



**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ**

**CARRERA INFORMÁTICA**

**TESIS PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERA  
EN INFORMÁTICA**

**TEMA:**

**SOFTWARE DE CONTROL GANADERO EN EL HATO BOVINO  
DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA  
DE MANABÍ MANUEL FÉLIX LÓPEZ**

**AUTORAS:**

**MURILLO LOOR GEMA EMPERATRIZ  
ZAMBRANO ROSADO MARÍA XIMENA**

**TUTOR**

**ING. JORGE ANTONIO PÁRRAGA ÁLAVA, MGS.**

**CALCETA, JULIO 2013**

## **DERECHO DE AUTORÍA**

Gema Emperatriz Murillo Loor y María Ximena Zambrano Rosado, declaran bajo juramento que el trabajo aquí descrito es de su autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y que se han consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos los derechos de propiedad intelectual a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Feliz López, según lo establecido por la Ley de Propiedad Intelectual y su reglamento.

.....  
**GEMA E. MURILLO LOOR**

.....  
**MARÍA X. ZAMBRANO ROSADO**

## CERTIFICACIÓN

Jorge Antonio Párraga Álava certifica haber tutelado la tesis **SOFTWARE DE CONTROL GANADERO EN EL HATO BOVINO DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ**, que ha sido desarrollado por Murillo Loor Gema Emperatriz y Zambrano Rosado María Ximena, previo a la obtención del título de Ingeniera en Informática, de acuerdo al **REGLAMENTO PARA LA ELABORACIÓN DE TESIS DE GRADO DE TERCER NIVEL** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....  
**ING. JORGE A. PÁRRAGA ÁLAVA, MGS.**

## **APROBACIÓN**

Los suscritos integrantes del tribunal correspondiente, declaran que han **APROBADO** la tesis **SOFTWARE DE CONTROL GANADERO EN EL HATO BOVINO DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ**, que ha sido propuesta, desarrollada y sustentada por Murillo Loor Gema Emperatriz y Zambrano Rosado María Ximena, previa la obtención del título de Ingeniera en Informática, de acuerdo al Reglamento **PARA LA ELABORACIÓN DE TESIS DE GRADO DEL TERCER NIVEL** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....  
**ING. ÁNGEL A. VÉLEZ MERO, Mg.**  
**MIEMBRO**

.....  
**LIC. ISABEL MATILLA BLANCO, Mg.**  
**MIEMBRO**

.....  
**ING. JESSICA J. MORALES CARRILLO Mg. Sc.**  
**PRESIDENTE**

## **AGRADECIMIENTO**

El agradecimiento es la gratitud se ejerce cuando una persona experimenta aprecio y reconocimiento por otra que le ha prestado su ayuda. Dios, tú mi única verdad ser supremo que ha permitido alcanzar muchos objetivos en la vida, uno de ellos es ver culminado los estudios de tercer nivel.

A nuestros padres fuentes inspiradas de superación y esfuerzo diario, su apoyo incondicional en lo moral como lo económico han sido el empuje para lograr todos nuestros anhelos. Así como también muestra familia y amigos que de una u otra manera colaboran con la superación en nuestros estudios.

A la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, institución que me acogió y a quien le debemos ser unas profesionales. A los docentes que nos impartieron sus conocimientos en las aulas de clase formando un profesional con capacidad de servicio en el mundo laborar.

**LAS AUTORAS**

## **DEDICATORIA**

Este trabajo de investigación está dedicado en especial a Dios, quien me llena de fortaleza ante cualquier adversidad. A mis padres Ángel Modesto Murillo Cevallos y Nelly Emperatriz Loor Zambrano, por su apoyo incondicional en todas las etapas de mi vida y que sabiamente me han sabido guiar por el camino del bien. Así mismo dedico este logro a mis hermanos María Dolores y Ángel Carlos, y a mi novio Fernando, quienes han sido mi ejemplo de superación y me ayudaron desinteresadamente.

A la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, que me dio la oportunidad de capacitarme y a todos los docentes que compartieron sus conocimientos para poder culminar el presente trabajo de tesis y por ende alcanzar una de mis más preciadas metas que es ser una profesional.

Murillo Loor Gema Emperatriz

## **DEDICATORIA**

Este trabajo de investigación está dedicado a todas las personas que han contribuido al desarrollo del mismo, en especial a Dios que es la fortaleza de vida; a seguir adelante cumpliendo sueños y siendo feliz con cada logro.

Detrás de cada logro existe el apoyo, confianza y cariño de la familia que une lazos de amor compartidos a lo largo de la vida; tomando los problemas como parte de su vida para luego compartir su sabiduría con consejos productivos que sirvieron de mucha ayuda para culminar con éxito la tesis. Así mismo agradecer a mis padres que contribuyeron con enseñanzas que hicieron de mí un mejor ser humano, dejando como herencia los estudios que son fuente fundamental para un mejor bienestar. A la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, que es una institución presta a servir a cada estudiante que ingrese a ser parte de ella, así también como su entorno.

Zambrano Rosado María Ximena

## CONTENIDO GENERAL

CARÁTULA: .....	I
DERECHO DE AUTORÍA.....	II
CERTIFICACIÓN .....	III
APROBACIÓN .....	IV
AGRADECIMIENTO.....	V
DEDICATORIA.....	VI
DEDICATORIA.....	VII
CONTENIDO GENERAL.....	VIII
CONTENIDO DE CUADROS Y FIGURAS.....	X
RESUMEN .....	XII
PALABRAS CLAVES .....	XII
ABSTRACT .....	XIII
CAPÍTULO I. ANTECEDENTES .....	1
1.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA.....	1
1.2. JUSTIFICACIÓN .....	2
1.3. OBJETIVOS .....	3
1.3.1. OBJETIVO GENERAL.....	3
1.3.2. OBJETIVOS ESPECÍFICOS .....	3
1.4. IDEAS A DEFENDER .....	3
CAPÍTULO II. MARCO TEÓRICO.....	4
2.1. LA GANADERÍA Y SU IMPORTANCIA EN EL MUNDO .....	4
2.1.1. ECUADOR EN SU ACTIVIDAD GANADERA .....	6
2.1.2. MANEJO REPRODUCTIVO DEL HATO GANADERO .....	8
2.2. FUNDAMENTOS DE INGENIERÍA DE SOFTWARE.....	10
2.2.1. DEFINICIÓN .....	10
2.2.2. IMPORTANCIA .....	10
2.2.3. TÉCNICAS DE PROGRAMACIÓN .....	11



2.2.4.	LENGUAJES DE PROGRAMACIÓN .....	14
2.2.5.	DIAGRAMAS UML (UNIFIED MODELING LANGUAGE).....	17
2.2.6.	PRUEBAS DE SOFTWARE .....	21
2.2.7.	METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	27
2.3.	GENERALIDADES DE GESTORES DE BASES DE DATOS .....	36
2.3.1.	DEFINICIÓN .....	36
2.3.2.	IMPORTANCIA .....	36
	CAPÍTULO III. DESARROLLO METODOLÓGICO .....	40
3.1.	MÉTODOS CIENTÍFICOS.....	40
3.2.	MÉTODO INFORMÁTICO.....	40
3.2.1.	FASE I: PLANEACIÓN.....	40
3.2.2.	FASE II: DISEÑO .....	51
3.2.2.1.	INTERFAZ DEL SOFTWARE DE CONTROL GANADERO .....	60
3.2.3.	FASE III: DESARROLLO DEL CÓDIGO .....	60
3.2.4.	FASE IV: PRUEBAS .....	61
	CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....	64
4.1.	RESULTADOS .....	64
4.2.	DISCUSIÓN.....	71
	CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES .....	73
5.1.	CONCLUSIONES.....	73
5.2.	RECOMENDACIONES .....	74
	BIBLIOGRAFÍA .....	75
	ANEXOS .....	80

## CONTENIDO DE CUADROS Y FIGURAS

Figura 2.1. Proceso XP .....	30
Fuente: Orjuela y Rojas (2008) .....	30
Figura 2.2. Fases de la Metodología XP. ....	31
Fuente: .....	31
Figura 3.1. Procesos del ciclo vital ganadero en el hato bovino de la ESPAM MFL.....	42
Cuadro 3.1. Historia del registro de animales (Iteración1). ....	43
Cuadro 3.2. Historia del registro animal (Iteración 2). ....	43
Cuadro 3.3. Historia del registro animal (Iteración 3). ....	44
Cuadro 3.4. Historia del registro de eventos (Iteración 1). ....	44
Cuadro 3.5. Historia del registro de eventos (Iteración 2). ....	45
Cuadro 3.6. Historia del registro de eventos (Iteración 3). ....	45
Cuadro 3.7. Historia del registro de producción (Iteración 1). ....	46
Cuadro 3.8. Historia del registro de producción (Iteración 2). ....	46
Cuadro 3.9. Historia del registro de producción (Iteración 3). ....	47
Cuadro 3.10. Historia del registro de reproducción (Iteración 1). ....	48
Cuadro 3.11. Historia del registro de reproducción (Iteración 2). ....	48
Cuadro 3.12. Historia del registro de reproducción (Iteración 3). ....	49
Cuadro 3.13. Historia de reporte (Iteración 1). ....	49
Cuadro 3.14. Historia de reporte (Iteración 2). ....	50
Cuadro 3.15. Historia de reporte (Iteración 3). ....	51
Cuadro 3.16. Diccionario de datos de la tabla "Animal". ....	52
Cuadro 3.17. Diccionario de datos de la tabla "Evento". ....	52
Cuadro 3.18. Diccionario de datos de la tabla "Producción". ....	53
Cuadro 3.19. Diccionario de datos de la tabla "Reproducción". ....	53
Cuadro 3.20. Diccionario de datos de la tabla "HistorialClinico". ....	54
Cuadro 3.21. Diccionario de datos de la tabla "Suplementacion". ....	54
Cuadro 3.22. Diccionario de datos de la tabla "Venta". ....	55
Cuadro 3.23. Diccionario de datos de la tabla "Enfermedad". ....	55
Cuadro 3.24. Diccionario de datos de la tabla "Farmacos". ....	56
Cuadro 3.25. Diccionario de datos de la tabla "Tipo_evento". ....	56

Cuadro 3.26. Diccionario de datos de la tabla “Tipo_fecundacion”	56
Cuadro 3.27. Diccionario de datos de la tabla “Estado”	57
Cuadro 3.28. Diccionario de datos de la tabla “Repro_animal”	57
Cuadro 3.29. Diccionario de datos de la tabla “Suple_animal”	57
Cuadro 3.30. Diccionario de datos de la tabla “Historial_animal”	58
Cuadro 3.31. Diccionario de datos de la tabla “Evento_animal”	58
Cuadro 3.32. Diccionario de datos de la tabla “Usuario”	58
Cuadro 3.33. Diccionario de datos de la tabla “Ingreso_sistema”	59
Cuadro 3.34. Diccionario de datos de la tabla “Tipo_parto”	59
Cuadro 3.35. Diccionario de datos de la tabla “Raza”	59
Cuadro 3.36. Diccionario de datos de la tabla “Sexo”	60
Figura 4.1. Base de datos del Software de control ganadero del Hato Bovino de la ESPAM MFL	65
Figura 4.2. Ventana de inicio de sesión	66
Figura 4.3. Ventana del menú principal	66
Figura 4.4. Ventana de ingreso y modificación de datos principales de los bovinos.	67
Figura 4.5. Ventana de consulta de datos de los bovinos.	67
Figura 4.6. Ventana de reporte de datos de producción de los bovinos.	68
Cuadro 4.1. Pruebas de tiempos comparativos con respecto al uso manual y del sistema.	69
Grafico 4.1. Estadística del tiempo invertido en el registro de procesos de los bovinos.	70

## **RESUMEN**

El objetivo primordial de esta investigación fue crear un sistema informático que permita gestionar y controlar el registro de bovinos en la ESPAM MFL, con el firme propósito de agilizar los procedimientos y así facilitar las actividades en el hato bovino de la ESPAM. El modelo de desarrollo que se utilizó fue el modelo XP (extreme programming) a través de sus cuatro fases como: planificación, diseño, desarrollo del código y prueba, el cual permitió la recopilación de información para definir los requisitos y la arquitectura del sistema. Así mismo se elaboraron las historias de usuario y diagramas de flujos de datos que ayudaron a describir las funcionalidades del sistema. Luego se procedió a la elaboración de la base de datos en Microsoft SQL Server 2008 R2 que se acopló al trato y manipulación de la información. Seguido a esto se creó la interfaz gráfica del sistema y se incorporó la lógica de la aplicación mediante la codificación, para el efecto se utilizó la plataforma de desarrollo Microsoft Visual Basic 2010. Una vez validada la aplicación, se comprobó la optimización del 43% en el ingreso de datos de los bovinos, 16% en el ingreso de producción lechera, 85% en consultas de las actividades ganaderas y el 55% en generar reportes, obteniendo como ventaja el uso del sistema ganadero INFOHATO de la ESPAM MFL.

## **PALABRAS CLAVES**

Hato bovino, herramienta de programación, producción lechera, historial veterinario, insumos pecuarios.

## **ABSTRACT**

The primary objective of this investigation was to create a computerized system that allows manage and control the registration of bovines in the ESPAM MFL, with the firm purpose of speeding up the procedures and this way to facilitate the activities in the ESPAM cattle herd. The development model used was the model XP (extreme programming) through its four phases as planning, design, development and testing code, which allowed the collection of information to define the requirements and system architecture. Likewise were developed history user and data flow diagrams that helped describe the system's functionality. Then he proceeded to the development of the database in Microsoft SQL Server 2008 R2 which was coupled to the treatment and handling of information. Following this GUI is created and incorporated system of the application logic by encoding, for the effect is the development platform used Microsoft Visual Basic 2010. Once validated the implementation, optimization was found 43% in data entry of cattle, 16% in revenue from milk production, 85% in inquiries from livestock and 55% in generating reports, with the advantage the use of INFOHATO system of ESPAM MFL.

## **KEYWORDS**

Bovine Hato, programming tool, milk production, veterinary record, livestock supplies.

# **CAPÍTULO I. ANTECEDENTES**

## **1.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA**

La ganadería bovina representa un sector de mucha importancia para el país, ya que contribuye a la fertilidad de los campos, por esta razón el hato se extendió rápidamente, creando una gran fuente de riqueza a través de los parámetros productivos y reproductivos que pueden incrementarse mediante la introducción de cambios tecnológicos.

El hato bovino de la ESPAM MFL se dedica principalmente a la cría y levante de terneros, las vacas en producción se ordeñan diariamente y la leche obtenida pasa a ser parte de la planta de lácteos de la misma institución. Además, colabora con la comunidad, al brindar a los estudiantes de pecuaria un espacio para realizar prácticas veterinarias, que le permitan desarrollar destrezas y habilidades.

Para un manejo adecuado del ganado es conveniente llevar el registro minucioso de información relevante en cuanto a procesos de inseminación, partos, gestación, tratamiento de enfermedades, cantidad de animales disponibles, entre otros; todo de forma eficiente a través de herramientas que agilicen tal situación, sin embargo se ha observado que en el hato bovino estos son registrados manualmente, trayendo como consecuencia la pérdida de información y con ello el desconocimiento de aspectos importantes del ciclo vital ganadero. Con los antecedentes expuestos las autoras de la presente tesis se plantean la siguiente interrogante:

¿De qué manera optimizar el manejo ganadero en el hato bovino de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López?

## 1.2. JUSTIFICACIÓN

En toda explotación pecuaria el objetivo central es lograr las máximas utilidades y estabilidad de la empresa en el tiempo. Para ello, es necesario que se realice una buena administración de los recursos y se desarrolle social y económicamente a sus integrantes. Para un manejo adecuado del ganado es necesario llevar un control de todos los registros de cada bovino, de cada tratamiento realizado en determinada fecha.

La Ley Orgánica del Régimen de la Soberanía Alimentaria (LORSA, 2010) publicada en el Registro Oficial No. 583 del 5 de Mayo de 2009, entre los Derechos de sanidad e inocuidad alimentaria establece en el Art. 25 que: “el Estado prevendrá y controlará la introducción y ocurrencia de enfermedades de animales y vegetales; así mismo promoverá prácticas y tecnologías de producción, industrialización, conservación y comercialización que permitan alcanzar y afianzar la inocuidad de los productos.” Así mismo el Art. 8 de la Ley de Educación Superior (LOES, 2006), literal f indica: “fomentar y ejecutar programas de investigación de carácter científico, tecnológico y pedagógico que coadyuven al mejoramiento y protección del ambiente y promuevan el desarrollo sustentable nacional”; razones por la cual las autoras elaboraron el trabajo de investigación titulado SOFTWARE DE CONTROL GANADERO EN EL HATO BOVINO DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ.

Con la implementación del software de control ganadero se mejoró el registro de información y la gestión económica de los vacunos en el hato bovino de la ESPAM MFL, lo que benefició al personal encargado, estudiantes, profesores y comunidad en general, gracias a la entrega veraz y oportuna de información sobre la etapa de la cadena productiva del ciclo ganadero; evitando, entre otras situaciones, la dosificación excesiva de fármacos y por ende la disminución de químicos utilizados en productos lácteos para el consumo humano, lo que repercute en la rebaja paulatina de gastos por concepto de compra de medicamentos para los bovinos.

### **1.3. OBJETIVOS**

#### **1.3.1. OBJETIVO GENERAL**

Crear un software de control ganadero en el hato bovino de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López para optimizar los registros de producción y reproducción.

#### **1.3.2. OBJETIVOS ESPECÍFICOS**

- Recopilar datos del proceso de control ganadero que se lleva a cabo en el hato bovino de la ESPAM MFL.
- Diseñar una base de datos que se acople a los requerimientos y manipulación de información.
- Codificar el software con sus respectivas funcionalidades y según requerimientos establecidos.
- Realizar pruebas para verificar el correcto funcionamiento de la aplicación desarrollada.
- Implementar el software de gestión y control ganadero en hato bovino de la ESPAM MFL.

### **1.4. IDEAS A DEFENDER**

- La creación del software de control en el hato bovino de la ESPAM MFL coadyuvará al proceso de automatización tecnológica que se lleva a cabo en la institución.
- El software desarrollado permitirá llevar un mejor manejo de los registros de producción y reproducción del ganado en el hato bovino de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.



## **CAPÍTULO II. MARCO TEÓRICO**

### **2.1. LA GANADERÍA Y SU IMPORTANCIA EN EL MUNDO**

La ganadería ha permitido a muchas personas desarrollarse económicamente en algunos países del mundo, generando empleo y alimentos nutritivos para la población, tanto en su materia prima, como productos elaborados. En los últimos años, se ha incrementado el interés por la aplicación de sistemas de producción animal sostenibles, de los que se espera proporcionen una relación equilibrada entre factores medioambientales, socioculturales y económicos (Espinoza *et al.*, 2007).

La tecnología se ha vuelto muy importante en todas las actividades que realiza el hombre, ya que facilita sus tareas y le permite adquirir conocimientos. Un sistema de control ganadero le permite conocer y analizar de manera práctica toda una serie de indicadores individuales y poblacionales, esto constituye una excelente herramienta de gestión, selección y mejoramiento animal, con la que su negocio ganadero debe contar para mejorar e incrementar productividad y rentabilidad (Miranda, 2009). Así mismo es importante mejorar el sistema de identificación de los animales para conocer su procedencia, y hacer un seguimiento para investigar las posibles enfermedades (REDEVET, 2006).

Serralvo y Acevedo (s.f.) manifiesta que los métodos de investigación basados en la producción animal han supuesto una mejora en la calidad de vida y el bienestar de la sociedad ocasionados por la abundancia y la diversificación de la alimentación de los individuos, los factores que contribuyen a la subsistencia y el progreso de las sociedades es el que se refiere al desarrollo de los métodos de cría, engorde y reproducción de animales para obtener un provecho de ellos. Peláez (s.f.) considera que la investigación en ganadería busca la optimización de recursos para lograr producir el mayor número de productos ganaderos y sus derivados teniendo en cuenta entre otras cuestiones, temas como el desarrollo tecnológico.

En el último siglo, el sector ganadero ha experimentado un notable avance gracias al progreso tecnológico que ha facilitado el trabajo a los ganaderos. Las

decisiones son mejor tomadas cuando se tiene acceso a suficiente información. Y en las actividades agropecuarias es necesaria mucha información para tomar las mejores decisiones y así anticipar la resolución de problemas o el aprovechamiento de oportunidades (Serralvo y Acevedo, s.f.).

Clemens (2007) citado por Valdez (2011). En la actualidad es cada vez mayor la sensibilización del consumidor respecto a la seguridad alimentaria, lo cual ha provocado un mayor nivel de exigencia y, por tanto, un desplazamiento de la importancia de los agentes que intervienen en la cadena hacia la mesa. Por ello es muy importante que los administradores de ranchos, haciendas, fincas o hatos estén informados de cada proceso que se lleva con el ganado, como la alimentación, el periodo de gestación y otros, para prevenir enfermedades y no arriesgar la salud de todos los consumidores.

Avilez *et al.*, (2010) expresa que los sistemas de producción de leche se caracterizan por su alta complejidad y su influencia por el medio ambiente. García y Aguilar (2004) opina que los sistemas de informática y cómputo en la administración y manejo del hato, han permitido incrementar también los rendimientos por animal.

Hernández *et al.*, (2011) menciona que la cadena productiva de la ganadería bovina de carne muestra una limitada vinculación entre los diferentes eslabones (productores, engordadores, procesadores y comercializadores), ya que éstos generalmente han trabajado de manera independiente, lo que ha impedido que operen de manera integrada que contribuya a aumentar su competitividad.

La preocupación central del ganadero, como cualquier otro empresario, es cómo lograr un mayor ingreso económico, por lo que la tecnología que se desarrolle debe proveer valor agregado a las actividades del productor. Con la utilización de un sistema automatizado se incrementa la confiabilidad de los resultados, se humaniza el trabajo y disminuye el tiempo de procesamiento de la información (Lora *et al.*, 2011).

La gerencia de fincas es una actividad compleja que requiere grandes capitales para operar, además de estar expuesta a los cambios en las políticas agrícolas y constantes innovaciones tecnológicas, así como la variabilidad de condiciones ambientales que enmarcan el proceso productivo de estos sistemas, exige la presencia de gerentes capaces de adaptarse a todas esas fuentes de cambios y tomar decisiones de manera rápida y oportuna que les permita alcanzar el propósito de producción, rentabilidad y crecimiento del negocio (Silva, *et al.*, 2010).

### **2.1.1. ECUADOR EN SU ACTIVIDAD GANADERA**

La ganadería en el Ecuador es una actividad familiar de varias décadas, la misma que ha venido arrastrándose a través del tiempo de generación en generación, hasta nuestro días. Con el pasar del tiempo se ha venido tecnificándolo que es producción ya que setiene mayor acceso a tecnología e insumos de calidad. Lo cual hace que se incremente la eficiente de la producción.

La ganadería en el territorio nacional, básicamente está basada en la producción de carne y leche, desglosando se obtiene que en las regiones de costa y oriente, se cría en mayor porcentaje el ganado de carne y en la regiónsierra se enfoca más la producción lechera. El ganado lechero necesita más mano de obra por animal, y se ve afectado mayormente por el nivel de manejo que reciben que cualquier otro tipo de animales domésticos, es decir necesitan un elevado nivel de atención y cuidado, lo cual influye en el precio final de la leche.

La producción de leche por la calidad de sus productos, presenta niveles de competitividad en los mercados regionales del país, sin embargo, se ve afectada por los altos costos de producción debido a los bajos niveles de productividad de las fincas y la falta de incorporación de mejoras tecnológicas de manera integral, asociado a los procesos de industrialización que restringuen la recepción de leche a los productores.

Entre los aspectos a considerar para la manufactura de leche, en el sector ganadero los más importantes son:

- Pastizales
- Sobrealimentación
- Tecnología
- Medicinas

El incremento de la cantidad de ganado en el país ha incidido en la producción ya que poco a poco ha ido agrandando la frontera agrícola, de manera que en la actualidad el 50% del territorio nacional es productivo o trabajado para distintas funciones agropecuarias. Principalmente los Brahman tienen influencia en la Costa y en el Oriente.

Los Brahman, son reses más grandes que necesitan más espacio por su peso y porque, precisamente, es ganadería de carne. En la Sierra, por el contrario, el pasto crece menos y las razas no necesitan tanto espacio para vivir”. En los últimos años, los ganaderos de la Costa incluyen a las razas de doble propósito para optimizar la producción. Estas producen leche y carne. En Manabí, la importación de reproductores de la raza Brahman empezó en los 50. Los hatos en Bahía y Chone, tuvieron el liderazgo (Salazar y Vera, 2009).

Manabí se caracteriza por tener un clima privilegiado y estar ubicado en una zona geográfica idónea para el desarrollo, por lo cual es considerada una Provincia agrícola, pecuaria y pesquera. En esta, existe gran cantidad de áreas de pasto, apto para crías de ganado, sea este dedicado a la cría de por lo menos una de estas especies para sostener a sus familias. Dentro de las diferentes especies de ganado que se crían en esta provincia tenemos que el vacuno es el más desarrollado.

Los hacendados aprovechan al máximo los productos que pueden obtener de estos animales. La existencia de industrias procesadoras de lácteos, carne, viseras y cuero del animal, hacen de este tipo de industria una de las que mayor tecnología posee para la explotación de sus productos, para poder brindar mejor calidad y tener mayor productividad (Dueñas y Muñoz, 2001).

### **2.1.2. MANEJO REPRODUCTIVO DEL HATO GANADERO**

El mejoramiento ganadero consiste en hacer un buen manejo del ganado, mejorando la raza poco a poco, seleccionando siempre la mejor vaca y descartando a la vaca con mayor cantidad de partos. La mayoría de productores mantiene su ganado todo el año en sus pastizales, sin darle mucho cuidado. Por tanto, obtienen una producción muy baja.

Cuando el ganado está descuidado se cree que es de mala raza, pero no siempre es así: haciendo un buen manejo habrá mejor producción y ganancias (Blanco *et al.*, 2003). Cuando no se ha llevado un manejo adecuado o cuando se va a iniciar con un programa de manejo, lo primero que se debe hacer es diagnosticar el estado reproductivo del hato.

- Porcentaje de vacas vacías.
- Porcentaje de vacas gestantes.
- Edad de la gestación de cada vaca.
- Número de vacas problema.
- Conocer la eficiencia de los técnicos en inseminación artificial
- Evaluar el desempeño de los toros.
- Predecir los partos y su distribución durante el año.
- Calcular parámetros reproductivos
- Conocer el periodo abierto de cada vaca año tras año, permite identificar a las vacas más fértiles.
- Calcular el periodo abierto promedio del hato.
- Identificar y desechar las vacas improproductivas.
- Suplementar a costos convenientes y mejorar la nutrición.

- Formar grupos de vacas por tiempo de gestación.

Las vacas problema son las vacas vacías que manifiestan una alteración en el sistema reproductor o que tienen más de 85 días posparto sin celo, monta o Inseminación Artificial. El examen del aparato reproductor por palpación transrectal a todas las vacas y vaquillas en edad reproductiva es de gran ayuda, ya que la información que se obtiene nos ayuda a tomar buenas decisiones y a realizar acciones que mejoran la rentabilidad (Iñiguez, s.f.).

Prieto *et al.*, (2002) manifiesta que una de las tecnologías más aplicadas en los hatos es la inseminación artificial, con el objetivo de obtener crías machos para la venta como reproductores y hembras de alto valor genético para remplazo de vientres. Así también García y Aguilar (2004) opina que la granja lechera promedio tiene una tecnificación moderada y utilizan métodos como la inseminación artificial, selección genética y sistemas de ordeño. Los sistemas de informática y cómputo en la administración y manejo del hato, han permitido incrementar también los rendimientos por animal.

Una parte de la producción se genera en Instituciones educativas como la ESPAM MLF, que posee un hato bovino el cual se dedica principalmente a la producción de leche, que se envía a la planta de lácteos de la misma institución para procesar queso semi pasteurizado, yogurt, requesón y otros derivados del lácteo, lo que aumenta la fuente de trabajo favoreciendo al crecimiento del país. Así mismo se realiza mejoramiento genético por medio de la sincronización e inseminación artificial.

La Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López ESPAM MFL en conjunto con el Ministerio de Agricultura, Ganadería, Acuacultura y Pesca (MAGAP) y el Instituto Nacional Autónomo de Investigaciones Agropecuarias (INIAP) se reunieron el 28 de Julio del 2010 en las oficinas de la Dirección Provincial Agropecuaria de Manabí con el objetivo de analizar el Proyecto Ganadero: Aplicación de alternativas tecnológicas en mejoramiento de pastos, genética, salud animal, administración, comercialización para el fortalecimiento de las ganaderías de pequeños y

medianos productores ganaderos de la provincia de Manabí, mismo que es llevado a cabo por las tres instituciones (ESPAM MFL, 2010).

## **2.2. FUNDAMENTOS DE INGENIERÍA DE SOFTWARE**

### **2.2.1. DEFINICIÓN**

La Ingeniería de software es la rama de la ingeniería que crea y mantiene las aplicaciones de software Ingeniería de Software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos (Sandoval, s.f.).

### **2.2.2. IMPORTANCIA**

La importancia de la Ingeniería de Software hace referencia a los actuales problemas que presenta su utilización y en muchos casos, su empleo totalmente empírico. Establece las principales situaciones como la incompreensión entre desarrolladores y usuarios o clientes, el largo tiempo de desarrollo, la corrección de fallas innecesariamente. Propone la enseñanza de la Ingeniería de Software aparejada con la de la programación como vía para eliminar la causa principal de lo planteado. Ejemplifica las ventajas del desarrollo de productos informáticos mediante el uso de una metodología de Ingeniería de Software con el Sistema de Negociación Comercial y el Sistema de Administración de las Capacidades. Hace especial énfasis en cómo pueden desarrollarse de forma simultánea dos sistemas que tienen intercambio de información por medio de la utilización de la Ingeniería de Software (Ruíz y Aguilera 2007).

El manejo de sistemas informáticos es tan importante para los planes de las empresas, que en ocasiones se contratan gerentes con la única responsabilidad de administrarlos. Comprar la tecnología adecuada es la parte fácil; el reto es adaptarla a las necesidades de la organización y asimilarla; es

por ello que alcanzar un alto grado de adaptación es un aspecto fundamental para el éxito de la compañía.

Es este precisamente el entorno donde se desarrolla la Ingeniería del Software, disciplina de la informática que integra proceso, métodos y herramientas para desarrollar y mantener software de calidad que resuelvan problemas de todo tipo. Ahora bien, la gestión de proyectos es una actividad importante dentro de la Ingeniería del Software. Empieza antes de iniciar cualquier actividad técnica y continúa a lo largo de todo el ciclo de vida del software. Comprende estimaciones, análisis de riesgos, planificación, seguimiento y control (Linares y Geizzelez, 2007).

Es importante medir el proceso de ingeniería de software y el producto que se elabora porque es la forma más objetiva de comprender y mejorar el proceso de desarrollo y el producto que se elabora. Si no se realizan mediciones, no hay forma de determinar si se está mejorando, las decisiones se basan sólo en evaluaciones subjetivas, lo que puede llevar a malas estimaciones o interpretaciones erróneas del proceso. Para establecer objetivos de mejora es necesario conocer el estado actual de desarrollo del software (Salazar, 2009).

### **2.2.3. TÉCNICAS DE PROGRAMACIÓN**

El diseño de un programa que se realiza sin seguir una metodología puede funcionar, pero se debe tener en cuenta que con el tiempo se convertirá en un conjunto de instrucciones. Es decir que las consecuencias de no utilizar un método determinado llevará a cometer errores que pueden costar el buen funcionamiento del mismo. Las diferentes etapas del programa suelen tener discontinuidad y son difícilmente identificables. En consecuencia existe una difícil fase de desarrollo y mantenimiento.

Aquí se identifican algunos de los problemas que suelen presentarse:

- Se presenta un exceso en la rigidez del programa, lo que implica que sean difícilmente adaptables a cualquier tipo de configuración.



- Se pierde mucho tiempo en la corrección de errores.
- Los programas generalmente son propios de cada programador, lo que implica que no sean muy manejables por otros.
- Para cuando se realice la documentación final, existirán deficiencias por presentarse la ausencia de diagramas, habrán descripciones que no estén completas o simplemente no las habrá, y la documentación no estará actualizada.

Existe una larga lista de problemas que pueden presentarse pero en este caso sólo se han citado algunos. Es de suma importancia poder prevenir las modificaciones que puedan realizarse en el futuro, así como también la actualización de la documentación. Para esto, se citan algunas que son importantes como:

- Incrementar el volumen de datos y estructuras.
- Modificación en la forma como se organiza la información.
- Modificación por actualización de los documentos.
- Ampliación, reducción o sustitución en el sistema del proceso de datos.

La creación de programas debe tener la flexibilidad suficiente para ser modificables en el momento en que se requiera. Estos deben ser claros, simples, con el fin de poder ser leídos e interpretados de forma fácil, para la programación deberá asumirse ciertas normas que permitan la estandarización de la programación, implicando una disminución en costos, independencia del programador y seguridad.

Cuando existen problemas con cierto grado de complejidad, el diseño del algoritmo requiere de una reducción y simplificación en la legibilidad del algoritmo. Las técnicas de programación que sirven de gran ayuda para la solución de problemas de este tipo, consiguiendo mayor rapidez y eficacia tenemos:

- Estructurada
  - Secuencial
  - Repetitiva
  - Selección
- Modular

### **2.2.1.1. PROGRAMACIÓN MODULAR**

Este es uno de los métodos para el diseño más flexible y de mayor actuación para la productividad de un programa. En este tipo de programación el programa es dividido en módulos, cada uno de las cuales realiza una tarea específica, codificándose independientemente de otros módulos. Cada uno de éstos son analizados, codificados y puestos a punto por separado.

Los programas contienen un módulo denominado módulo principal, el cual supervisa todo lo que sucede, transfiriendo el control a submódulos (los que son denominados subprogramas), para que puedan realizar sus funciones. Sin embargo, cada submódulo devolverá el control al módulo principal una vez completada su tarea. Si las tareas asignadas a cada submódulo son demasiado complejas, se procederá a una nueva subdivisión en otros módulos. Dada la ventaja de ser independientes el programa puede ser trabajado por diferentes programadores a la vez. Con ello se ahorra tiempo en el diseño del algoritmo y en su posterior codificación. También un módulo puede ser codificado sin afectar a los demás, incluso sin alterar su función principal. más pequeños aún.

Este procedimiento se realiza hasta que cada uno de los módulos ejecute tareas específicas. Estas pueden ser entrada, salida, manipulación de datos, control de otros módulos o alguna combinación de éstos. Puede ser que un módulo derive el control otro mediante un proceso denominado bifurcación, pero se debe tomar en cuenta que esta derivación deberá ser devuelta a su módulo original.

- **MEDIDA DE LOS MÓDULOS**

La determinación del tamaño de los módulos significa un obstáculo para la programación modular. No existe un estándar que determine qué tamaño debe tener un módulo, pero existen alternativas que deben tomarse en cuenta, como una aproximación del tamaño físico de una página, es por eso que los programadores tienen la tarea difícil de poder hacer que los módulos tengan aproximadamente esta medida.

- **IMPLEMENTACIÓN DE LOS MÓDULOS**

Este tipo de programación puede implementarse utilizando módulos que toman diferentes nombres, según el lenguaje de programación en el cual estén expresados: como módulos en todos los lenguajes (Murillo, 1990).

## **2.2.4. LENGUAJES DE PROGRAMACIÓN**

### **2.2.4.1. DEFINICIÓN**

Un lenguaje de programación desde el punto de vista del usuario es una notación que sirve para escribir algoritmos que puedan ser ejecutados en una máquina; una notación precisa, rigurosa y con la pretensión de ausencia de ambigüedad. Un lenguaje tiene un conjunto de componentes elementales e indivisibles: variables, literales, palabras reservadas, símbolos separadores y símbolos aritméticos que pueden formar parte de un programa fuente.

### **2.2.4.2. IMPORTANCIA**

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. Aunque muchas veces se usan los términos 'lenguaje de programación' y 'lenguaje informático' como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de

programación y a otros más, como, por ejemplo, el HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación).

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje Léxico.

Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa. Los procesadores usados en las computadoras son capaces de entender y actuar según lo indican programas escritos en un lenguaje fijo llamado lenguaje de máquina. Todo programa escrito en otro lenguaje puede ser ejecutado de dos maneras:

- Mediante un programa que va adaptando las instrucciones conforme son encontradas. A este proceso se le llama interpretar y a los programas que lo hacen se los conoce como intérpretes.
- Traduciendo este programa, al programa equivalente escrito en lenguaje de máquina. A ese proceso se le llama compilar y al programa traductor se le denomina compilador (Balderrama, s.f.).

López, (s.f.) considera que el paradigma de programación ha evolucionado, y entre muchos avances que se han producido, destacan tres grandes pasos: el primero es cuando se inventó la programación de computadoras como tal; el segundo paso es cuando se gestó la primera gran evolución, a la que se le nombró Programación Estructurada; y el tercer paso, es cuando se generó otra gran evolución, a la que se le denomina Programación Orientada a Objetos.

Existen varios lenguajes de programación entre ellos:

- FORTRAN
- COBOL
- PASCAL, C,
- C++
- JAVA
- VISUAL BASIC

#### **2.2.4.3. VISUAL STUDIO 2010**

Los desarrolladores que han utilizado las versiones anteriores de Visual Studio han podido comprobar que las diferentes ventanas de Visual Studio (Explorador de Soluciones, de Servidores, Tareas, Resultados, Vista de Clases, etc...) estaban obligadas a visualizarse dentro del propio espacio de visualización de Visual Studio, con el problema que esto supone si por ejemplo se trabaja con varios monitores.

En Visual Studio 2010 esto ya no sucede, ahora se puede seguir ubicando las ventanas en el espacio de visualización del IDE o bien optar por desplazarlas a otra parte del escritorio o a otro monitor adaptando la presentación a nuestros gustos. Aunque las ventanas tengan la posibilidad de flotar entre escritorios siguen presentando datos actualizados del estado del entorno, por lo que si cambiamos alguna propiedad de un control notaremos dicho cambio inmediatamente en cualquier ventana relacionada con dicho control (Rayo, 2010).

#### **2.2.4.4. VISUAL BASIC .NET**

Con la aparición de .NET Framework, y de Visual Basic .NET, como una de las herramientas estrella para el desarrollo sobre esta nueva plataforma de trabajo, estamos asistiendo a una evolución/revolución sin precedentes en el mundo de la informática, que sitúa a este clásico de la programación en una posición difícil de igualar y menos aún de superar.

Visual Basic .NET (VB.NET a partir de ahora), como cada nueva versión de las que han aparecido en el mercado de este producto, incorpora, como es natural, un buen conjunto de novedades. Sin embargo, la inclusión de Visual Basic en el entorno de .NET , añade también un compendio de drásticos cambios para los programadores de versiones anteriores, derivados en su conjunto, de la necesidad de afrontar con garantías de éxito el desarrollo de la nueva generación de aplicaciones para Internet, objetivo perseguido por todas las herramientas de desarrollo actuales (Blanco, 2002).

Visual Basic .NET, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o GUI (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual) (Quiroz, 2010).

En algunos experimentos con humanos se ha eliminado la necesidad de usar una interfaz programando el experimento en Visual Basic (VB) y registrando las respuestas directamente en el teclado. El uso de VB se ha extendido en diferentes laboratorios debido a que permite la presentación de imágenes complejas en un monitor y reduce el equipo experimental con humanos a una computadora. VB es probablemente la herramienta más útil que tienen a la mano los investigadores interesados en desarrollar experimentos con una computadora (Escobar y Lattal, 2010).

## **2.2.5. DIAGRAMAS UML (UNIFIED MODELING LANGUAGE)**

### **2.2.5.1. DIAGRAMA DE CASOS DE USO**

Un Diagrama es una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo conexo de arcos (relaciones) y vértices (otros elementos del modelo). Un diagrama no es un elemento semántico, un diagrama muestra representaciones de elementos semánticos

del modelo, pero su significado no se ve afectado por la forma en que son representados (Giraldo *et al.*, 2011)

En la especificación de la Superestructura del Unified Modeling Language (UML), el diagrama de casos de uso se define como el “diagrama que muestra las relaciones entre los actores y el sujeto (sistema) y los casos de uso.

El diagrama de casos de uso, describe los requisitos funcionales del sistema en términos de las secuencias de acciones”, se presentan los siguientes elementos de su especificación:

- **Casos de uso:** son las especificaciones de un conjunto de acciones realizadas por el actor sobre el sistema.
- **Actores:** son los roles que los usuarios desempeñan respecto del sistema y que emplean los casos de uso.
- **Relaciones:** identifican la comunicación existente entre actores y casos de uso. Las relaciones pueden ser de cuatro tipos:
  - Asociación: se establece entre los actores y casos de uso.
  - <<include>>: se presenta cuando el caso de uso origen incluye también el comportamiento descrito por el caso de uso destino.
  - <<extend>>: ocurre cuando el caso de uso origen extiende el comportamiento del caso de uso destino.
  - <<inheritance>>: un caso de uso origen hereda la especificación del caso de uso destino y posiblemente la modifica y/o amplía (Zapata *et al.*, 2009).

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. Los elementos básicos son:

**AUTORES:** Un actor en un diagrama de casos de uso representa un rol que alguien puede estar jugando, no un individuo particular por lo tanto puede

haber personas particulares que puedan estar usando el sistema de formas diferentes en diferentes ocasiones: socio de biblioteca y bibliotecario.

**RELACIONES:** Por defecto, en UML se supone que las relaciones son bidireccionales, excepto cuando están presentes las puntas de flecha para restringirlas. Entre los elementos de un diagrama de casos de uso se pueden presentar tres tipos de relaciones:

**RELACIÓN DE ASOCIACIÓN:** Relación entre un actor y un caso de uso. Una asociación entre un actor y un caso de uso indica que el actor y el caso de uso se comunican entre sí, enviándose y recibiendo mensajes. Se trata de una relación bidireccional, que se representa por una línea continua pero sin flechas en el origen y en el destino. Sin embargo, hay circunstancias en las que sí es necesario limitar la comunicación pues sólo se utiliza en una sola dirección, en cuyo caso el sentido de la misma se indica mediante la punta de una flecha.

**RELACIÓN DE DEPENDENCIA:** Relación entre casos de uso. Se representa por una línea discontinua dirigida entre ellos (del elemento dependiente al independiente). Hay dos tipos:

- Incluye «includes» o Utiliza «uses». Significa que una instancia del caso de uso fuente comprende también el comportamiento descrito por el caso de uso destino. Es decir, denota la inclusión del comportamiento de un escenario en otro.
- Extiende «extends». Significa que el caso de uso fuente extiende el comportamiento del caso de uso destino. Es decir, denota cuando un caso de uso es una especialización de otro.

**RELACIÓN DE GENERALIZACIÓN:** Relación entre casos de uso y raras veces entre actores. La generalización entre casos de uso es igual que la generalización entre clases. Aquí significa que el caso de uso hijo hereda el comportamiento y el significado del caso de uso padre; el hijo puede añadir o incluso invalidar el comportamiento de su padre. Se representa por una línea continua dirigida entre ellos (del hijo al padre) con la punta de una flecha en forma triangular.



**ESCENARIO:** es una interacción entre el sistema y los actores, que puede ser descrito mediante una secuencia de mensajes. Un caso de uso es una generalización de un escenario (Otero, s.f.).

### 2.2.5.2. REQUERIMIENTOS

Los requerimientos de software pueden dividirse en 2 categorías: requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema.

Por otra parte los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, entre otros (Arias, 2005).

### 2.2.5.3. DIAGRAMA DE FLUJOS

Es la representación gráfica de un sistema que ilustra como fluyen los datos a través de distintos procesos. Los diagramas de flujo de datos (DFD) se realizan a distintos niveles de abstracción, detallando procesos concretos que aparecen como elementos simples en DFD de nivel superior (DECSAI, s.f.). Los elementos de un diagrama de flujos de datos (DFD) son:

- **ENTIDAD EXTERNA:** Persona, grupo de personas o unidad de negocio que entrega y recibe información.
- **PROCESO:** Conjunto de actividades de negocio que explican que se hace y como se llevan a cabo.

- **FLUJO DATOS:** Señala el flujo de datos de una entidad externa a un proceso y viceversa, de un proceso a otro, y de un proceso a un almacén de datos y viceversa.
- **ALMACÉN DATOS:** Lugar físico donde se almacenan los datos procesados o desde donde se recuperan para apoyar un proceso (Zuloaga, s.f.).

## **2.2.6. PRUEBAS DE SOFTWARE**

### **2.2.6.1. DEFINICIÓN**

El término “prueba” se ha utilizado a lo largo de los años para referenciar diferentes conceptos: haciendo mención a técnicas para realizar pruebas, dando nombre a diferentes actividades y objetivos en la forma de aplicar las pruebas (unitarias, integración, aceptación o sistema). se puede considerar que las pruebas alcanzan objetivos diferentes. Esto nos permite afirmar que las pruebas del software han ido evolucionado para poder desempeñar nuevos papeles en el desarrollo de software y sistemas (Yagüe y Garbajosa, 2009).

La fase de pruebas es una de las más costosas del ciclo de vida software. En sentido estricto, deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación (p.ej., la base de datos). Obviamente, se aplican diferentes técnicas de prueba a cada tipo de producto software.

### **2.2.6.2. TIPOS DE PRUEBAS DE SOFTWARE**

A continuación se describen las principales tipos pruebas que se pueden realizar a cualquier tipo de software. Cada prueba contendrá como mínimo a siguiente información:

- Objetivo de la prueba
- Descripción de la prueba

- Técnica
- Criterio de Completitud
- Consideraciones Especiales

#### **a) PRUEBAS UNITARIAS**

##### **OBJETIVO DE LA PRUEBA:**

Se focaliza en ejecutar cada módulo (o unidad mínima a ser probada, ej = una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores. Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido.

##### **DESCRIPCIÓN DE LA PRUEBA:**

- Particionar los módulos en pruebas en unidades lógicas fáciles de probar.
- Por cada unidad hay que definir los casos de prueba (pruebas de caja blanca).
- Para esto los casos de prueba deben diseñarse de forma tal que se recorran todos los caminos de ejecución posibles dentro del código bajo prueba; por lo tanto el diseñador debe construirlos con acceso al código fuente de la unidad a probar.
- Los aspectos a considerar son los siguientes: Rutinas de excepción, Rutinas de error, Manejo de parámetros, Validaciones, Valores válidos, Valores límites, Rangos, Mensajes posibles.

##### **TÉCNICA:**

- Comparar el resultado esperado con el resultado obtenido.
- Si existen errores, reportarlos.

##### **CRITERIO DE COMPLETITUD:**

- Todas las pruebas planeadas han sido ejecutadas.

- Todos los defectos que se identificaron han sido tenidos en cuenta.

### **CONSIDERACIONES ESPECIALES:**

Para la elaboración de pruebas unitarias en java se puede utilizar el UNIT y CACTUS.

### **b) PRUEBAS DE INTEGRACIÓN**

#### **OBJETIVO DE LA PRUEBA:**

- Identificar errores introducidos por la combinación de programas probados unitariamente.
- Determina cómo la base de datos de prueba será cargada.
- Verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.
- Verificar que las especificaciones de diseño sean alcanzadas.
- Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.

#### **DESCRIPCIÓN DE LA PRUEBA:**

- Describe cómo verificar que las interfaces entre las componentes de software funcionan correctamente.
- Determina cómo la base de datos de prueba será cargada.
- Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.
- Decide qué acciones tomar cuando se descubren problemas.

Por cada Caso de Prueba ejecutado:

Comparar el resultado esperado con el resultado obtenido.

#### **TÉCNICA:**

- Utilizar la técnica top-down. Se empieza con los módulos de nivel superior, y se verifica que los módulos de nivel superior llaman a los de nivel inferior de manera correcta, con los parámetros correctos.
- Utilizar la técnica down-top. Se empieza con los módulos de nivel inferior, y se verifica que los módulos de nivel inferior llaman a los de nivel superior de manera correcta, con los parámetros correctos.

**CRITERIO DE COMPLETITUD:**

- Todas las pruebas planeadas han sido ejecutadas.
- Todos los defectos que se identificaron han sido tenidos en cuenta.

**c) PRUEBAS DE CARGA****OBJETIVO DE LA PRUEBA:**

- Verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga.

**DESCRIPCIÓN DE LA PRUEBA:**

- Las pruebas de carga miden la capacidad del sistema para continuar funcionando apropiadamente bajo diferentes condiciones de carga.
- La meta de las pruebas de carga es determinar y asegurar que el sistema funciona apropiadamente aún más allá de la carga de trabajo máxima esperada. Adicionalmente, las pruebas de carga evalúan las características de desempeño (tiempos de respuesta, tasas de transacciones y otros aspectos sensibles al tiempo).

**TÉCNICA:**

- Use los scripts desarrolladas para Pruebas del Negocio.
- Modifique archivos de datos (para incrementar el número de transacciones o veces que cada transacción ocurre).

**CRITERIO DE COMPLETITUD:**

- Múltiples transacciones, múltiples usuarios. Se completaron las pruebas de los scripts sin ninguna falla y dentro del tiempo esperado.
- Consideraciones Especiales:
- Las pruebas de carga deben ser ejecutadas en una máquina dedicada o en un tiempo dedicado. Esto permite control total y medidas precisas.
- La Base de datos utilizada para pruebas de desempeño debe ser de un tamaño real o proporcionalmente más grande que la diseñada.

**d) PRUEBAS DE STRESS****OBJETIVO DE LA PRUEBA:**

- Verificar que el sistema funciona apropiadamente y sin errores, bajo estas condiciones de stress:
- Memoria baja o no disponible en el servidor.
- Máximo número de clientes conectados o simulados (actuales o físicamente posibles)
- Múltiples usuarios desempeñando la misma transacción con los mismos datos.
- El peor caso de volumen de transacciones (ver pruebas de desempeño).

NOTAS: La meta de las pruebas de stress también es identificar y documentar las condiciones bajo las cuales el sistema FALLA.

**DESCRIPCIÓN DE LA PRUEBA:**

Las pruebas de stress se proponen encontrar errores debidos a recursos bajos o completitud de recursos. Poca memoria o espacio en disco puede revelar defectos en el sistema que no son aparentes bajo condiciones normales. Otros defectos pueden resultar de incluir recursos compartidos, como bloqueos de base de datos o ancho de banda de la red. Las pruebas de stress identifican la carga máxima que el sistema puede manejar.

El objetivo de esta prueba es investigar el comportamiento del sistema bajo condiciones que sobrecargan sus recursos. No debe confundirse con las pruebas de volumen: un esfuerzo grande es un pico de volumen de datos que se presenta en un corto período de tiempo. Puesto que la prueba de esfuerzo involucra un elemento de tiempo, no resulta aplicable a muchos programas, por ejemplo a un compilador o a una rutina de pagos. Es aplicable, sin embargo, a programas que trabajan bajo cargas variables, interactivas, de tiempo real y de control de proceso.

Aunque muchas pruebas de esfuerzo representan condiciones que el programa encontrará realmente durante su utilización, muchas otras serán en verdad situaciones que nunca ocurrirán en la realidad. Esto no implica, sin embargo, que estas pruebas no sean útiles. Si se detectan errores durante estas condiciones “imposibles”, la prueba es valiosa porque es de esperar que los mismos errores puedan presentarse en situaciones reales, algo menos exigentes.

#### **TÉCNICA:**

- Use los scripts utilizados en las pruebas de desempeño.
- Para probar recursos limitados, las pruebas se deben correr en un servidor con configuración reducida (o limitada).
- Para las pruebas de stress restantes, deben utilizarse múltiples clientes, ya sea corriendo los mismos scripts o scripts complementarios para producir el peor caso de volumen de transacciones.

#### **CRITERIO DE COMPLETITUD:**

Todas las pruebas planeadas han sido ejecutadas y excedidas sin que el sistema falle. (O si las condiciones en que el sistema falle ocurren por fuera de las condiciones especificadas).

**CONSIDERACIONES ESPECIALES:**

- Producir stress en la red puede requerir herramientas de red para sobrecargarla de tráfico.
- El espacio en disco utilizado para el sistema debe ser reducido temporalmente para limitar el espacio disponible para el crecimiento de la Base de datos.
- Sincronización de varios clientes accediendo simultáneamente los mismos registros (Abad, 2005).

**2.2.7. METODOLOGÍAS DE DESARROLLO DE SOFTWARE**

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo según Avison (1995) citado por Tinoco (2010).

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software. Se podrían clasificar en dos grandes grupos:

- Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas Metodologías Pesadas.
- Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente



funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas Metodologías ligeras/ágiles (Carrillo *et al.*, 2008).

### **2.2.7.1. METODOLOGÍAS ÁGILES**

Desde el inicio de la definición, se tenía claro que era necesario adaptar un proceso clásico, a uno dentro de la corriente ágil, basando la carga del proceso en herramientas y metodologías ágiles. Este espíritu ágil no surge de la dirección hacia la estructura, sino de forma de inversa y espontáneamente años atrás: los propios programadores y técnicos propusieron dichas mejoras en sus procesos, siendo estas escuchadas y valoradas por la dirección (Navarro y Garzás, 2010).

Las metodologías ágiles contemplan el desarrollo de software de manera integral, con un énfasis especial en la entrega de valor al cliente, en la generación de negocio y el retorno de la inversión (ROI). Sólo hay una manera efectiva de crear software que funcione, y es de manera colaborativa. La colaboración entre cliente y desarrolladores es indispensable: se debe fomentar y apoyar. Díaz (2009) manifiesta que las metodologías ágiles nos están proporcionando un marco en el que lograr una calidad satisfactoria es parte integral del proceso de desarrollo.

Rodríguez *et al.*, (2010) menciona que Agile Software Development (ADS) define un grupo de metodologías de desarrollo software basadas en un desarrollo iterativo, donde los requisitos y las soluciones evolucionan a través de una estrecha colaboración entre equipos inter-funcionales y autoorganizados. La mayor parte de los estudios sobre metodologías ágiles que existen en la actualidad relatan experiencias, normalmente positivas, en la aplicación de este tipo de metodologías en organizaciones y proyectos concretos, y por tanto, difícilmente generalizables.

Giugni y Loaiza (2008) manifiestan que se realizó un estudio comparativo entre las metodologías ágiles y las tradicionales, el cual reflejó algunas de las ventajas de los procesos ágiles: facilitan la generación rápida de prototipos y de versiones previas a la entrega, además garantizan un riesgo menor ante la posibilidad de cambios en los requisitos.

Entre las metodologías ágiles más populares están:

- DSDM (Dynamic Systems Development Method)
- FDD (Feature Driven Development)
- MSF (Microsoft Solution Framework)
- DSDM (Dynamic Systems development methods)
- CMMI (Capability Maturity Model Integration)
- SCRUM
- XP (eXtreme Programming)

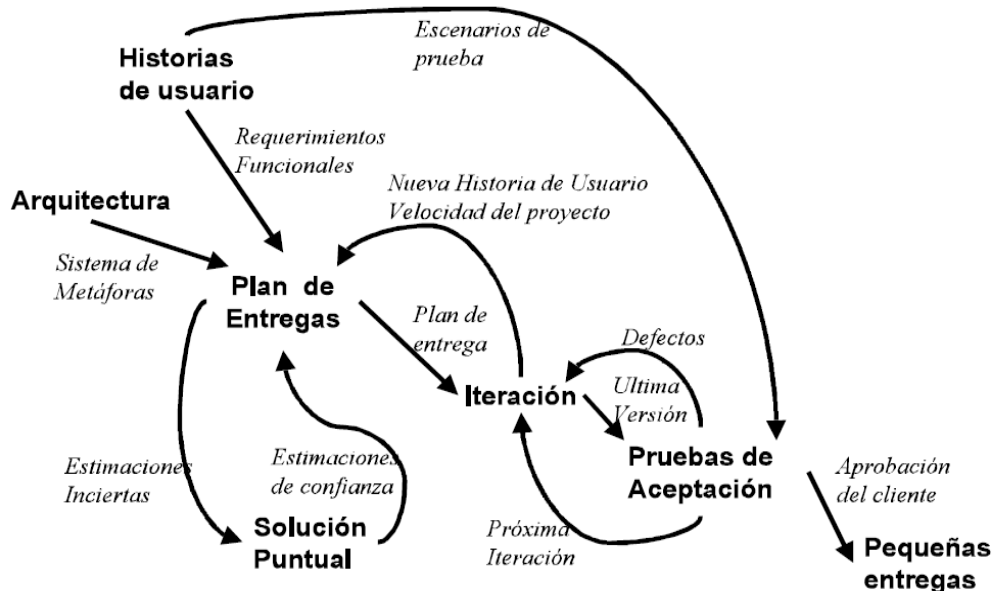
#### **2.2.7.1.1. EXTREME PROGRAMING (XP)**

XP (eXtreme Programing) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los programadores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes (Orjuela y Rojas, 2008).

XP (eXtreme Programing) nace como nueva disciplina de desarrollo de software hace aproximadamente unos seis años, y ha causado un gran revuelo entre el colectivo de programadores del mundo. Kent Beck, su autor, es un programador que ha trabajado en múltiples empresas y que actualmente lo hace como programador en la conocida empresa automovilística DaimlerChrysler. Con sus teorías ha conseguido el respaldo de gran parte de la

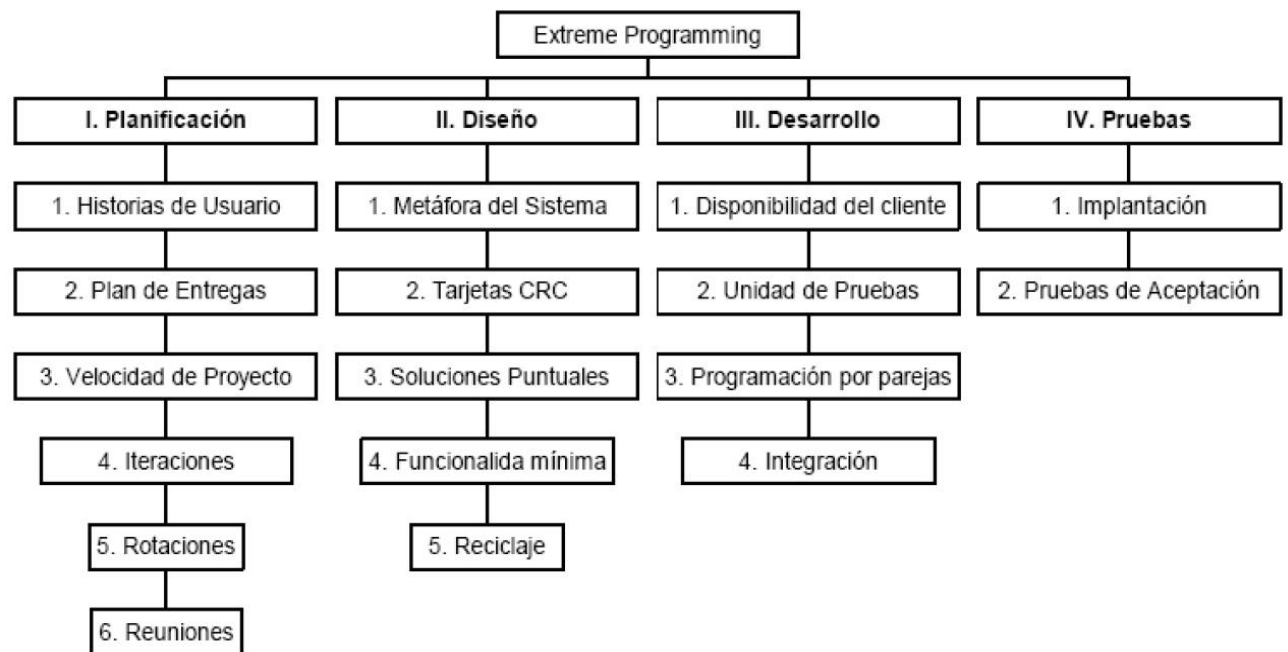
industria del software y el rechazo de otra parte (Calero, 2003). Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuándo lo necesita. Por tanto, debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación.

El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software. La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica (Delgado, 2008).



**Figura 2.1.** Proceso XP

**Fuente:** Orjuela y Rojas (2008)



**Figura 2.2.** Fases de la Metodología XP.

**Fuente:** Joskowicz, 2008

### **FASE I: Planificación**

La metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba (“*spikes*”), para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas (“*Release Plan*”) en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios”.

Los conceptos básicos de esta planificación son los siguientes:

### **a) Historias de usuarios**

Las “Historias de usuarios” (*“User stories”*) sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido.

Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.

### **b) Plan de entregas (“Release Plan”)**

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.). XP denomina a esta reunión “Juego de planeamiento” (*“Planning game”*), pero puede denominarse de la manera que sea más apropiada al tipo de empresa y cliente (por ejemplo, Reunión de planeamiento, *“Planning meeting”* o *“Planning workshop”*).

Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

### **c) Plan de iteraciones (“Iteration Plan”)**

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación.

Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

### **d) Reuniones diarias de seguimiento (“Stand-up meeting”)**

El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie.

## **FASE II: Diseño**

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

### **a) Simplicidad**

Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

### **b) Soluciones “spike”**

Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “spike”<sup>1</sup>), para explorar diferentes soluciones. Estos programas son únicamente para probar o evaluar una solución, y suelen ser desechados luego de su evaluación.

## **FASE III: Desarrollo del código**

### **a) Disponibilidad del cliente**

Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP. Al comienzo del proyecto, el cliente debe proporcionar las historias de usuarios. Pero, dado que estas historias son expresamente cortas y de “alto nivel”, no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo. No se requieren de largos documentos de especificaciones, sino que los detalles son proporcionados por el cliente, en el momento adecuado, “cara a cara” a los desarrolladores.

### **b) Programación en pares**

XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que ésta práctica duplica el tiempo asignado al proyecto (y por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

## **FASE IV: Pruebas**

### **a) Pruebas unitarias**

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (*“Test-driven programming”*). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

### **b) Detección y corrección de errores**

Cuando se encuentra un error (“bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

### **c) Pruebas de aceptación**

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra” (*“Black box system tests”*). Los clientes son responsables de verificar que los resultados de éstas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución.

Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de



aceptación, de manera que todo el equipo esté al tanto de esta información (Joskowicz, 2008).

## **2.3. GENERALIDADES DE GESTORES DE BASES DE DATOS**

### **2.3.1. DEFINICIÓN**

Los sistemas de Gestión de Bases de Datos, son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporciona un acceso controlado a la misma (Gil *et al.*, 2005).

### **2.3.2. IMPORTANCIA**

Para tener conocimiento de la evolución real de los datos en un sistema se requiere tener información sobre el tiempo. Ejemplo de tales aplicaciones son la administración de oficinas, asuntos bancarios, registro de personal, registros médicos, administración de inventarios, monitoreo del tiempo, etc. Las bases de datos tradicionales carecen de capacidades para registrar y manejar las variaciones dependientes del tiempo del mundo que modelan. Es por ello que han aparecido diferentes soluciones para dotar a los Sistemas de gestión de bases de datos (SGBD) de posibilidades de operar con el tiempo. Una base de datos temporal soporta algún aspecto de tiempo. Una base de datos relacional es temporal si contiene una o más relaciones temporales. Las bases de datos temporales permiten el almacenamiento y recuperación de todos los estados asumidos por un objeto durante su tiempo de vida, y con ello registran su evolución en el tiempo. Se caracterizan por la incorporación de uno o más atributos temporales, que pueden reflejar el momento en que un hecho fue actualizado en la base de datos (tiempo de transacción), o el momento en el que realmente ocurrieron los hechos que se modelan (tiempo válido), o ambos inclusive. En dependencia de la información temporal con que se trabaje, estas bases de datos se clasifican como Bases de datos tradicionales (Snapshot),

Bases de datos de tiempo de transacción (BDTT), Bases de datos de tiempo válido (BDTV) y Bases de datos bitemporales (BDBT) (Martínez y Díaz, 2010).

Control sobre la redundancia de datos: Los sistemas de archivos almacenan varias copias de los mismos datos en archivos distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos. En los sistemas de bases de datos todos estos archivos están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos, o bien es necesaria para mejorar las prestaciones.

Los SGBD es la aplicación que interactúa con los usuarios de los programas de aplicación y la base de datos. Algunos de los SGBD más conocidos son:

- DB2
- SLQ/DS
- ORACLE
- INGRES
- INFORMIX
- SQL

Entre los objetivos de un sistema de gestión de base de datos (SGBD) se detallan:

- Definir la Base de Datos mediante el Lenguaje de Definición de Datos, el cual permite especificar la estructura, tipo de datos y las restricciones sobre los datos, almacenándolo todo en la base de datos.
- Separar la descripción y manipulación de la data, permitiendo un mayor entendimiento de los objetos, además de flexibilidad de consulta y actualización de los datos:

- Permitir la inserción, eliminación, actualización, consulta de los datos mediante el Lenguaje de Manejo de Datos, lo que permite resolver el problema que presentan los sistemas de archivos, donde hay que trabajar con un conjunto fijo de consultas o la necesidad de tener muchos programas de aplicaciones (Gil *et al.*, 2005).

### **2.3.2.1. MICROSOFT SQL EXPRESS 2008 R2**

Microsoft SQL Server 2008 R2 Express RTM es un sistema de administración de datos eficaz y confiable que ofrece un variado conjunto de características, protección de datos y rendimiento para aplicaciones incrustadas, sitios web ligeros y almacenes de datos locales. Diseñada para una implementación fácil y la agilización de creación de prototipos, SQL Server Express se ha diseñado para facilitar la implementación y agilizar la creación de prototipo, e integrarse a la perfección con otras inversiones de infraestructura de servidor.

SQL 2008 r2 Express Edition es una versión light con licencia gratuita que se distribuye con Management Studio Express y Reporting Services. SQL Server Management Studio es una potente herramienta utilizada por desarrolladores y administradores para trabajar y gestionar bases de datos. Con Management Studio puede realizar varias tareas entre ellas: procesar objetos de modelo de minería de datos, examinar modelos de minería de datos, crear scripts de objetos de minería de datos, crear consultas de predicción, eliminar objetos de minería de datos de la base de datos.

SQL Server 2008 Management Studio Express combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos que permiten a desarrolladores y administradores de todos los niveles obtener acceso a SQL Server, incluye una gran cantidad de nuevas características que permiten una gestión más racional y eficaz del mismo, aumentan el rendimiento, la escalabilidad y la estabilidad del servidor y, permiten una configuración avanzada a nivel de servicios, seguridad del servidor, etc.

Entre las nuevas características, quizás la más llamativa sea el soporte para compatibilidad con .NET. Esto permite la programación de ensamblados en C# o Visual Basic 2008 y su ejecución en SQL Server, lo que abre un gran abanico de posibilidades complementando la funcionalidad que proporciona T-SQL (López y Aroa, 2009).

Para tener acceso a una base de datos almacenada en un SQL utilizando Visual Studio .Net 2008 o 2010 con DataSets, existen diversas maneras de realizarlo, la mayor parte de ellas basadas en programación compleja. Una manera de realizarlo es agregar la carpeta ASP.NET App\_Code, añadir un nuevo elemento del tipo DataSet, anexar la o las tablas de la bases de datos, crear las consultas SQL, diseñar los Webforms y escribir el código para llamar a las consultas para el usuario (Rivera, 2011).

## **CAPÍTULO III. DESARROLLO METODOLÓGICO**

La presente tesis se realizó en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López en el hato bovino de la carrera de Pecuaria, la misma que está situado en la ciudad de Calceta en el sitio El Limón, lo que optimizó el manejo en la producción de leche para la elaboración de productos lácteos y reproducción de cada ganado registrado en el hato, sirviendo de ayuda en la toma de decisiones.

### **3.1. MÉTODOS CIENTÍFICOS**

Los métodos utilizados en el desarrollo del software fueron el inductivo y deductivo, pues se investigó las necesidades del hato bovino de la ESPAM MFL y se dedujo que es necesario crear un software de control ganadero para ordenar los registros y a la vez para mantener información actualizada de cada uno de los miembros del hato, al mismo tiempo se dialogó con el coordinador del hato bovino para conocer más a fondo la situación del procesamiento de los datos ganaderos.

### **3.2. MÉTODO INFORMÁTICO**

La Metodología informática XP (eXtreme Programming) es la que guió el desarrollo del software, permitió una visión general del producto detallando las funciones que tienen mayor prioridad en el desarrollo en un tiempo breve. Este método consta de cuatro fases: Planeación, Diseño, Desarrollo del código y Pruebas.

#### **3.2.1. FASE I: PLANEACIÓN**

En cumplimiento a la primera fase de metodología XP, se realizaron visitas en el hato bovino de la ESPAM MLF en las que se establecieron conversaciones

con el Dr. Raúl Guevara Viera (Anexo 01) encargado de tal dependencia, además se empleó una entrevista (Anexo 02) para conocer los procesos llevados a cabo en el mismo lugar. Mediante las actividades mencionadas se obtuvo la información necesaria para diseñar el diagrama de flujo de datos (DFD), como se muestra en la figura 3.1.

El diagrama de flujo de datos explica el ciclo vital del ganado desde su gestación hasta su descarte o venta. Inicia detallando cuando el animal ha entrado en estado de preñez (vaca gestante), después de los 283 días procede al parto, y por tanto se obtiene el ternero y la madre, esta última entra al período de lactancia, la cual es ordeñada dos veces al día y la producción pasa a la planta de lácteos de la ESPAM MFL.

Una vez que se observa la producción de esta res, se resuelve si es conveniente volverla a seleccionar para la fecundación o no, si tiene algún problema se procede a vender, por el contrario si es una buena ejemplar se prepara; si presenta celo natural se la conduce al apareamiento, si se consigue la gestación es necesario el destete y luego el secado. Si no presenta celo se realizará una sincronización para continuar con la inseminación artificial, así mismo si se da la gestación se llevarán a cabo los procesos ya mencionados.

Por otro lado los terneros después del destete reciben una suplementación alimenticia para seleccionar las novillas de remplazo, las restantes se venden, en el caso de las hembras. Si son machos se escoge el procreador generalmente uno de cada cinco generaciones y los demás se castran y después de un tiempo se descartan.

Al contar con información clara de los procesos, se elaboraron las historias de usuario (Cuadro 3.1-3.15), que especifican los procesos que va a realizar el sistema, y que cuenta con la aprobación del representante del hato bovino.

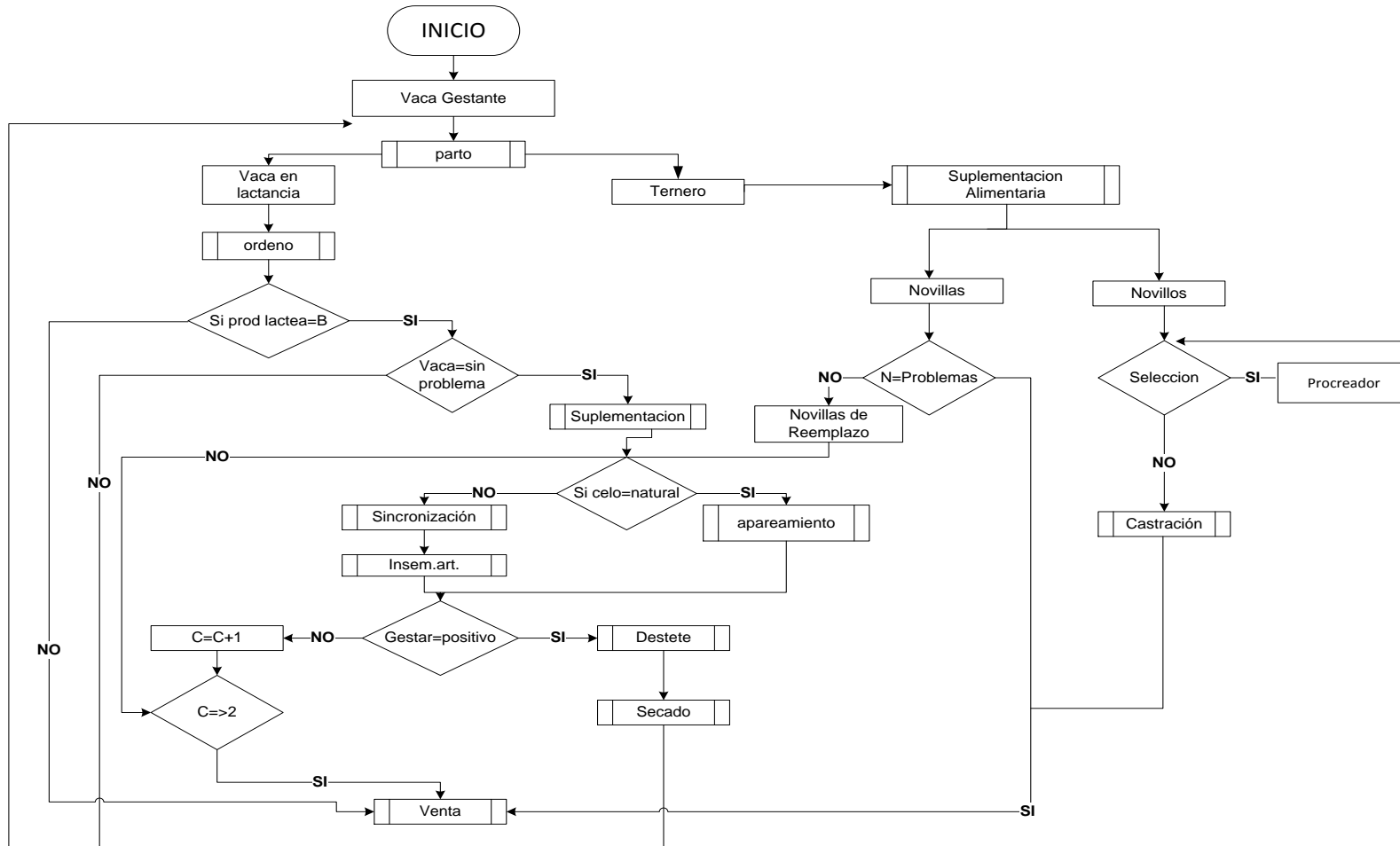


Figura 3.1. Procesos del ciclo vital ganadero en el hato bovino de la ESPAM MFL.

**Cuadro 3.1.** Historia del registro de animales (Iteración1).

<b>Historia de Usuario</b>	
Número:1	Usuario: Administrador del Hato Bovino
Nombre historia: Registro de datos del animal	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:1
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El Administrador ingresa los datos del animal, el sistema procesa los datos y los guarda en la base de datos.	
Observaciones: CONFIRMADO con el cliente	

**Cuadro 3.2.** Historia del registro animal (Iteración 2).

<b>Historia de Usuario</b>	
Número:1	Usuario: Administrador del Hato Bovino
Nombre historia: Registro de datos del animal	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:2
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El Administrador puede de ingresar los datos del animal, generar un listado de todos los ejemplares guardados y además realizar una búsqueda específica de cada animal, mediante el código o nombre que lo identifica.	
Observaciones: CONFIRMADO con el cliente	



**Cuadro 3.3.** Historia del registro animal (Iteración 3).

<b>Historia de Usuario</b>	
Número:1	Usuario: Administrador del Hato Bovino
Nombre historia: Registro de datos del animal	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:3
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El administrador puede ingresar los datos de cada animal, generar un listado de ellos, realizar una búsqueda específica y modificar los registros que considere necesario. Además puede imprimir un reporte individual con los datos del bovino.	
Observaciones: CONFIRMADO con el cliente	

**Cuadro 3.4.** Historia del registro de eventos (Iteración 1).

<b>Historia de Usuario</b>	
Número:2	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar eventos reproductivos o veterinarios de cada bovino	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:1
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El Administrador puede generar la lista de bovinos ingresados en la base de datos, escoger uno y asignarle un determinado evento y luego guardarlo,	

Observaciones: CONFIRMADO con el cliente
---

**Cuadro 3.5.** Historia del registro de eventos (Iteración 2).

<b>Historia de Usuario</b>	
Número:2	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar eventos reproductivos o veterinarios de cada bovino	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:2
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El administrador ingresa los eventos de cada animal, los modifica y guarda los cambios.	
Observaciones: CONFIRMADO con el cliente	

**Cuadro 3.6.** Historia del registro de eventos (Iteración 3).

<b>Historia de Usuario</b>	
Número:2	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar eventos reproductivos o veterinarios de cada bovino	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:3
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	

<p>Descripción: El administrador ingresa eventos del animal y modifica los mismos. Además genera e imprime reportes con los eventos de cada vacuno.</p>
<p>Observaciones: CONFIRMADO con el cliente</p>

**Cuadro 3.7.** Historia del registro de producción (Iteración 1).

<b>Historia de Usuario</b>	
Número:3	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar datos de producción láctea	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:1
<p>Programador responsable:</p> <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
<p>Descripción: El administrador ingresa los datos de producción de los respectivos bovinos.</p>	
<p>Observaciones: CONFIRMADO con el cliente</p>	

**Cuadro 3.8.** Historia del registro de producción (Iteración 2).

<b>Historia de Usuario</b>	
Número:3	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar datos de producción láctea	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:2

Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>
Descripción: El administrador ingresa los datos de producción, los modifica y guardar los cambios.
Observaciones: CONFIRMADO con el cliente

**Cuadro 3.9.** Historia del registro de producción (Iteración 3).

<b>Historia de Usuario</b>	
Número:3	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar datos de producción láctea	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:3
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El administrador puede ingresar datos, modificarlos e imprimir el reporte de un determinado animal con sus respectivos datos de producción.	
Observaciones: CONFIRMADO con el cliente	

**Cuadro 3.10.** Historia del registro de reproducción (Iteración 1).

<b>Historia de Usuario</b>	
Número:4	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar datos de reproducción (crías) del bovino pertinente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:1
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El administrador ingresa los datos de reproducción de los respectivos bovinos.	
Observaciones: CONFIRMADO con el cliente	

**Cuadro 3.11.** Historia del registro de reproducción (Iteración 2).

<b>Historia de Usuario</b>	
Número:4	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar datos de reproducción (crías) del bovino pertinente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:2
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El administrador ingresa los datos de reproducción, los modifica y guardar los cambios.	

Observaciones:  
CONFIRMADO con el cliente

**Cuadro 3.12.** Historia del registro de reproducción (Iteración 3).

<b>Historia de Usuario</b>	
Número:4	Usuario: Administrador del Hato Bovino
Nombre historia: Registrar datos de reproducción (crías) del bovino pertinente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:3
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El administrador puede ingresar datos, modificarlos e imprimir el reporte de un determinado animal con sus respectivos datos de reproducción.	
Observaciones: CONFIRMADO con el cliente	

**Cuadro 3.13.** Historia de reporte (Iteración 1).

<b>Historia de Usuario</b>	
Número:5	Usuario: Administrador del Hato Bovino
Nombre historia: Reporte General de datos de cada bovino	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:1
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	

<p><b>Descripción:</b> El sistema le permite al administrador generar un reporte de cada animal con todos los procesos por los cuales haya pasado.</p>
<p><b>Observaciones:</b> CONFIRMADO con el cliente</p>

**Cuadro 3.14.** Historia de reporte (Iteración 2).

<b>Historia de Usuario</b>	
Número:5	Usuario: Administrador del Hato Bovino
Nombre historia: Reporte General de datos de cada bovino	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:2
<p>Programador responsable:</p> <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
<p><b>Descripción:</b> El administrador puede imprimir el reporte generado de un determinado bovino con todos los procesos de cada animal generado</p>	
<p><b>Observaciones:</b> CONFIRMADO con el cliente</p>	

**Cuadro 3.15.** Historia de reporte (Iteración 3).

<b>Historia de Usuario</b>	
Número:5	Usuario: Administrador del Hato Bovino
Nombre historia: Reporte General de datos de cada bovino	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada:3
Programador responsable: <ul style="list-style-type: none"> <li>• Murillo Loor Gema Emperatriz</li> <li>• Zambrano Rosado María Jimena</li> </ul>	
Descripción: El administrador puede generar reporte, imprimirlo.	
Observaciones: CONFIRMADO con el cliente	

### 3.2.2. FASE II: DISEÑO

Con los requerimientos obtenidos se procedió a crear la base de datos en Microsoft SQL Server 2008, ya que proporciona una plataforma productiva e inteligente. La base de datos (figura 4.1) consta de 21 tablas y están interrelacionadas entre sí según corresponde. Las entidades principales son: “Animal”, “Reproducción”, “Producción”, “Suplementación”, “HistorialClinico” y “Venta”. Y las características de las mismas se detallan en el diccionario de datos (Cuadro 3.16 - 3.36).



**Cuadro 3.16.** Diccionario de datos de la tabla “Animal”.

<b>TABLA: “Animal”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_animal	Int		PK	Código que identifica al animal
fecha_nacimiento	Date			Fecha en que nació el animal
Sexo	Varchar	(50)		Sexo del animal (Hembra o Macho)
Raza	Varchar	(50)		Nombre de la raza de donde proviene el animal
Peso	Varchar	(50)		Peso del animal
nombre_padre	Varchar	(50)		Nombre de la madre
nombre_madre	Varchar	(50)		Nombre de padre
Estado	Int		FK	Clave foránea que relaciona con la tabla “Estado”

**Cuadro 3.17.** Diccionario de datos de la tabla “Evento”.

<b>TABLA: “Evento”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_evento	Int		PK	Código que identifica el evento reproductivo o veterinario del animal
fecha_evento	Date			Fecha en que se ocurre un evento reproductivo o veterinario del animal
tipo_evento	Int		FK	Clave foránea que relaciona con la tabla “Tipo_evento”
Observación	Varchar	(50)		Opinión del evento efectuado

**Cuadro 3.18.** Diccionario de datos de la tabla “Producción”.

<b>TABLA: “Producción”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_produccion	Int		PK	Código que identifica el registro de producción láctea
fecha_produccion	Date			Fecha del ordeño de las vacas
litros_manana	Float			Números de litros de leche ordeñados en la mañana
litros_tarde	Float			Números de litros de leche ordeñados en la tarde.
cod_animal	Int		FK	Clave foránea que relaciona con la tabla “Animal”

**Cuadro 3.19.** Diccionario de datos de la tabla “Reproducción”.

<b>TABLA: “Reproducción”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
od_reproduccion	Int		PK	Código que identifica el registro de reproducción animal
fecha_parto	Date			Fecha de nacimiento de la cría
sexo_cria	Varchar			Sexo de la cría (macho o hembra)
tipo_fecundacion	Int		FK	Clave foránea que relaciona con la tabla “Tipo_fecundacion”
Tipo_parto	Varchar	(50)		Forma de parto (normal, cesárea, prematuro)
estado_cria	Int			Clave foránea que relaciona con la tabla “Estado”

**Cuadro 3.20.** Diccionario de datos de la tabla “HistorialClinico”.

<b>TABLA: “Historial Clínico”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_historial	Int		PK	Código que identifica el registro clínico.
enfermedad	Int		FK	Clave foránea que relaciona con la tabla “Enfermedad”
fármaco_utilizado	Int		FK	Clave foránea que relaciona con la tabla “Farmacos”
cantidad_dosis	Float			Cantidad de fármaco que se aplicó.
fecha_tratamiento	Date			Fecha del tratamiento del animal.
Resultado	Varchar	(50)		Qué se obtuvo después del tratamiento

**Cuadro 3.21.** Diccionario de datos de la tabla “Suplementacion”.

<b>TABLA: “Suplementacion”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_suplementacion	Int		PK	Código que identifica cada registro.
tipo_alimento	varchar	(50)		Fecha de nacimiento del animal
Cantidad	Int			Cantidad de alimento suministrado.
fecha_inicio	Date			Fecha en que inicia el tratamiento alimenticio.
fecha_fin	Date			Fecha en que termina el tratamiento alimenticio.

**Cuadro 3.22.** Diccionario de datos de la tabla “Venta”.

<b>TABLA: “Venta”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_animal	Int		PK	Código del animal
fecha_venta	Date			Fecha en la que se realizó la venta del animal.
Precio	Float			Valor de venta del animal.
Cliente	Varchar	(50)		Nombre del cliente.
id_cliente	Int			Identificación del cliente.
cod_animal	Int			Clave foránea que relaciona con la tabla Animal.

**Cuadro 3.23.** Diccionario de datos de la tabla “Enfermedad”.

<b>TABLA: “Enfermedad”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_enfermedad	Int		PK	Código que identifica cada enfermedad del animal
Nombre	Varchar	(50)		Nombre de una determinada enfermedad del animal
Síntomas	Varchar	(50)		Determina los síntomas de la respectiva enfermedad

**Cuadro 3.24.** Diccionario de datos de la tabla “Farmacos”.

<b>TABLA: “Farmacos”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_farmaco	Int		PK	Código que identifica cada fármaco
Nombre	Varchar	(50)		Nombre de un determinado fármaco utilizable en los bovinos
Dosis	Varchar	(50)		Cantidad de fármaco administrable a los bovinos

**Cuadro 3.25.** Diccionario de datos de la tabla “Tipo\_evento”.

<b>TABLA: “Tipo_evento”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_tipo_evento	Int		PK	Código que identifica cada tipo de evento
Nombre_evento	Varchar	(50)		Nombre de un determinado evento

**Cuadro 3.26.** Diccionario de datos de la tabla “Tipo\_fecundacion”.

<b>TABLA: “Tipo_fecundacion”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_tipo_fecundacion	Int		PK	Código del animal
Nombre_fecundacion	Varchar	(50)		Nombre del tipo de fecundación

**Cuadro 3.27.** Diccionario de datos de la tabla “Estado”.

<b>TABLA: “Estado”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_estado	Int		PK	Código que identifica cada estado del animal
nombre_estado	Varchar	(50)		Nombre del estado en que se encuentra el animal

**Cuadro 3.28.** Diccionario de datos de la tabla “Repro\_animal”.

<b>TABLA: “Repro_animal”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
Cod_reproduccion	Int		PK	Código que identifica cada registro de la tabla reproducción
cod_animal	Int			Código que identifica cada registro de la tabla animal

**Cuadro 3.29.** Diccionario de datos de la tabla “Suple\_animal”.

<b>TABLA: “Suple_animal”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_suple	Int		PK	Código que identifica cada registro de la tabla suplementación
cod_animal	Int			Código que identifica cada registro de la tabla animal

**Cuadro 3.30.** Diccionario de datos de la tabla “Historial\_animal”.

<b>TABLA: “Historial_animal”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_historial	Int		PK	Código que identifica cada registro de la tabla HistorialClinico
cod_animal	Int			Código que identifica cada registro de la tabla animal

**Cuadro 3.31.** Diccionario de datos de la tabla “Evento\_animal”.

<b>TABLA: “Evento_animal”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_evento	Int		PK	Código que identifica cada registro de la tabla evento
cod_animal	Int			Código que identifica cada registro de la tabla animal

**Cuadro 3.32.** Diccionario de datos de la tabla “Usuario”.

<b>TABLA: “Usuario”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_usuario	Int		PK	Código que identifica al usuario
Nombre	Varchar	(50)		Nombre del usuario
Apellido	Varchar	(50)		Apellido del usuario
Clave	Varchar	(50)		Contraseña del usuario

**Cuadro 3.33.** Diccionario de datos de la tabla “Ingreso\_sistema”.

<b>TABLA: “Ingreso_sistema”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
cod_ingreso	Int		PK	Código que identifica cada ingreso
fecha_ingreso	Date			Fecha de ingreso del usuario al sistema
hora_inicio	Time			Hora en que el usuario ingresó al sistema
hora_fin	Time			Hora en que el usuario salió del sistema
cod_usuario	Int			Código del usuario que ingresó al sistema

**Cuadro 3.34.** Diccionario de datos de la tabla “Tipo\_parto”.

<b>TABLA: “Tipo_parto”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
id_parto	Int		PK	Código que identifica cada parto
Nombre	varchar(50)			Nombre del tipo de parto

**Cuadro 3.35.** Diccionario de datos de la tabla “Raza”.

<b>TABLA: “Raza”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
id_raza	Int		PK	Código que identifica cada raza
Nombre	varchar(50)			Nombre de las razas.



**Cuadro 3.36.** Diccionario de datos de la tabla “Sexo”.

<b>TABLA: “Sexo”</b>				
<b>CAMPOS</b>	<b>TIPO DE DATOS</b>	<b>LONGITUD</b>	<b>CLAVE</b>	<b>DESCRIPCIÓN</b>
id_sexo	Int		PK	Código que identifica cada sexo (hembra y macho)
Nombre	varchar(50)			Nombre del sexo del (hembra y macho).

### **3.2.2.1. INTERFAZ DEL SOFTWARE DE CONTROL GANADERO**

El software de control ganadero se manejó con interfaces amigables que facilitan la interacción del administrador con el sistema. Al iniciar el sistema se carga una interfaz de inicio de sesión donde se debe escribir el usuario y contraseña correcta, luego se activa el menú ubicado en la parte superior mediante el componente Menustrip, el mismo que sirvió de enlace para acceder a los otros formularios. Cada formulario tenía adjuntados controles básicos como buttons, label, textbox, combobox, datagridview, entre otros que sirvieron de complemento para manipular eficientemente la información. Entre las ventanas principales del sistema de control ganadero se encuentran las siguientes: Acceso al sistema (Figura 4.2), ventana de menú (Figura 4.3), formulario de registro de bovinos (Figura 4.4), formulario de consulta (Figura 4.5) y reporte (Figura 4.6).

### **3.2.3. FASE III: DESARROLLO DEL CÓDIGO**

Durante todo el desarrollo del software de control ganadero se trabajó con programación en pares, las autoras trabajaron en un mismo ordenador coordinando de una manera eficaz y eficiente la elaboración del código e interfaz del mismo, disminuyendo errores y obteniendo un sistema de mejor calidad al trabajar con este tipo de programación.

El software fue codificado mediante programación modular, la cual permite realizar y probar cada módulo individualmente, en este caso se crearon los siguientes: conexión que es donde se indica la ruta de la base de datos, conectar, insertar, llenar, modificar y filtrar datos; se realizaron todos estos módulos para cada formulario como bovinos, reproducción, historial, eventos, entre otros.

**Llenar datos:** Se realizaron consultas en SQL Server 2008 R2 mediante el comando select, que llame los datos de una determinada tabla, para mostrar los registros en un datagridview.

**Conectar datos:** Se realizaron consultas en SQL Server 2008 R2 mediante el comando select, que llame los datos de acuerdo al registro seleccionado en el datagridview para mostrarlo en los textbox correspondientes.

**Insertar datos:** Las consultas realizadas en SQL Server 2008 R2 mediante el comando insert into, permitió guardar los datos en la tabla correspondientes.

**Modificar datos:** Para la modificación de los datos de acuerdo al registro seleccionado en el datagridview SQL Server 2008 R2 se utilizó el comando update.

**Filtrar datos:** La filtración de datos se realizó a través de consultas en SQL Server 2008 R2, mediante el comando select, para mostrar un registro determinado en el datagridview, siempre y cuando exista, de acuerdo al textbox de búsqueda.

#### **3.2.4. FASE IV: PRUEBAS**

Después de desarrollar los formularios y las funcionalidades del sistema se procedió a realizar las diferentes pruebas de software. En primer lugar se realizaron las pruebas unitarias, es decir que cada módulo funcione correctamente, para luego hacer una prueba en general logrando la aceptación del cliente, bajo estas condiciones:

- **Identificar errores introducidos por la combinación de programas probados unitariamente.**

En este punto fue donde se pudo corroborar que la conexión establecida entre la plataforma de desarrollo Visual Basic .net 2010 y SQL Server 2008 R2 había sido correctamente estructurada, permitiendo tener una interacción y manipulación correcta de la información sin presentarse errores al momento de realizar los diferentes ingresos y consultas.

- **Verificar que las interfaces y las aplicaciones funcionen correctamente.**

Se ingresó información ficticia de las entidades para realizar las pruebas en la cual se detectaron muchos errores que poco a poco se fueron corrigiendo. En las tablas donde se generaba una nueva tabla por la relación de varios a varios como por ejemplo entre la entidad “Animal” relacionada con “Eventos”, ocurría el error de que no podía almacenar dos tablas a la vez. Por lo cual se creó un distributivo para que no se presente este tipo de inconvenientes. Esta misma acción se realizó para todas las tablas que lo requerían.

- **Comprobar que las especificaciones de diseño sean alcanzadas.**

La interfaz gráfica resultó ser amigable y de fácil manipulación ya que contiene controles que le indican claramente para que sirven. Se optó por conseguir a una persona en particular, la cual fue capacitada de manera rápida para comprobar la facilidad del sistema. Los resultados fueron positivos ya que el usuario pudo realizar las tareas sin mayor dificultad.

- **Evidenciar cuáles son los requerimientos de implementación pertinentes para instalar el sistema.**

El objetivo de este contexto fue identificar el comportamiento de la aplicación informática bajo condiciones de sus recursos. Para el efecto se establecieron los requisitos mínimos que debe tener el equipo en donde se instalará el sistema, los cuales se detallan continuación:

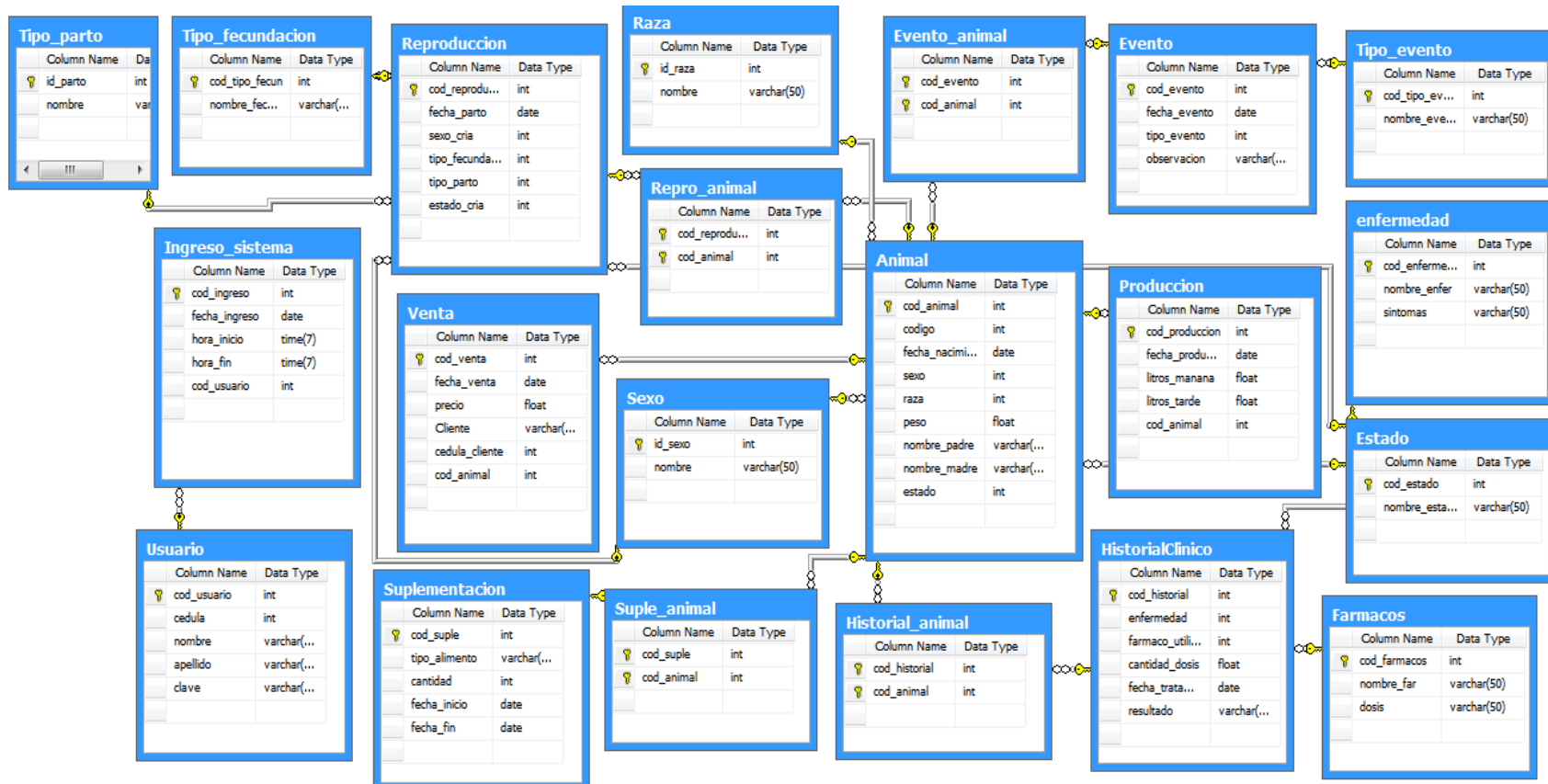
- ✓ Procesador arquitectura de 32 o 64 bits con Intel dual core a 2ghz o superior.
- ✓ Memoria de 2 Gb o más.
- ✓ Espacio libre en disco 10 Gb o más.
- ✓ Sistema operativo Windows Vista o superior.

## **CAPÍTULO IV. RESULTADOS Y DISCUSIÓN**

### **4.1. RESULTADOS**

Para el desarrollo del software se inició realizando visitas al hato bovino, donde se establecieron conversaciones con el representante (Anexo 01) y se planteó una entrevista (Anexo 02), lo cual permitió conocer los procesos de control ganadero que se llevan a cabo en el mismo, con la información adquirida se elaboró el diagrama de flujo de datos (figura 3.1). Así mismo se analizó las funcionalidades del sistema obteniendo como resultado las historias de usuario con la respectiva aprobación de la persona interesada, éstos se encuentran representados desde el cuadro 3.1 hasta cuadro 3.15.

Los requerimientos obtenidos sirvieron de referencia para la creación la base de datos (figura 4.1) del software de control ganadero, utilizando como gestor SQL server 2008 express R2.



**Figura 4.1.** Base de datos del Software de control ganadero del Hato Bovino de la ESPAM MFL

Seguidamente se diseñaron las interfaces gráficas del sistema las cuales resultaron de fácil manipulación para el administrador del mismo.



Figura 4.2. Ventana de inicio de sesión



Figura 4.3. Ventana del menú principal

Software Ganadero Infohato

Archivo Ingreso de datos Bovinos Consultas Reportes Usuario Ayuda

REGISTRO DE DATOS BOVINOS

Registros: 13

BUSCAR:   Todos

CÓDIGO NOMBRE	FECHA NACIMIENTO	SEXO	RAZA	PESO EN kg.	PADRE	MADRE	ESTADO	SITUACIÓN	EDAD AÑOS	CLASIFICACIÓN	PRECIO ESTIMADO	LOTE
98	03/06/2013	Macho	Jersey	5,12	107	103	Viva	Sano	0	Temero	50	Temeros_lactancia
105	28/11/2012	Hembra	Gyrolando	1	90	101	Viva	Sano	1	Temera	90	Temeros_lactancia
106	03/08/2011	Hembra	Hostein	14	100	102	Viva	Sano	2	Vacona	300	Produccion
107	20/01/2010	Hembra	Gyr	3,76	104	105	Viva	Sano	3	Vacona	600	Produccion
109	05/01/2011	Hembra	Gyr	11	108	90	Viva	Enfemo	2	Vacona	400	Produccion
110	04/11/2011	Hembra	Brown swiss	12	109	108	Viva	Sano	2	Vacona	300	Produccion
112	07/06/2013	Hembra	Brown swiss	12	90	102	Viva	Sano	0	Temera	50	Temeros_lactancia
120	06/06/2013	Hembra	Jersey	1	1000	103	Viva	Sano	0	Temera	50	Temeros_lactancia
130	10/06/2013	Hembra	Brown swiss	44	90	102	Viva	Sano	0	Temera	50	Temeros_lactancia
133	14/07/2011	Macho	Brown swiss	34	101	90	Viva	Sano	2	Torete	300	Temeros
134	23/12/2011	Hembra	Gyrolando	10,5	133	107	Viva	Sano	2	Temera	200	Reemplazo
140	24/06/2013	Hembra	Gyrolando	10,5	133	106	Viva	Sano	0	Temera	50	Temeros_lactancia
1000	15/07/2005	Hembra	Jersey	100	102	90	Viva	Sano	8	Vaca	600	Produccion

MUERTAS Y VENDIDAS

INGRESO DE DATOS

Código nombre: 98    Peso: 5,12 kg.    Situación: Sano

Fecha nacimiento: 03/06/2013    Raza: Jersey    Padre: 103

Sexo: Macho    Estado: Viva    Madre: 107

Lote: Temeros\_lactancia

0 Años/0 Meses/21 Días

ESPAM MFL HATO BOVINO    Fecha: 24/06/2013 2:31:45

Figura 4.4. Ventana de ingreso y modificación de datos principales de los bovinos.

Software Ganadero Infohato

Archivo Ingreso de datos Bovinos Consultas Reportes Usuario Ayuda

CONSULTAS DE DATOS BOVINOS

BUSCAR:   Todos

CÓDIGO NOMBRE	FECHA NACIMIENTO	SEXO	RAZA	PESO	PADRE	MADRE	ESTADO	SITUACIÓN	EDAD AÑOS	CLASIFICACIÓN	PRECIO ESTIMADO
90	14/02/2000	Hembra	Gyr	15	107	102	Vendida	Sano	13	Vaca	600
98	03/06/2013	Macho	Jersey	5,12	107	103	Viva	Sano	0	Temero	50
100	02/12/2012	Hembra	Hostein	10,86	101	90	Vendida	Recuperado	1	Temera	90
101	11/06/2013	Macho	Brown swiss	12,67	0	0	Vendida	Sano	0	Temero	50
102	03/06/2013	Macho	Jersey	10	0	0	Vendida	Sano	0	Temero	50
103	29/10/2012	Macho	Brown swiss	9	102	105	Aboto	Sano	1	Temero	90

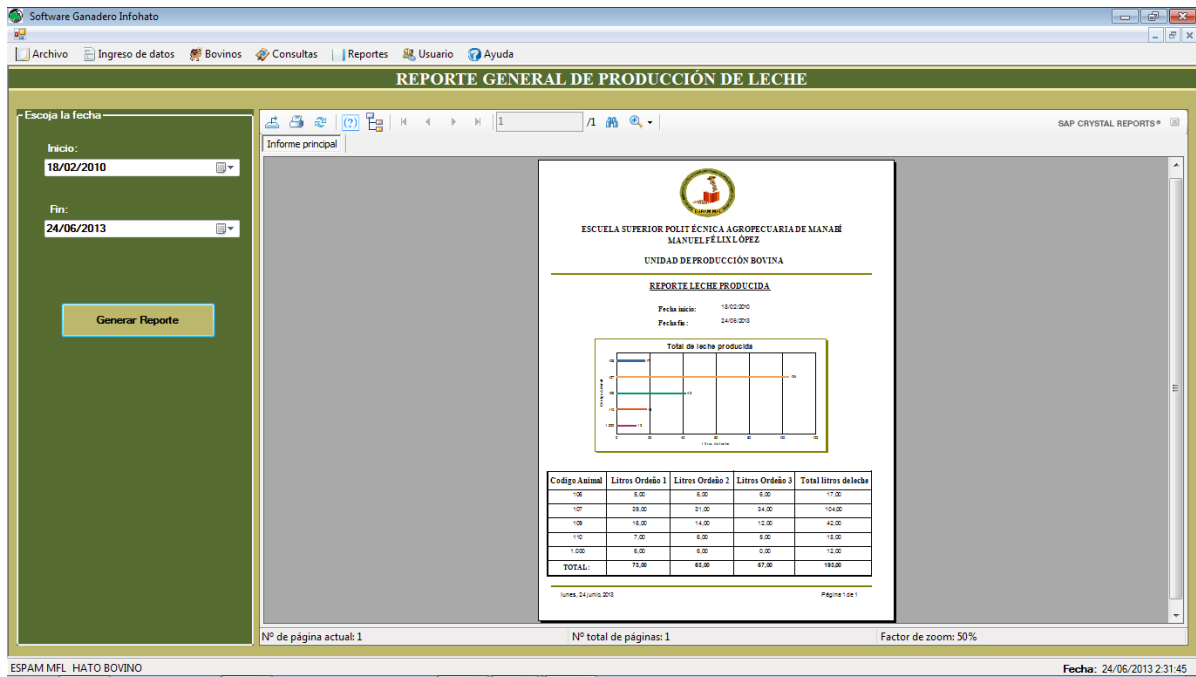
DATOS A VISUALIZAR

Reproducción  Producción  Suplementación  Historial Clínico  Eventos  Venta

ESPAM MFL HATO BOVINO    Fecha: 24/06/2013 2:31:45

Figura 4.5. Ventana de consulta de datos de los bovinos.





**Figura 4.6.** Ventana de reporte de datos de producción de los bovinos.

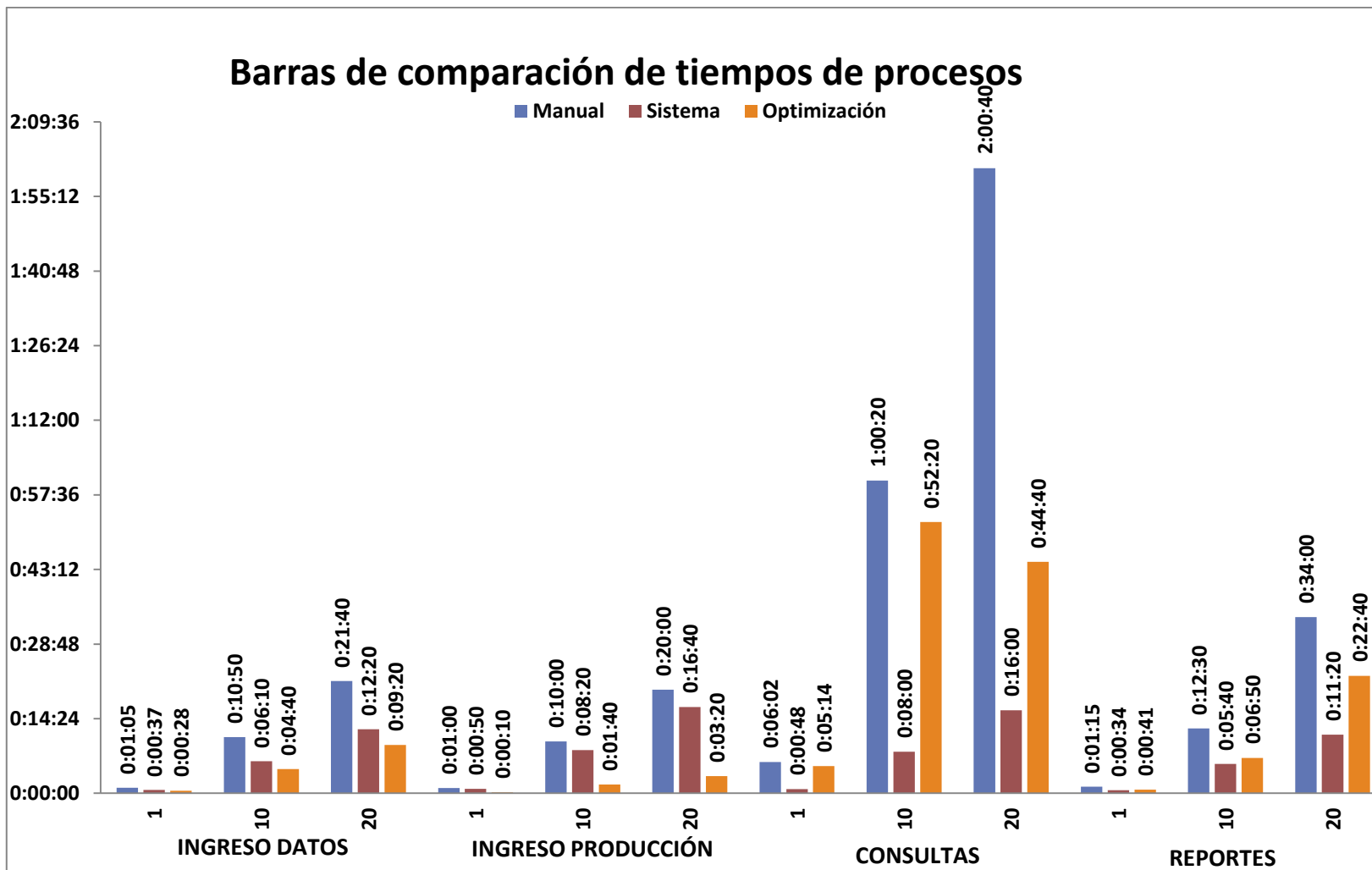
Luego se elaboró la programación de las funcionalidades que fueron desarrolladas en visual Basic .net 2010 (figura 4.2- 4.6), después de tener el software terminado se ejecutaron las pruebas necesarias para ultimar detalles.

Por medio de la implementación del sistema en el hato bovino de la ESPAM MFL se logró mejorar la eficiencia en los procesos que se realizan en esta institución dada la disminución de los tiempos utilizados en los procesos (ingresos, consulta y reportes de registros bovinos) que se detallan a continuación:

**Cuadro 4.1.** Pruebas de tiempos comparativos con respecto al uso manual y del sistema.

Proceso	Nº Registros	Tiempo Empleado		Optimización	
		Manual	Sistema	Minutos	Porcentaje
Ingreso de datos Bovinos	1	0:01:05	0:00:37	0:00:28	43%
	10	0:10:50	0:06:10	0:04:40	
	20	0:21:40	0:12:20	0:09:20	
Ingreso de producción lechera	1	0:01:00	0:00:50	0:00:10	16%
	10	0:10:00	0:08:20	0:01:40	
	20	0:20:00	0:16:40	0:03:20	
Consulta de actividades bovinos	1	0:06:02	0:00:48	0:05:14	87%
	10	1:00:20	0:08:00	0:52:20	
	20	2:00:40	0:16:00	0:44:40	
Generar reportes	1	0:01:15	0:00:34	0:00:41	55%
	10	0:12:30	0:05:40	0:06:50	
	20	0:34:00	0:11:20	0:22:40	

En comparación del tiempo empleado en realizar procesos manuales y con el sistema ganadero INFOHATO se obtuvieron los siguientes resultados de optimización. El 43% en el ingreso de datos de los bovinos, 16% en el ingreso de producción lechera, 85% en consultas de las actividades ganaderas y el 55% en generar reportes.



**Gráfico 4.1.** Estadística del tiempo invertido en el registro de procesos de los bovinos.

## 4.2. DISCUSIÓN

En toda institución la información es el recurso más apreciado, por lo cual es importante almacenarla de manera ordenada y tenerla accesible y disponible para quienes la utilizan. En la actualidad existen avances tecnológicos que sirven de apoyo para automatizar todo tipo de información. Haciendo uso de estas mejoras informáticas se desarrolló un sistema de información de la Finca ganadera San Antonio (SIFG) en Bogotá (Baracaldo, 2008), del cual se comprobó que contenía varias funcionalidades como ingreso de registros de los animales, pero no se pueden almacenar de forma detallada todos los procesos que se realizan en una finca, ya que contiene una base de datos sencilla de ocho entidades. Así mismo, las interfaces que se presentaban ante el usuario eran poco amigables y carentes de información concreta de lo que se requiere.

Otro sistema de referencia es “Análisis, Diseño, Implementación e Implantación de un Sistema en Ambiente Web, para la Administración Ganadera de La Hacienda The María” (Garzón y Rocha, 2012), del mismo se comprobó que contiene procesos de ingresos y reportes de datos bovinos, pero no se habían incluido opciones de búsqueda de ninguna actividad ganadera. Además los reportes son limitados mostrándose solo en el formulario sin acceso a imprimir.

Debido a ello las autoras tomaron como referencia los aspectos mencionados, y desarrollaron el software de control ganadero en el hato bovino de la ESPAM MFL con interfaces interactivas para el usuario, además cuenta con todas las entidades necesarias en la base de datos para almacenar de manera detallada los procesos que se realizan en esta institución desde el registro de animales, datos de reproducción, historial clínico, eventos veterinarios, venta y más. Otro aspecto importante de este sistema es que posee opciones de búsqueda, para que mediante filtros establecidos se obtenga de forma más eficiente la información. Así mismo posee la ventaja de generar reportes por cada evento ganadero, utilizando la herramienta Crystal Report, lo cual permite optimizar el rendimiento de los informes distribuidos en el entorno; de manera que se aprovechen las mejoras realizadas en el proceso. Estos reportes se pueden

imprimir para plasmar en un medio físico la información de las actividades que se realizan en el hato bovino.

Por otro lado, para un óptimo desarrollo de software se necesita una metodología informática adecuada, que permita facilitar la estructura de los métodos y procedimientos a utilizar. Pero SIFG no cuenta con una, utiliza un modelo de desarrollo UML (Lenguaje Unificado de Modelado), la cual le sirve sólo para representar las entidades con las que cuenta el sistema. Por el contrario el software de control ganadero del hato bovino de la ESPAM MFL desarrollada por las autoras, se basó en la metodología de desarrollo Extreme Programming (XP), la que permitió elaborar el sistema mediante fases disminuyendo errores y el tiempo de trabajo al momento de presentarlo al cliente.

Con las comparaciones realizadas se demostró que el software de control ganadero en el hato bovino de la ESPAM MFL, posee mayores ventajas frente a otros sistemas que realizan procesos similares, permitiendo la optimización del tiempo empleado en el registro y búsqueda de la información. De igual manera facilita la toma de decisiones al momento de realizar una actividad ganadera.

# **CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES**

## **5.1. CONCLUSIONES**

- La recolección de información de base de datos para la realización de un análisis exhaustivo, permitió plantear los diferentes requerimientos establecidos entre los directivos del hato bovino de la ESPAM MFL.
- La elaboración de una base de datos lógicamente organizada y estructura, facilitó la manipulación de datos permitiendo efectuar consultas, ingresos, actualizaciones, entre otros procesos.
- En la codificación del software se realizaron pruebas para optimizar las funcionalidades requeridas al ingresar los datos y en la manipular de los diferentes controles, localizando así como las falencias y corregirlas acertadamente.
- El software realizado permitió mejorar la gestión económica al reportar datos de producción lechera y valoración patrimonial del hato bovino.

## 5.2. RECOMENDACIONES

- Una vez recolectados los datos, se ordenen clasifiquen y se seleccionan los necesarios, para no tener exceso de información al momento de realizar el análisis de requerimiento.
- En el diseño de la base de datos se deben emplear tablas y campos relevantes, aplicando políticas de seguridad para conservar la integridad de la información.
- El desarrollo de una aplicación informática debe presentar una interfaz detallada de todos los procedimientos a realizar, para que personas inexpertas puedan ser uso de sus funcionalidades.
- Verificar un correcto funcionamiento del software de todas las especificaciones antes expuestas por el cliente, realizando pruebas constantes que ostenten su eficiencia antes de la entrega total.
- Emplear un computador con los requerimientos apropiados que permita el óptimo funcionamiento del sistema.

## BIBLIOGRAFÍA

- Abad, J. 2005. Pruebas de Software. (En línea). Consultado, 28 de Noviembre. Formato HTML. Disponible en <http://ing-sw.blogspot.com/>
- Arias, M. 2005. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. CR. InterSedes: Revista de las Sedes Regionales, Vol. 6. N° 10. p 1-2.
- Avilez, J; Escobar, P; Von Fabeck, G; Villagran, K; García, F; Matamoros, R. y García, A. 2010. Caracterización productiva de explotaciones lecheras empleando metodología de análisis multivariado. Maracaibo, VE. Revista Científica, FCV-LUZ. Vol. XX, N° 1. p 74 - 80.
- Balderrama, C. s.f. Introducción al lenguaje de programación. (En línea). Consultado, 15 de Noviembre. Formato PDF. Disponible en [http://www.sistemas.edu.bo/cbalderrama/sis%201100/PRESENTACION/programaci\\_n%20Modular.pdf](http://www.sistemas.edu.bo/cbalderrama/sis%201100/PRESENTACION/programaci_n%20Modular.pdf).
- Baracaldo, L. 2008. Sistema de información de la Finca ganadera San Antonio (SIFG). Tesis Ing. Informática. Corporación Universitaria Minutos de Dios. Bogotá, CO. p 29-40.
- Blanco, M. 2002. Programación en Visual Basic .Net. (En línea). Consultado, 16 de agosto 2012. Formato PDF. Disponible en <http://www.jp.unlugar.com/Visual.pdf>.
- Blanco, M; Malaver, M; Pezo, S. 2003. Manual práctico de ganadería: alimentación animal, sanidad animal, mejoramiento Ganadero. (En línea). Consultado, 28 de Octubre. Formato PDF. Disponible en <http://www.infolactea.com/descargas/biblioteca/142.pdf>.
- Calero, M. 2013. Una explicación de la programación extrema (XP). (En línea). Consultado, 18 de Noviembre. Formato PDF. Disponible en [www.willydev.net/descargas/prev/ExplicaXP.pdf](http://www.willydev.net/descargas/prev/ExplicaXP.pdf).
- Carrillo, I; Pérez, R. y Rodríguez, A. 2008. Metodología de Desarrollo del Software. (En línea). Consultado, 1 de Julio. Formato PDF. Disponible en [http://www.google.com.ec/url?sa=t&rct=j&q=metodologia+de+desarrollo+de+software&source=web&cd=3&sqi=2&ved=0CFoQFjAC&url=https%3A%2F%2Fsolusoftg11.googlecode.com%2Ffiles%2FMetodologias%2520de%2520desarrollo.pdf&ei=f3vwT7DjPIr8ATluLSHAg&usg=AFQjCNFqUa1Qgmx3GUFnb6D\\_\\_7UyTvPWoa&cad=rja](http://www.google.com.ec/url?sa=t&rct=j&q=metodologia+de+desarrollo+de+software&source=web&cd=3&sqi=2&ved=0CFoQFjAC&url=https%3A%2F%2Fsolusoftg11.googlecode.com%2Ffiles%2FMetodologias%2520de%2520desarrollo.pdf&ei=f3vwT7DjPIr8ATluLSHAg&usg=AFQjCNFqUa1Qgmx3GUFnb6D__7UyTvPWoa&cad=rja).
- DECSAI (Departamento de Ciencia de la Educación). s.f. Elementos de los diagramas de flujos de datos. (En línea). Consultado, 15 de Noviembre. Formato PDF. Disponible en <http://elvex.ugr.es/idbis/db/docs/design/X-DFDs.pdf>.



- Delgado E. 2008. Metodologías de desarrollo de software. Matanzas, CU. Revista de Arquitectura e Ingeniería, Vol. 2, N° 3.
- Díaz, J. 2009. Las metodologías ágiles como garantía de calidad del software. Madrid, ES. REICIS Revista Española de Innovación, Calidad e Ingeniería del Software. Vol. 5, N° 3. p 2-3.
- Dueñas, L. y Muñoz, C. 2001. Análisis del impacto socioeconómico del cooperativismo como una alternativa de desarrollo para la provincia de Manabí. (En línea). Consultado, 28 de Octubre. Formato PDF. Disponible en <http://www.dspace.espol.edu.ec/bitstream/123456789/3559/1/6086.pdf>.
- Escobar, R. y Lattal, A. 2010. Interfaz de bajo costo usando un puerto paralelo y Visual Basic. MX. Revista Mexicana de análisis de la conducta. Vol. 36, N° 3. p 3.
- ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López). 2010. ESPAM MFL, MAGAP E INIAP emprenden Proyecto Ganadero. (En línea). Consultado, 21 de Noviembre. Formato PDF. Disponible en <http://www.espam.edu.ec/index.php?id=ver&codigo=114>
- Espinoza, J; Palacios, A; Ávila, N; Guillén, A; De Luna, R; Ortega, R. y Murillo B. 2007. La ganadería orgánica, una alternativa de desarrollo pecuario para algunas regiones de México: una revisión. Caracas, VE. INTERCIENCIA. Vol. 32, N° 006.
- García, L y Aguilar, A. 2004. Explotación Lechera y Productividad: Tendencias Mundiales. Torreón, MX. Revista Mexicana de Agronegocios. Vol. 14 N° 014. p 8.
- Garzón, F. y Rocha, P. 2012. Análisis, Diseño, Implementación e Implantación de un Sistema en Ambiente Web, para la Administración Ganadera de La Hacienda The María. Tesis. Ing. en Sistemas. Universidad Politécnica Salesiana. Quito, EC. p 42-87.
- Gil, F; Albrigo J; Do Rosario J. 2005. Sistemas de Gestión de Base Datos. (En línea). Consultado, 15 de Noviembre. Formato PDF. Disponible en <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r61632.pdf>.
- Giraldo, G; Acevedo, J; Moreno, D. 2011. Una ontología para la representación de conceptos de diseño de software. Medellín. CO. Revista Avances en Sistemas e Informática, Vol. 8, N° 3. p 104.
- Giugni, M. y Loaiza, R. 2008. Metodología para el desarrollo de portales centrada en el usuario: una evaluación empírica. VE. Telematique, Vol. 7, N° 3. p 58.
- Hernández, J; Rebollar, S; González, F; Guzmán, E; Albarrán, B. y García, A. 2011. La cadena productiva de ganado bovino en el sur del estado de México. Torreón, MX. Revista Mexicana de Agronegocios. Vol. XV, N° 29. p 672-680.

- Iñiguez, F. s.f. Manejo reproductivo del hato ganadero. (En línea). Consultado, 28 de Octubre. Formato PDF. Disponible en <http://www.webveterinaria.com/virbac/news24/manejoreproductivo.pdf>.
- Joskowicz, J. 2008. Reglas y Prácticas en eXtreme Programming. (En línea). Consultado, 18 de Noviembre. Formato PDF. Disponible en <http://books.openlibra.com/pdf/XP%20-%20Jose%20Joskowicz.pdf>.
- Linares, J. y Geizzelez, M. 2007. Administración de proyectos en ingeniería del software. VE. Telos, Vol. 9, Nº. 1. p 27.
- López, L. s.f. Metodologías para la Enseñanza Aprendizaje de la Programación Estructurada y Orientada a Objetos. (En línea). Consultado, 30 de Octubre. Formato PDF. Disponible en [http://www.iiisci.org/journal/CV\\$/risci/pdfs/HPP102ES.pdf](http://www.iiisci.org/journal/CV$/risci/pdfs/HPP102ES.pdf).
- López, P. y Aroa, S. 2009. Administración de bases de datos con SQL Server 2008. (En línea). Consultado, 1 de Junio. Formato PDF. Disponible en: <http://www.luarna.com/Documentos%20compartidos/Ejemplos%20de%20lectura/Administraci%C3%B3n%20de%20bases%20de%20datos%20con%20SQL%20Server%202008%20%28ejemplo%29.pdf>.
- Lora, D; Sotto, P; Fernández, M; Fuentes, N. y Wong, M. 2011. Impacto de la aplicación de un software para la programación y control de los medios mecanizados en una unidad productora del municipio Güira de Melena. La Habana. CU. Revista Ciencias Técnicas Agropecuarias, Vol. 20, Nº 1. p 7.
- Martínez, J. y Díaz, J. 2010. Interfaz para la gestión de bases de datos temporales (IGBDT). Santiago de Cuba. CU. Ciencia en su PC, Vol. 1, Nº 1. p 48.
- Miranda, E. 2009. Prototipo de registro de y control de transacciones con bovinos ganadería HR. (En línea). Consultado, 1 de Junio. Formato PDF. Disponible en: [http://dspace.uniminuto.edu:8080/jspui/bitstream/10656/355/1/TTI\\_MirandaCamposLibardo\\_08.pdf](http://dspace.uniminuto.edu:8080/jspui/bitstream/10656/355/1/TTI_MirandaCamposLibardo_08.pdf).
- Murillo, F. 1990. Técnica de programación. (En línea). Consultado, 15 de Noviembre. Formato PDF. Disponible en: [http://blearning.itmina.edu.mx/dep/sada/carreras/Ingenieria%20Electronica/3er%20Semestre/Programacion%201/Programacion1\\_IE/unidad1.pdf](http://blearning.itmina.edu.mx/dep/sada/carreras/Ingenieria%20Electronica/3er%20Semestre/Programacion%201/Programacion1_IE/unidad1.pdf).
- Navarro, J. y Garzás, J. s.f. Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías ágiles y software libre. Madrid, ES. REICIS Revista Española de Innovación, Calidad e Ingeniería del Software, Vol. 6, Nº 1. p 6.
- Orjuela, A. y Rojas, M. 2008. Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. CO. Avances en Sistemas e Informática. Vol. 5, Nº 2, p 162.

- Otero, M. s.f. Elementos del diagrama de casos de uso. (En línea). Consultado, 15 de Noviembre. Formato PDF. Disponible en <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r33017.pdf>
- Peláez, P. s.f. Caracterización de los quesos con leche de cabra en la Isla de Tenerife. Influencia de factores ambientales en el desarrollo de sus características organolépticas y físico-químicas.(En línea). Consultado, 1 de Julio. Formato PDF. Disponible en [http://www.google.com.ec/url?sa=t&rct=j&q=&source=web&cd=4&ved=0CFQQFjAD&url=http%3A%2F%2Fwww.saltodelpastorcanario.org%2Fweb%2Fwp-content%2Fuploads%2F2011%2F09%2FLos-quesos-de-cabra-en-la-isla-de-Tenerife..pdf&ei=nXTwT9G\\_CYOs9ASH85W8DQ&usg=AFQjCNHCCbKM onnoG5ChPiF11qyTbOK2oQ&cad=rja](http://www.google.com.ec/url?sa=t&rct=j&q=&source=web&cd=4&ved=0CFQQFjAD&url=http%3A%2F%2Fwww.saltodelpastorcanario.org%2Fweb%2Fwp-content%2Fuploads%2F2011%2F09%2FLos-quesos-de-cabra-en-la-isla-de-Tenerife..pdf&ei=nXTwT9G_CYOs9ASH85W8DQ&usg=AFQjCNHCCbKM onnoG5ChPiF11qyTbOK2oQ&cad=rja).
- Prieto, E; González, M; Espitia, A. 2002. Restricción del amamantamiento en vacas del sistema cría libre. Montería, CO. Revista MVZ Córdoba. Vol. 7, N° 001. p 157-161.
- Quiroz, L. 2010. Diseño y Desarrollo de un Sistema Informático para el Control de Usuario de la Biblioteca de la Facultad de Ciencias Informáticas e Implementación de un Ambiente Tecnológico de Aprendizaje. (En línea). Consultado, 5 de junio 2012. Formato PDF. Disponible en [http://repositorio.utm.edu.ec/bitstream/123456789/44/1/TESIS%202010-06 14.pdf](http://repositorio.utm.edu.ec/bitstream/123456789/44/1/TESIS%202010-06%2014.pdf).
- Rayo, A. 2010. Visual Studio 2010. Novedades. (En línea).ES. Consultado, 16 de agosto 2012. Formato PDF. Disponible en [http://www.luarna.com/Documentos%20compartidos/Ejemplos%20de%20lectura/Visual%20Studio%202010.%20Novedades%20\(ejemplo\).pdf](http://www.luarna.com/Documentos%20compartidos/Ejemplos%20de%20lectura/Visual%20Studio%202010.%20Novedades%20(ejemplo).pdf).
- REDVET (Revista Electrónica de Veterinaria). 2006. Consideraciones sobre el ciclo de *Tenia saginata* en humanos y bovinos de la provincia Villa Clara. Cuba. Villa Clara, CU. REDVET. Vol. VII, N° 04.
- Rivera, J. 2011. Acceso a Datos con DataSets en Visual Web Developer 2008/2010. Aguascalientes, MX. Conciencia Tecnológica. Vol. 41, p 2.
- Rodríguez, P; Musat, D; Yagüe, A; Turhan, B; Rohunen, A; Kuvaja, P; Oivo, M. 2010. Adopción de metodologías ágiles: un estudio comparativo entre España y Europa. Madrid, ES. REICIS Revista Española de Innovación, Calidad e Ingeniería del Software. Vol. 6, N° 4. p 11,17.
- Ruíz, J. y Aguilera, O. 2007. Importancia de la Ingeniería de Software en la producción de software. Holguín, CU. Ciencias Holguín, Vol. XIII, N° 2. p 2.
- Salazar, G. 2009. Estimación de proyectos de software: un caso práctico.CO. Ingeniería y Ciencia, Vol. 5, N° 9. p 125.

- Salazar, I. y Vera, C. 2009. Análisis de la producción y comercialización de los productos lácteos de Indulac S.A. y su participación en las ventas del cantón Portoviejo. (En línea). Consultado, 29 de Octubre. Formato PDF. Disponible en: <http://repositorio.utm.edu.ec/bitstream/123456789/174/1/INDUSTRIAS%20LACTEAS28TESIS%29.pdf>.
- Sandoval, C. s.f. Ingeniería de software. (En línea). Consultado, 15 de Noviembre. Formato PDF. Disponible en <http://www.csandoval.net/files/intro.pdf>.
- Serralvo, M. y Acevedo, D. c2010. Sistema para la gestión de la información de los Proyectos agropecuarios en el sector ganadero. (en línea). Consultado, 15 de noviembre. Formato PDF. Disponible en <http://www.eumed.net/cursecon/ecolat/cu/2011/scac.htm>.
- Silva, D; Peña, M; Urdanet, F. 2010. Registros de control e indicadores de resultados en ganadería bovina de doble propósito. Revista Científica, FCV-LUZ. Vol. XX, N° 1. p 90.
- Tinoco, O; Rosales, P; Salas, J. 2010. Criterios de selección de metodologías de desarrollo de software. Lima, PE. Industrial Data. Vol. 13, N° 2. p 70.
- Valdez, G; Torres, G; Sánchez, A; Paz, R; Vázquez, M. y Pardo, D. 2011. Acortando la brecha digital para la trazabilidad sanitaria: el problema de la transferencia tecnológica en la ganadería sonorensis, caso SITAGAN. Hermosillo, MX. Estudios Sociales. Vol. 19, N° 37. p 10,11 y 15.
- Yagüe, A y Garbajosa, J. 2009. Comparativa práctica de las pruebas en entornos tradicionales y ágiles, REICIS Revista Española de Innovación, Calidad e Ingeniería del Software. Vol. 5, N° 4. p 3, 8 y 9.
- Zapata, C; Tamayo, P; Arango, F. 2007. Conversión de esquemas preconceptuales a diagrama de casos de uso empleando atom3. Medellín, CO. Dyna. Vol. 74, N° 153. p 238 y 239.
- Zuloaga, L. s.f. Diagrama de flujo de datos (DFD). (En línea). Consultado, 15 de Noviembre. Formato PDF. Disponible en <http://www.galeon.com/zuloaga/Doc/ADS04.pdf>.

# **ANEXOS**

**ANEXO 1**  
**CONVERSACIÓN CON EL COORDINADOR DEL HATO BOVINO**  
**DR. RAÚL GUEVARA**



**ANEXO 1-A ESTABLECIENDO REQUERIMIENTO**



**ANEXO 1-B MOSTRANDO EL FUNCIONAMIENTO DEL SOFTWARE.**

**ANEXO 2**

**ENTREVISTA AL COORDINADOR DEL HATO BOVINO  
DR. RAÚL GUEVARA**



## ENTREVISTA

Esta entrevista se realizó con el objetivo de obtener los requerimientos necesarios para el desarrollo del software ganadero INFOHATO de la ESPAM MFL, así mismo para determinar la importancia de la implementación de este sistema en la institución mencionada.

- **¿CUÁL ES LA CAUSA PRINCIPAL PARA QUE DESEE IMPLEMENTAR UN SOFTWARE DE CONTROL EN EL HATO BOVINO?**

La causa principal para implementar un software de control ganadero es la obtención de un sistema productivo de información precisa y confiable que permita una mejor toma de decisiones a la hora de evaluar procesos, además como una base de información, investigación y acciones que se conducen en una organización.

- **¿CÓMO CONTROLA LOS DATOS EN EL HATO BOVINO?**

Los datos en el hato bovino se controlan diariamente llevando un registro manual de todos los datos que se generan, como la producción de leche, salud, suplementación entre otras. Proceso que lo realiza la el coordinador en conjuntos con demás docentes en el hato.

- **¿QUÉ NECESIDADES DEBE CUMPLIR EL SISTEMA?**

Entre las necesidades a cumplir por el sistema está la posibilidad de dar informes diarios, mensuales y anuales de los procesos llevados en el hato como producción animal, incidencia de enfermedades, rendimientos de leche y consumo de alimento entre otros.

- **¿QUIÉNES O QUIEN MANIPULAN LOS DATOS DE LOS BOVINOS?**

La persona responsable de manipular los datos de los bovinos es la Dr. Ana María quien tiene accesos a la información individualizada de los animales del hato.

**ANEXO 3**  
**MANUAL DE USUARIO**



# MANUAL DE USUARIO

SOFTWARE DE CONTROL  
GANADERO INFOHATO

## ÍNDICE

INTRODUCCIÓN .....	88
1. ACCESO AL SISTEMA .....	88
2. COMPONENTES DE LA VENTANA.....	89
2.1. MENÚ PRINCIPAL.....	89
2.1.1. MENÚ INGRESO DE DATOS .....	90
2.2 MENÚ BOVINOS .....	91
2.2.1 OPCIÓN LOTE .....	91
2.2.2. REPRODUCCIÓN.....	93
2.2.3 PRODUCCIÓN.....	94
2.2.4 SUPLEMENTACIÓN .....	94
2.2.5 HISTORIAL CLÍNICO.....	95
2.2.6 EVENTOS .....	95
2.2.7 VENTAS.....	96
2.2.8 PASTOREO .....	96
2.3 MENÚ CONSULTAS.....	97
2.4 REPORTE .....	98
2.5 USUARIO.....	100
2.6 AYUDA.....	100

## INTRODUCCIÓN

El manual de usuario indica los pasos para manipular adecuadamente la información del registro de producción y reproducción de los bovinos. El sistema INCOATE posee algunos formularios que facilitan el ingreso de los datos de los bovinos y permite imprimir reportes de los mismos.

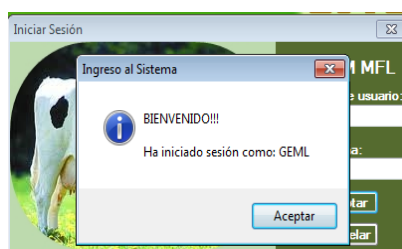
El software maneja principalmente los procesos de registros de bovinos cada uno identificado por un código, así mismo se guardarán los datos de reproducción, producción lechera diaria, historial clínico y eventos veterinarios, de los cuales se realizarán consultas que ayudarán en la toma de decisiones, al momento de realizar actividades bovinas y aplicar ciencias expertas. A continuación se detallan las funcionalidades del software de control ganadero.

### 1. ACCESO AL SISTEMA

Muestra la ventana de inicio de sesión, el cual permite dos accesos de administrador y otro. Para ambos casos debe conocer su usuario y contraseña para poder ingresar al sistema.



En caso de ser incorrecto el ingreso se muestra un mensaje de error, caso contrario se visualizará el mensaje de bienvenida y se activará el menú para realizar los procesos requeridos.



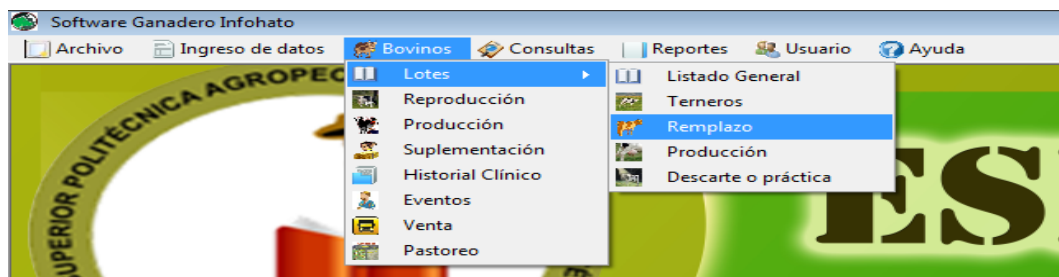
## 2. COMPONENTES DE LA VENTANA

### 2.1. MENÚ PRINCIPAL



- ❖ Archivo
- ❖ Ingreso de datos
- ❖ Bovinos
- ❖ Consultas
- ❖ Reportes
- ❖ Usuario
- ❖ Ayuda

Para tener acceso solo de clic en el menú que desee y rápidamente se desplegarán las opciones que contiene cada uno de ellos.



### 2.1.1. MENÚ INGRESO DE DATOS

Se encuentra en el menú principal del sistema, dentro de ella se hallan las opciones de ingresar enfermedades, tipo de fecundación, tipo de parto, raza, potreros e insumos.



Al ingresar a cualquiera de ellas se activa un cuadro de diálogo donde muestra un formulario con acciones de nuevo, editar, cancelar y guardar los datos requeridos.



## 2.2 MENÚ BOVINOS

Esta es la tercera pestaña del Menú Principal, aquí encontramos opciones de ingreso de bovinos por lotes, reproducción, producción, suplementación, historial clínico, eventos, ventas y pastoreo.



### 2.2.1 OPCIÓN LOTE

Este menú a la vez contiene: listado general, terneros, reemplazo producción y descarte, al escoger cualquiera de estas opciones se muestra un formulario, allí encontramos un datagridview que nos muestra los registros ingresados y adicional el precio estimado de acuerdo a la edad, así mismo en la parte superior de los botones se destaca el año, mes y día exacto de vida de cada animal, a continuación se describen los botones:

- ❖ Nuevo: Permite ingresar un nuevo registro con los datos principal del bovino.

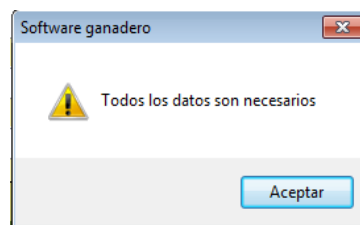
- ❖ Editar: Permite modificar los datos del bovino que se han ingresado incorrectamente, o corregir un error de ortografía al momento de escribir los datos.



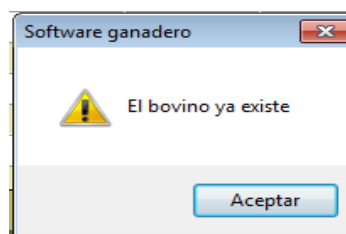
**INGRESO DE DATOS**

Código nombre:	98	Peso:	5,12 kg.	Situación:	Sano
Fecha nacimiento:	03/06/2013	Raza:	Jersey	Padre:	103
Sexo:	Macho	Estado:	Viva	Madre:	107
0 Años/0 Meses/21 Días				Lote:	Temeros_lactancia

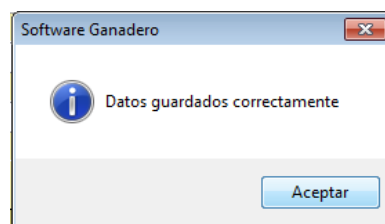
- ❖ Cancelar: Cancela la operación que se está realizando.
- ❖ Guardar: Guarda los datos generados en el formulario dentro de la base de datos, en caso de no llenar todos los campos se mostrar un mensaje de error, en cual indica que se debe llenar todos los campos.



Si por error usted ingresa el mismo código de un bovino se visualiza un mensaje que ya existe el registro.



Caso contrario si ingresa los datos correctamente aparecerá el siguiente mensaje.



Además existe un filtro para la búsqueda rápida de información necesaria que el usuario requiera de uno o varios bovinos, esta búsqueda se puede realizar por cualquier categoría del registro. Así mismo contiene una casilla de verificación el cual muestra el ganado vendido y muerto.

MUERTAS Y VENDIDAS

## 2.2.2. REPRODUCCIÓN

Software Ganadero Infohato

Archivo Ingreso de datos Bovinos Consultas Reportes Usuario Ayuda

REGISTRO DE LA REPRODUCCIÓN GANADERA

Registros: 6

BUSCAR:  Todos

ANIMAL	FECHA PARTO	SEXO	TIPO FECUNDACIÓN	TIPO PARTO	ESTADO	NÚMERO DE PARTO
106	04/06/2013	Hembra	Inseminacion	Cesárea	Viva	1
107	25/04/2012	Macho	Normal	Cesárea	Viva	1
107	30/05/2013	Hembra	Inseminacion	Normal	Viva	2
109	25/03/2013	Hembra	Normal	Normal	Viva	0
110	16/05/2013	Macho	Inseminacion	Cesárea	Viva	0
1000	10/06/2013	Hembra	Normal	Cesárea	Viva	0

INGRESO DE DATOS

Animal:

Fecha parto: 04/06/2013

Tipo fecundación: Inseminacion

Tipo parto: Cesárea

Sexo cria: Hembra

Estado cria: Viva

MIENÚ

NUEVO

EDITAR

CANCELAR

GUARDAR

ESPAM MFL HATO BOVINO

Fecha: 24/06/2013 2:31:45

En este formulario se muestran los datos de la reproducción ganadera que existe en el hato, con las opciones de nuevo, editar, cancelar y guardar, además de un filtro donde se agiliza la búsqueda del bovino que se desee mostrar, esta búsqueda se realiza por código y fecha de parto. También le brinda la posibilidad de ingresar un nuevo tipo de parto y de fecundación, mediante los botones que están a la derecha del campo correspondiente.

Tipo fecundación:

Tipo parto:

## 2.2.3 PRODUCCIÓN

En este formulario se muestran los datos de la producción lechera que diariamente se produce, existe las opciones de nuevo, editar, cancelar y guardar, además de un filtro donde se agiliza la búsqueda del bovino que se desee mostrar.

**REGISTROS DE LA PRODUCCIÓN GANADERA**

Registros: 11

ANIMAL	FECHA PRODUCCIÓN	HORA	LITROS ORDENO 1	HORA	LITROS ORDENO 2	HORA	LITROS ORDENO 3	TOTAL	DIAS LACTANCIA	ALIMENTO ENTREGADO	CANTIDAD ALIMENTO
109	31/05/2013	07:00	7	01:00	6	06:00	5	18	67	Pasto verde	10.25
107	31/05/2013	08:00	6	01:00	5	06:00	4	15	1	Pasto verde	10.5
109	01/06/2013	07:00	9	01:00	8	06:00	7	24	68	Pasto verde	10.9
107	01/06/2013	08:00	6	01:00	5	06:00	4	15	2	Pasto verde	10.5
107	10/06/2013	07:00	7	12:00	6	16:00	5	18	11	pasto	3.4
107	11/06/2013	07:00	5	12:00	5	17:00	6	16	12	pasto	4
107	12/06/2013	07:00	7	12:00	4	16:00	8	19	13	pasto	3
107	13/06/2013	07:00	8	12:00	6	16:00	7	21	14	pasto	3
106	17/06/2013	07:00	5	12:00	6	17:00	6	17	13	pasto	2
110	17/06/2013	07:00	7	12:00	6	17:00	5	18	32	pasto	4
1000	17/06/2013	07:00	6	12:00	6	00:00	0	12	7	pasto	3

**INGRESO DE DATOS**

Animal: 109 Fecha producción: 31/05/2013

Hora ordeño 1: 07:00 Litros ordeño 1: 7 t. Días lactancia: 67  
 Hora ordeño 2: 01:00 Litros ordeño 2: 6 t. Alimento entregado: Pasto verde  
 Hora ordeño 3: 06:00 Litros ordeño 3: 5 t. cantidad alimento: 10.25

NUEVO EDITAR CANCELAR GUARDAR

ESPAM MFL HATO BOVINO Fecha: 24/06/2013 2:31:45

## 2.2.4 SUPLEMENTACIÓN

En este formulario se muestran los datos de la suplementación que se administra diariamente a cada animal, existe las opciones de nuevo, editar, cancelar y guardar, además de un filtro donde se agiliza la búsqueda del bovino que se desee mostrar.

**REGISTROS DE LA SUPLEMENTACIÓN ALIMENTICIA**

Registros: 8

ANIMAL	CANTIDAD DIARIA	FECHA INICIO	FECHA FIN
90	0	31/05/2013	27/05/2013
90	0	31/05/2013	27/05/2013
90	0	02/06/2013	29/04/2013
90	0	31/05/2013	27/05/2013
90	0	02/06/2013	02/06/2013
90	0	02/06/2013	01/06/2013
100	16	03/06/2013	21/06/2013
90	0	06/06/2013	07/06/2013

**INGRESO DE DATOS**

Animal: 90 Fecha Inicio: 31/05/2013

Cantidad diaria: 0 kg. Fecha Fin: 27/05/2013

NUEVO EDITAR CANCELAR GUARDAR

ESPAM MFL HATO BOVINO Fecha: 24/06/2013 2:31:45

## 2.2.5 HISTORIAL CLÍNICO

En este formulario se muestran los datos del historial clínico de cada animal se especifica la enfermedad y el fármaco utilizado, existe las opciones de nuevo, editar, cancelar y guardar, además de un filtro donde se agiliza la búsqueda del bovino que se desee mostrar.

The screenshot displays the 'REGISTROS DEL HISTORIAL CLÍNICO' window. At the top, there is a search bar and a 'Todos' button. Below it is a table with the following data:

ANIMAL	ENFERMEDAD	FÁRMACO UTILIZADO	CANTIDAD DOSIS	MEDIDA	FECHA TRATAMIENTO	OBSERVACION
90	AftosaA	Oxibetraciclina	5.00	cm.	03/06/2013	aaa
100	rabies	paracetamol	10.00	ml.	03/06/2013	# g hhhhhhhhhh
101	AftosaA	Oxibetraciclina	3.00	ml.	06/06/2013	ssssss

Below the table is the 'INGRESO DE DATOS' form with the following fields:

- Animal: 90
- Enfermedad: AftosaA
- Fecha tratamiento: 03/06/2013
- Fármaco utilizado: Oxibetraciclina
- Cantidad dosis: 5.00
- Posología: cm.
- Observación: aaa

At the bottom of the form are buttons for 'NUEVO', 'EDITAR', 'CANCELAR', and 'GUARDAR'. The status bar at the bottom indicates 'ESPAM MFL HATO BOVINO' and 'Fecha: 24/06/2013 2:31:45'.

## 2.2.6 EVENTOS

En este formulario se muestran los datos de los eventos que se realizan en el hato bovino por cada animal, existe las opciones de nuevo, editar, cancelar y guardar, además de un filtro donde se agiliza la búsqueda del bovino que se desee mostrar.

The screenshot displays the 'REGISTROS DE EVENTOS VETERINARIOS' window. At the top, there is a search bar and a 'Todos' button. Below it is a table with the following data:

ANIMAL	FECHA EVENTO	TIPO EVENTO	OBSERVACIÓN
90	01/02/2012	Desparasitacion	AAAAAAAAAuhh
100	01/02/2012	Desparasitacion	SSSSSSSSSS
90	20/03/2013	Descome	HHHHHH
107	10/08/2012	Inseminación Artificial	Primera inseminación

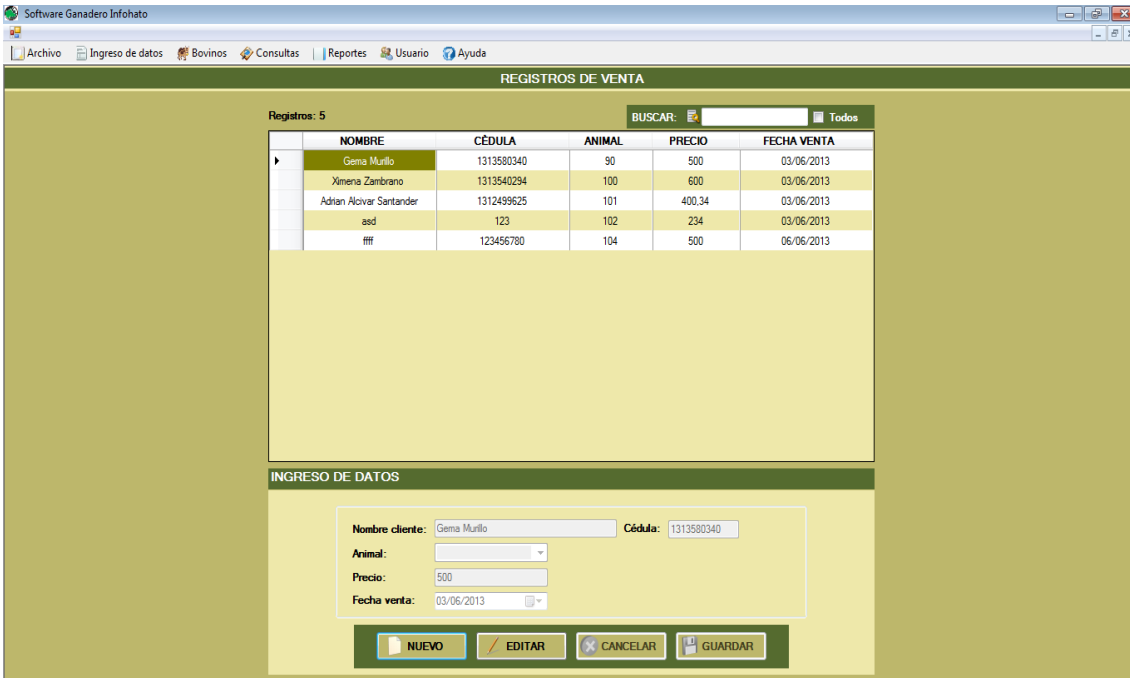
Below the table is the 'INGRESO DE DATOS' form with the following fields:

- Animal: 90
- Tipo de evento: Desparasitacion
- Fecha evento: 01/02/2012
- Observación: AAAAAAAAAuhh

At the bottom of the form are buttons for 'NUEVO', 'EDITAR', 'CANCELAR', and 'GUARDAR'. The status bar at the bottom indicates 'ESPAM MFL HATO BOVINO' and 'Fecha: 24/06/2013 2:31:45'.

## 2.2.7 VENTAS

En este formulario se muestran los registros que definen la venta de un determinado bovino, posee las opciones de nuevo, editar, cancelar y guardar, además de un filtro donde se agiliza la búsqueda del bovino que se desee mostrar. Al ingresar una nueva venta en el combo animal se cargan sólo los animales que no han sido vendidos.



The screenshot shows the 'Software Ganadero Infohato' application window. The main title is 'REGISTROS DE VENTA'. Below the title, there is a search bar labeled 'BUSCAR:' and a 'Todos' button. The table below shows 5 records with the following data:

NOMBRE	CÉDULA	ANIMAL	PRECIO	FECHA VENTA
Gema Murillo	1313580340	90	500	03/06/2013
Ximena Zambrano	1313540294	100	600	03/06/2013
Adrian Alviar Santander	1312499625	101	400,34	03/06/2013
asd	123	102	234	03/06/2013
###	123456780	104	500	06/06/2013

Below the table is the 'INGRESO DE DATOS' section, which includes a form with the following fields:

- Nombre cliente: Gema Murillo
- Cédula: 1313580340
- Animal: (dropdown menu)
- Precio: 500
- Fecha venta: 03/06/2013

At the bottom of the form are four buttons: NUEVO, EDITAR, CANCELAR, and GUARDAR.

## 2.2.8 PASTOREO

En este formulario se muestran los datos de pastoreo, es decir se indica en que potrero se encuentra cada lote bovino en la fecha determinada, existe las opciones de nuevo, editar, cancelar y guardar, además de un filtro donde se agiliza la búsqueda del bovino que se desee mostrar.

Software Ganadero Infohato

Archivo Ingreso de datos Bovinos Consultas Reportes Usuario Ayuda

### REGISTRO DE ESTADÍA DE LOS BOVINOS

Registros: 1 BUSCAR:  Todos

	FECHA INGRESO	UNIDAD ANIMAL	PASTO	ALTURA PASTO INGRESO	ALTURA PASTO SALIDA	POTRERO	DÍAS DE OCUPACIÓN	DÍAS DE RECUPERACIÓN	TEMP. °C	H.R. %	PRECT. ml.	ÁREA EN HECTÁREA	CANTIDAD PASTO (kg/ha)
▶	02/02/2013	Temeros_lactancia	sss	3	2	UPF 1	1	3	25	5	3	1	1
	02/02/2013	Temeros	www	3	2	UPF 10	1	5	25	5	4	1	1
	03/02/2013	Reemplazo	dd	1	1	UPF 20	1	1	1	1	1	11	1

#### INGRESO DE DATOS

Fecha ingreso:	<input type="text" value="02/02/2013"/>	Días ocupación:	<input type="text" value="1"/>
Unidad Animal:	<input type="text" value="Temeros_lactancia"/>	Días de recuperación:	<input type="text" value="3"/>
Especie de pasto:	<input type="text" value="sss"/>	Temp. °C :	<input type="text" value="25"/>
Altura de pasto ingresar:	<input type="text" value="3"/>	H.R. %:	<input type="text" value="5"/>
Altura de pasto salida:	<input type="text" value="2"/>	Prect. ml.:	<input type="text" value="3"/>
Potrero:	<input type="text" value="UPF 1"/>	Área:	<input type="text" value="1"/>
		Cantidad pasto:	<input type="text" value="1"/>

ESPAM MFL HATO BOVINO Fecha: 24/06/2013 2:31:45

## 2.3 MENÚ CONSULTAS

Al escoger consulta se visualiza un formulario con la Datagridview donde se cargan los datos de cada animal, luego seleccionando un determinado animal, y una opción en el menú de procesos, se muestran los datos respectivos en otra grilla; cuenta además con un filtro que agiliza la búsqueda del registro que se requiera.

Software Ganadero Infohato

Archivo Ingreso de datos Bovinos Consultas Reportes Usuario Ayuda

### CONSULTAS DE DATOS BOVINOS

BUSCAR:  Todos

CÓDIGO NOMBRE	FECHA NACIMIENTO	SEXO	RAZA	PESO	PADRE	MADRE	ESTADO	SITUACION	EDAD AÑOS	CLASIFICACIÓN	PRECIO ESTIMADO
90	14/02/2000	Hembra	Gyr	15	107	102	Vendida	Sano	13	Vaca	600
98	03/06/2013	Macho	Jersey	5.12	107	103	Viva	Sano	0	Temero	50
100	02/12/2012	Hembra	Hostein	10.86	101	90	Vendida	Recuperado	1	Temera	90
101	11/06/2013	Macho	Brown swiss	12.67	0	0	Vendida	Sano	0	Temero	50
102	03/06/2013	Macho	Jersey	10	0	0	Vendida	Sano	0	Temero	50
103	29/10/2012	Macho	Brown swiss	9	102	105	Aborto	Sano	1	Temero	90

**DATOS A VISUALIZAR**  
 Reproducción
  Producción
  Suplementación
  Historial Clínico
  Eventos
  Venta

ESPAM MFL HATO BOVINO Fecha: 24/06/2013 2:31:45

## 2.4 REPORTE



En el menú reporte animal se mostrará un formulario con los datos generales de los bovinos, donde se genera el reporte respectivo de acuerdo a la búsqueda por cualquier campo que tiene cada animal.

CÓDIGO NOMBRE	FECHA NACIMIENTO	SEXO	RAZA	PESO	PADRE	MADRE	ESTADO	SITUACION	EDAD AÑOS	CLASIFICACIÓN	PRECIO ESTIMADO
50	14/02/2000	Hembra	Gyr	15	107	102	Vendida	Sano	13	Vaca	600
90	03/06/2013	Macho	Jersey	5.12	107	103	Viva	Sano	0	Temero	50
100	02/12/2012	Hembra	Hoaltein	10.86	101	90	Vendida	Recuperado	1	Temero	90
101	11/06/2013	Macho	Brown swiss	12.67	0	0	Vendida	Sano	0	Temero	50
102	03/06/2013	Macho	Jersey	10	0	0	Vendida	Sano	0	Temero	50
103	29/10/2012	Macho	Brown swiss	9	102	105	Aborto	Sano	1	Temero	90
104	27/02/2009	Macho	Brown swiss	8	103	105	Vendida	Sano	4	Toro	600
105	28/11/2012	Hembra	Gyrolando	1	90	101	Viva	Sano	1	Temera	90
106	03/08/2011	Hembra	Hoaltein	14	100	102	Viva	Sano	2	Vacona	300
107	20/01/2010	Hembra	Gyr	3.76	104	105	Viva	Sano	3	Vaca	600
108	24/11/2005	Macho	Hoaltein	7	106	103	Muerta	Sano	8	Toro	600
109	05/01/2011	Hembra	Gyr	11	108	90	Viva	Enfermo	2	Vacona	400
110	04/11/2011	Hembra	Brown swiss	12	109	108	Viva	Sano	2	Vacona	300
112	07/06/2013	Hembra	Brown swiss	12	90	102	Viva	Sano	0	Temera	50
120	06/06/2013	Hembra	Jersey	1	1000	103	Viva	Sano	0	Temera	50
130	10/06/2013	Hembra	Brown swiss	44	90	102	Viva	Sano	0	Temera	50
133	14/07/2011	Macho	Brown swiss	34	101	90	Viva	Sano	2	Torete	300
134	23/12/2011	Hembra	Gyrolando	10.5	133	107	Viva	Sano	2	Temera	200
140	24/06/2013	Hembra	Gyrolando	10.5	133	106	Viva	Sano	0	Temera	50
1000	15/07/2005	Hembra	Jersey	100	102	90	Viva	Sano	8	Vaca	600

REPORTE DE BOVINOS

Informe principal

ESCAM MFL

ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

HATO BOVINO

REPORTE DE REGISTROS BOVINOS

CODIGO ANIMAL	FECHA NACIMIENTO	SEXO	RAZA	PADRE	MADRE	CLASIFICACIÓN	VALOR
98	03/06/2013	Macho	Jersey	107	103	Temera	50,00
105	28/11/2012	Hembra	Gyrlando	90	101	Temera	90,00
106	03/08/2011	Hembra	Holstein	100	102	Vacana	300,00
107	20/01/2010	Hembra	Gyr	104	105	Vaca	600,00
109	05/01/2011	Hembra	Gyr	108	90	Vacana	400,00
110	04/11/2011	Hembra	Brown swiss	109	108	Vacana	300,00
112	07/06/2013	Hembra	Brown swiss	90	102	Temera	50,00
120	06/06/2013	Hembra	Jersey	1000	103	Temera	50,00
130	10/06/2013	Hembra	Brown swiss	90	102	Temera	50,00
133	14/07/2011	Macho	Brown swiss	101	90	Temera	300,00
134	23/12/2011	Hembra	Gyrlando	133	107	Temera	200,00
140	24/06/2013	Hembra	Gyrlando	133	106	Temera	50,00
1000	15/07/2005	Hembra	Jersey	102	90	Vaca	600,00
Registros: 13	TOTAL INVERTIDO						3840,00

Nº de página actual: 1      Nº total de páginas: 1      Factor de zoom: 75%

En el menú reporte producción se mostrará un formulario donde se puede escoger la fecha inicio y fin de la producción lechera, de todos los bovinos que se encuentren en ese rango, los resultados se generarán en forma individual y grupal.

Software Ganadero Infohato

Archivo Ingreso de datos Bovinos Consultas Reportes Usuario Ayuda

REPORTE GENERAL DE PRODUCCIÓN DE LECHE

Escoja la fecha

Inicio: 18/02/2010

Fin: 24/06/2013

Generar Reporte

Informe principal

ESCAM MFL

ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

UNIDAD DE PRODUCCIÓN BOVINA

REPORTE LECHE PRODUCIDA

Fecha inicio: 18/02/2010  
Fecha fin: 24/06/2013

Título de leche producida

Código Animal	Litros Ordeno 1	Litros Ordeno 2	Litros Ordeno 3	Total litros de leche
106	5,00	6,00	6,00	17,00
107	33,00	21,00	24,00	78,00
108	13,00	14,00	12,00	39,00
110	7,00	6,00	6,00	19,00
1000	6,00	6,00	6,00	18,00
TOTAL:	74,00	69,00	67,00	210,00

lunes 24 junio 2013      Página 1 de 1

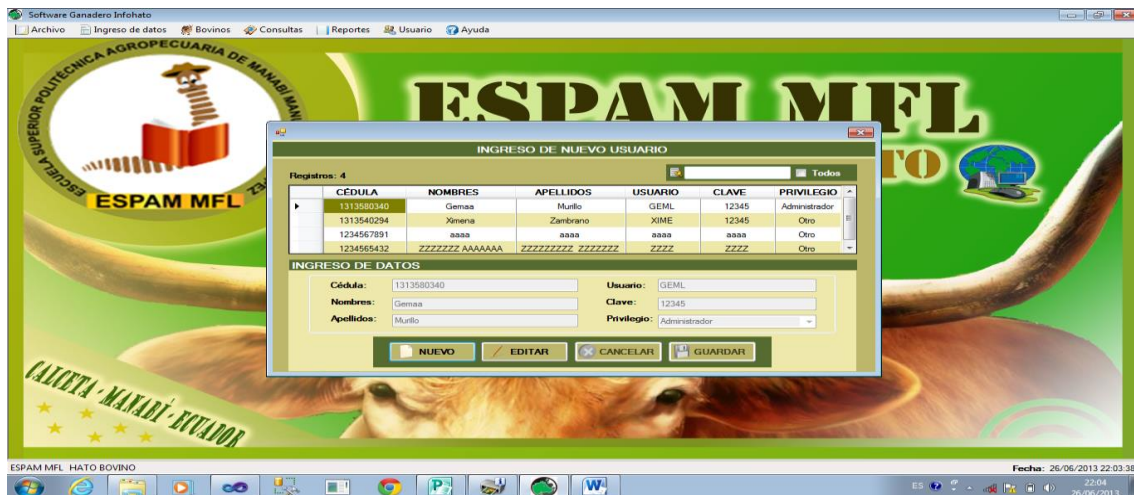
Nº de página actual: 1      Nº total de páginas: 1      Factor de zoom: 50%

ESPAM MFL HATO BOVINO      Fecha: 24/06/2013 2:31:45

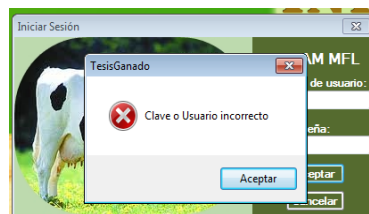


## 2.5 USUARIO

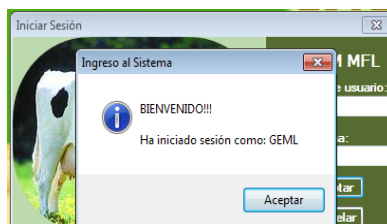
En este formulario se podrá registrar los usuarios que realizarán el ingreso al sistema de acuerdo a la categoría ya sea administrador u otro, si es el primero tendrá la posibilidad de ingresar datos, realizar consultas y generar reportes caso contrario se deshabilitarán las opciones de ingreso, pero si podrá realizar los otros procesos.



Cada uno de estos usuarios accederá por un nombre de usuario y contraseña, en caso de ser incorrectas se presentará a el siguiente mensaje.



Caso contrario se presentará un mensaje de bienvenida.



## 2.6 AYUDA

Aparecerá un pdf del manual de usuario.

**ANEXO 4**

**CERTIFICACIÓN DE CULMINACIÓN DE LA TESIS POR PARTE DEL  
TUTOR ING. JORGE A. PÁRRAGA ÁLAVA MGS.**



**ANEXO 5**

**CERTIFICADO DE FINALIZACIÓN DEL DESARROLLO DE LA TESIS POR  
PARTE DEL COORDINADOR DEL HATO BOVINO  
DR. RAÚL GUEVARA**

