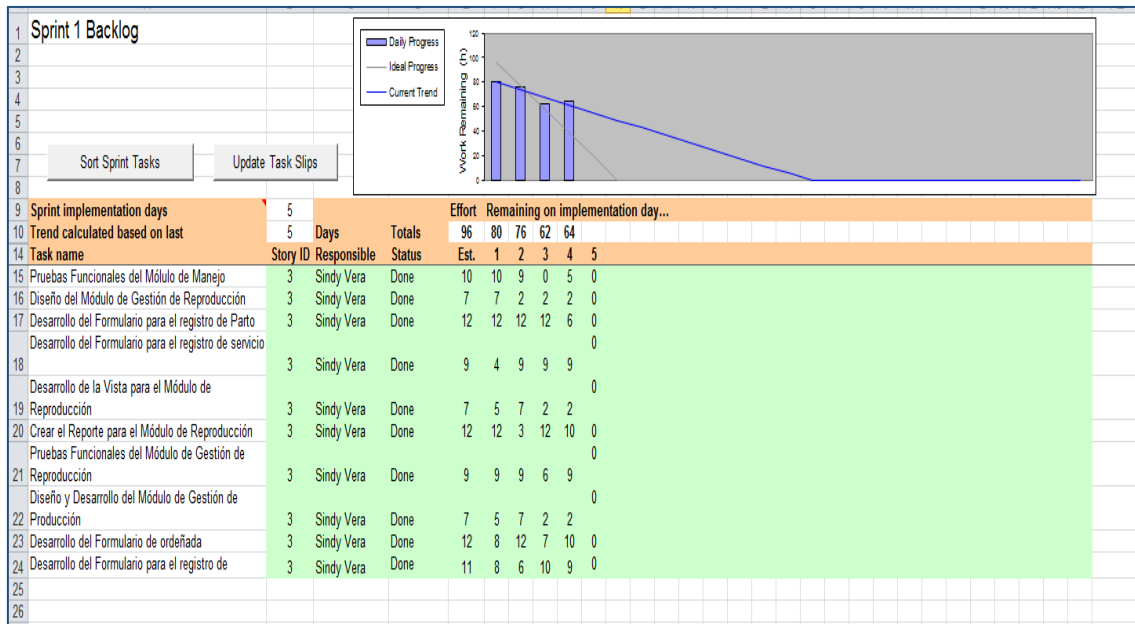
## ANEXO 3

### SEGUNDO SPRINT PROYECTO FARMWARE



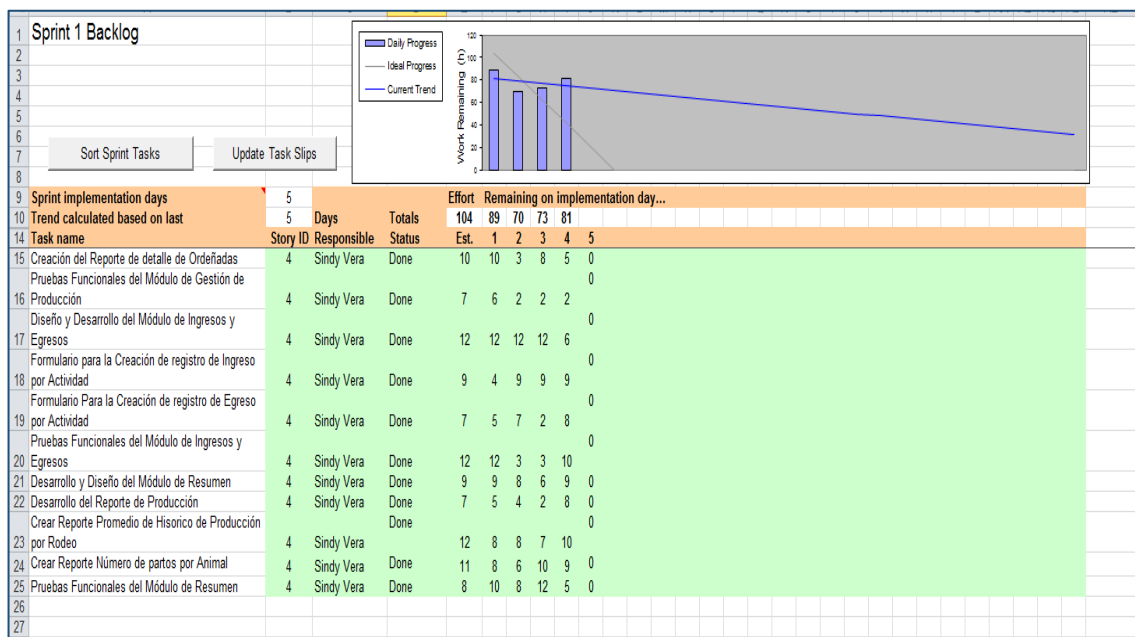
## ANEXO 4

### TERCER SPRINT PROYECTO FARMWARE



## ANEXO 5

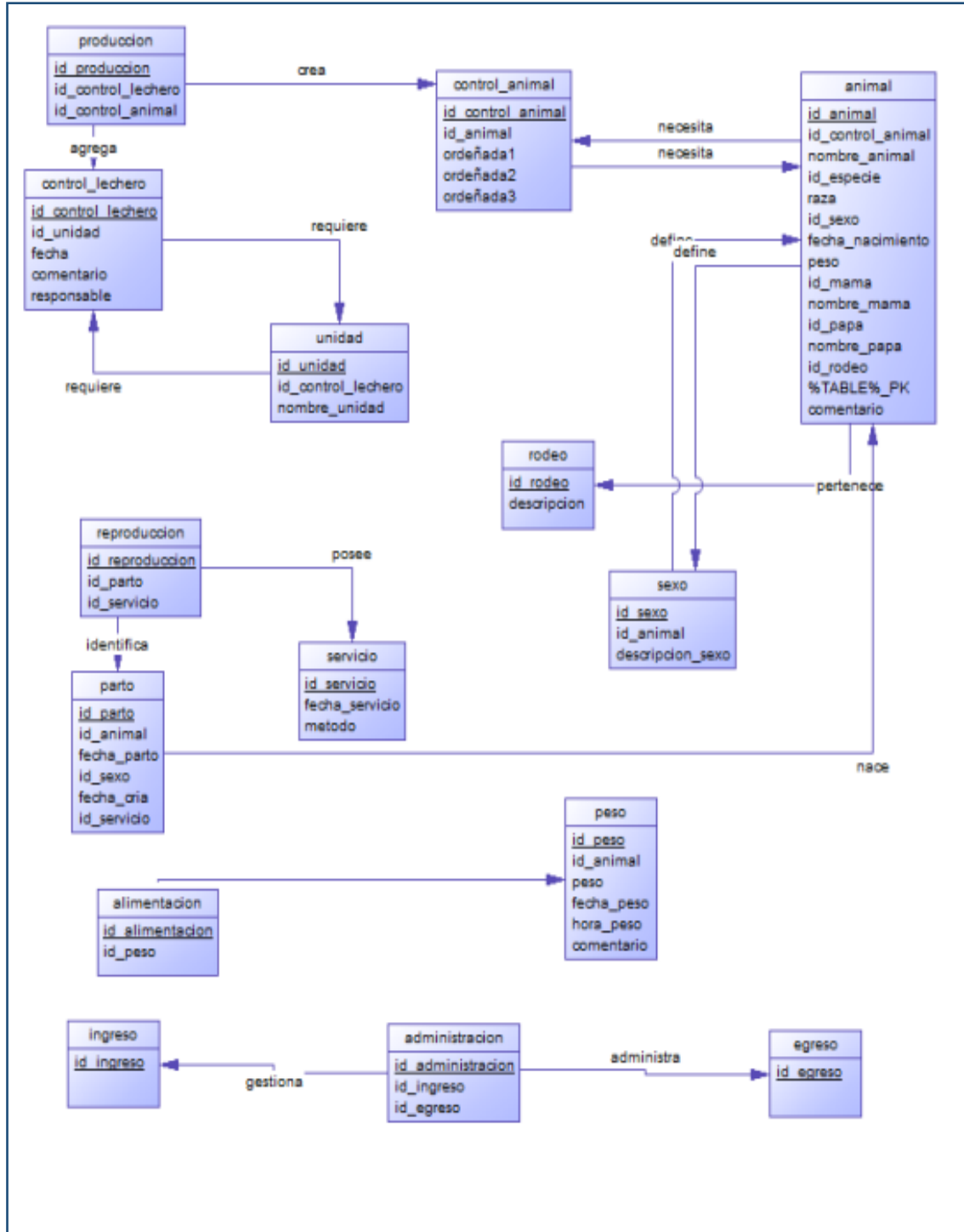
### CUARTO SPRINT PROYECTO FARMWARE



## ANEXO 6

### DIAGRAMA FÍSICO FARMWARE

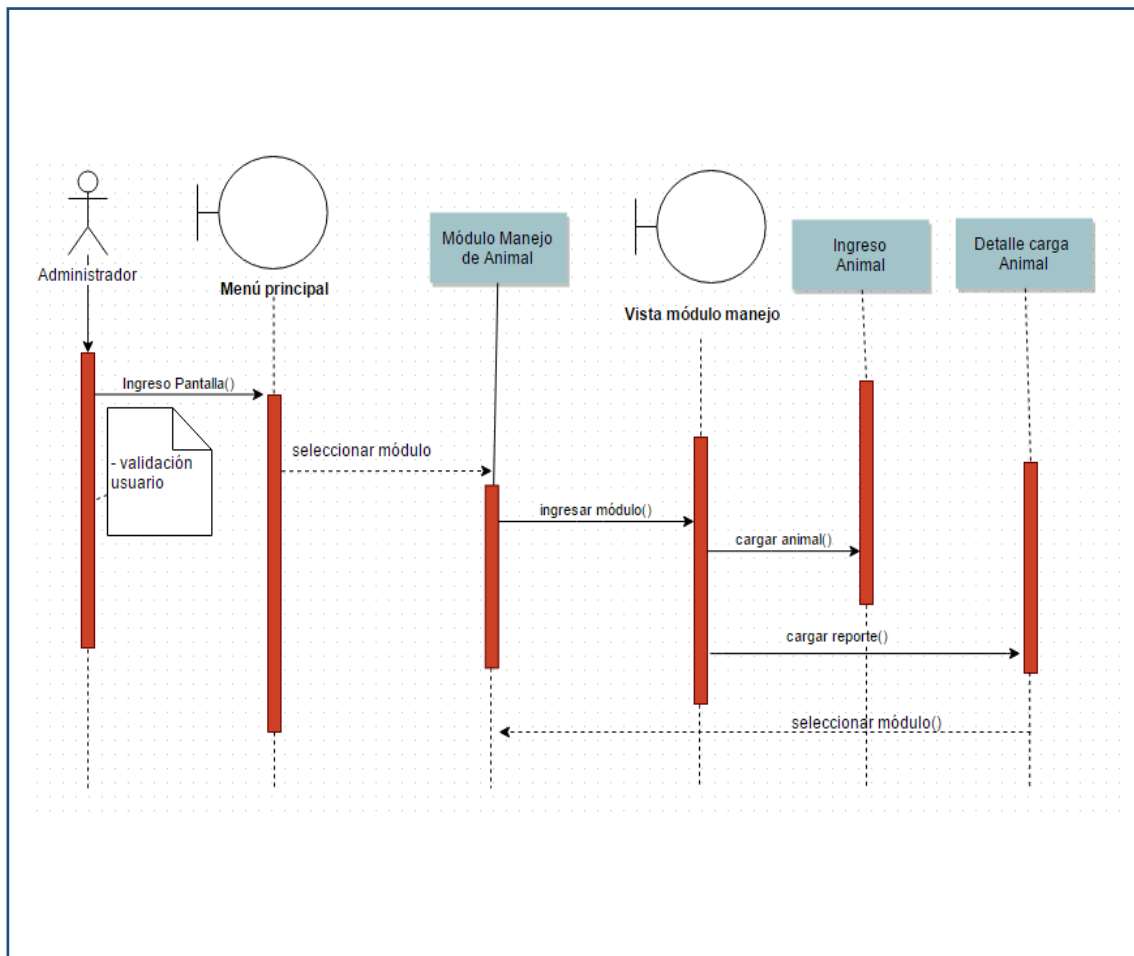
Diagrama Físico del sistema de la aplicación.



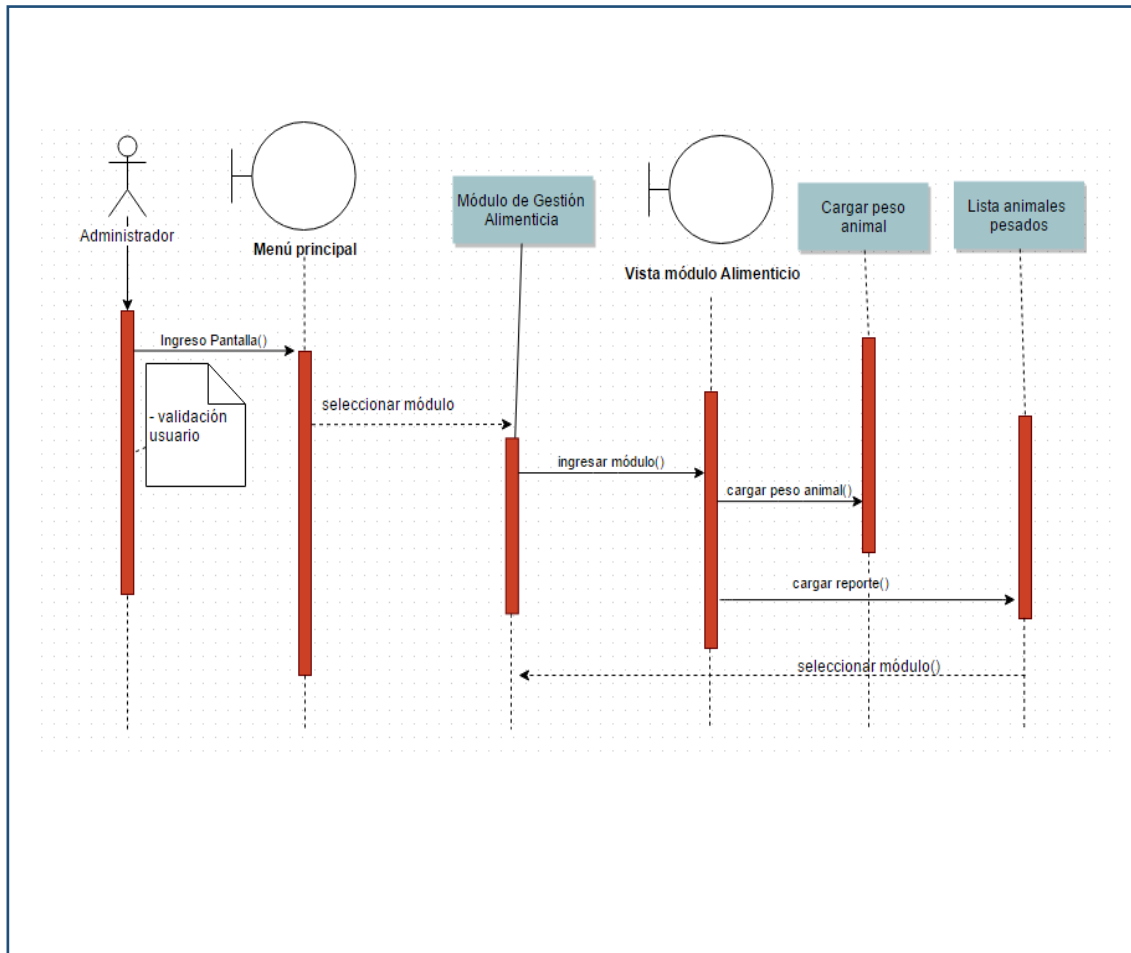
## ANEXO 7

### DIAGRAMA DE SECUENCIA

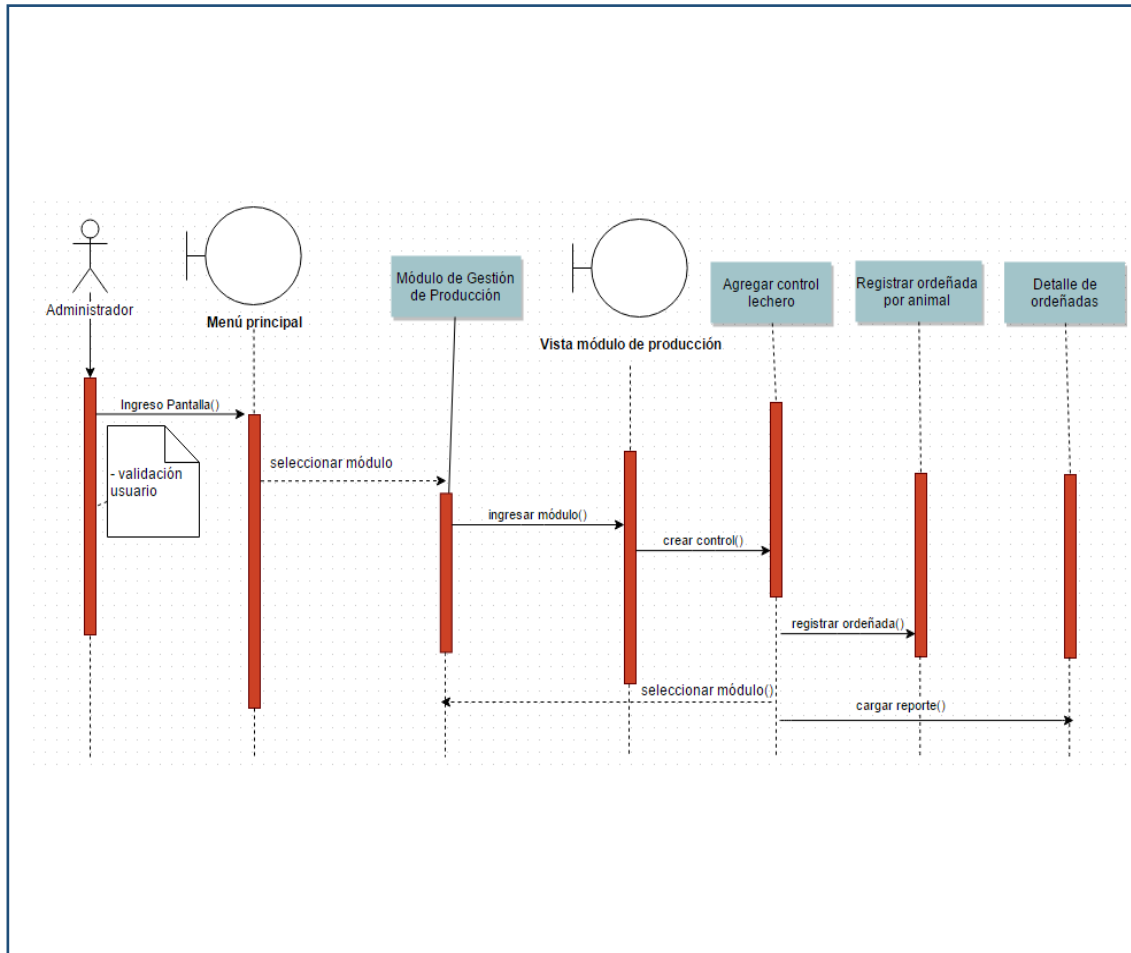
Diagrama de secuencia de administración del módulo de manejo de animal



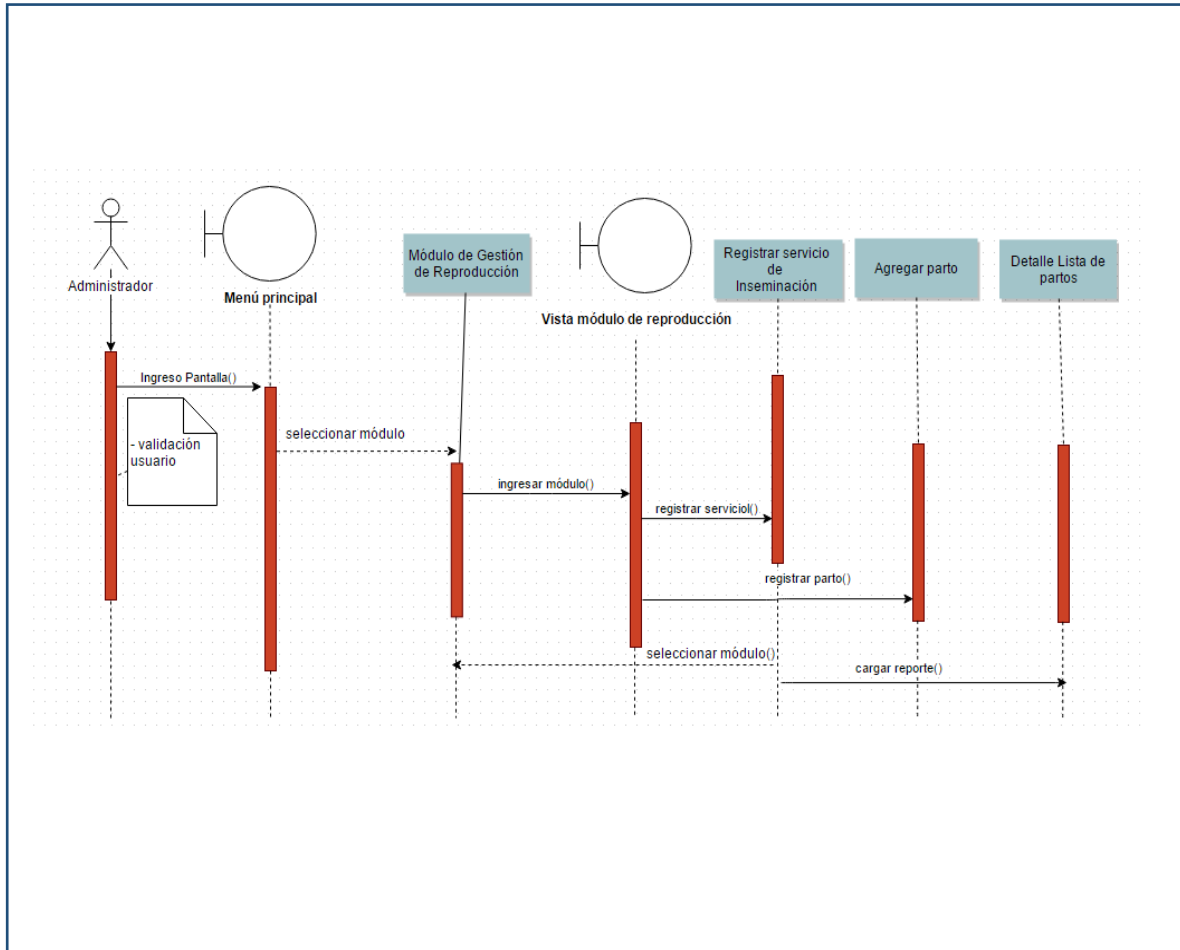
## Diagrama de secuencia de administración del módulo de gestión alimenticia



## Diagrama de secuencia de administración del módulo de gestión de producción

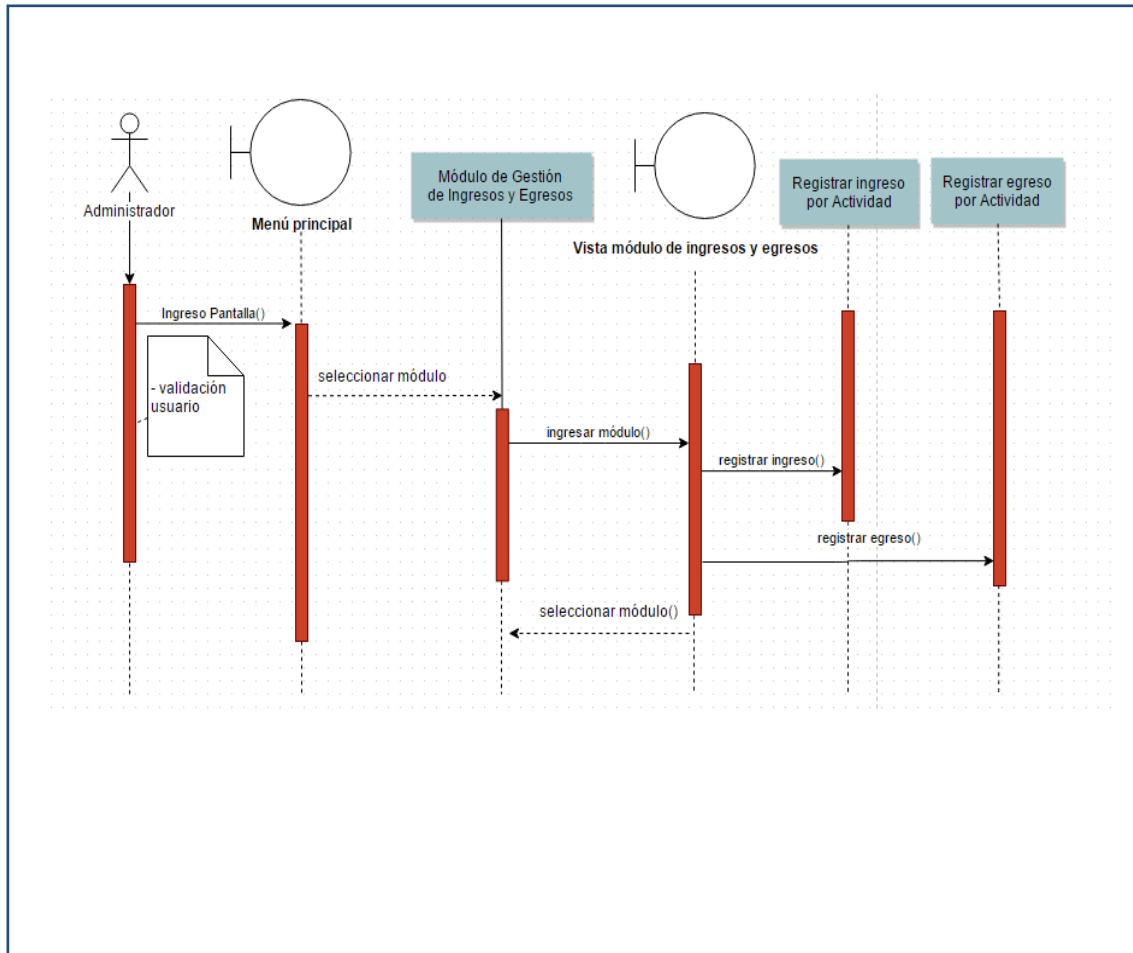


## Diagrama de secuencia de administración del módulo de gestión de reproducción

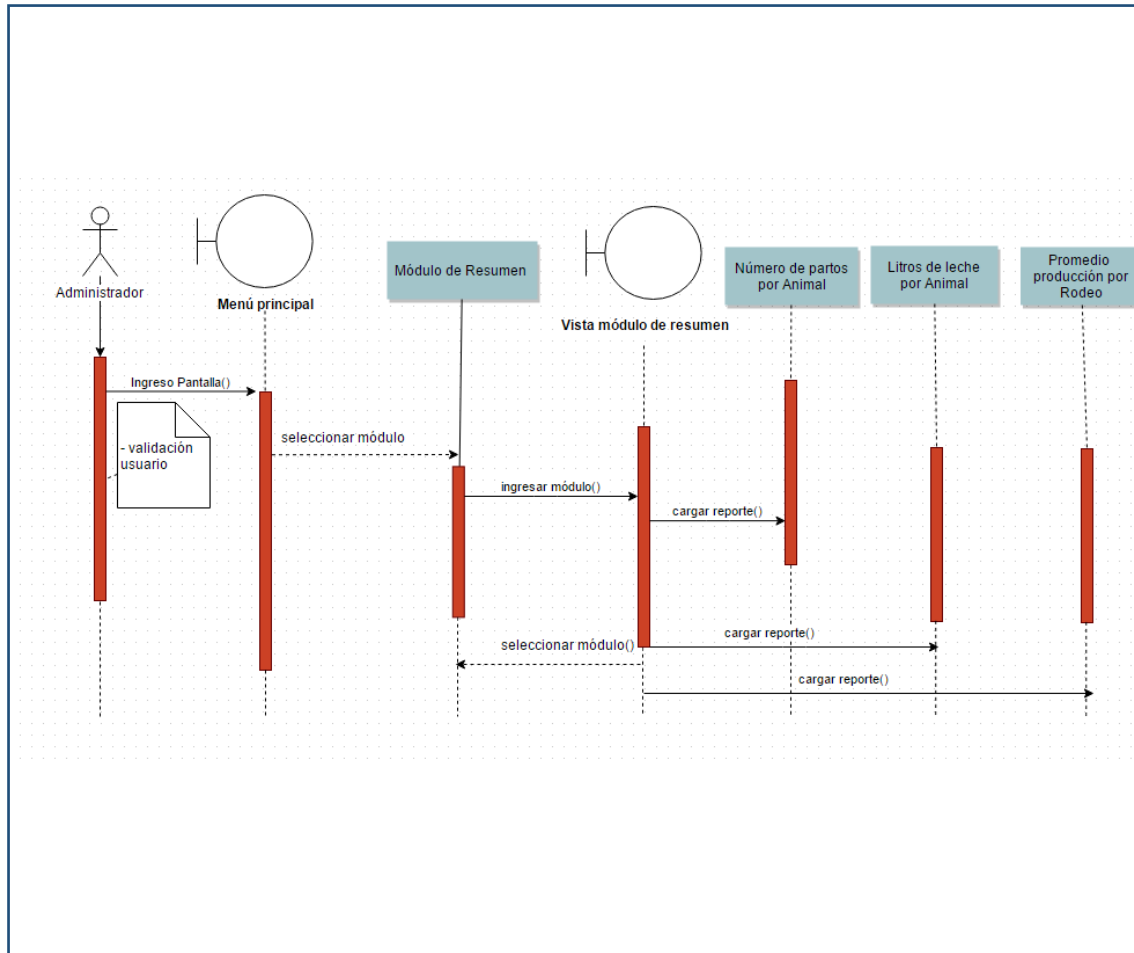




## Diagrama de secuencia de administración del módulo de ingresos y egresos



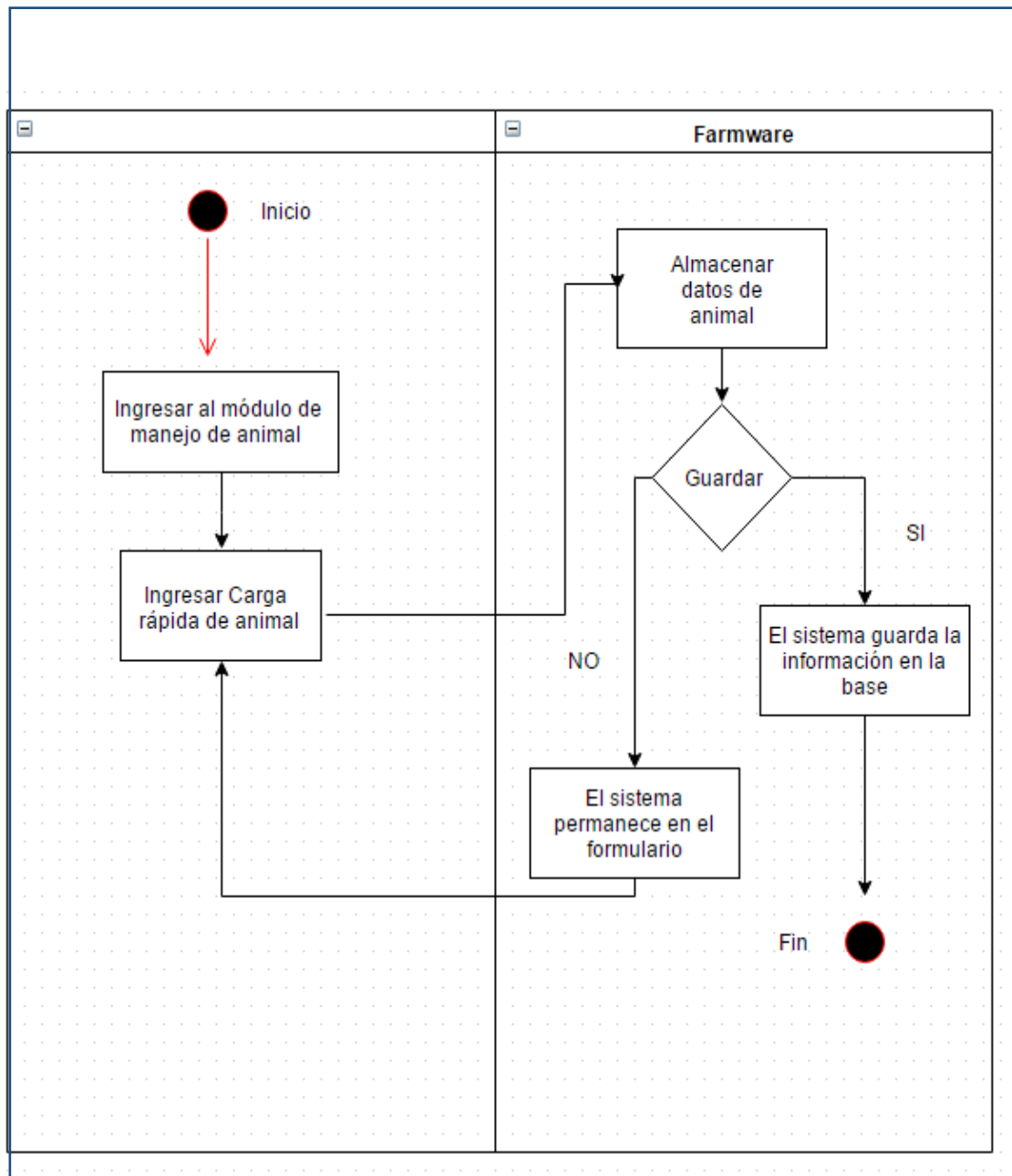
## Diagrama de secuencia de administración del módulo de resumen



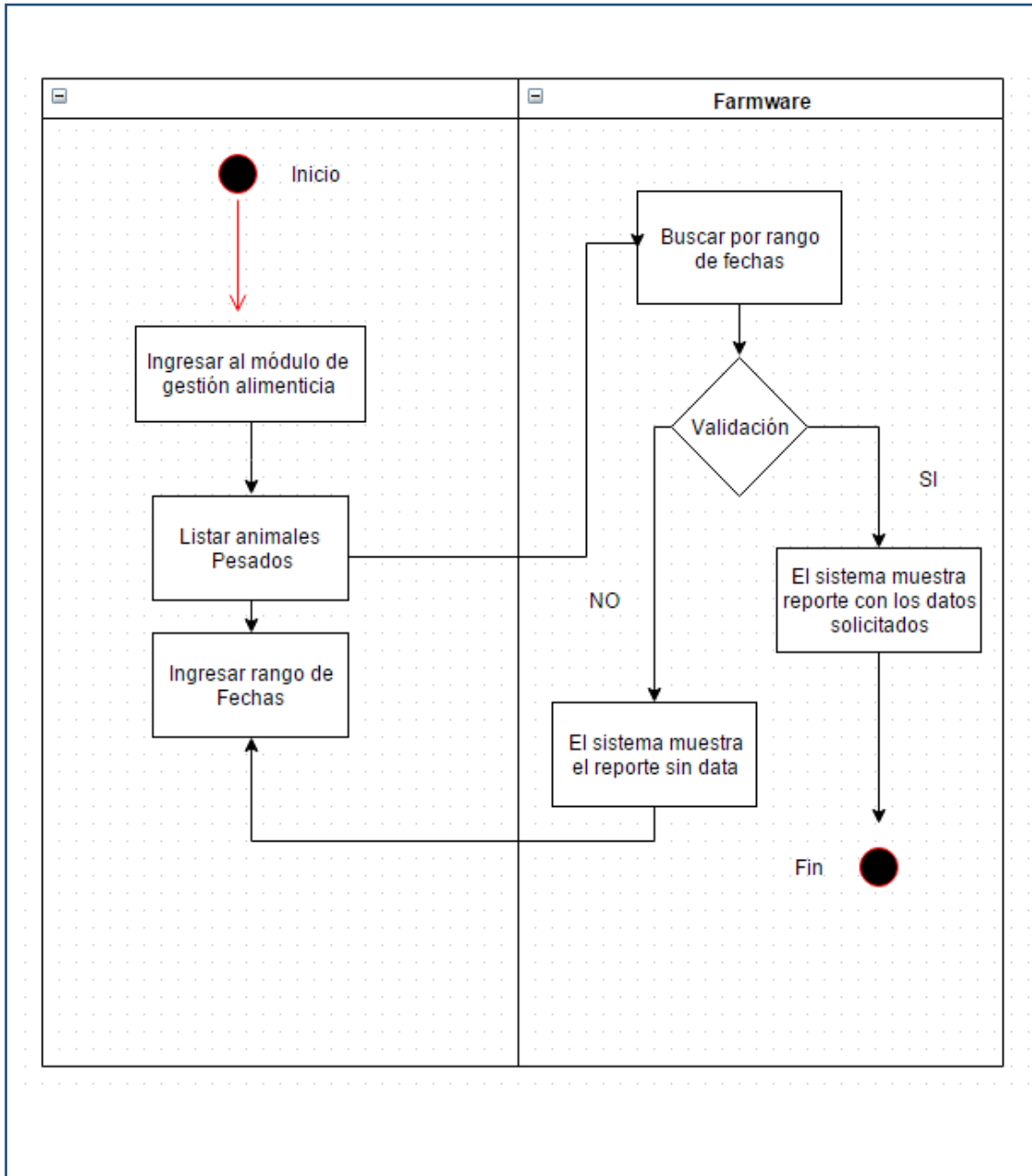
## ANEXO 8

### DIAGRAMA DE ACTIVIDAD

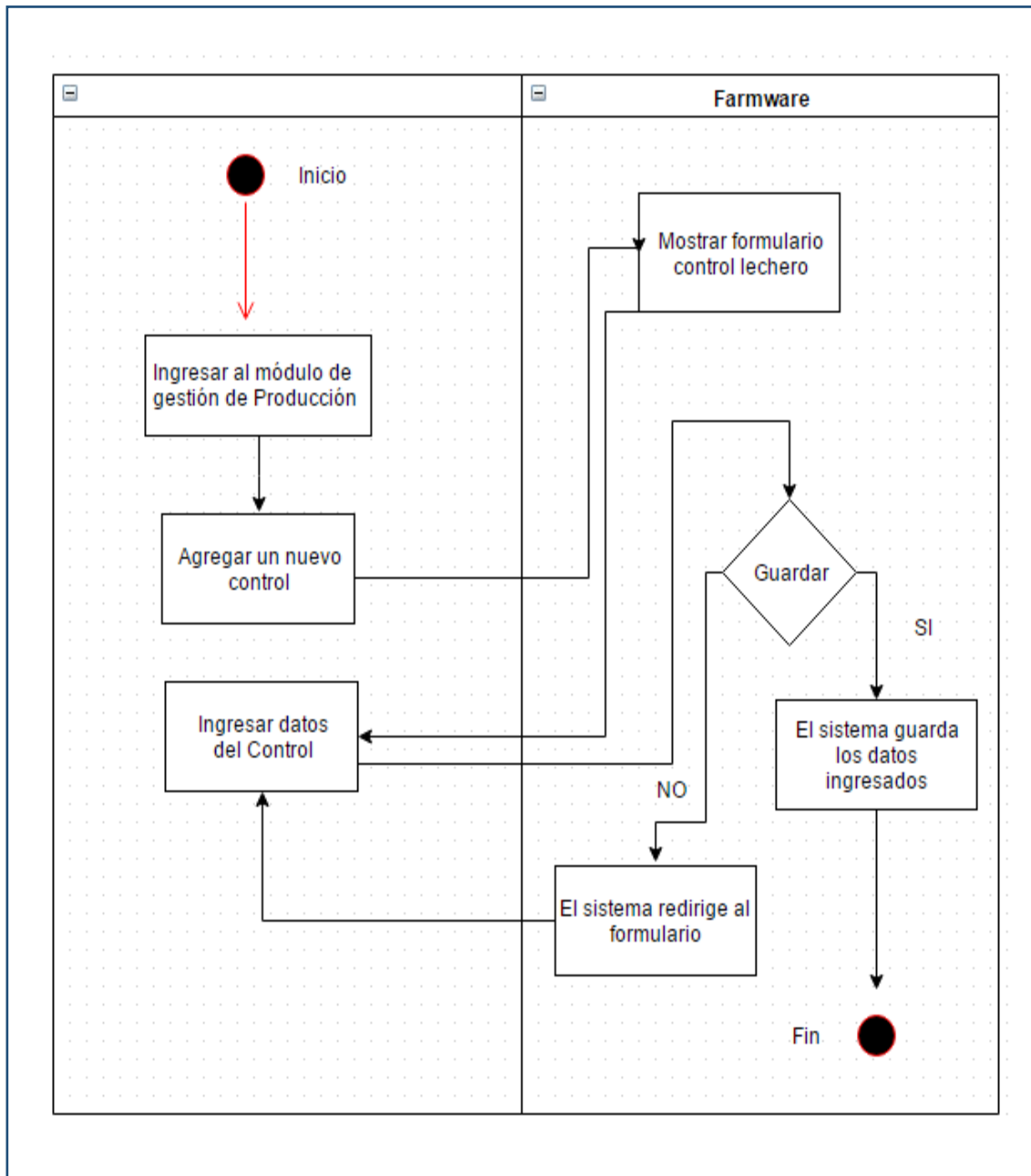
Diagrama de actividad ingresar animales en el módulo de manejo



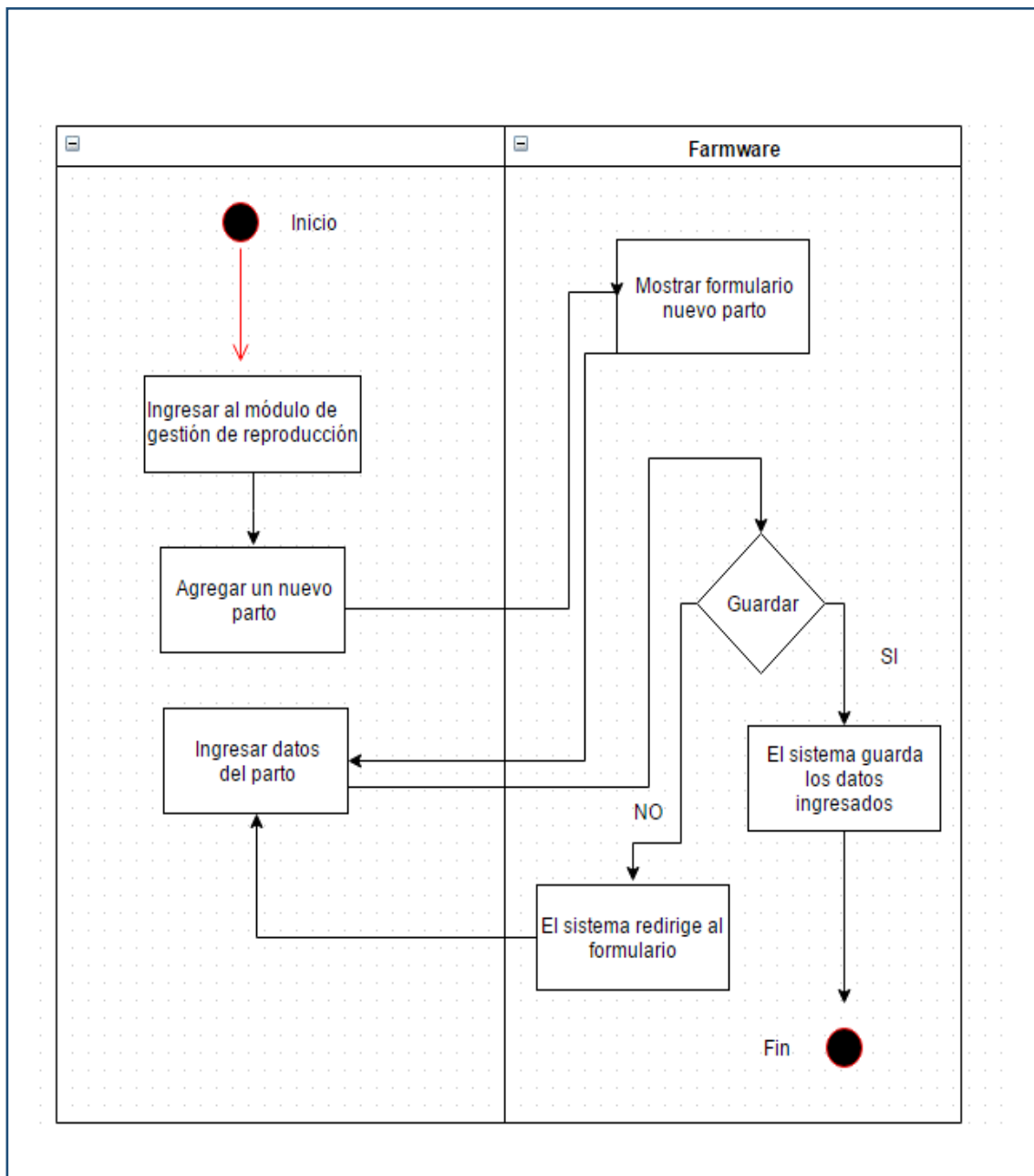
## Diagrama de actividad listar animales pesados en módulo alimenticio



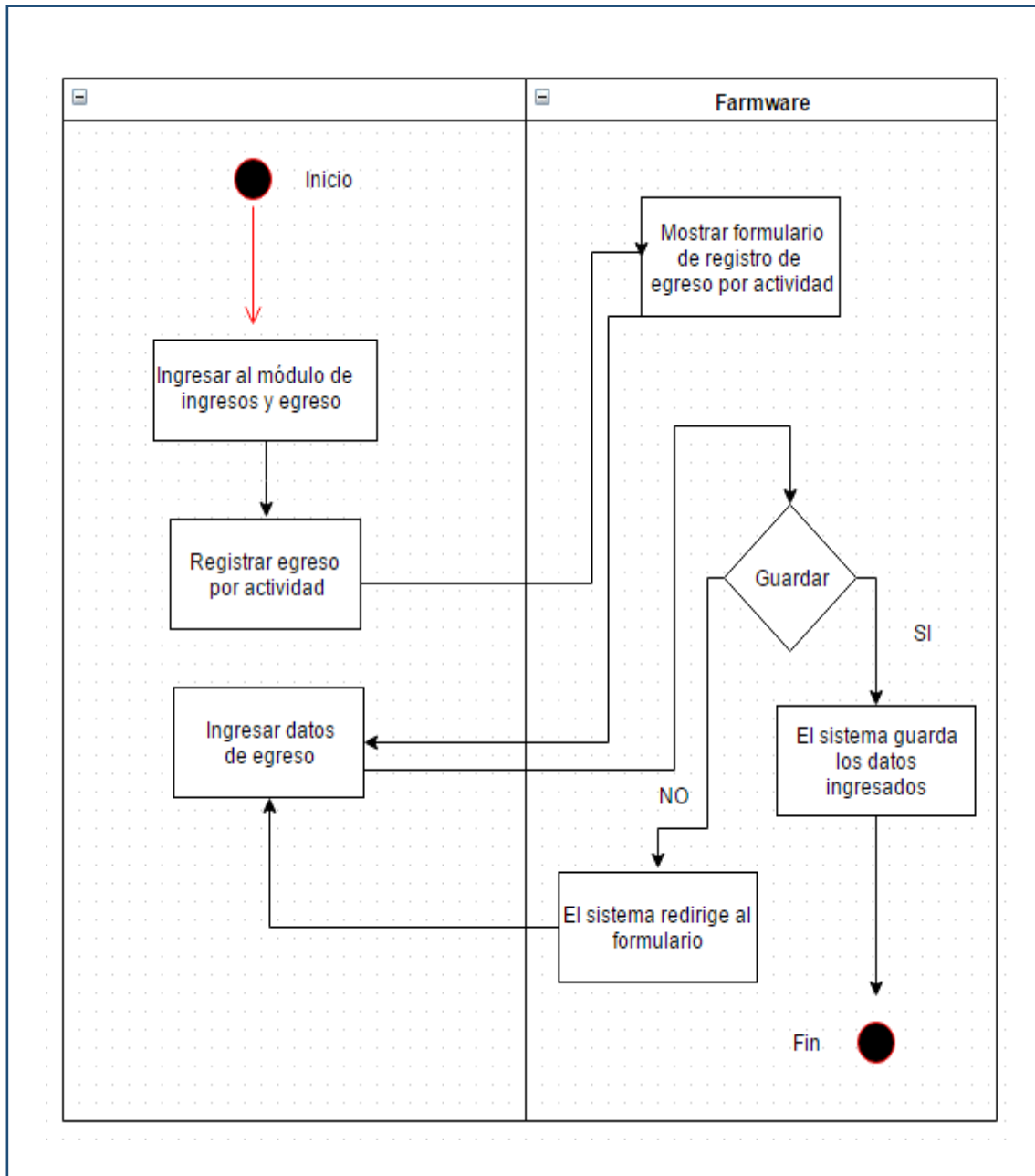
## Diagrama de actividad para crear control lechero en módulo de gestión de producción



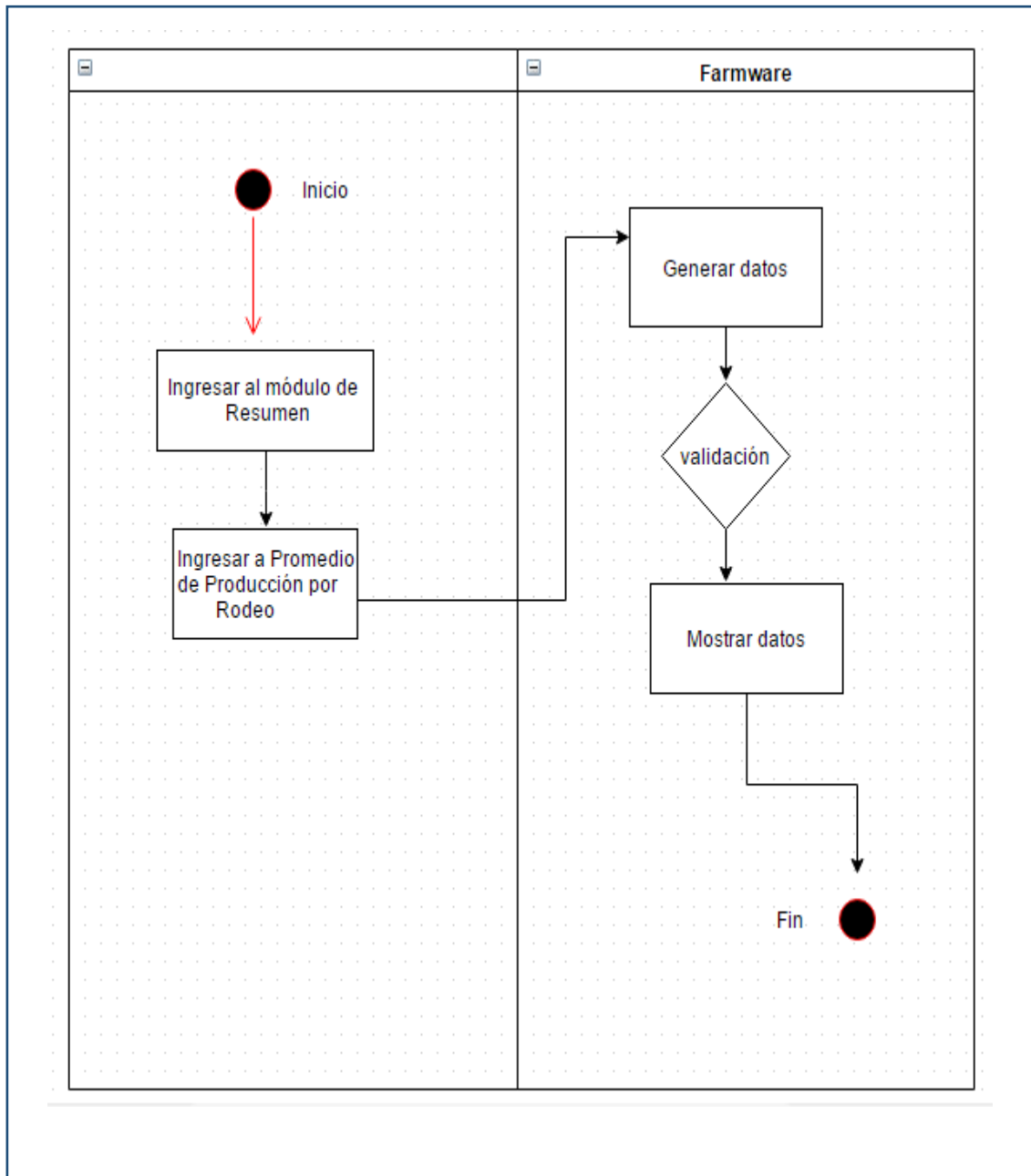
## Diagrama de actividad para registrar un nuevo parto en el módulo de gestión de reproducción



## Diagrama de actividad de registro de egreso por actividad en el módulo de ingresos y egresos



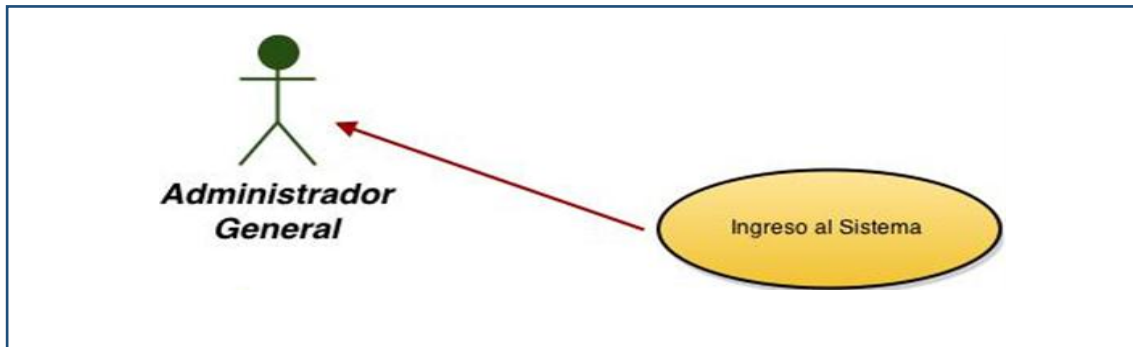
**Diagrama de actividad para generar reporte de Promedio de promedio de producción de rodeo en el módulo de resumen**





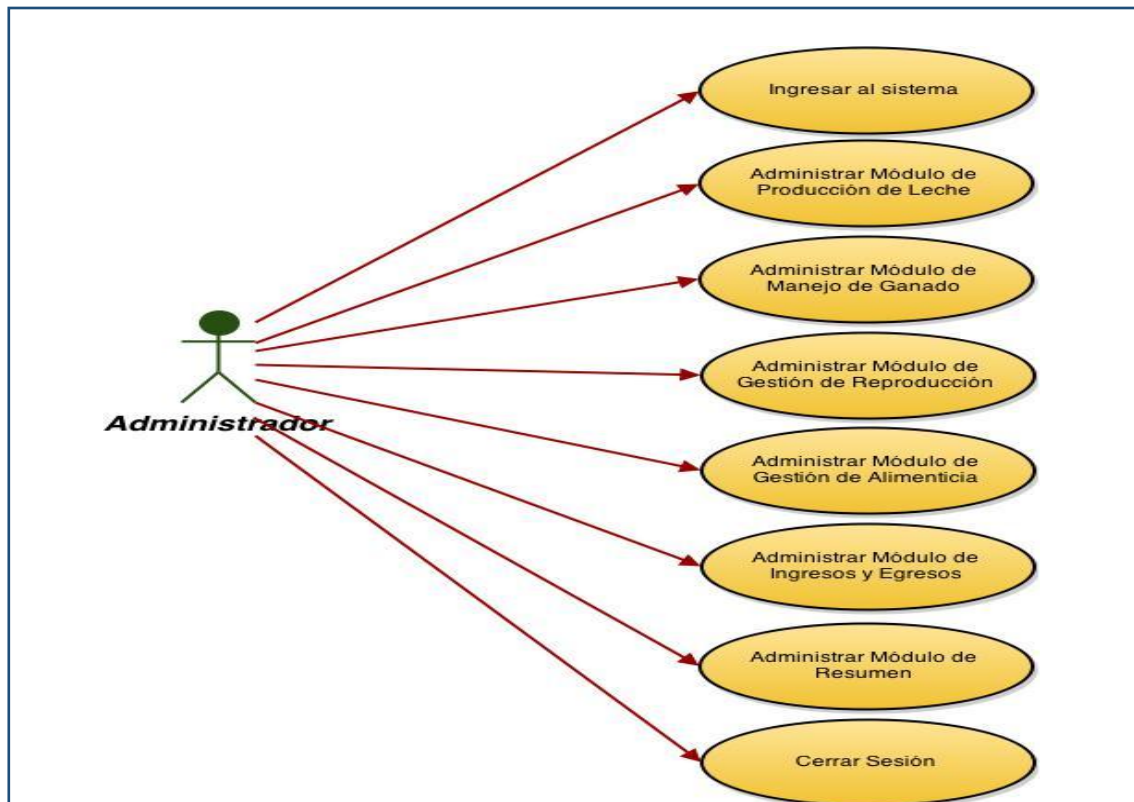
## ANEXO 9

### Diagrama de Caso de uso para ingresar al sistema



<b>Caso de Uso</b>	Ingresar al sistema.
<b>Descripción</b>	El administrador general tendrá un nombre de usuario y contraseña que les permitirá autenticarse e ingresar al sistema.
<b>Actores</b>	El administrador general.
<b>Precondiciones</b>	Los actores deben estar registrados en el sistema para poder acceder a la aplicación.
<b>Flujo Normal</b>	<ul style="list-style-type: none"><li>• El actor ingresa a la página inicial del sistema.</li><li>• El actor debe registrar su nombre y contraseña en el sistema.</li><li>• El actor da clic en el botón de ingresar.</li><li>• Se muestra la pantalla principal del sistema con sus módulos.</li></ul>
<b>Flujo Alternativo</b>	El actor ingresa de manera errónea su usuario y contraseña.  El actor da clic en el botón de ingresar y aparece el mensaje de "Usuario o clave incorrecta"
<b>Pos condiciones</b>	Ingreso al menú principal del sistema.

## Diagrama de caso de uso del administrador general



<b>Caso de Uso</b>	Administración de la información.
<b>Descripción</b>	El administrador general podrá crear y manejar la data que se encuentra en cada módulo del sistema.
<b>Actores</b>	El administrador general.
<b>Precondiciones</b>	El actor debe ingresar al sistema de manera satisfactoria.
<b>Flujo Normal</b>	<p><b>Producción de Leche,</b></p> <p>Crear Unidad:</p> <ul style="list-style-type: none"> <li>• El Administrador general selecciona la opción unidad.</li> <li>• El Administrador general presiona el botón crear Unidad.</li> <li>• El sistema muestra la pantalla con sus campos para ingresar los datos.</li> <li>• El Administrador general ingresa los datos.</li> <li>• Da clic en el botón guardar.</li> <li>• El sistema guarda la información y se actualiza en el módulo de producción de Leche en el campo</li> </ul>

unidad.

**Modificar Unidad:**

- El Administrador general marca la opción de unidad.
- El Administrador general selecciona el botón Editar Unidad.
- El sistema muestra la pantalla con los datos originales
- El Administrador general selecciona el campo y modifica los datos.
- El Administrador general da clic en el botón guardar.
- El sistema guarda la información modificada y se actualiza en el módulo de producción de leche en el campo unidad.

**Eliminar Unidad:**

- El Administrador general marca la opción unidad.
- El sistema muestra la pantalla de unidad.
- El Administrador general da clic sobre el botón Eliminar.
- El sistema elimina el registro de la unidad y se actualiza en la lista de unidades.

**Manejo del Ganado,**

**Crear Rodeo:**

- El Administrador general selecciona la opción rodeo.
- El Administrador general presiona el botón crear rodeo.
- El sistema muestra la pantalla con sus campos para ingresar los datos.
- El Administrador general ingresa los datos.
- Da clic en el botón guardar.
- El sistema guarda la información y se actualiza en el módulo de manejo del ganado en el campo rodeo.

**Modificar Rodeo:**

- El Administrador general marca la opción de rodeo.
- El Administrador general selecciona el botón Editar rodeo.

- El sistema muestra la pantalla con los datos originales
- El Administrador general selecciona el campo y modifica los datos.
- El Administrador general da clic en el botón guardar.
- El sistema guarda la información modificada y se actualiza en el módulo manejo del ganado en el campo rodeo.

#### Eliminar Rodeo:

- El Administrador general marca la opción rodeo.
- El sistema muestra la pantalla de rodeo.
- El Administrador general da clic sobre el botón eliminar.
- El sistema elimina el registro del rodeo seleccionado y se actualiza en la lista de rodeos.

#### **Gestión de Reproducción**

##### Crear sexo de la cría:

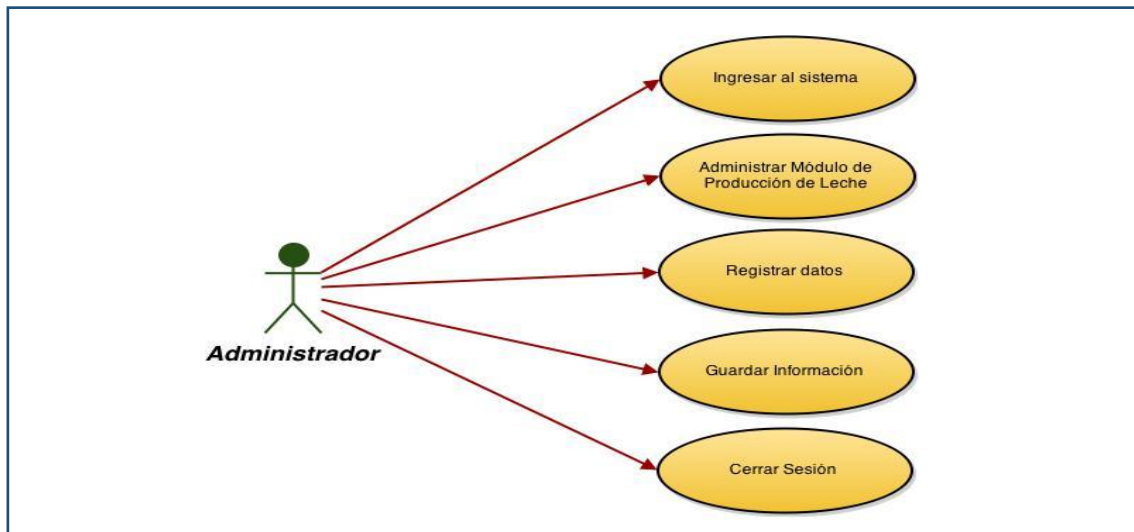
- El Administrador general selecciona la opción sexo.
- El Administrador general presiona el botón crear sexo de la cría.
- El sistema muestra la pantalla con sus campos para ingresar los datos.
- El Administrador general ingresa los datos.
- Da clic en el botón guardar.
- El sistema guarda la información y se actualiza en el módulo de gestión de reproducción en el campo sexo de la cría.

##### Modificar sexo de la cría:

- El Administrador general marca la opción de sexo de la cría.
- El Administrador general selecciona el botón Editar sexo de la cría.
- El sistema muestra la pantalla con los datos originales
- El Administrador general selecciona el campo y modifica los datos.
- El Administrador general da clic en el botón guardar.
- El sistema guarda la información y se actualiza en el módulo de gestión de reproducción en el campo

	<p>sexo de la cría.</p> <p>Eliminar sexo de la cría:</p> <ul style="list-style-type: none"> <li>• El Administrador general marca la opción sexo de la cría.</li> <li>• El sistema muestra la pantalla de sexo de la cría.</li> <li>• El Administrador general da clic sobre el botón eliminar.</li> <li>• El sistema elimina el registro del sexo de la cría seleccionado y se actualiza en la lista de sexo de la cría.</li> </ul>
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El actor selecciona la opción crear, editar o eliminar.</li> <li>• Al momento de guardar presiona la opción cancelar.</li> <li>• El sistema muestra la pantalla inicial de los módulos.</li> </ul>
<b>Pos condiciones</b>	Ingreso al menú principal del sistema.

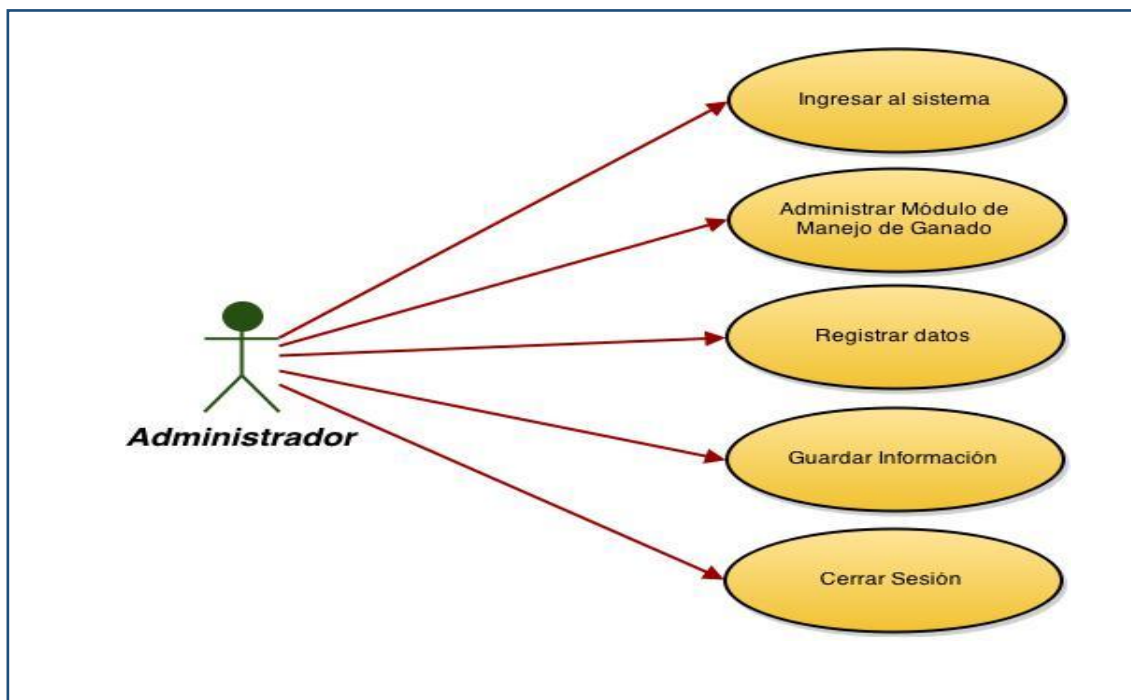
## Diagrama de Caso de uso administrar la producción de leche



<b>Caso de Uso</b>	Administrar la producción de leche.
<b>Descripción</b>	El administrador podrá crear la información que se encuentra en el módulo de producción de leche.
<b>Actores</b>	El administrador.
<b>Precondiciones</b>	El actor debe ingresar al sistema de manera exitosa.
<b>Flujo Normal</b>	<p>Crear control lechero</p> <ul style="list-style-type: none"> <li>• El administrador debe ingresar al módulo de producción de leche.</li> <li>• El administrador da clic en agregar un nuevo control lechero.</li> <li>• El administrador debe ingresar los siguientes datos: fecha, comentarios, responsable, unidad.</li> <li>• El administrador debe dar clic en agregar control.</li> </ul> <p>Crear control por animal</p> <ul style="list-style-type: none"> <li>• El usuario debe seleccionar la opción agregar registro de control por animal.</li> <li>• El sistema mostrará un formulario para agregar el registro de control por animal.</li> <li>• El administrador debe ingresar los datos: animal, ordeñada 1, ordeñada 2, ordeñada 3.</li> <li>• El administrador debe hacer clic en el botón crear.</li> <li>• El sistema le regresará a la página del módulo de</li> </ul>

	producción de leche.
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El actor selecciona la opción de crear la información que se encuentra en el módulo de producción de leche.</li> <li>• Al momento de guardar presiona la opción cancelar.</li> <li>• El sistema muestra la pantalla inicial del módulo de producción de leche.</li> </ul>
<b>Pos condiciones</b>	Ingreso a la página principal del módulo de producción de leche.

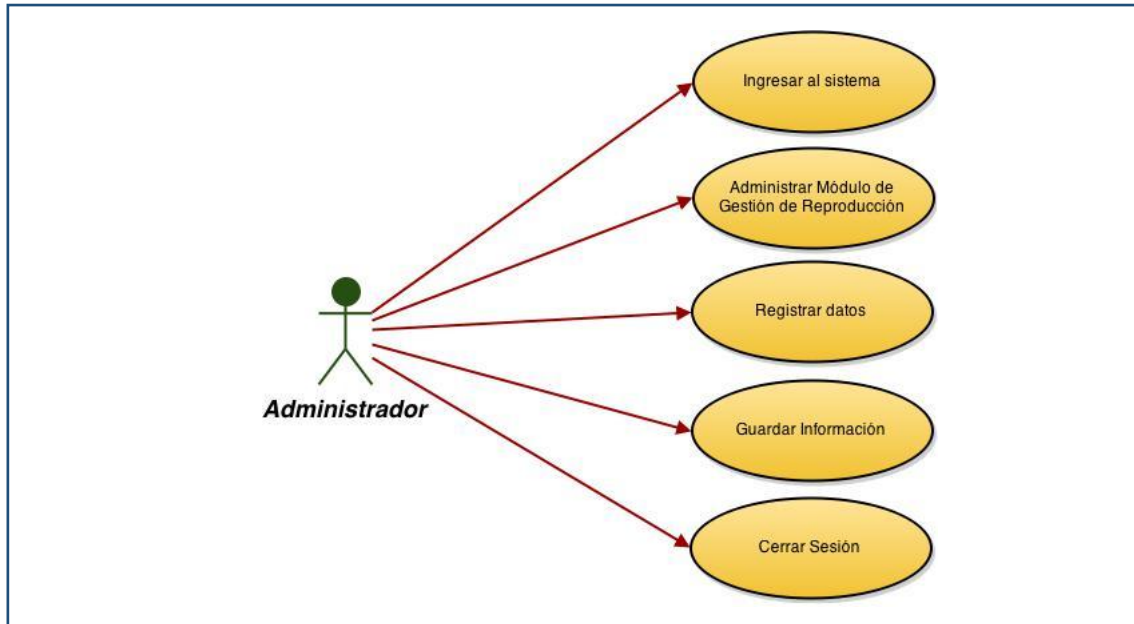
### Diagrama de Caso de uso administrar manejo de ganado



<b>Caso de Uso</b>	Administrar manejo de Ganado.
<b>Descripción</b>	El administrador podrá crear y manejar la información que se encuentra en el módulo de manejo de ganado.
<b>Actores</b>	El administrador.
<b>Precondiciones</b>	El actor debe ingresar al sistema de manera exitosa.
<b>Flujo Normal</b>	<p>Crear nuevo animal</p> <ul style="list-style-type: none"> <li>• El administrador debe ingresar al módulo de manejo de ganado.</li> <li>• El sistema mostrará en la página principal del módulo el resumen de todos los animales creados.</li> <li>• El administrador presionará la opción crear nuevo animal.</li> <li>• El sistema le direccionará a un formulario para crear un nuevo animal.</li> <li>• El administrador debe ingresar los siguientes datos: nombre, especie (Bovina), macho o hembra, fecha de nacimiento, peso, nombre madre, nombre progenitor, rodeo, comentarios.</li> <li>• El administrador da clic en el botón crear.</li> <li>• El sistema regresará a la página principal del módulo de manejo de ganado.</li> </ul>
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El actor selecciona la opción de crear la información que se encuentra en el módulo de manejo de ganado.</li> <li>• Al momento de guardar presiona la opción cancelar.</li> <li>• El sistema muestra la pantalla inicial del módulo de manejo de ganado.</li> </ul>
<b>Pos condiciones</b>	Ingreso a la página principal de módulo de manejo de ganado.



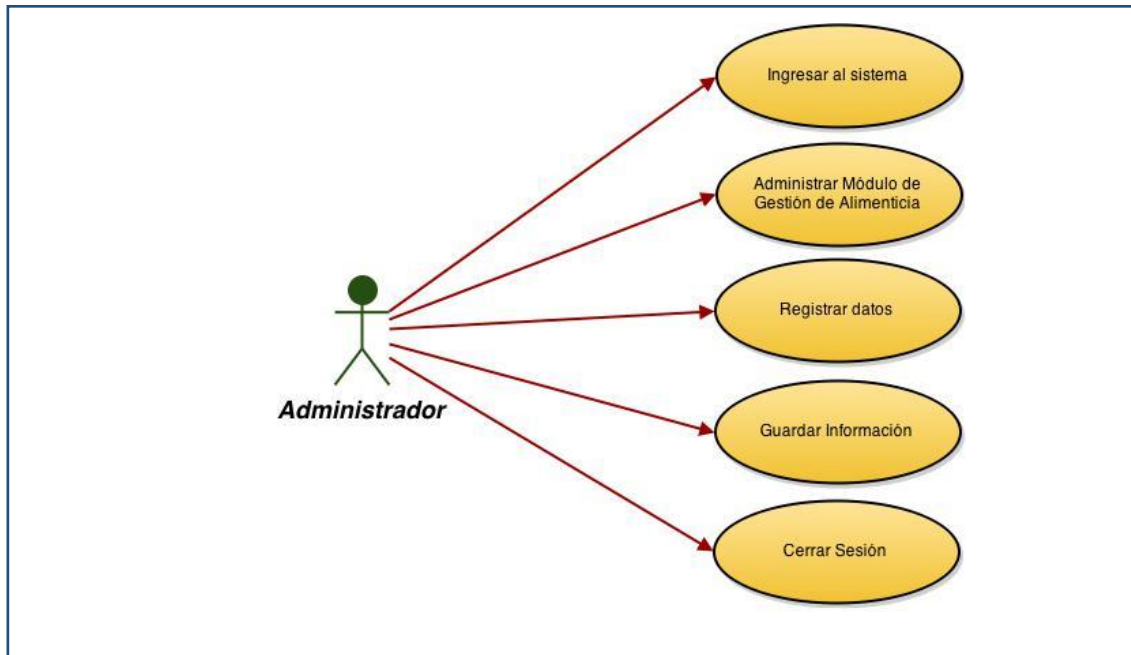
## Diagrama de Caso de uso administrar gestión de reproducción



<b>Caso de Uso</b>	Administrar gestión de reproducción.
<b>Descripción</b>	El administrador podrá crear la información que se encuentra en el módulo de gestión de reproducción.
<b>Actores</b>	El administrador.
<b>Precondiciones</b>	El actor debe ingresar al sistema de manera exitosa.
<b>Flujo Normal</b>	<p>Agregar un parto:</p> <ul style="list-style-type: none"> <li>• El administrador debe ingresar al módulo de gestión de reproducción</li> <li>• El sistema muestra en la pantalla principal del módulo dos opciones (agregar un parto y agregar un servicio).</li> <li>• El administrador da clic en agregar un parto.</li> <li>• El sistema muestra un formulario en el cual el administrador debe ingresar los siguientes datos: fecha de parto, sexo de la cría, fecha de servicio.</li> <li>• El administrador debe ingresar todos los datos en los campos de manera obligatoria.</li> <li>• El administrador debe dar clic en guardar.</li> <li>• El sistema guardará los datos y los presentara según la categoría en la pantalla principal del módulo de gestión de reproducción.</li> </ul>

	<p>Agregar un servicio:</p> <ul style="list-style-type: none"> <li>• El administrador debe ingresar al módulo de gestión de reproducción.</li> <li>• El sistema muestra en la pantalla principal del módulo dos opciones (agregar un parto y agregar y un servicio).</li> <li>• El administrador da clic en agregar un servicio.</li> <li>• El sistema muestra un formulario en el cual el administrador debe ingresar los siguientes datos: fecha de servicio y método de reproducción.</li> <li>• El administrador debe ingresar todos los datos en los campos de manera obligatoria.</li> <li>• El administrador debe dar clic en el botón Agregar servicio.</li> <li>• El sistema guardará los datos y los presentara según la categoría en la pantalla principal del módulo de gestión de reproducción.</li> </ul>
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El actor selecciona la opción de crear la información que se encuentra en el módulo de gestión de reproducción.</li> <li>• Al momento de guardar presiona la opción cancelar.</li> <li>• El sistema muestra la pantalla inicial del módulo de gestión de reproducción.</li> </ul>
<b>Pos condiciones</b>	Ingreso a la página principal de módulo de gestión de reproducción.

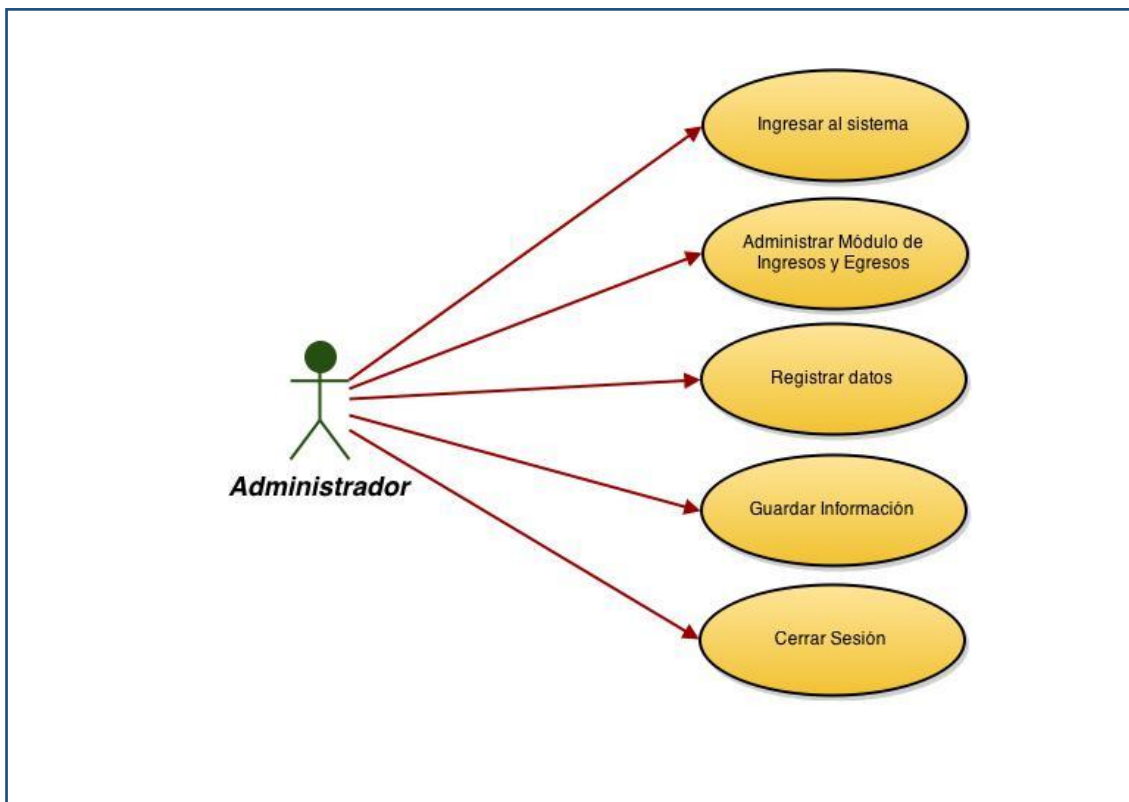
## Diagrama de Caso de uso administrar gestión alimenticia



<b>Caso de Uso</b>	Administrar gestión alimenticia.
<b>Descripción</b>	El administrador podrá crear la información que se encuentra en el módulo de gestión alimenticia.
<b>Actores</b>	El administrador.
<b>Precondiciones</b>	El actor debe ingresar al sistema exitosamente.
<b>Flujo Normal</b>	<p>Cargar peso de animal:</p> <ul style="list-style-type: none"> <li>• El administrador debe ingresar al módulo de gestión alimenticia.</li> <li>• El administrador da clic en el icono de cargar peso de un animal.</li> <li>• El sistema traerá un formulario; el administrador debe ingresar de manera obligatoria los siguientes campos: animal, fecha de peso, peso, comentarios.</li> <li>• El administrador da clic en crear.</li> <li>• El sistema guarda la información</li> <li>• El administrador da clic en el icono peso de los animales.</li> <li>• El sistema muestra la base de todos los pesos de</li> </ul>

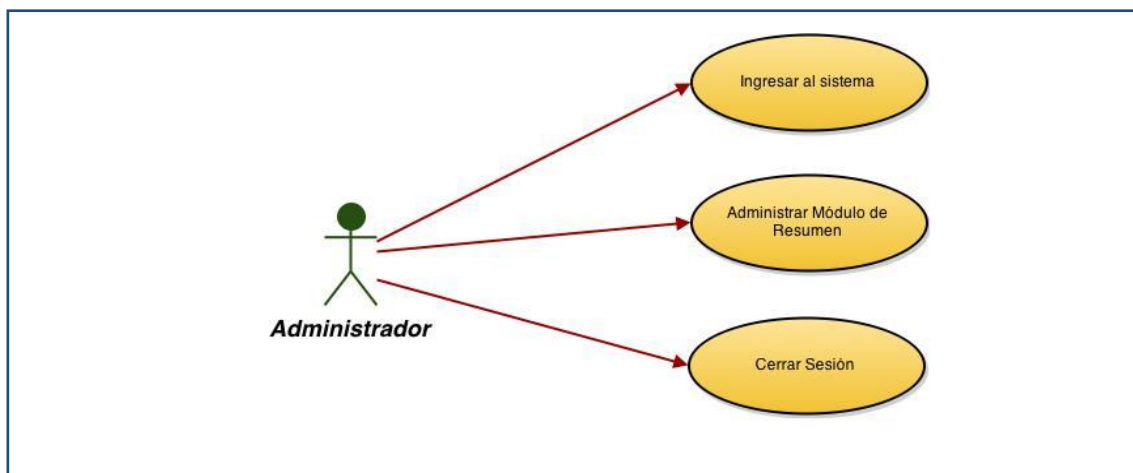
	los animales registrados.
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El actor selecciona la opción de crear la información que se encuentra en el módulo de gestión alimenticia.</li> <li>• Al momento de guardar presiona la opción cancelar.</li> <li>• El sistema muestra la pantalla inicial del módulo de gestión alimenticia.</li> </ul>
<b>Pos condiciones</b>	Ingreso a la página principal de módulo de gestión alimenticia.

### Diagrama de Caso de uso administrar ingresos y egresos



<b>Caso de Uso</b>	Administrar Ingresos y Egresos.
<b>Descripción</b>	El administrador podrá registrar la información que se encuentra en el módulo de ingresos y egresos.
<b>Actores</b>	El administrador.
<b>Precondiciones</b>	El actor debe ingresar al sistema de manera exitosa.
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• En el módulo se registra los gastos e ingresos, ventas de la producción de leche.</li> </ul>
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El actor se encarga de crear la información que se encuentra en el módulo de ingresos y egresos.</li> <li>• Al momento de guardar presiona la opción Cancelar.</li> <li>• El sistema muestra la pantalla inicial del módulo de ingresos y egresos.</li> </ul>
<b>Pos condiciones</b>	Ingreso a la página principal de módulo de ingresos y egresos.

### Diagrama de Caso de uso administrar resumen



<b>Caso de Uso</b>	Administración resumen
<b>Descripción</b>	El administrador podrá crear la información que se encuentra en el módulo de resumen.
<b>Actores</b>	El administrador.
<b>Precondiciones</b>	El administrador debe ingresar al sistema.
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El administrador ingresa a la página inicial del sistema.</li> <li>• El administrador debe seleccionar en el menú el Módulo resumen.</li> <li>• El sistema muestra los siguientes iconos: Reporte de leche, Reporte de reproducción, Reporte de producción.</li> <li>• El administrador al dar clic en cualquiera de estos iconos podrá generar el resumen de la categoría seleccionada.</li> <li>• El sistema mostrará un gráfico de la categoría seleccionada y un listado de la información.</li> </ul>
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El actor selecciona la opción de crear la información que se encuentra en el módulo de resumen.</li> <li>• Al momento de guardar presiona la opción Cancelar.</li> <li>• El sistema muestra la pantalla inicial del módulo de resumen.</li> </ul>
<b>Pos condiciones</b>	Ingreso a la página principal de módulo de resumen.

ANEXO 10

## Sybase PowerDesigner Physical Data Model Report

*Model: farmware\_fisico*  
*Report: Farmware*



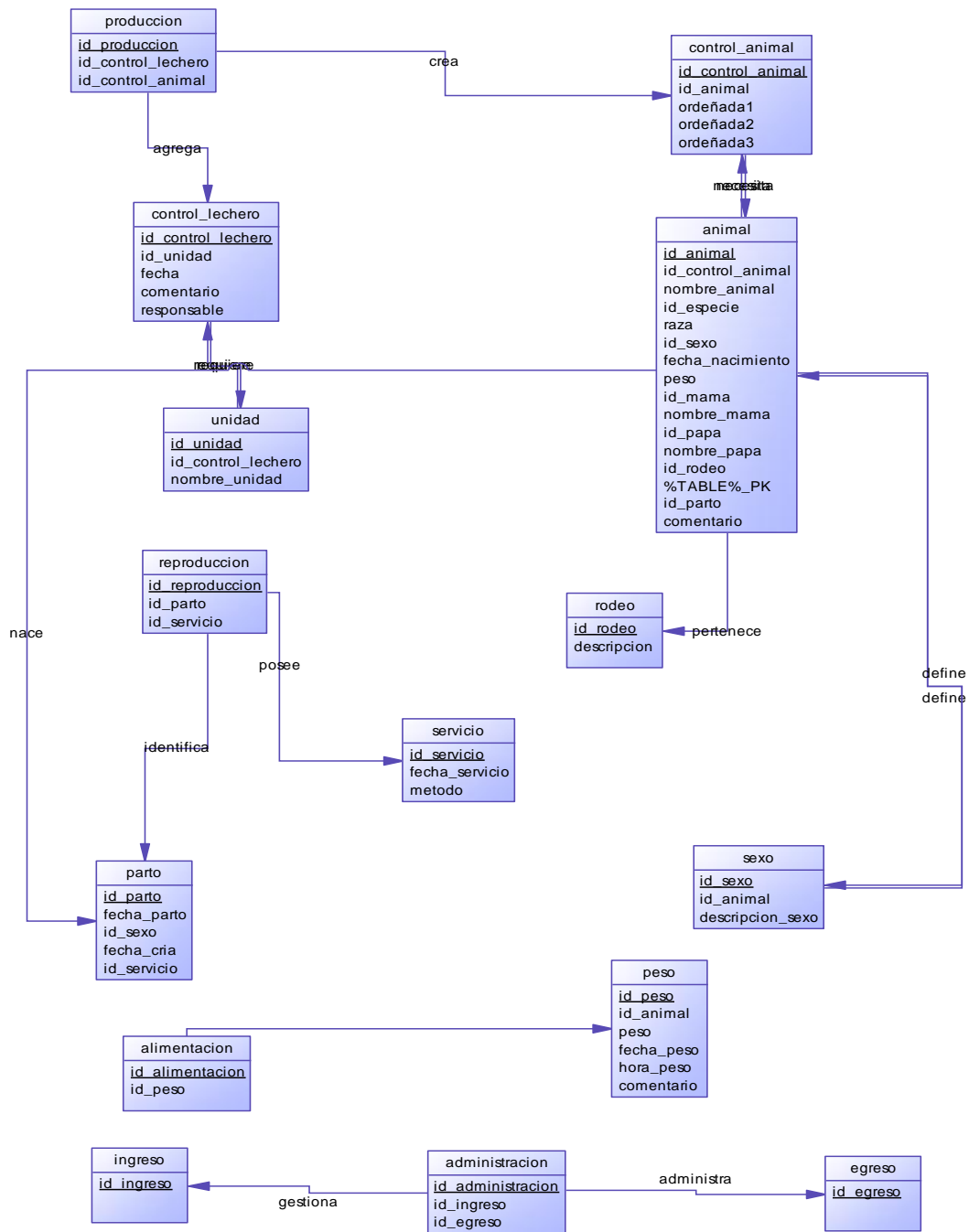
**Autora:** Sindy Vera  
**Año:** 2015

## List of tables

Name	Code
administracion	ADMINISTRACION
alimentacion	ALIMENTACION
animal	ANIMAL
control_animal	CONTROL_ANIMAL
control_lechero	CONTROL_LECHERO
egreso	EGRESO
ingreso	INGRESO
parto	PARTO
peso	PESO
produccion	PRODUCCION
reproduccion	REPRODUCCION
rodeo	RODEO
servicio	SERVICIO
sexo	SEXO
unidad	UNIDAD

## DIAGRAMA





## administracion

### List of columns of the table administracion

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_administracion	ID_ADMINISTRACION	integer		X		administracion	
id_ingreso	ID_INGRESO	integer			X	administracion	
id_egreso	ID_EGRESO	integer			X	administracion	

## Table alimentacion

### List of columns of the table alimentacion

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_alimentacion	ID_ALIMENTACION	integer		X		alimentacion	
id_peso	ID_PESO	integer			X	alimentacion	

## Table animal

### List of columns of the table animal

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_animal	ID_ANIMAL	integer		X		animal	
id_control_animal	ID_CONTROL_ANIMAL	integer		X	X	animal	
nombre_animal	NOMBRE_ANIMAL	varchar(20)	20			animal	

id_especie	ID_ESPECIE	integer					ani mal
raza	RAZA	varchar(20)	20				ani mal
id_sexo	ID_SEXO	varchar(20)	20				ani mal
fecha_nacimiento	FECHA_NACIMIENTO	timestamp					ani mal
peso	PESO	decimal					ani mal
id_mama	ID_MAMA	integer					ani mal
nombre_mama	NOMBRE_MAMA	varchar(50)	50				ani mal
id_papa	ID_PAPA	integer					ani mal
nombre_papa	NOMBRE_PAPA	varchar(50)	50				ani mal
id_rodeo	ID_RODEO	integer			X		ani mal
%TABLE% _PK	__TABLE__PK	integer		X	X		ani mal
id_parto	ID_PARTO	integer			X		ani mal
comentario	_13	varchar(100)	100				ani mal

### Table control\_animal

#### List of columns of the table control\_animal

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_control_animal	ID_CONTROL_ANIMAL	integer		X		control_animal	
id_animal	ID_ANIMAL	integer		X	X	control_animal	
ordeñada1	ORDENADA1	integer				control_animal	
ordeñada2	ORDENADA2	integer				control_animal	
ordeñada3	ORDENADA3	integer				control_animal	

## Table control\_lechero

### List of columns of the table control\_lechero

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_control_lechero	ID_CONTROL_LECHERO	integer		X		control_lechero	
id_unidad	ID_UNIDAD	integer		X	X	control_lechero	
fecha	FECHA	timestamp				control_lechero	
comentario	COMENTARIO	varchar (100)	100			control_lechero	
responsable	RESPONSABLE	varchar (50)	50			control_lechero	

## Table egreso

### List of columns of the table egreso

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_egreso	ID_EGRESO	integer		X		egreso	

## Table ingreso

### List of columns of the table ingreso

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_ingreso	ID_INGRESO	integer		X		ingreso	

## Table parto

### List of columns of the table parto

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_parto	ID_PARTO	integer		X		parto	

fecha_parto	FECHA_PARTO	timestamp				parto	
id_sexo	ID_SEXO	integer				parto	
fecha_criatura	FECHA_CRIATURA	timestamp				parto	
id_servicio	ID_SERVICIO	integer				parto	

### Table peso

#### List of columns of the table peso

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_peso	ID_PESO	integer		X		peso	
id_animal	ID_ANIMAL	integer				peso	
peso	PESO	decimal				peso	
fecha_peso	FECHA_PESO	date				peso	
hora_peso	HORA_PESO	time				peso	
comentario	COMENTARIO	varchar(100)	100			peso	

### Table produccion

#### List of columns of the table produccion

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_produccion	ID_PRODUCION	integer		X		produccion	
id_control_luchero	ID_CONTROL_LUCHERO	integer			X	produccion	
id_control_animal	ID_CONTROL_ANIMAL	integer			X	produccion	

### Table reproduccion

#### List of columns of the table reproduccion

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_reproduccion	ID_REPRODUCCION	integer		X		reproduccion	
id_parto	ID_PARTO	integer			X	reproduccion	
id_servicio	ID_SERVICIO	integer			X	reproduccion	

### Table rodeo

#### List of columns of the table rodeo

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_rodeo	ID_RODEO	integer		X		rodeo	
descripcion	DESCRIPCION	varchar(30)	30			rodeo	

### Table servicio

#### List of columns of the table servicio

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_servicio	ID_SERVICIO	integer		X		servicio	
fecha_servicio	FECHA_SERVICIO	timestamp				servicio	
metodo	METODO	varchar(20)	20			servicio	

### Table sexo

#### List of columns of the table sexo

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
------	------	-----------	--------	-----------	-------------	-------	-------------

					Key		
id_sexo	ID_SEXO	integer		X		sex	
id_animal	ID_ANIMAL	integer		X	X	o	
descripcion_sexo	DESCRIPCION_SEXO	varchar (20)	20			sex	
						o	

### Table unidad

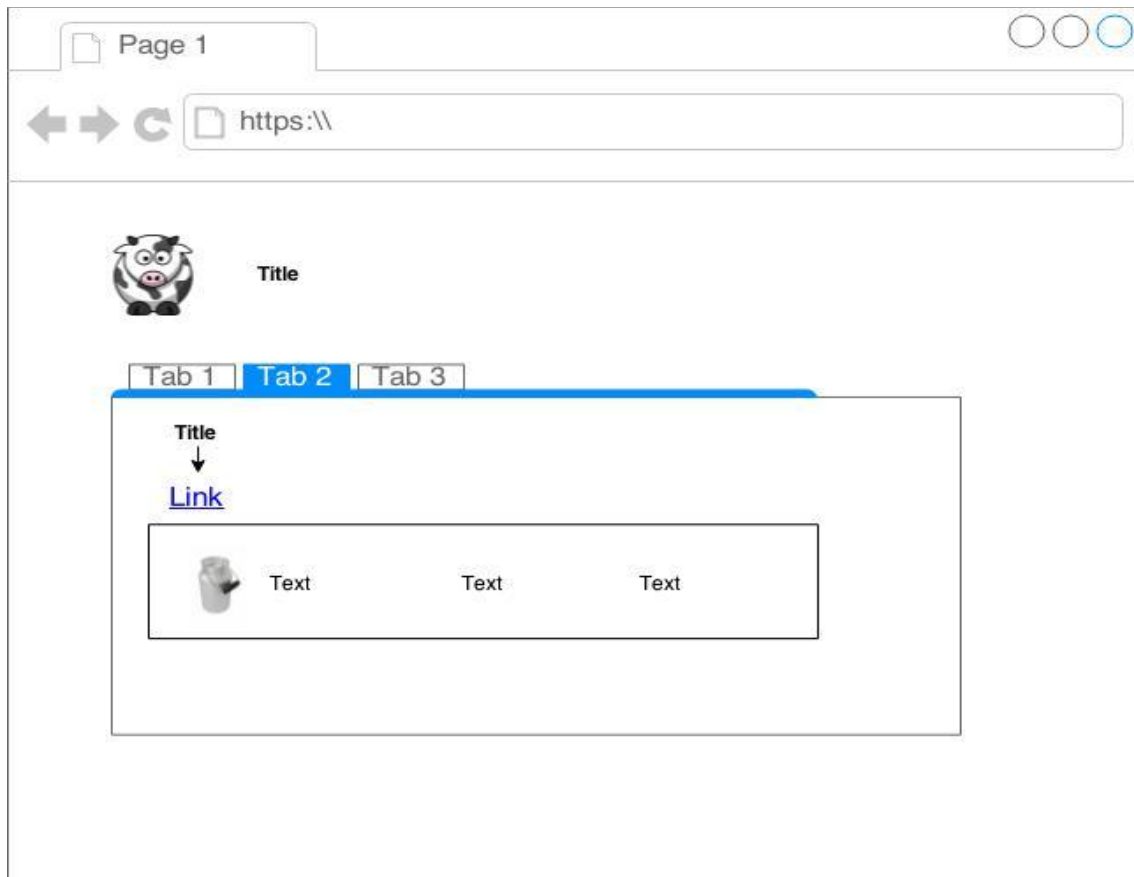
#### List of columns of the table unidad

Name	Code	Data Type	Length	Mandatory	Foreign Key	Table	Description
id_unidad	ID_UNIDAD	integer		X		unidad	
id_control_l echero	ID_CONTROL_L ECHERO	integer		X	X	unidad	
nombre_unidad	NOMBRE_UNIDAD	varchar(30)	30			unidad	

## ANEXO 11

### DISEÑO WEB (WIREFRAMES)

#### Página principal del Módulo producción de leche

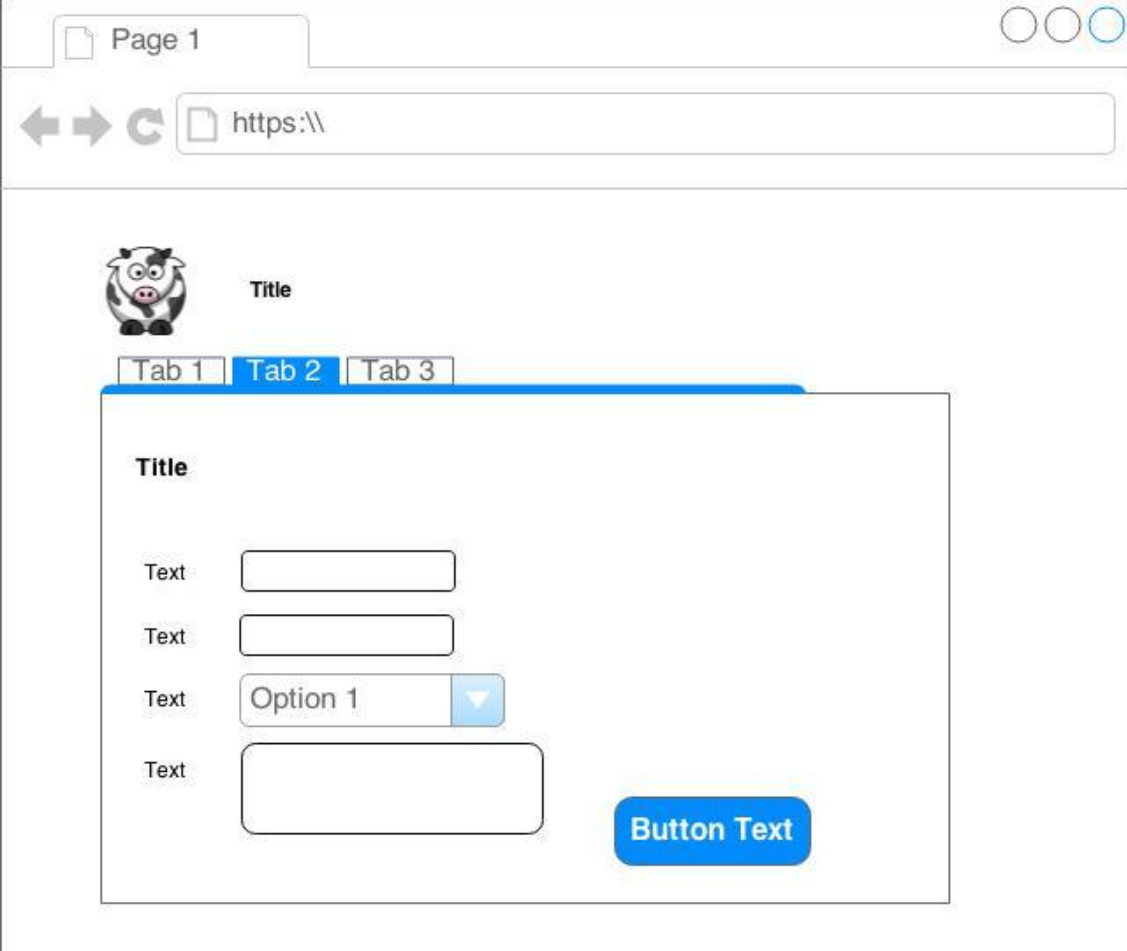


Al escoger el módulo de producción de mostrará la siguiente página. En la pantalla se presentara a manera de resumen el control lechero registrado de los animales de la Granja. Al dar clic en cualquier control lechero creado se presentara el detalle del control con los siguientes datos:

- Fecha
- Comentarios
- Responsable



## Agregar control lechero



The screenshot shows a web browser window with a single tab labeled "Page 1". The address bar contains "https:\\". The main content area displays a cow icon and a "Title" label. Below this is a tabbed interface with three tabs: "Tab 1", "Tab 2", and "Tab 3". The "Tab 2" tab is active and contains a form with the following elements:

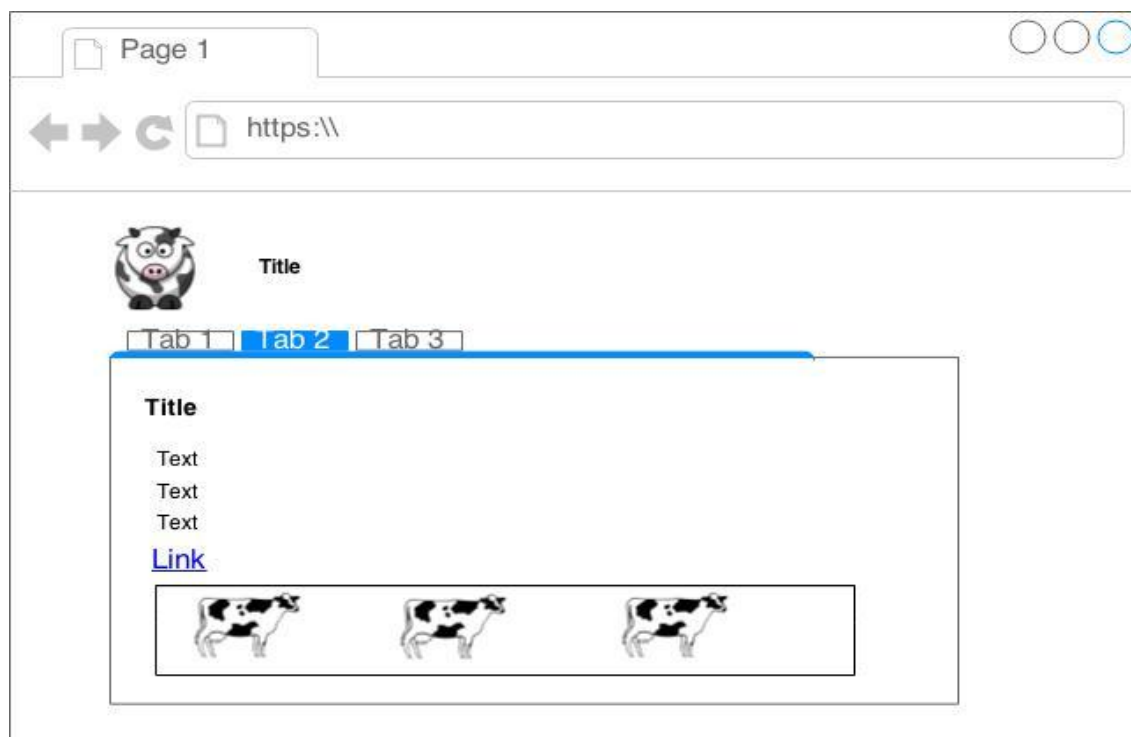
- A "Title" label.
- Four "Text" labels, each followed by an input field.
- A "Text" label followed by a dropdown menu with "Option 1" selected.
- A "Text" label followed by a large, empty text input field.
- A blue button labeled "Button Text".

Para agregar un nuevo control el administrador debe dar clic en el link **Agregar Control lechero**, se mostrará el siguiente formulario donde el usuario deberá ingresar los siguientes datos:

- Fecha, se refiere a la fecha del control actual.
- Comentarios, si existió algún percance al momento de realizar el control.
- Responsable, la persona encargada de realizar el control.

Una vez se registren todos los datos el administrador deberá dar clic en el botón **Agregar** y el sistema automáticamente guardara los datos.

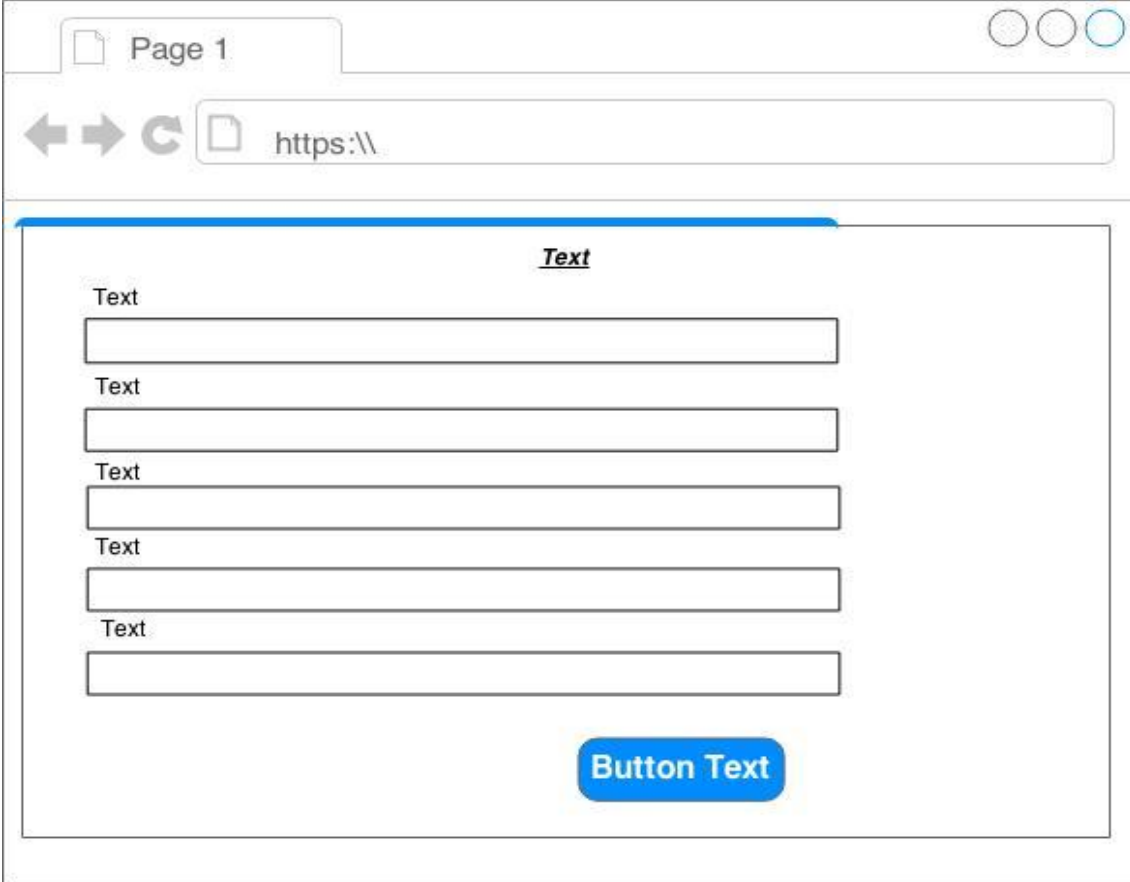
## Página del Control lechero creado



Al escoger el control lechero creado aparecerán los datos registrados por el administrador.

Para poder realizar el registro del control por animal el usuario deberá seleccionar la opción Agregar control por animal y de esta manera se mostrara el detalle de todos los animales ordeñados en cada control lechero.

## Agregar control por animal



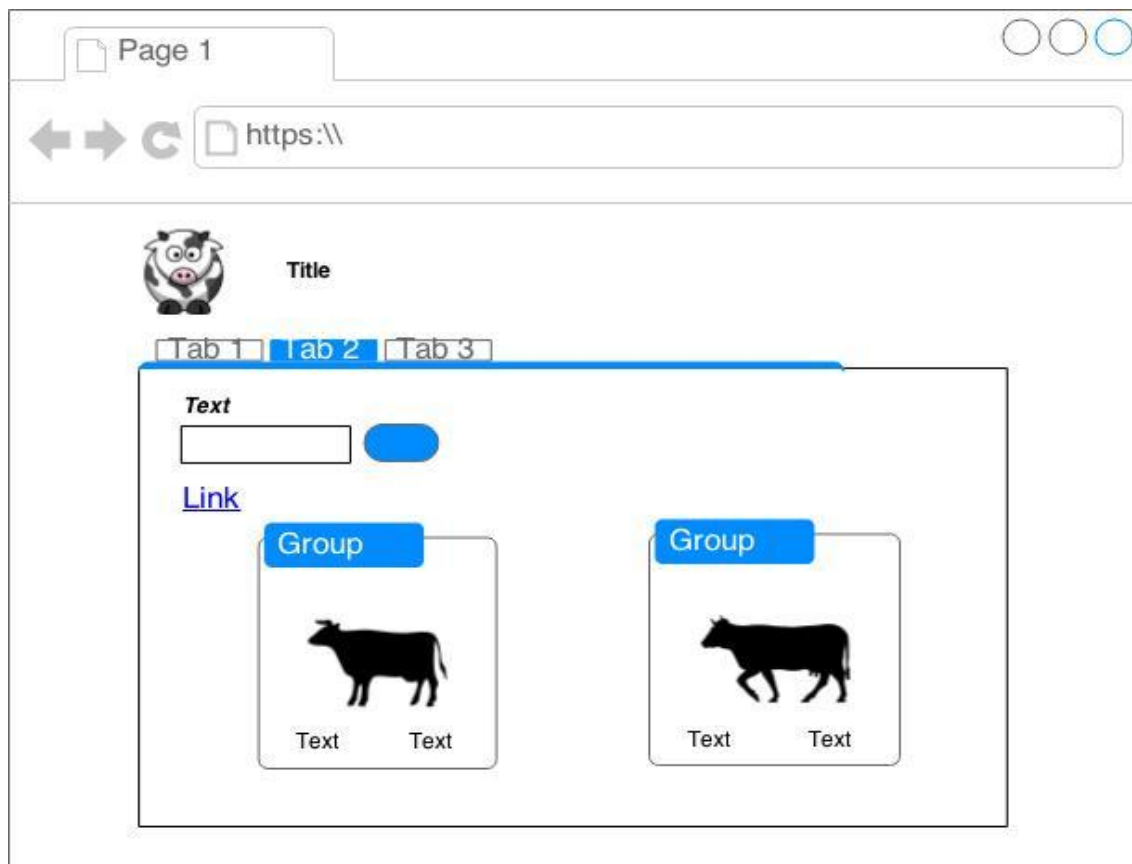
The image shows a web browser window with a single tab labeled 'Page 1'. The address bar contains 'https:\\'. The main content area displays a form with the following elements:

- A title 'Text' centered at the top of the form.
- Five text input fields, each preceded by the label 'Text'.
- A blue button labeled 'Button Text' positioned at the bottom right of the form.

Al dar clic en el link **Agregar control por animal**, el sistema direccionará a un formulario el mismo que el Administrador deberá llenar de manera obligatoria.

- Animal, se registrará el nombre del animal.
- Ordeñada 1, Cuanto se obtuvo en la primera ordeñada, este dato se ingresa de acuerdo a la unidad seleccionada.
- Ordeñada 2, Cuanto se obtuvo en la segunda ordeñada, este dato se ingresa de acuerdo a la unidad seleccionada.
- Ordeñada 3, Cuanto se obtuvo en la tercera ordeñada, este dato se ingresa de acuerdo a la unidad seleccionada.

## Página principal del Módulo de manejo de ganado



En la pantalla principal del Módulo de ganado se mostrarán todos los animales que se encuentran registrados en la Base de Datos del sistema.


El administrador podrá realizar la búsqueda de un animal en particular por medio del nombre.

En el recuadro aparecerá el nombre del animal, el nombre de sus progenitores, la especie y la raza a la que pertenece.

## Crear nuevo animal

Page 1

https:\\

 Title

Tab 1 Tab 2 Tab 3

**Text**

Text

Text

Text

Text

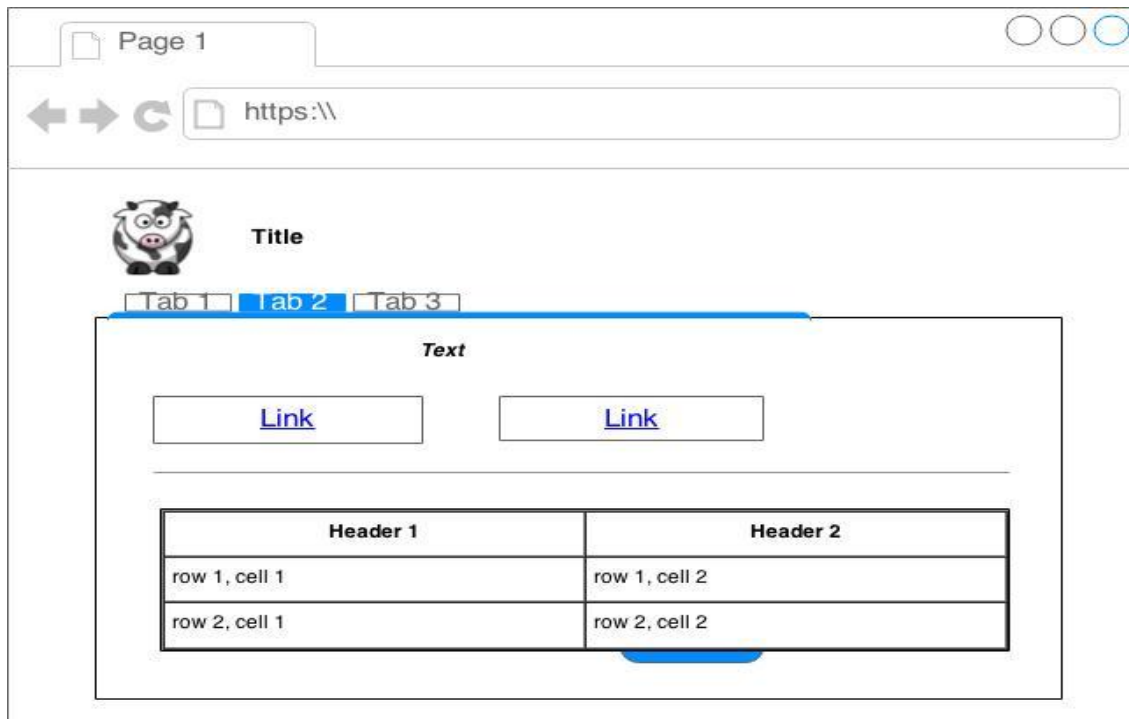
Text

Para ingresar un nuevo animal el administrador debe dar clic en el link **Crear Nuevo Animal**, el sistema mostrara un formulario en el cual se deben ingresar los siguientes datos:

1. Nombre, es el nombre de cada animal.
  - Especie, en este caso aparecerá solo la especie Bovina.
  - Raza, se debe ingresar la raza del animal.
  - Macho o Hembra, se debe escoger el género al que pertenece el animal.
  - Fecha de nacimiento, se ingresara la fecha de nacimiento del animal.
  - Peso, el peso al nacer de la cría.
  - Nombre Madre, el nombre de la madre.
  - Nombre Progenitor, se debe registrar el nombre del padre
  - Rodeo, el rodeo al que pertenece.
  - Comentarios, se registrara cualquier percance y observación sobre el animal.

Una vez que se encuentren llenos los campos el administrador deberá dar clic en el botón de Crear y el sistema se encargara de guardar toda la información registrada.

### **Página principal del Módulo de gestión de reproducción**

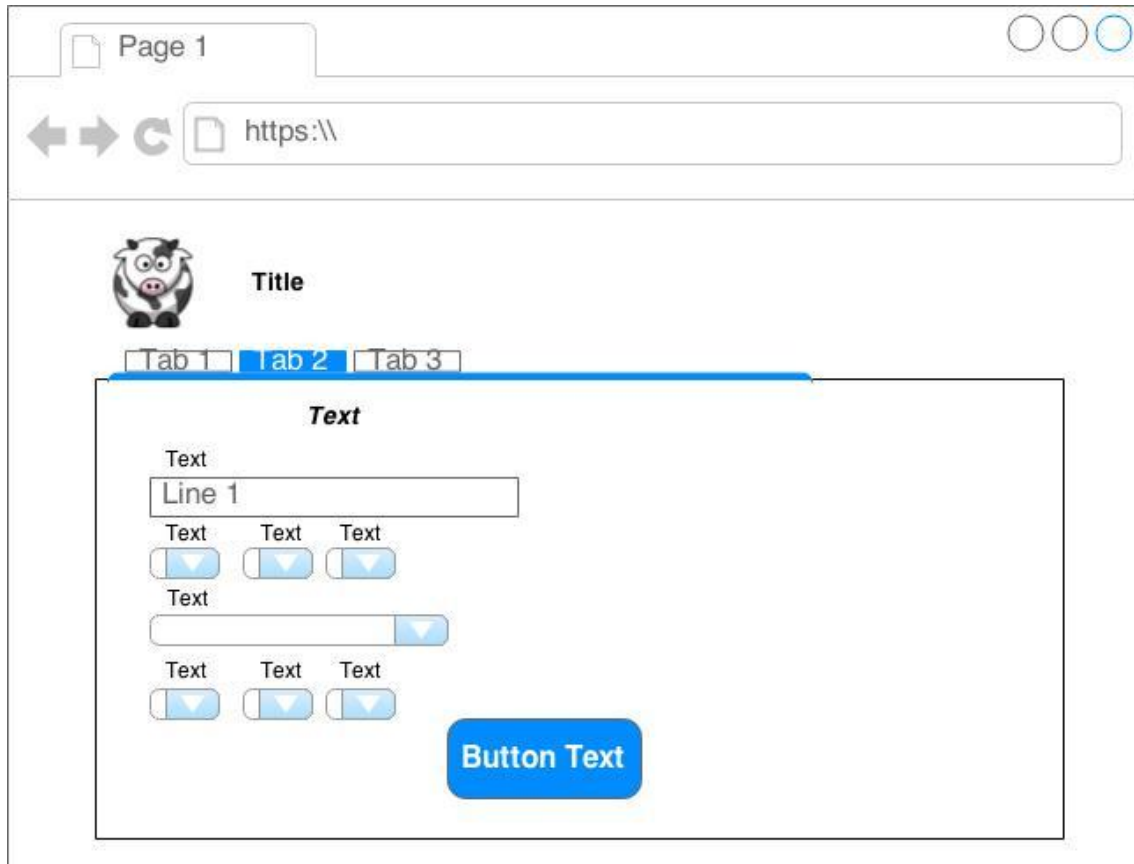


En el módulo de Gestión de Reproducción se mostrarán las siguientes opciones en la página principal:

- Agregar un parto
- Agregar un servicio.

Adicional se mostrará un detalle de los partos y servicios de los animales registrados.

## Agregar un parto



Para agregar un nuevo parto el administrador deberá clic en el link Agregar un parto se deben ingresar los siguientes datos:

- Fecha de parto, la fecha de parto del animal.
- Sexo de la cría, el sexo de la cría.
- Fecha de servicio, es la fecha del inicio del ciclo de reproducción.

El administrador deberá ingresar todos los datos en los campos de manera obligatoria y dar clic en el botón Agregar un Parto.

### **Agregar un servicio**

The screenshot displays a web browser window with a single tab titled "Page 1". The address bar shows a partial URL "https:\\". The main content area contains a cow icon, a "Title" label, and a tabbed interface with three tabs: "Tab 1", "Tab 2" (which is active and highlighted in blue), and "Tab 3". Within the active tab, there is a section titled "Text" containing several form elements: a "Text" label above a "Line 1" text input field; three "Text" labels, each followed by a dropdown menu; another "Text" label followed by a dropdown menu; and a final "Text" label followed by a dropdown menu. A blue button labeled "Button Text" is positioned at the bottom right of the form area.

Para agregar un nuevo servicio el administrador deberá dar clic en el link Agregar un Servicio.

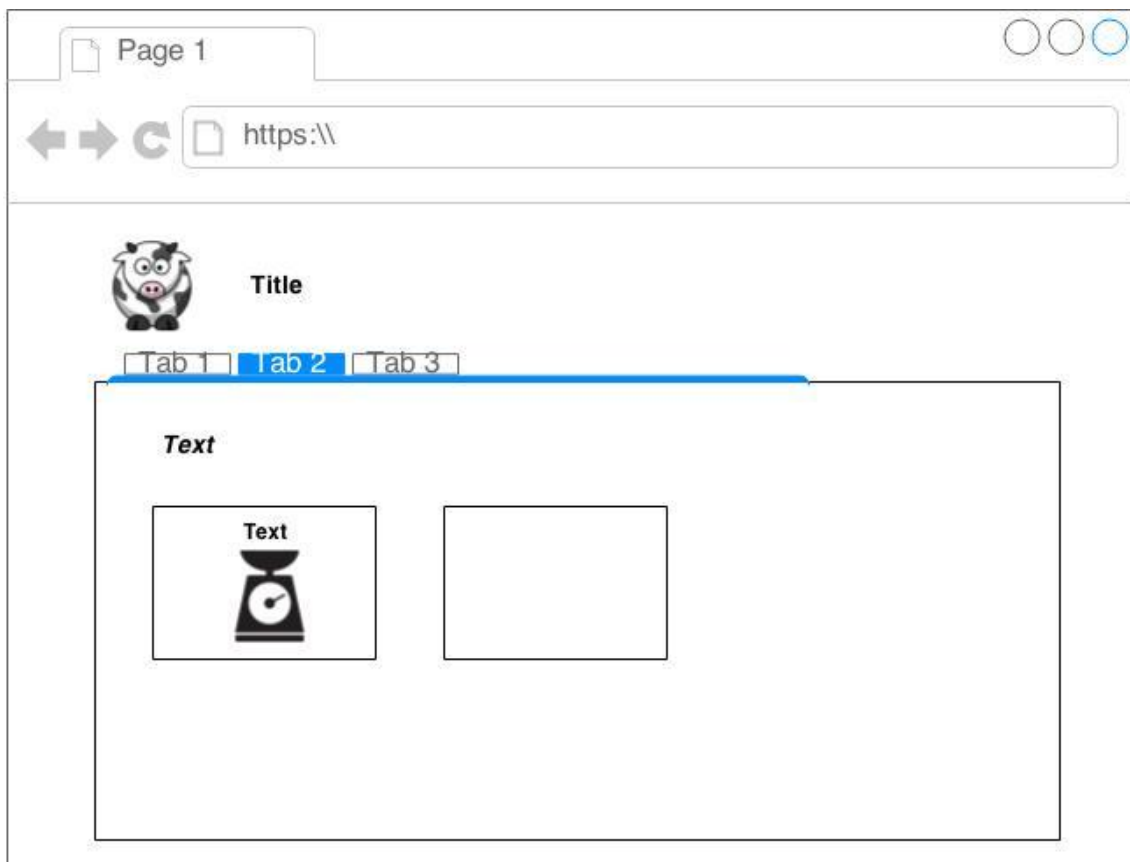
El sistema mostrara un formulario en el cual el administrador debe ingresar los siguientes datos:



- **Fecha de servicio**, es la fecha en la que inicia el acto de reproducción del animal.
- **Método de reproducción**, puede ser por monta natural, inseminación u otros.

El administrador deberá ingresar todos los datos en los campos de manera obligatoria y dar clic en el botón Agregar servicio.

### **Página principal del Módulo de gestión alimenticia**



En la pantalla principal del módulo de Gestión Alimenticia se muestra la opción para Cargar el peso de un animal.

El administrador deberá dar clic en la opción Cargar peso de animal para poder registrar el pesaje de cada uno de los animales de la granja.

## Cargar peso de animal

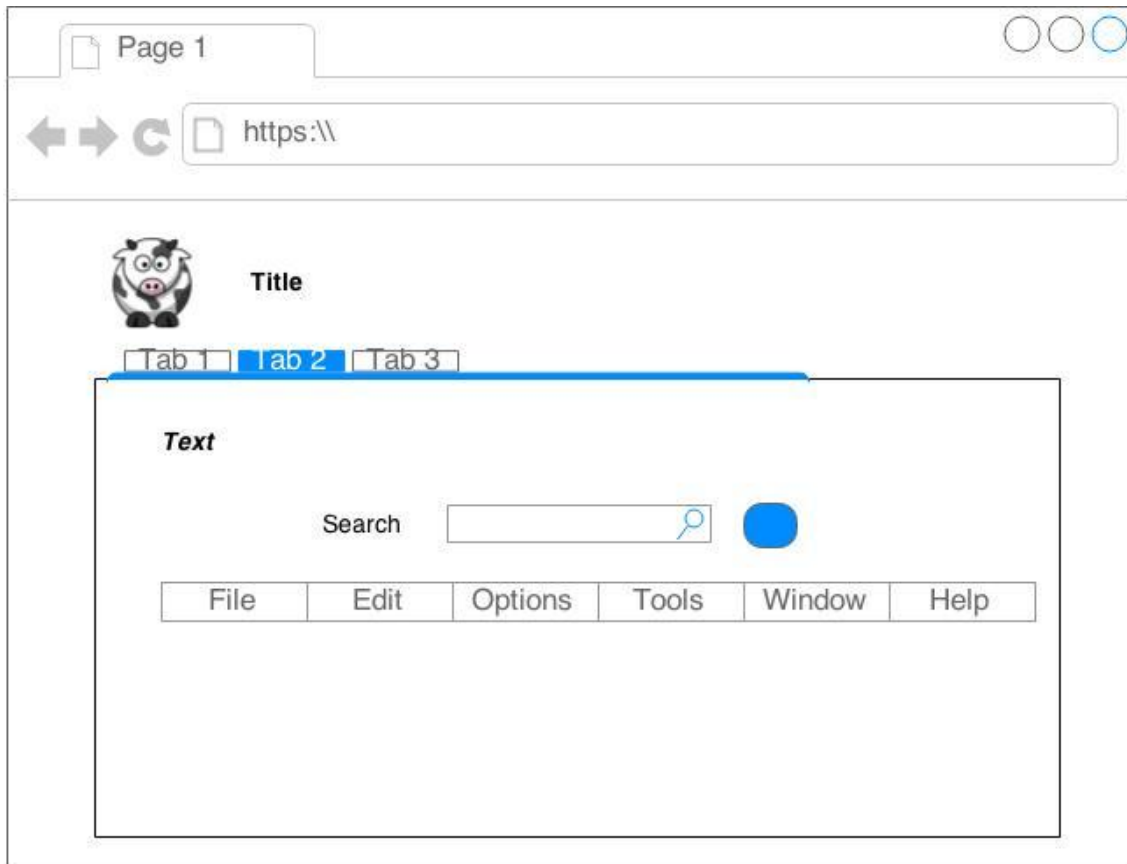
The image shows a web browser window with a tab labeled "Page 1" and a URL bar containing "https:\\". The page content includes a cow icon, a "Title" label, and three tabs: "Tab 1", "Tab 2", and "Tab 3". The main content area contains a form with several "Text" labels, input fields, and three dropdown menus. A blue button labeled "Button Text" is positioned to the right of the form.

El administrador deberá dar clic en el icono de Cargar peso de un animal y el sistema traerá el formulario en el cual se ingresaran los siguientes campos:

- Animal, se registrara el nombre del animal.
- Fecha de Peso, la fecha en que se está pesando el animal.
- Hora de Peso, la hora de pesaje.
- Peso, el peso en libras del animal.
- Comentarios, en caso de que exista alguna observación del animal al momento del pesaje.

Una vez registrados todos los campos el administrador dará clic en crear y el sistema guardará los datos.

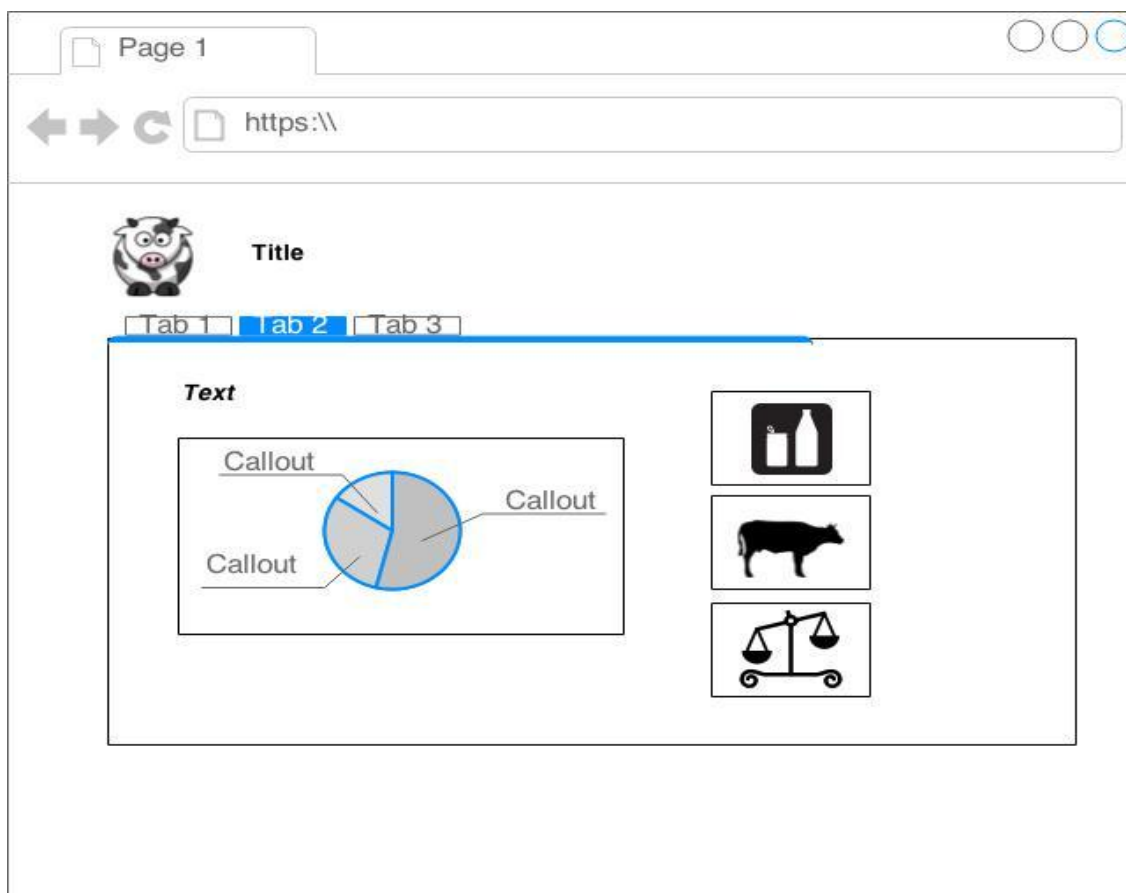
## Peso de un animal



En esta pantalla se mostrará el listado del peso de cada animal registrado.

El administrador podrá realizar la búsqueda ingresando el nombre del animal y dando clic en el botón buscar.

## Pantalla principal del Módulo de resumen



En la pantalla principal del módulo de Resumen se muestran los siguientes iconos:

- Reporte de Leche,
- Reporte de Reproducción,
- Reporte de Producción.

El administrador al dar clic en cualquiera de estos iconos podrá generar el reporte de la categoría seleccionada.

El sistema mostrará un gráfico de la categoría seleccionada y un listado de la información.

**ANEXO 12**



# **GUIA DE INSTALACIÓN DE FIRMWARE**

**Autor: Sindy Vera Cedeño**

**2015**

Para instalar Farmware se debe realizar lo siguiente:

## **PASO 1:**

### **Requerimientos de Hardware y de Software**

Requerimientos para la instalación del sistema web a partir de los archivos de aplicación.

Para compilar los archivos fuente se debe instalar lo siguiente:

- ✓ Sistema Operativo Windows 7 (Procesador 64 bits (x64), Memoria RAM de 4 (GB).
- ✓ Sql server 2008
- ✓ Visual Studio Ultimate 2012

## **PASO 2:**

### **Instalar Sql Server 2008**

1. Instalación de SQL Server 2008.
2. Hacer clic derecho en la aplicación SETUP.EXE y ejecutar como Administrador la aplicación.
3. Aparecerá la ventana del **Centro de instalación de SQL Server**, seleccionar la opción Instalación.
4. Hacer clic en la opción **Nueva instalación independiente de SQL Server**.
5. Se instalarán las **Reglas auxiliares del programa de instalación**.
6. Se ejecutaran las actualizaciones del producto, presionar siguiente para instalar los archivos de configuración.
7. En la ventana **Tipo de instalación**, seleccionar la opción **Realizar una nueva instalación de Sql Server 2008**, aparecerán las instancias que han sido instaladas.
8. Ingresar la **Clave del producto** y clic en aceptar los términos de licenciamiento.

9. En **Rol de instalación**, hacer clic en la opción **Instalación de características de SQL Server**, se instalará las Reglas de Instalación y se comprobará que todo se encuentre correcto.
10. En **Configuración de la Instancia**, hacer clic en el botón Siguiente.
11. Colocamos un nombre a la instancia de la base de datos, en este caso colocaremos SQL2008 y damos clic en Next.
12. En **Configuración del servidor** dejar las opciones que se encuentran por defecto.
13. En **Configuración del Motor de base de datos**, seleccionar la opción de **Agregar el usuario actual** y en **Analysis Services** seleccionar la opción Agregar el usuario actual.
14. La Aplicación está lista para ser instalado en la configuración del servidor, se mostrará el avance de la instalación e iniciar el motor de base de datos.

Una vez instalado el Sql Server 2008 es necesario **restaurar la copia de seguridad** de la base Farmware, la misma se encuentra en la carpeta aplicación.

1. Autenticarse en la instancia de SQL Server 2008 (login) mediante SQL Server Management Studio en este caso **PATRICIA\SQL2008**.
2. Crear Base de datos **Farmware** y dar clic derecho sobre la base. Seleccionar la opción de **Restaurar en Base de datos**.
3. Seleccionar la Base de datos que será restaurada desde un dispositivo. Ubicar el archivo .bak (farmware.bak) de origen, el respaldo de la base se encuentra en la carpeta TESIS\_VERA\Farmware\_backup.
4. Marcar el archivo de respaldo con la opción de **Restaurar**:
5. Ir al tab **Opciones**, y seleccionar la opción de **Overwrite the existing database** y escoger la dirección donde se va a restaurar nuestra base.
6. Hacer clic en el botón OK cuando el proceso de restauración haya culminado.

Cadena de Conexión: "Data Source=PATRICIA\\SQL2008;Initial Catalog=farmware;Integrated Security=True"

### PASO 3:

## Instalar IIS (*INTERNET INFORMATION SERVICES*) y las características de ASP.NET

Para la creación de un sitio web en IIS se debe instalar los módulos correspondientes a ASP.NET y luego se agregan los archivos de aplicación.

1. Instalar características ASP.NET en IIS (*Internet Information Services*).
2. En nuestra página de Inicio, hacer clic en Panel de Control.
3. Hacer clic en el icono **Programas y características**.
4. Al lado izquierdo del recuadro, hacer clic en Activar o desactivar las características de Windows.

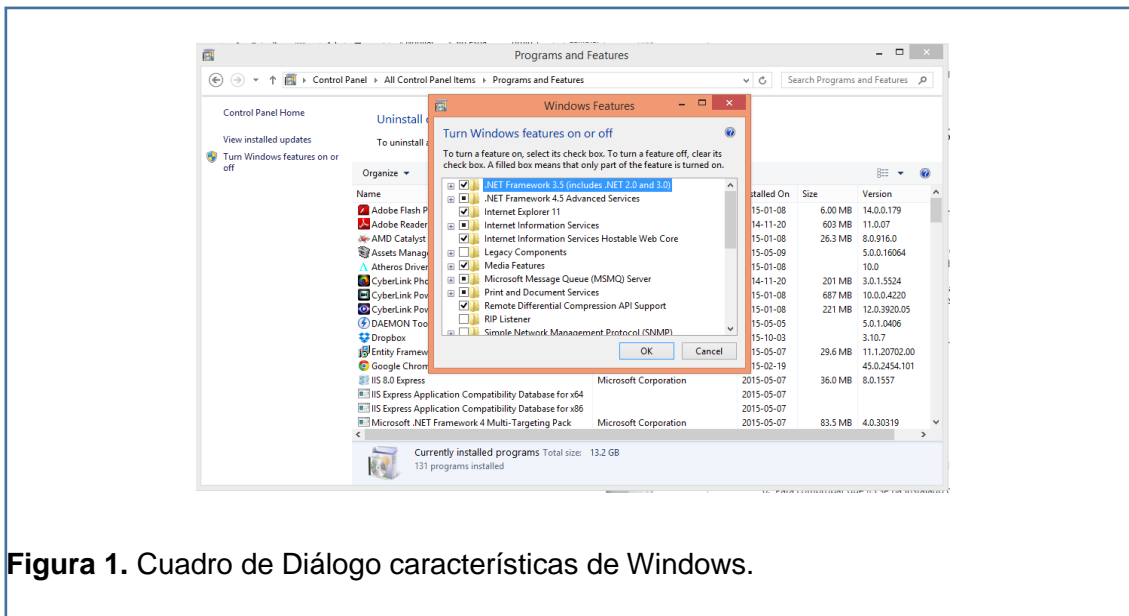


Figura 1. Cuadro de Diálogo características de Windows.

5. Dar clic en Internet Information Services para instalar los módulos correspondientes.
6. Expandir **Características de desarrollo de aplicaciones** y hacer clic en ASP.NET 4.5.



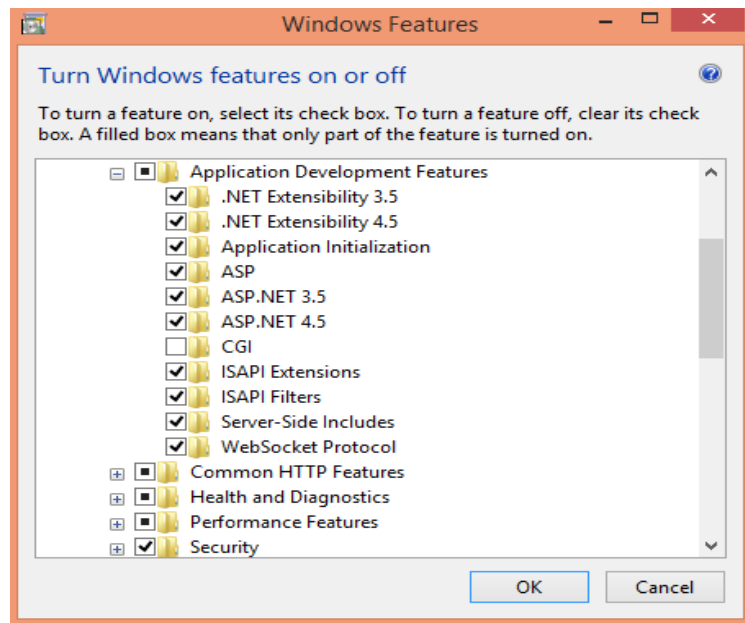


Figura 2. Cuadro de Características de desarrollo de aplicaciones.

7. Hacer clic en el botón **Aceptar** y automáticamente se cerrará el cuadro de dialogo de **Características de Windows**.
8. Para verificar que el IIS se instaló de manera correcta, se debe escribir en el explorador la url: <http://localhost>.

Debe aparecer la página de bienvenida de Internet Information Services.

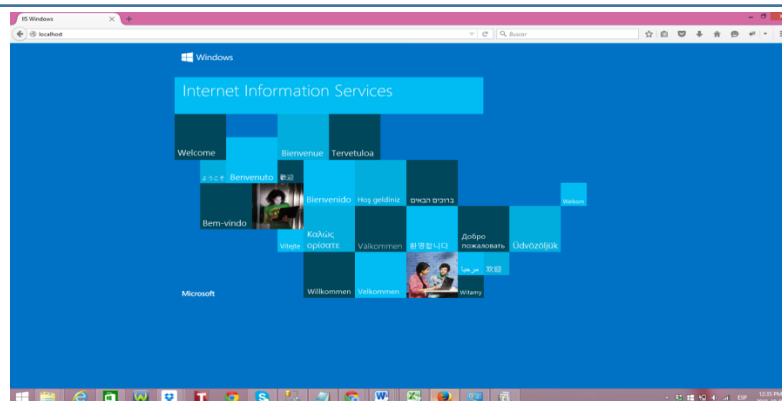


Figura 3. Pantalla de Bienvenida de IIS.

## PASO 4:

### INICIAR EL ADMINISTRADOR DE INTERNET INFORMATION SERVICES.

En este paso se publicara el sitio web local en nuestro servidor IIS:

1. En Inicio, hacer clic en **Ejecutar**.
- 5 En el cuadro de Abrir se debe escribir **Panel de Control**, seguidamente hacer clic en el botón Aceptar.
- 6 En el cuadro de Panel de Control, hacer clic en el icono de **Herramientas administrativas**.

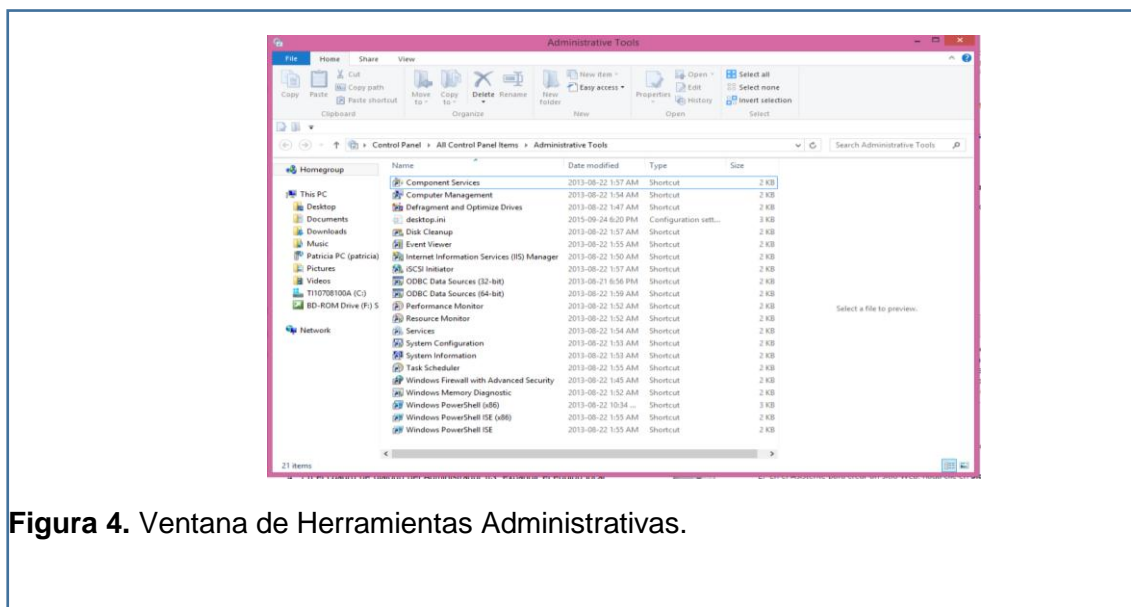
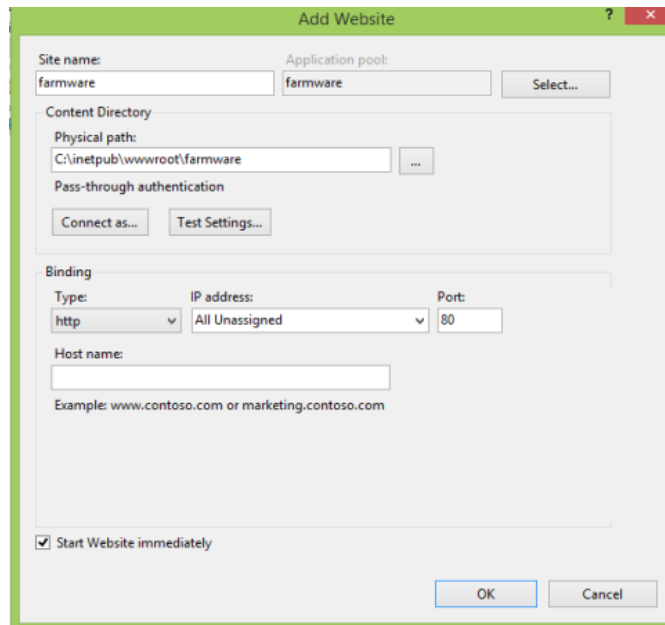


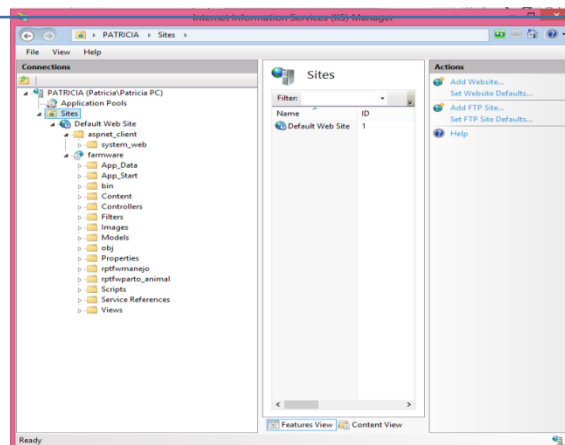
Figura 4. Ventana de Herramientas Administrativas.

- 7 En la ruta **C:\inetpub\wwwroot\** se deberá crear una nueva carpeta llamada **farmware**, la misma que tendrá la solución de la aplicación.
- 8 En el cuadro de dialogo del **Administrador IIS**, expandir el equipo local y en **Sitios Web** aparecerá una carpeta llamada **Farmware**.
- 9 Dar clic derecho sobre la carpeta y seleccionar la opción **Convertir en aplicación**, hacer clic en **Aceptar**.



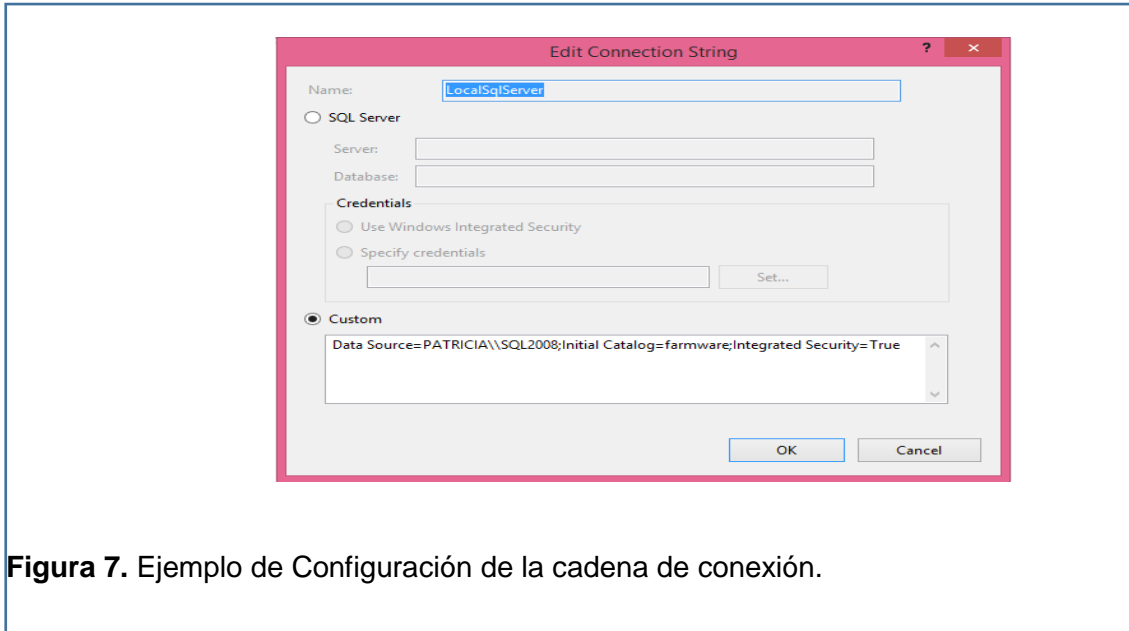
**Figura 5.** Ventana para agregar un nuevo sitio web.

10 Expandir el sitio Farmware y verificar que se muestren los archivos de aplicación.

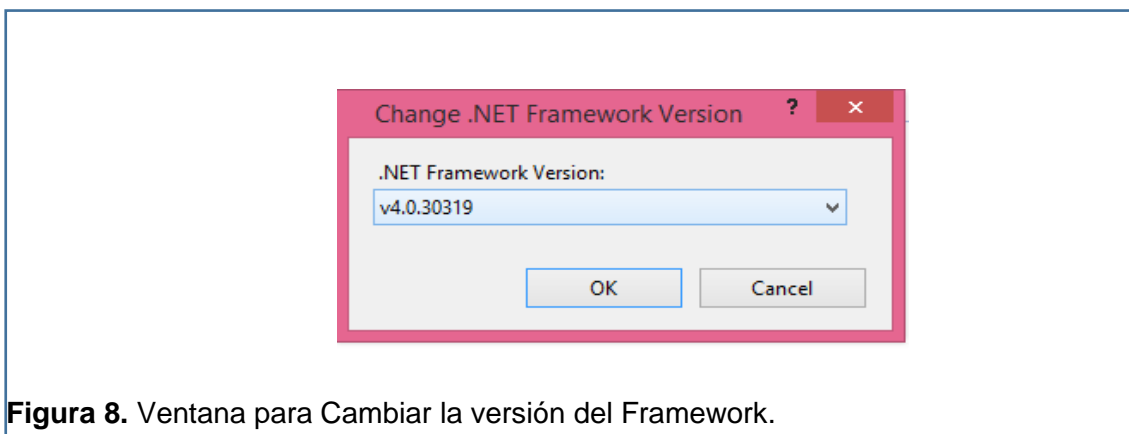


**Figura 6.** Ejemplo de la creación del sitio web.

8. Hacer clic en el icono de cadenas de conexión que se encuentra en el módulo ASP.NET, dar clic en la opción Modificar para configurar la cadena.



9. En el lado derecho del Administrador de IIS, en la sección **Acciones** hacer clic en **Cambiar la versión del .NET Framework** y escoger la 4.0, dar clic en **Aceptar**.



10. Hacer clic en **Grupo de aplicaciones**, seleccionar **DefaultAppPool** y dar clic derecho y luego escoger **Configuraciones Avanzadas**.
11. En la ventana de **Configuraciones Avanzadas** escoger en **Versión de .NET Framework** la v4.0, hacer clic en **Aceptar**.

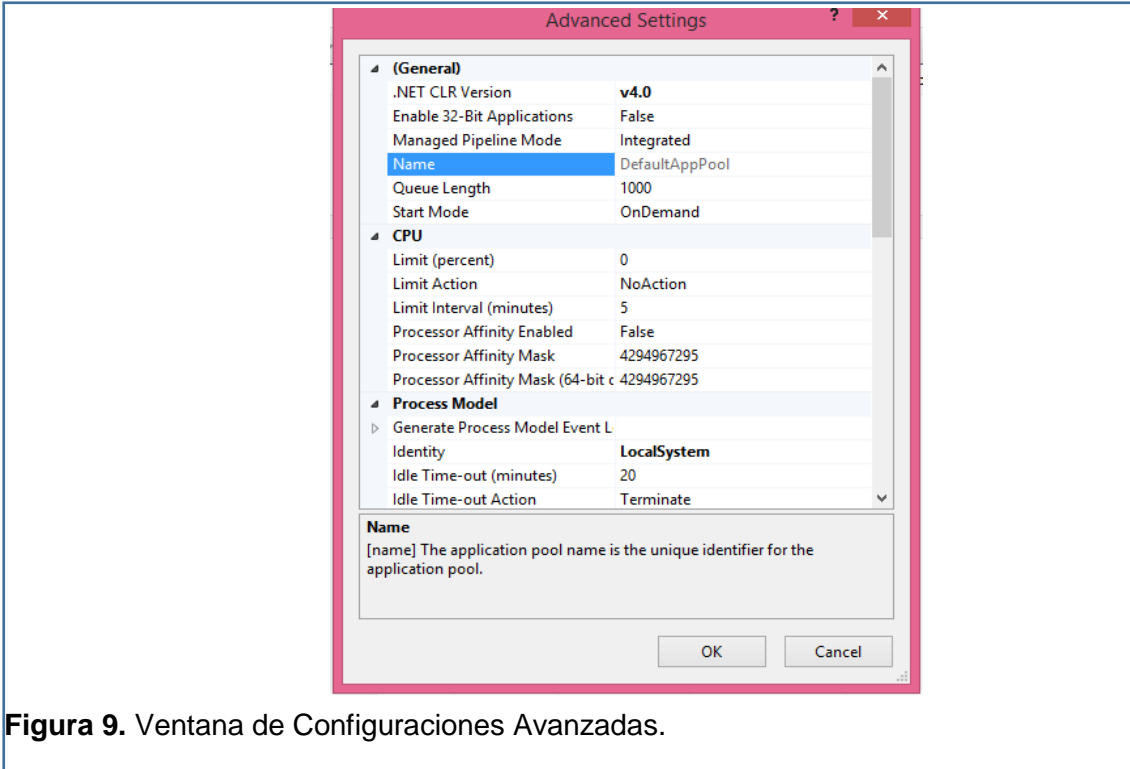


Figura 9. Ventana de Configuraciones Avanzadas.

- 12. Reiniciar el IIS.
- 13. Seleccionar la aplicación web publicada (**Farmware**), hacer clic en **Examinar \*:80 (http)** y deberá aparecer la página principal del sistema.

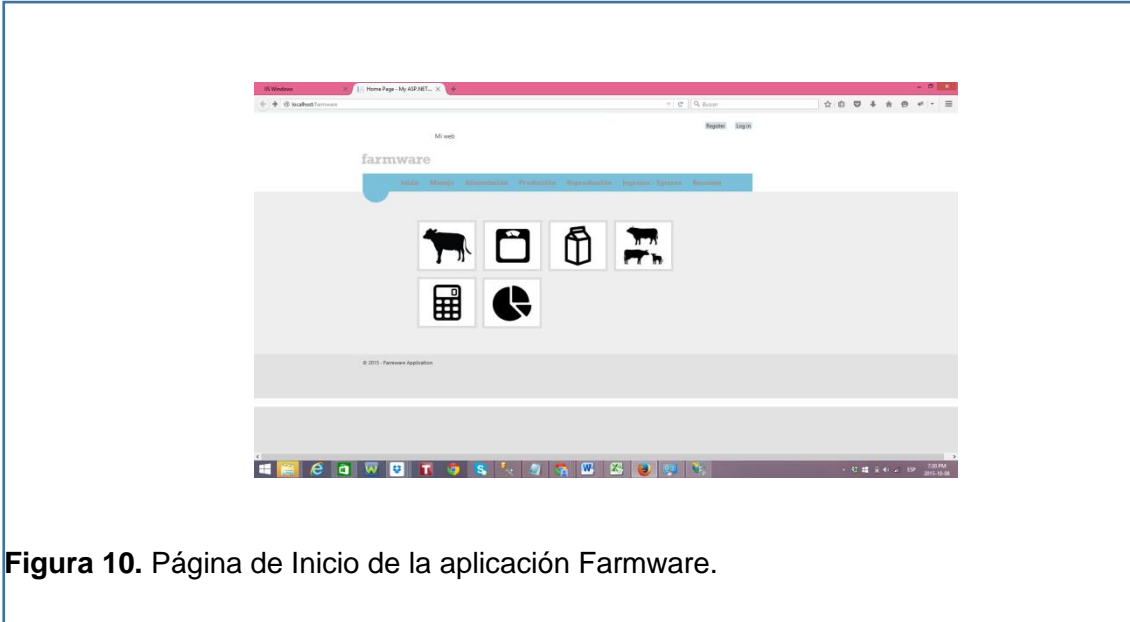


Figura 10. Página de Inicio de la aplicación Farmware.

14. Si al publicar la aplicación web .NET no se cargan los estilos css en todas las páginas se debe activar en el Administrador del IIS el módulo de presentación de contenido estático.

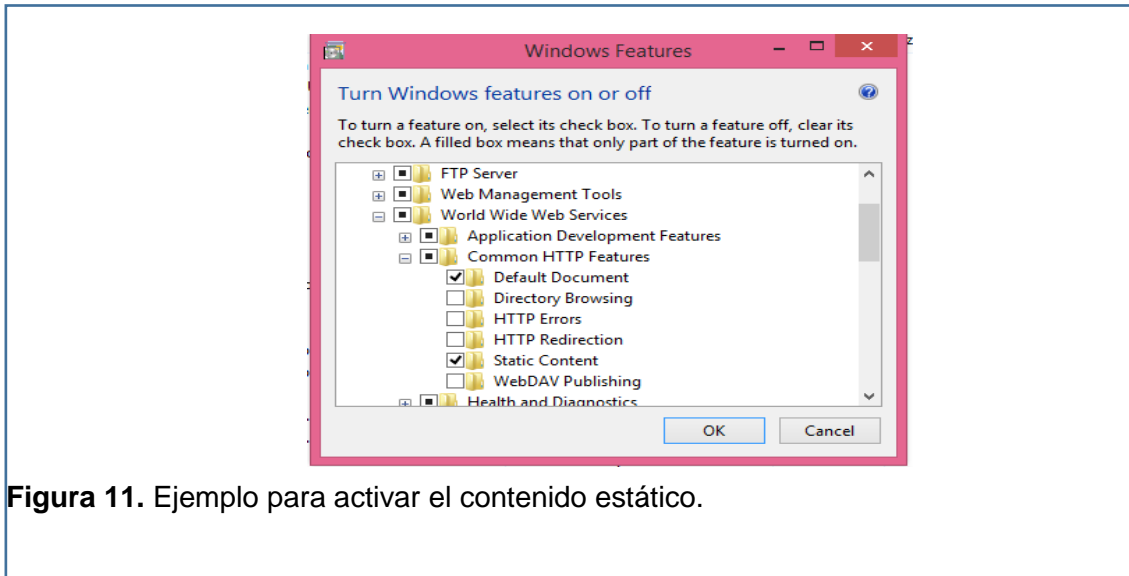


Figura 11. Ejemplo para activar el contenido estático.

## PASO 5:

### INSTALACIÓN VISUAL STUDIO ULTIMATE 2012.

1. Ejecutar la aplicación Visual Studio.
2. En la ventana siguiente aceptar los términos y condiciones de la licencia.
3. Seleccionar de la lista de características las que se desean agregar a la instalación.
4. Se iniciará el proceso de instalación y a continuación se observará una ventana con el mensaje de que la instalación ha concluido satisfactoriamente.
5. Al finalizar la instalación se copiarán los archivos fuentes del aplicativo los mismos que se encuentran en el cd de la tesis en la carpeta **Archivos Fuentes** en la siguiente ubicación TESIS\_VERA\farmware.

A continuación se procederá a abrir el aplicativo web.

6. Seleccionar la opción **Archivo – Abrir -- Proyecto o solución.**

7. Se abrirá una ventana en la que se debe seleccionar el proyecto **“farmware”** y seguidamente se cargará el proyecto.
8. Para ejecutar nuestra aplicación debemos seleccionar el explorador con el que se va a trabajar (**MOZILLA FIREFOX**).
9. Dar clic en el botón **Ejecutar**.

**ANEXO 13**



# **MANUAL DE USUARIO DE FARMWARE**

**Autor: Sindy Vera Cedeño**

**2015**



## TABLA DE GRÁFICOS

Figura 1. Representación de la página de autenticación del aplicativo Farmware donde el usuario debe colocar sus credenciales.

Figura 2. Página de inicio del aplicativo web y su Barra de Menú

Figura 3. Pantalla principal del módulo de Manejo de Animales.

Figura 4. Pantalla principal de Carga Rápida de Animal.

Figura 5. Pantalla Principal de la vista Detalle Carga de Animales.

Figura 6. Pantalla principal del Módulo de Gestión de Alimentación.

Figura 7. Pantalla principal de la Carga de Peso por Animal.

Figura 8. Pantalla Principal de la vista Detalle Animales Pesados.

Figura 9. Pantalla Principal del módulo de Gestión de Producción.

Figura 10. Pantalla principal para Agregar un nuevo control.

Figura 11. Pantalla del Módulo de Gestión de Producción.

Figura 12. Pantalla para Crear Ordeñada de cada Animal.

Figura 13. Pantalla para visualizar el Detalle de Ordeñada.

Figura 14. Pantalla para Crear Ordeñada de cada Animal.

Figura 15. Pantalla para visualizar el Detalle de Ordeñada.

Figura 16. Pantalla Principal de Módulo de Gestión de Reproducción.

Figura 17. Pantalla para Registrar un Servicio de Inseminación.

Figura 18. Formulario para Registrar un nuevo parto.

Figura 19. Pantalla para visualizar el Detalle de Partos.

Figura 20. Pantalla principal del Módulo de Ingresos y egresos.

Figura 21. Formulario para el registro de Ingreso por actividad.

Figura 22. Formulario para el registro de egreso por actividad.

Figura 23. Página principal del Módulo de Resumen.

## 1. Introducción.

### ¿Qué es Farmware?

Es un sistema que permite la gestión de ganado bovino mediante los siguientes módulos:

- ✓ **Producción de leche**

Lleva un registro del Control lechero y control por ordeñada de cada animal.

- ✓ **Manejo de ganado**

Permite ingresar los datos de un nuevo animal.

- ✓ **Gestión de Reproducción**

Agrega un parto y un nuevo servicio al sistema.

- ✓ **Gestión Alimenticia**

Almacena información sobre el pesaje de cada animal.

- ✓ **Ingresos y Egresos**

Se registran los gastos e ingresos de la venta de producción de leche.

- ✓ **Resumen**

Muestra un listado de reportes.

### ¿Cuál es el objetivo del Manual de Usuarios de Farmware?

El objetivo del presente manual, es dar a conocer a los usuarios que utilicen el aplicativo todas las características y el modo de funcionamiento de Farmware.

## 2. Requisitos del sistema

Para la utilización del aplicativo “**Farmware**” se necesita tener instalado localmente o en un servidor lo siguiente:

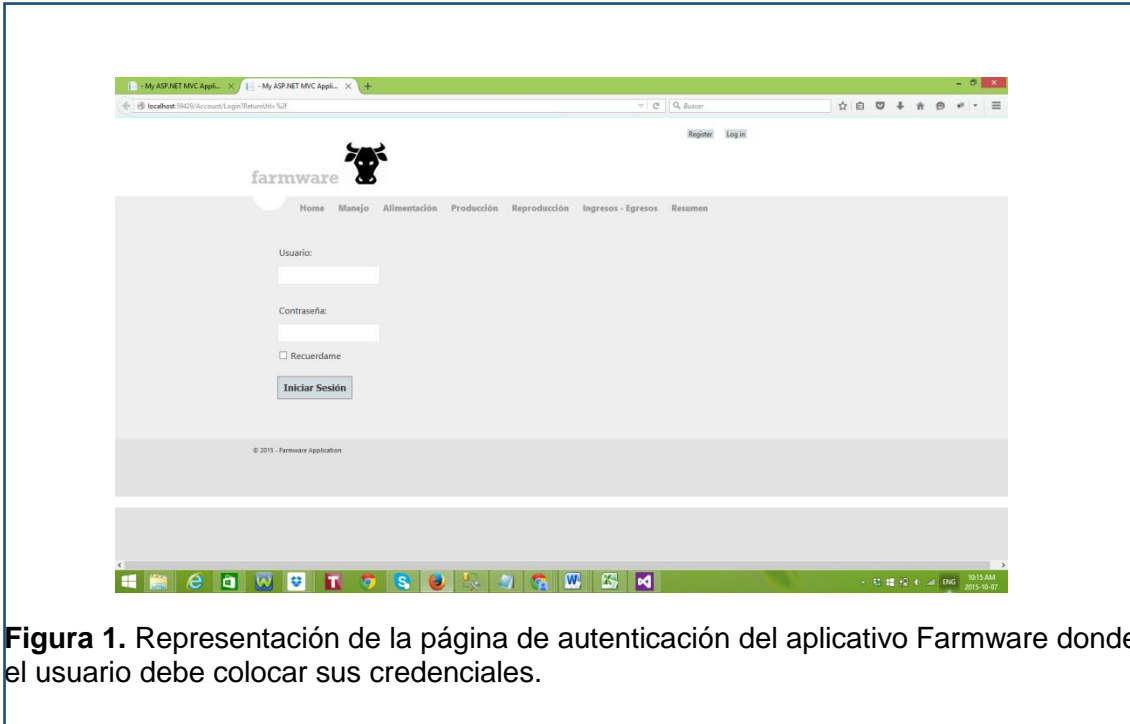
- ✓ SQL Server
- ✓ Habilitadas las características del IIS
- ✓ Navegador (Mozilla Firefox, Internet Explorer)

Farmware es una aplicación web, lo que facilita que una vez instalado en un computador, cualquier usuario pueda acceder desde otro equipo que se encuentre conectado a la misma red.

## 3. Ingreso a la Aplicación

Para acceder al sistema se debe acceder a la siguiente URL: <http://localhost/farmware/>, seguidamente se debe ingresar el usuario y contraseña entrar a la aplicación. Para ingresar a la aplicación desde un equipo del departamento donde se encuentre instalado localmente se debe escribir en el navegador la siguiente URL: <http://localhost/farmware/>. De manera automática se visualizará la pantalla de la Figura 1.

Esta pantalla es de autenticación del usuario; cabe mencionar que solo ingresan los usuarios que antes fueron registrados en el aplicativo, si no es usuario del sistema se deberá registrar en el link **Register** y llenar todos los campos mostrados en la pantalla, luego debe pulsar el botón **Iniciar Sesión** o presionar la tecla Enter del teclado, de esta manera el usuario ha ingresado al sistema.



**Figura 1.** Representación de la página de autenticación del aplicativo Farmware donde el usuario debe colocar sus credenciales.

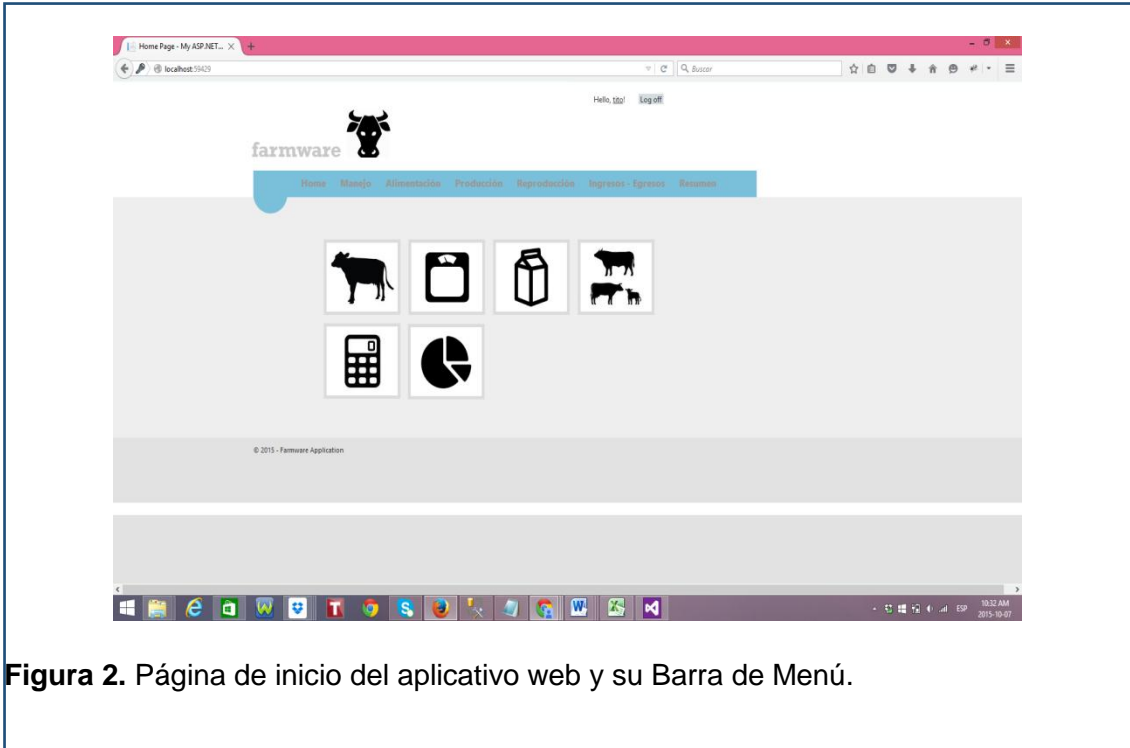
#### 4. Roles

El rol es la categorización que tienen los usuarios dentro de la aplicación, de esta manera se controla el trabajo de cada usuario otorgando ciertos privilegios según el rol.


- ✓ **Administrador:** este rol tiene todos los privilegios dentro del aplicativo son las personas autorizadas para manejar la información.

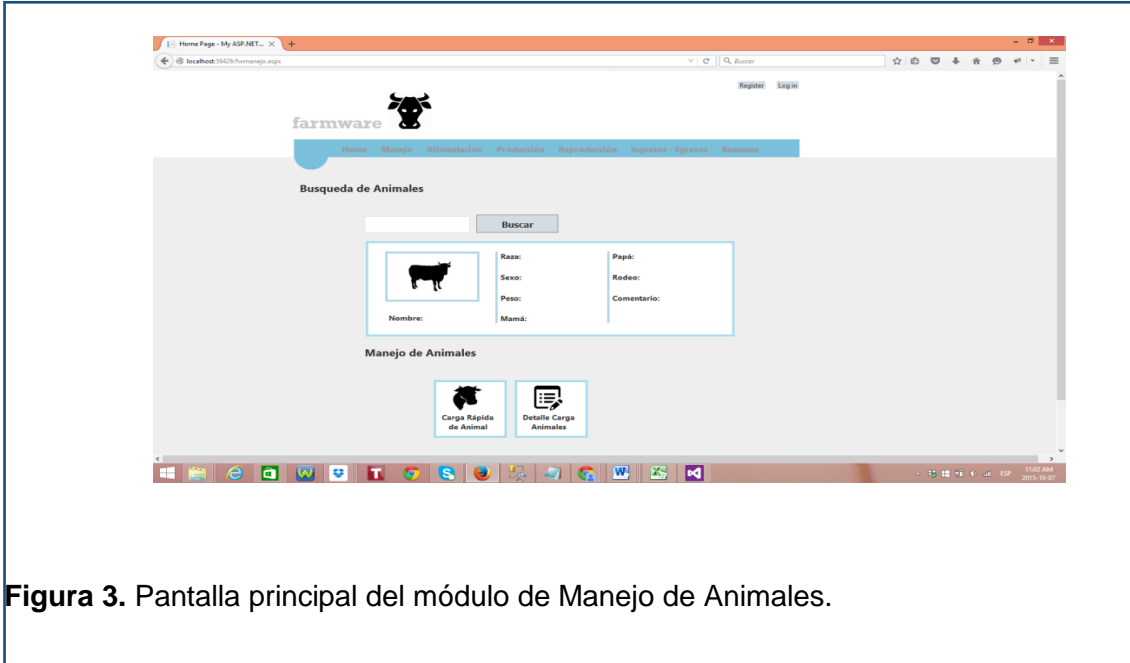
#### 5. Menú del Sistema

Una vez se inicie la sesión en el sistema automáticamente el sistema le dirige hacia la pantalla inicial donde se encuentra una barra de menú que se muestra en la Figura 3. Esta barra constará de forma permanente en el aplicativo, de tal manera que el usuario pueda acceder a cualquier otra página.



**Figura 2.** Página de inicio del aplicativo web y su Barra de Menú.

Al dar clic en el icono de manejo del animal  el usuario podrá realizar una búsqueda por nombre de los animales registrados y obtener la información referente del mismo. Adicional el administrador tiene la opción de realizar la Carga Rápida de un nuevo animal y una vista en donde se listarán los animales ingresados al aplicativo. Figura 3.



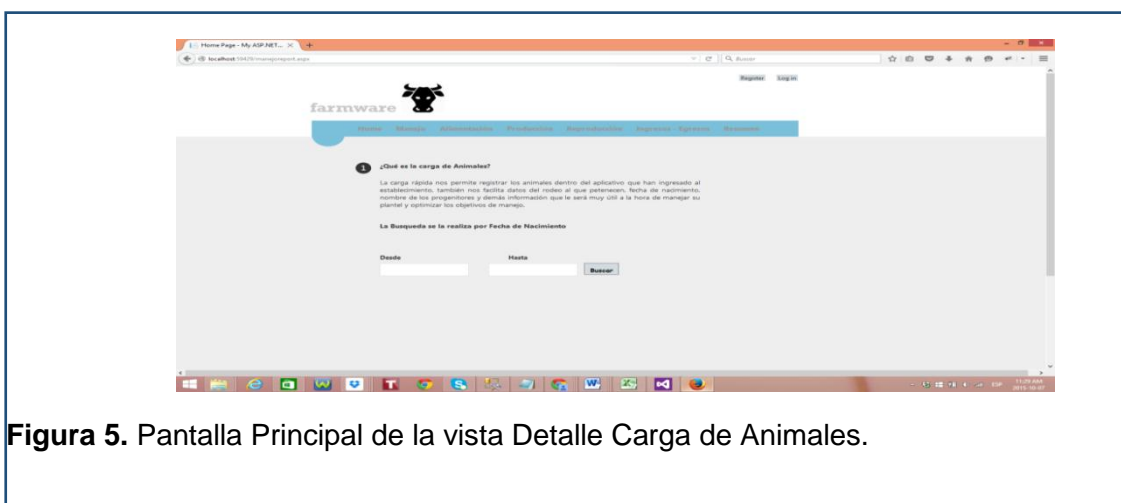
**Figura 3.** Pantalla principal del módulo de Manejo de Animales.

Para registrar todos los animales nuevos que han ingresado al establecimiento, el usuario deberá dar clic en la opción **Carga Rápida de Animal**; es de carácter obligatorio que todos los campos solicitados en el formulario sean ingresados, tal como se muestra en la Figura 4.




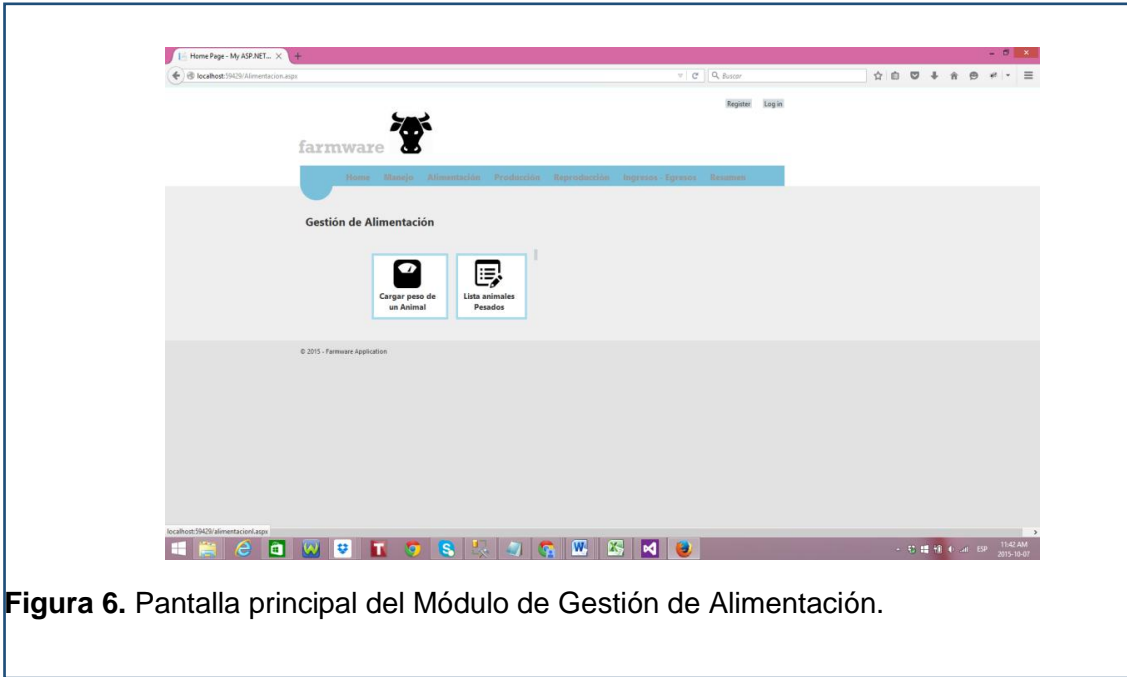
**Figura 4.** Pantalla principal de Carga Rápida de Animal.

Para observar un listado de todos los animales que han sido ingresados en el aplicativo, el administrador deberá dar clic en la opción **Detalle Carga de Animales** e ingresar el rango de fechas como parámetro de búsqueda, tal como se muestra en la Figura 5.



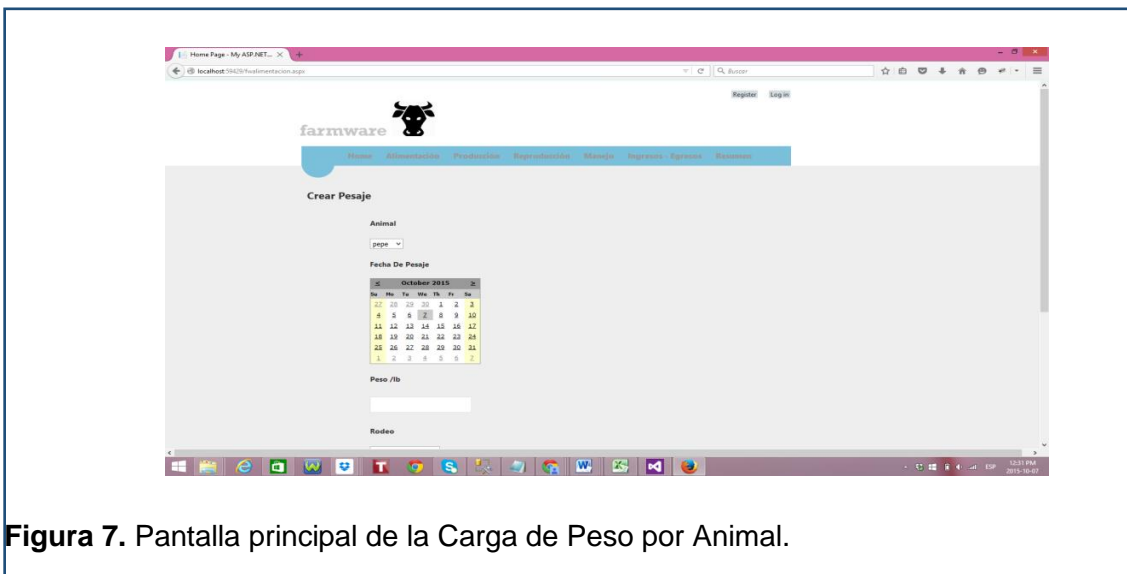
**Figura 5.** Pantalla Principal de la vista Detalle Carga de Animales.

Al dar clic en la opción del menú Gestión de Alimentación  el usuario podrá realizar una Carga de peso de cada animal y también podrá visualizar el listado de animales pesados en el enlace Lista Animales Pesados, como se muestra en la Figura 6.



**Figura 6.** Pantalla principal del Módulo de Gestión de Alimentación.

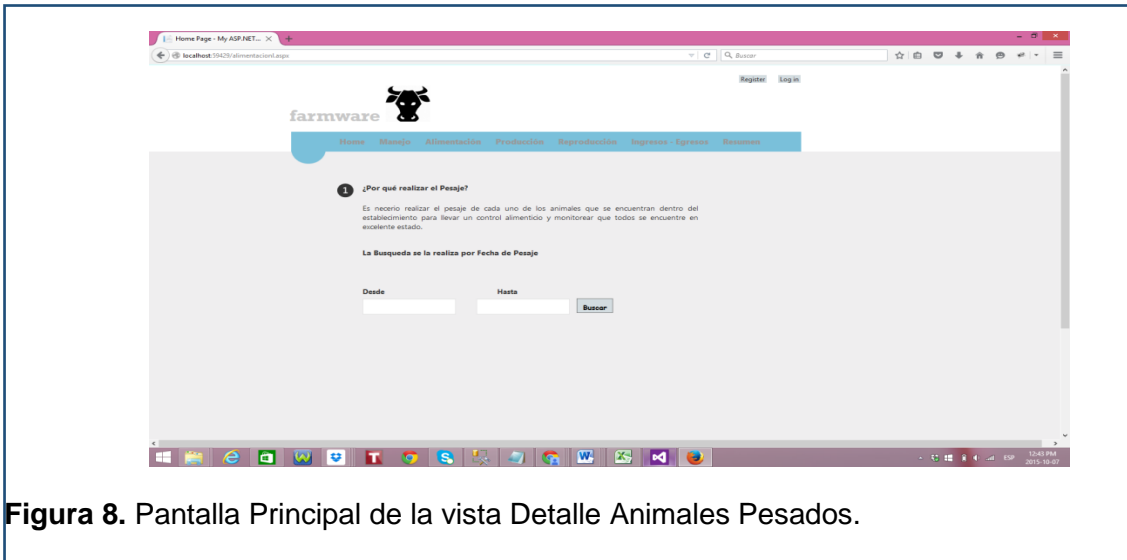
Para ingresar el pesaje de un animal del establecimiento, el usuario deberá hacer clic en la opción **Carga Peso de un Animal** y de manera obligatoria deberá llenar todos los campos solicitados por el sistema, tal como se muestra en la Figura 7.




**Figura 7.** Pantalla principal de la Carga de Peso por Animal.



Para observar un listado del pesaje de todos los animales que han sido registrados en el sistema, el administrador deberá dar clic en la opción **Lista Animales Pesados** para realizar la búsqueda es necesario ingresar el rango de fechas, tal como se muestra en la Figura 8.



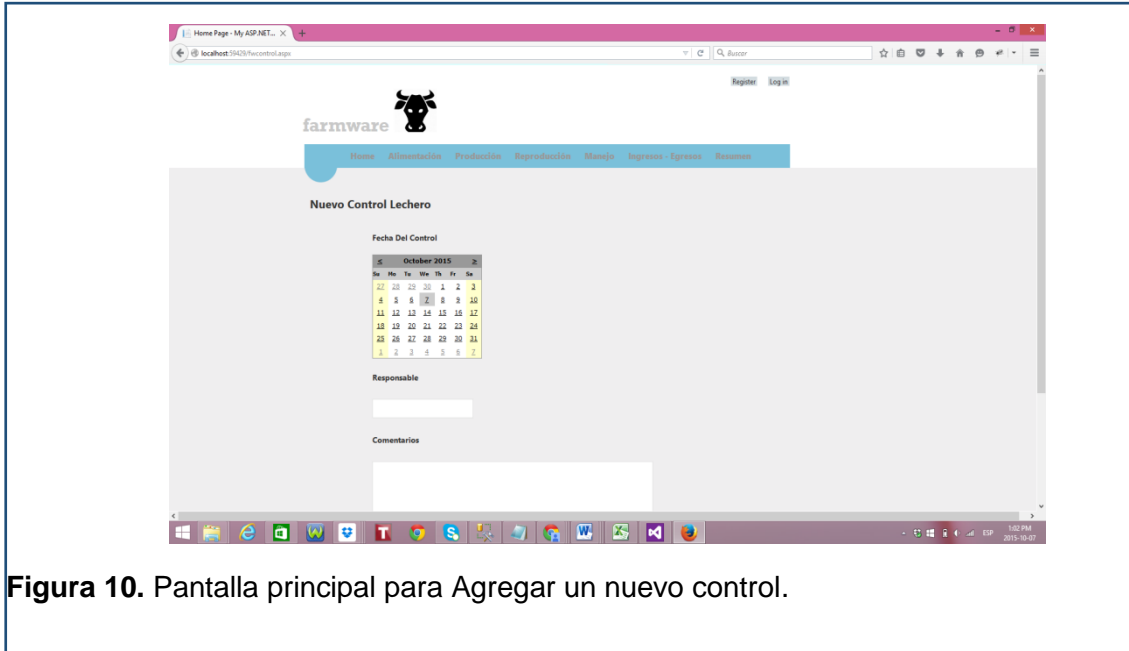
**Figura 8.** Pantalla Principal de la vista Detalle Animales Pesados.

Al dar clic en el botón del módulo Gestión de Producción  el sistema nos mostrará la siguiente pantalla Figura 9, presentándonos el link de agregar un nuevo control lechero al sistema.



**Figura 9.** Pantalla Principal del módulo de Gestión de Producción.

Para registrar un control lechero el usuario deberá dar clic en el link **Agregar un nuevo Control** y de manera obligatoria deberá ingresar todos los campos solicitados por el sistema, tal como se muestra en la Figura 10.



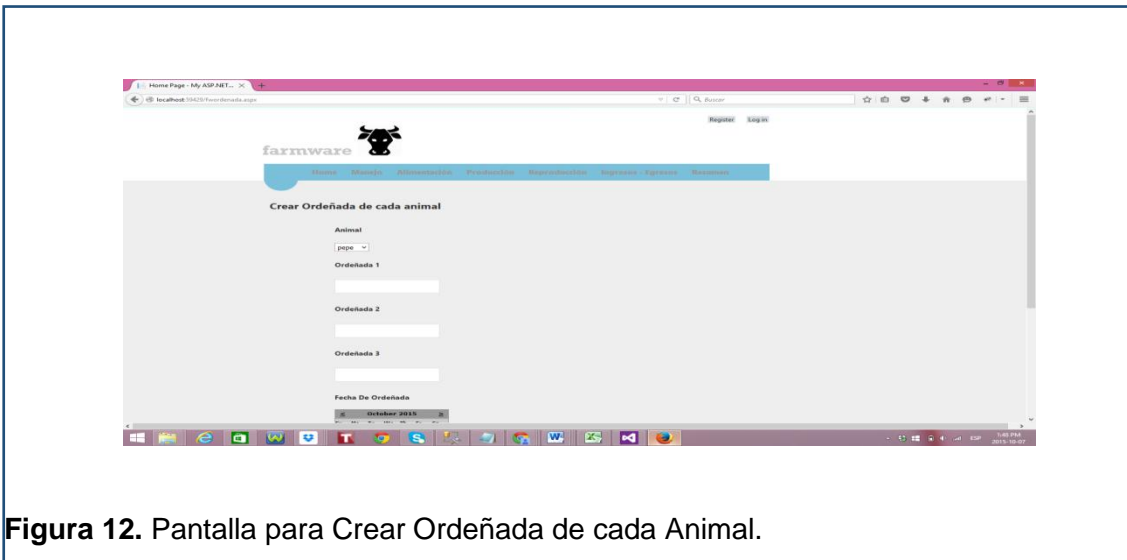
**Figura 10.** Pantalla principal para Agregar un nuevo control.

Al dar clic en el botón **Crear** se guardaran los datos ingresados, de manera automática el sistema nos envía a la página como se muestra en la figura 11. En la misma tenemos la opción de registrar el control de cada ordeñada por animal y visualizar el detalle de cada ordeñada.



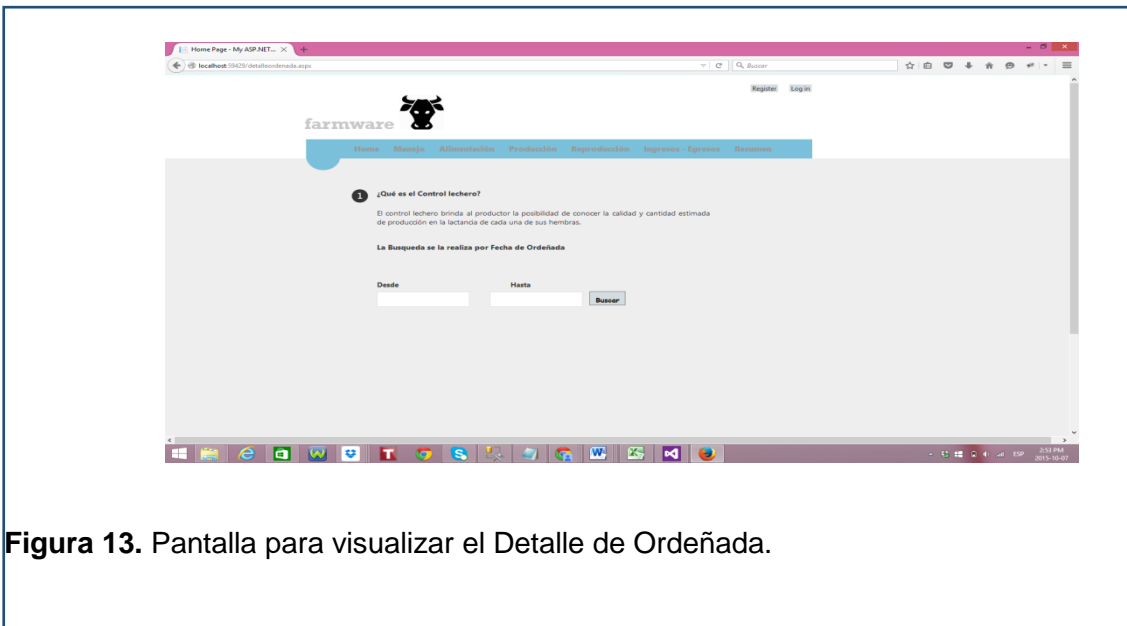
**Figura 11.** Pantalla del Módulo de Gestión de Producción.

Para registrar los litros obtenidos por ordeñada de cada animal el usuario deberá dar clic en la opción **Registrar Ordeñada por Animal**, es de carácter obligatorio llenar todos los campos que solicita el formulario, tal como se muestra la siguiente pantalla Figura 12.



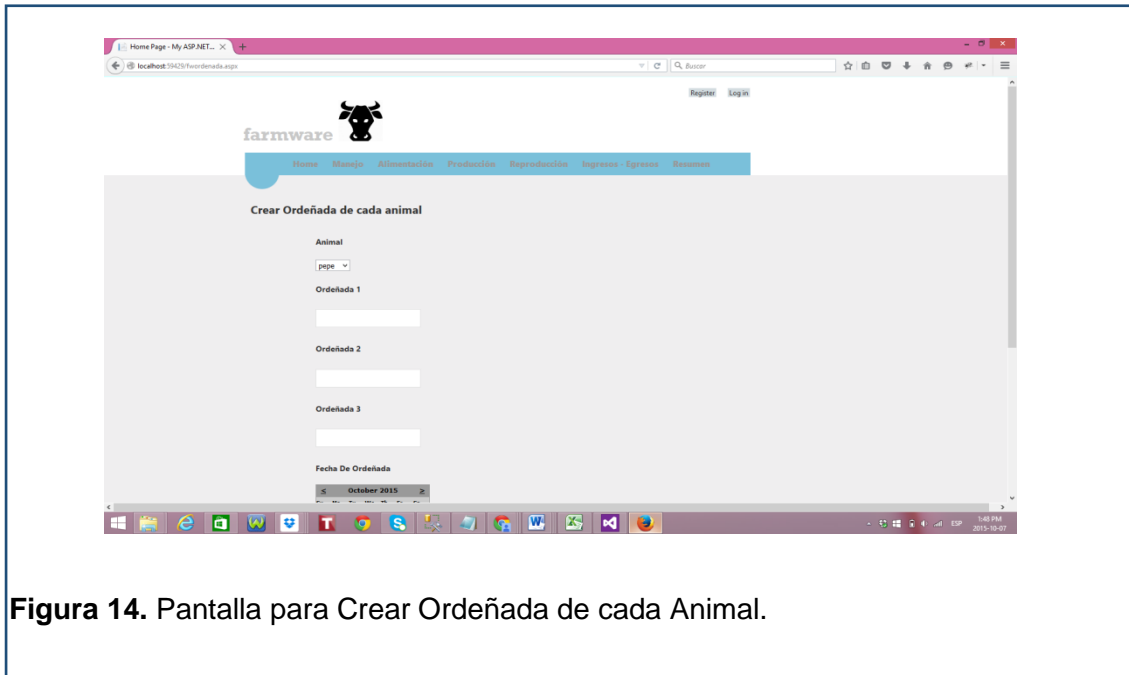
**Figura 12.** Pantalla para Crear Ordeñada de cada Animal.

Para observar un listado de todas las ordeñadas registradas en el sistema, el administrador deberá hacer clic en la opción **Detalle de Ordeñadas**; para realizar la búsqueda es necesario ingresar el rango de fechas, tal como se muestra en la Figura 13.



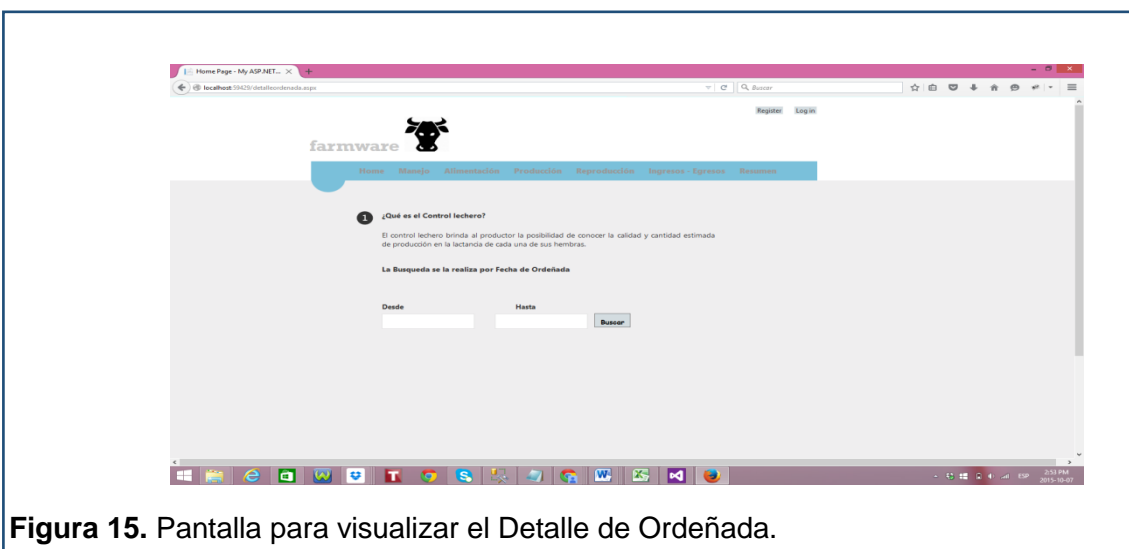
**Figura 13.** Pantalla para visualizar el Detalle de Ordeñada.

Para registrar los litros obtenidos por ordeñada de cada animal el usuario deberá dar clic en la opción **Registrar Ordeñada por Animal**, es de carácter obligatorio llenar todos los campos que solicita el formulario, tal como se muestra la siguiente pantalla Figura 14.




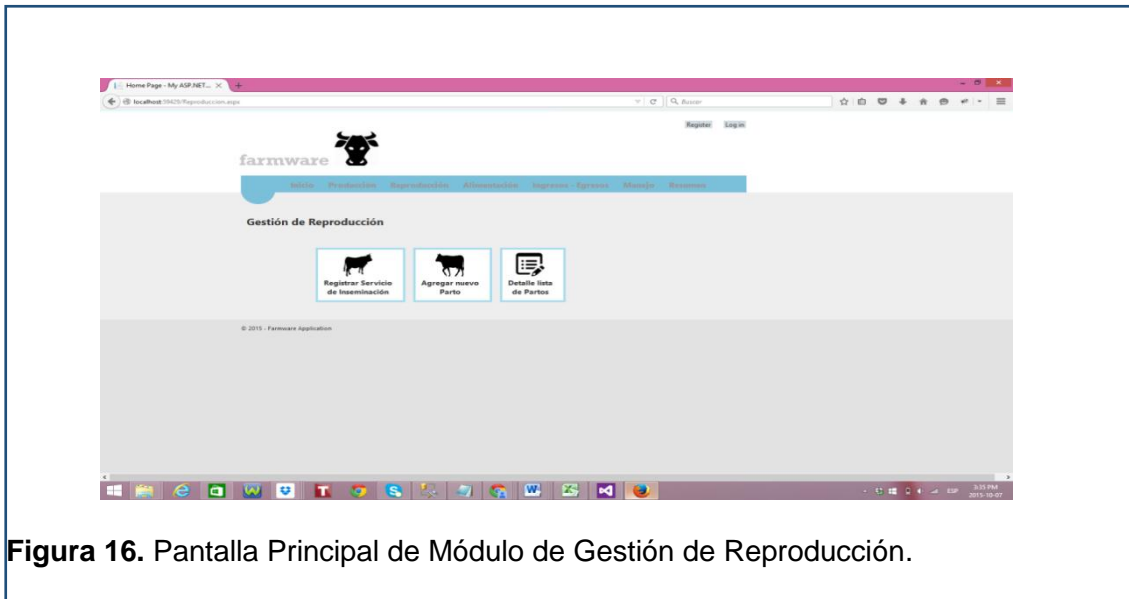
**Figura 14.** Pantalla para Crear Ordeñada de cada Animal.

Para observar un listado de todas las ordeñadas registradas en el sistema, el administrador deberá hacer clic en la opción **Detalle de Ordeñadas**; para realizar la búsqueda es necesario ingresar el rango de fechas, tal como se muestra en la Figura 15.



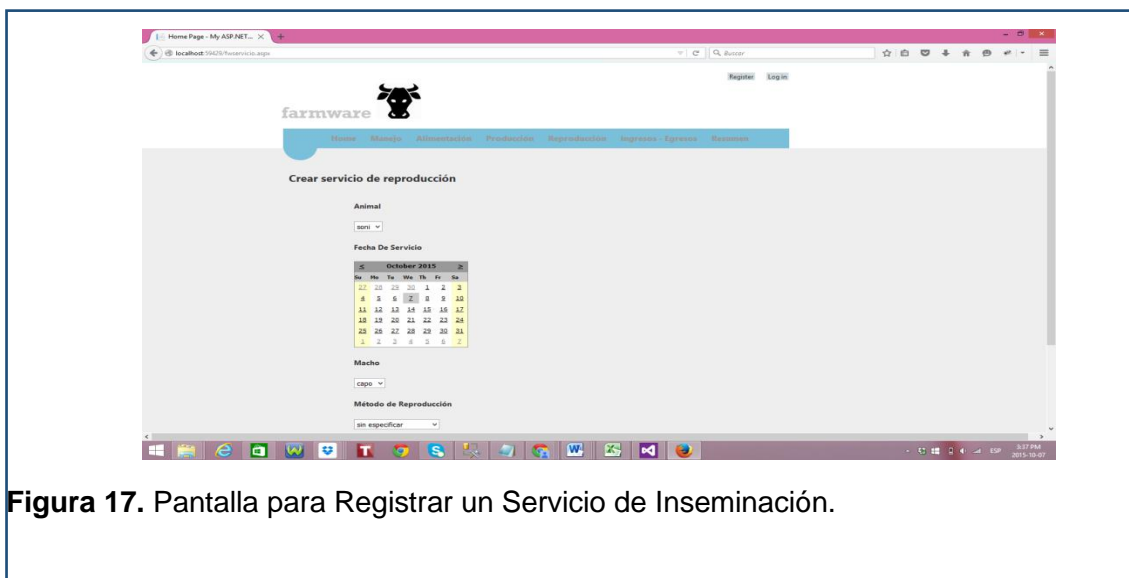
**Figura 15.** Pantalla para visualizar el Detalle de Ordeñada.

Al dar clic en el módulo de Gestión de Reproducción  el sistema mostrará la siguiente pantalla Figura 16. En esta página el usuario tendrá la opción de registrar un nuevo servicio, agregar un parto y visualizar el listado de los partos de cada una de sus hembras.



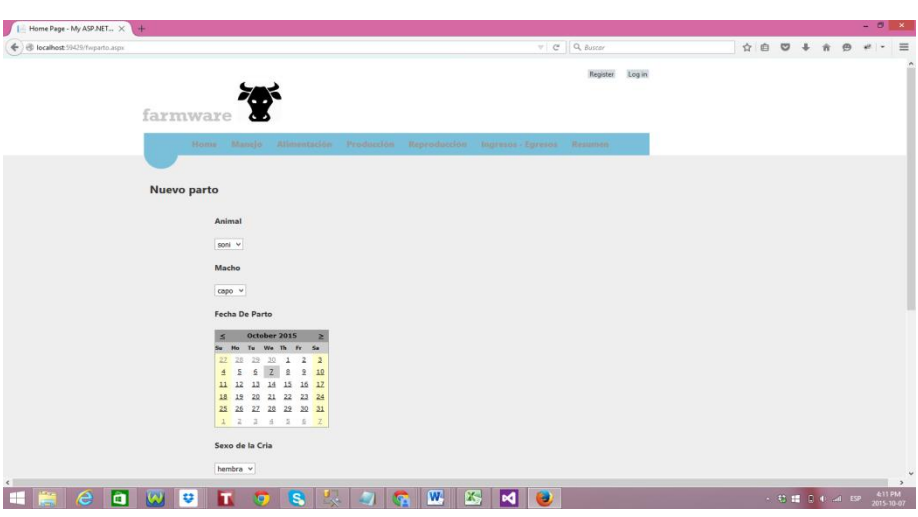
**Figura 16.** Pantalla Principal de Módulo de Gestión de Reproducción.

Para crear un servicio de reproducción el usuario deberá ingresar al enlace **Registrar Servicio de Inseminación** y digitar todos los campos solicitados en el formulario como se muestra en la Figura 17.



**Figura 17.** Pantalla para Registrar un Servicio de Inseminación.

Para agregar un nuevo parto el usuario deberá ingresar a la opción **Agregar nuevo Parto** y registrar todos los campos solicitados en la página.

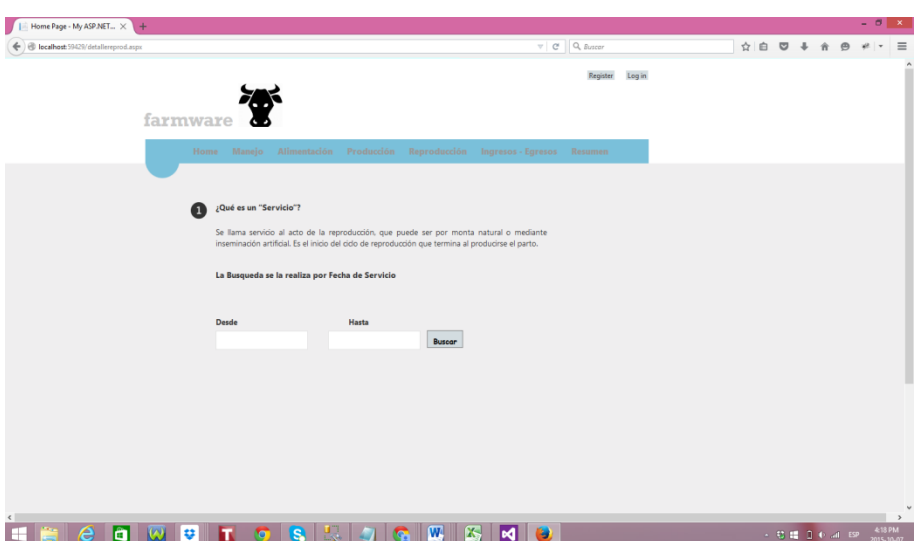


The screenshot shows a web browser window with the URL 'localhost:19429/farmware.aspx'. The page title is 'Home Page - My ASP.NET...'. The browser address bar shows 'localhost:19429/farmware.aspx'. The page features a navigation menu with items: Home, Manejo, Alimentación, Producción, Reproducción, Ingresos - Egresos, and Reservas. The main content area is titled 'Nuevo parto' and contains the following form fields:

- Animal: dropdown menu with '2001' selected.
- Macho: dropdown menu with 'cupo' selected.
- Fecha De Parto: calendar widget showing 'October 2015' with the 22nd highlighted.
- Sexo de la Cría: dropdown menu with 'hembra' selected.

Figure 18. Formulario para Registrar un nuevo parto.

Para visualizar el listado de todos los partos registrados dentro del establecimiento, el administrador deberá hacer clic en la opción **Detalle Lista de Partos** y para realizar la búsqueda es necesario ingresar el rango de fechas.




The screenshot shows a web browser window with the URL 'localhost:19429/detalleprod.aspx'. The page title is 'Home Page - My ASP.NET...'. The browser address bar shows 'localhost:19429/detalleprod.aspx'. The page features a navigation menu with items: Home, Manejo, Alimentación, Producción, Reproducción, Ingresos - Egresos, and Reservas. The main content area is titled '¿Qué es un "Servicio"?' and contains the following text and form fields:

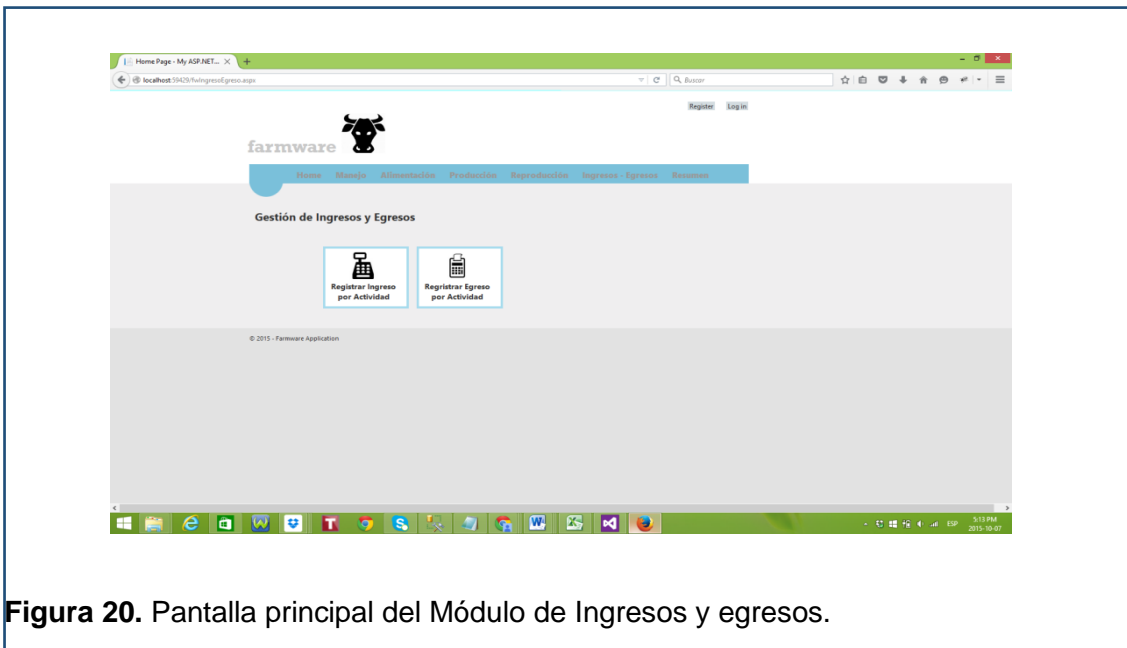
Se llama servicio al acto de la reproducción, que puede ser por monta natural o mediante inseminación artificial. Es el inicio del ciclo de reproducción que termina al producirse el parto.

La Búsqueda se realiza por Fecha de Servicio

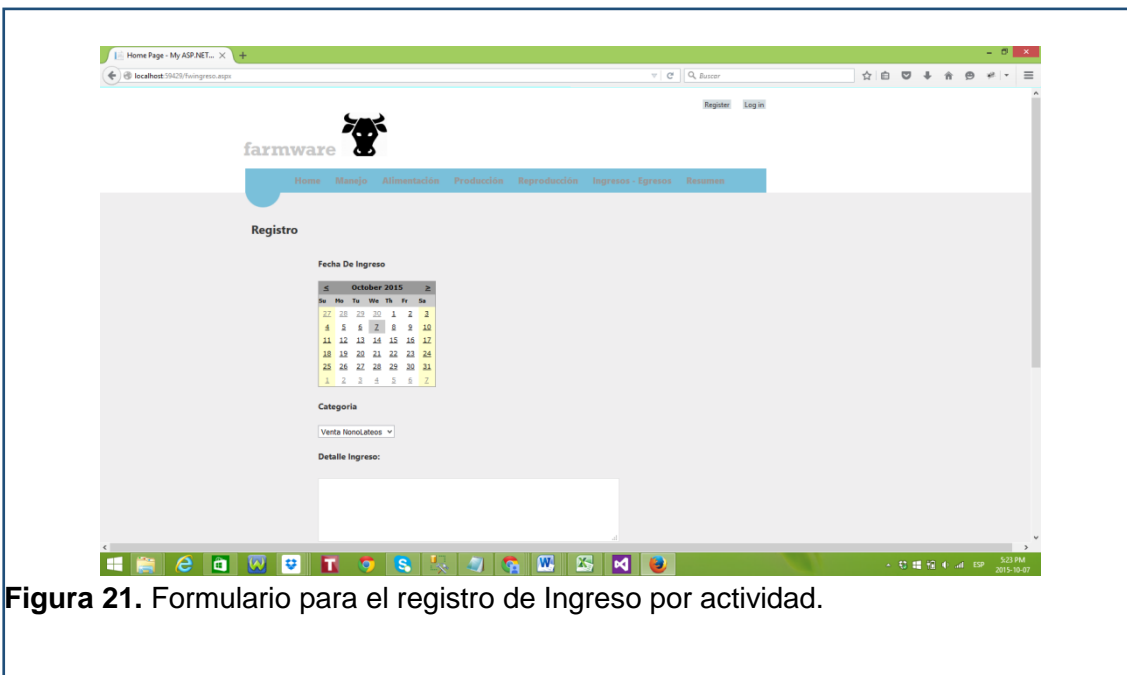
Desde:  Hasta:

Figure 19. Pantalla para visualizar el Detalle de Partos.

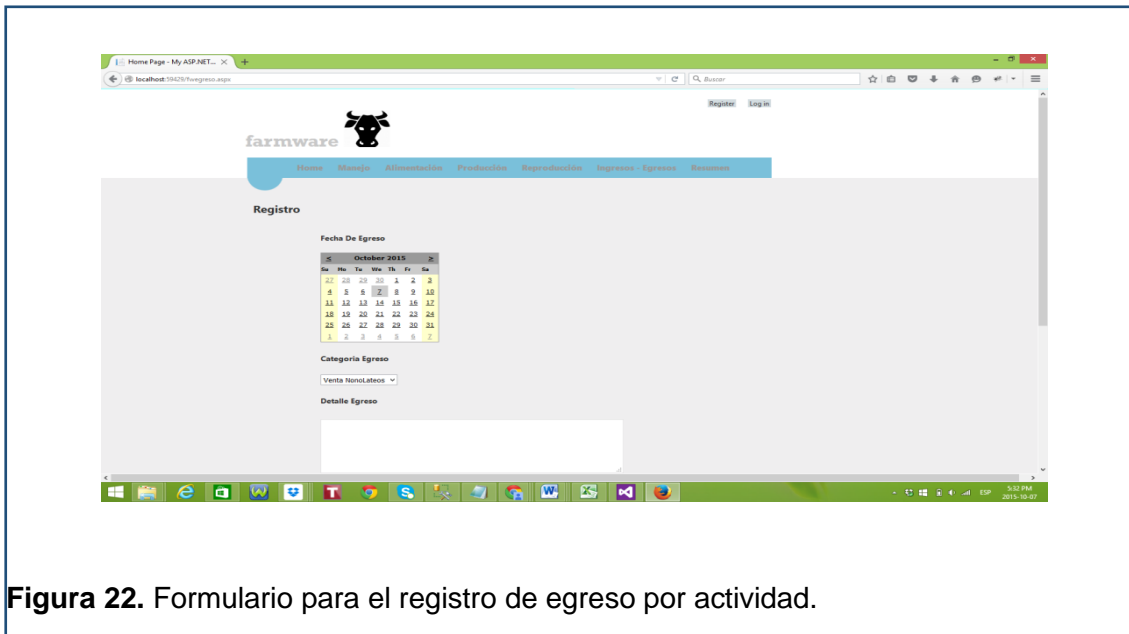
Al hacer clic en el módulo de Ingresos y egresos la  aplicación nos permitira registrar los ingresos y egresos obtenidos por cada actividad.




Para registrar los ingresos en el sistema el usuario debe dar clic en el botón de **Registrar Ingreso por actividad** como se muestra en la Figura 21.



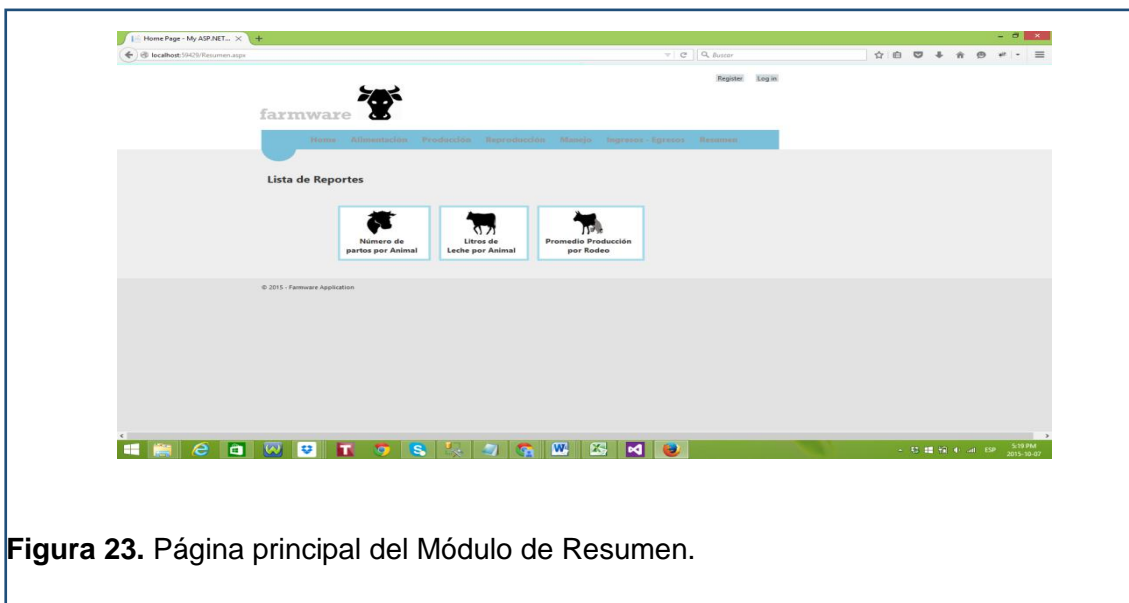
Para registrar los egresos dentro del aplicativo el usuario debe hacer clic en el botón de **Registrar egreso por actividad** como se muestra en la Figura 22.



**Figura 22.** Formulario para el registro de egreso por actividad.

Finalmente el aplicativo tendrá el Módulo de  Resumen el cual nos permitirá visualizar los reportes que se han parametrizado en el Sistema, tal como se muestra en la Figura 23; estos son:

- ✓ Números de partos por animal.
- ✓ Litros de leche por animal.
- ✓ Promedio Producción por Rodeo.



**Figura 23.** Página principal del Módulo de Resumen.



