

# UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

**Tema: “Sistema informático para la administración y gestión de información de la hacienda ProduAgro”**

Trabajo de Integración Curricular previo a la obtención del título de Ingenieros en Ciencias de la Computación.

AUTORES: Mora Herrera Jefferson Fernando.  
Portilla Yamberla Héctor Armando.  
TUTOR: Ing. Hidalgo Gujarro Jairo Vladimir, Msc.

Tulcán, 2024.

## **CERTIFICADO DEL TUTOR**

Certifico que los estudiantes Mora Herrera Jefferson Fernando y Portilla Yamberla Héctor Armando con el número de cédula 0402126296 y 1004820773 respectivamente han desarrollado el Trabajo de Integración Curricular: "Sistema informático para la administración y gestión de información de la hacienda ProduAgro"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva.

---

Ing. Hidalgo Guijarro Jairo Vladimir, MSc.

**TUTOR**

Tulcán, junio de 2024

## AUTORÍA DE TRABAJO

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Nosotros, Mora Herrera Jefferson Fernando y Portilla Yamberla Héctor Armando con cédula de identidad número 0402126296 y 1004820773 respectivamente declaramos que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que hemos llegado son de nuestra absoluta responsabilidad.



---

Mora Herrera Jefferson Fernando

**AUTOR**



---

Portilla Yamberla Héctor Armando

**AUTOR**

Tulcán, junio de 2024

## ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Nosotros Mora Herrera Jefferson Fernando y Portilla Yamberla Héctor Armando declaramos ser autor de los criterios emitidos en el Trabajo de Integración Curricular: "Sistema informático para la administración y gestión de información de la hacienda ProduAgro" y se exime expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.



---

Mora Herrera Jefferson Fernando  
**AUTOR**



---

Portilla Yamberla Héctor Armando  
**AUTOR**

Tulcán, junio de 2024

## AGRADECIMIENTO

En primer lugar, quiero expresar mi profundo agradecimiento a mis padres, hermanos/as y amigos por ser mi fuente constante de apoyo. Sus presencias han aportado inmensamente a mi vida de diversas maneras y les estoy agradecido por compartir esta etapa conmigo su respaldo ha sido esencial en mi desarrollo personal y profesional.

Son muchos los docentes que han sido parte de mi camino universitario y a todos ellos les quiero agradecer por transmitirme los conocimientos necesarios para hoy poder estar aquí, de igual manera agradezco a mi tutor el MSc. Jairo Hidalgo por su dedicación y paciencia, sin sus palabras y correcciones precisas no hubiese podido lograr llegar a esta instancia tan anhelada. Gracias por su guía y todos sus consejos. Finalmente quiero expresar mi sincero agradecimiento a Universidad Politécnica del Carchi, una institución que ha desafiado mis límites y al mismo tiempo ha sido la clave para alcanzar mi tan anhelado título.

Jefferson Fernando Mora Herrera

Agradezco a Dios, por bendecirme con salud y permitirme llegar a cumplir esta meta. A la Universidad Politécnica Estatal del Carchi por ser parte fundamental de mi formación académica y profesional, por brindarme las herramientas y conocimientos necesarios para construirme profesionalmente. Al Ing. Juan Pablo Grijalva, propietario de la Hacienda ProduAgro, por permitirme realizar este trabajo en sus instalaciones y brindarme la información necesaria para la elaboración de esta investigación. A mi tutor el Ing. Jairo Hidalgo, por el apoyo y conocimiento brindado a lo largo de la realización de este trabajo. A mis familiares, amigos, compañeros y cada una de las personas que de una u otra manera me apoyaron e impulsaron a lograr esta meta. Agradecimiento especial a mis padres, la Sra. Dolores Yamberla y el Sr. Leopoldo Portilla, todo lo que he logrado hasta hoy es gracias a ellos, sin su ejemplo de superación, trabajo, amor y coraje este trabajo no se hubiese completado. Ellos nunca dejaron de creer en mi ni en mis capacidades, nunca me faltaron sus palabras de apoyo cuando más lo necesitaba. Gracias papa y mama por ayudarme a cumplir mis metas y sueños.

Héctor Armando Portilla Yamberla

## DEDICATORIA

Dedico este trabajo de investigación a mi hermanita Lady Mora, quién me ha brindado el soporte emocional y económico, gracias por su apoyo incondicional que ha sido fundamental para lograr mis metas personales y académicas. Asimismo, quiero dedicar a mis padres, quienes con su cariño constante han sido mi fuente de inspiración alentándome a perseguir mis objetivos y superar cualquier adversidad en mi camino. Adicionalmente, quiero dedicar a mis compañeros que formaron parte de mi transcurso universitario, con quienes he compartido momentos memorables e inolvidables.

Esta meta la establezco como un logro personal, ya que atravesar estos años ha demandado esfuerzo, valentía y sacrificio.

Jefferson Fernando Mora Herrera

Este trabajo se lo dedico a mis padres, este trabajo es para ellos y por ellos, sin el apoyo incondicional, amor y coraje que me brindan, con seguridad diría que nunca lo hubiera logrado; sin su ejemplo de superación, trabajo arduo, perseverancia y empuje este trabajo no sería lo que es. Se lo dedico a Dios, por bendecirme cada día y permitirme llegar a este día. Se lo dedico a mis familiares, amigos, compañeros y cada una de las personas que, con sus consejos, ánimos y que de una u otra manera me impulsaron a cumplir esta importante meta.

Héctor Armando Portilla Yamberla

## ÍNDICE

<b>RESUMEN</b> .....	21
<b>ABSTRACT</b> .....	22
<b>INTRODUCCIÓN</b> .....	23
<b>I. EL PROBLEMA</b> .....	24
<b>1.1. PLANTEAMIENTO DEL PROBLEMA</b> .....	24
<b>1.2. FORMULACIÓN DEL PROBLEMA</b> .....	26
<b>1.3. JUSTIFICACIÓN</b> .....	26
<b>1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN</b> .....	27
1.4.1. Objetivo General .....	27
1.4.2. Objetivos Específicos.....	27
1.4.3. Preguntas de Investigación.....	28
<b>II. FUNDAMENTACIÓN TEÓRICA</b> .....	29
<b>2.1. ANTECEDENTES DE LA INVESTIGACIÓN</b> .....	29
<b>2.2. MARCO TEÓRICO</b> .....	32
2.2.1. Sistema Informático .....	32
2.2.2. Generalidades de un sistema informático.....	35
2.2.3. Metodologías de desarrollo de software .....	39
2.2.4. Herramientas para el desarrollo de software .....	42
2.2.5. Ingeniería de Procesos .....	55
2.2.6. Ingeniería de Software .....	56
2.2.7. Administrar y Gestionar información.....	59
2.2.8. Hacienda Ganadera .....	60
<b>III. METODOLOGÍA</b> .....	65
<b>3.1. ENFOQUE METODOLÓGICO</b> .....	65
3.1.1. Enfoque .....	65
3.1.2. Tipo de Investigación .....	65

3.2. IDEA A DEFENDER .....	68
3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES .....	68
3.4. MÉTODOS UTILIZADOS .....	72
IV. RESULTADOS Y DISCUSIÓN.....	77
4.1. RESULTADOS .....	77
4.2. PROPUESTA .....	92
4.3. DISCUSIÓN.....	100
V. CONCLUSIONES Y RECOMENDACIONES .....	102
5.1. CONCLUSIONES .....	102
5.2. RECOMENDACIONES .....	103
VI. REFERENCIAS BIBLIOGRÁFICAS .....	104
VII. ANEXOS .....	108

## ÍNDICE DE TABLAS

Tabla 1. Metodologías de desarrollo más utilizadas en el desarrollo de software.....	39
Tabla 2. Comparación de metodologías ágiles de desarrollo.....	40
Tabla 3. Cuadro comparativo de los lenguajes de programación más utilizados para aplicaciones web. ....	43
Tabla 4. Cuadro comparativo de entorno de desarrollo de un software. ....	46
Tabla 5. Cuadro comparativo de sistemas gestores de bases de datos.....	49
Tabla 6. Comparación de tipos de arquitecturas de software. ....	54
Tabla 7. Cuadro comparativo situación inicial y situación final. ....	63
Tabla 8. Operacionalización de variables: Variable independiente.....	70
Tabla 9. Operacionalización de variables: Variable dependiente.....	71
Tabla 10. Historial de Revisiones. ....	124
Tabla 11. Artículos y artefactos para entregar. ....	125
Tabla 12. Recursos humanos y profesionales.....	125
Tabla 13. Roles y responsabilidades. ....	126
Tabla 14. Plan de proyecto. ....	126
Tabla 15. Fases y líneas base.....	126
Tabla 16. Objetivos y riesgos de cada fase. ....	127
Tabla 17. Diagrama de Gantt. ....	127
Tabla 18. Calendario del proyecto. ....	127
Tabla 19. Control de desviación a la planificación.....	127
Tabla 20. Matriz de riesgo. ....	128
Tabla 21. Matriz de calor.....	128
Tabla 22. Recursos Hardware. ....	129
Tabla 23. Recurso de software.....	129
Tabla 24. Recursos de factibilidad económica.....	130
Tabla 25. Cuadro comparativo entre la Microsoft Excel, cuadernos, libretas físicas y el sistema informático.....	131
Tabla 26. Asignación de roles. ....	132
Tabla 27. Gestión de Usuarios.....	150
Tabla 28. Control de usuario.....	150
Tabla 29. Módulo de terneras y vaconas.....	150
Tabla 30. Historial veterinario. ....	150
Tabla 31. Módulo Ganado de Producción lechera. ....	151

Tabla 32. Inseminación.....	151
Tabla 33. Módulo Historial de mantenimiento de equipos agropecuarios.....	151
Tabla 34. Módulo Inventario veterinario.....	152
Tabla 35. Módulo Potreros y abonos.....	152
Tabla 36. Módulo Producción lechera.....	152
Tabla 37. Tarea de usuario 1 .....	153
Tabla 38. Tarea de usuario 2 .....	153
Tabla 39. Tarea de usuario 3 .....	153
Tabla 40. Tarea de usuario 4 .....	153
Tabla 41. Tarea de usuario 5 .....	154
Tabla 42. Tarea de usuario 6 .....	154
Tabla 43. Tarea de usuario 7 .....	154
Tabla 44. Tarea de usuario 8 .....	154
Tabla 45. Tarea de usuario 9 .....	154
Tabla 46. Tarea de usuario 10 .....	155
Tabla 47. Tarea de usuario 11 .....	155
Tabla 48. Tarea de usuario 12 .....	155
Tabla 49. Tarea de usuario 13 .....	155
Tabla 50. Tarea de usuario 14 .....	156
Tabla 51. Tarea de usuario 15 .....	156
Tabla 52. Tarea de usuario 16 .....	156
Tabla 53. Tarea de usuario 17 .....	156
Tabla 54. Tarea de usuario 18 .....	156
Tabla 55. Tarea de usuario 19 .....	157
Tabla 56. Tarea de usuario 20 .....	157
Tabla 57. Tarea de usuario 21 .....	157
Tabla 58. Tarea de usuario 22 .....	157
Tabla 59. Tarea de usuario 23 .....	157
Tabla 60. Tarea de usuario 24 .....	158
Tabla 61. Tarea de usuario 25 .....	158
Tabla 62. Tarea de usuario 26 .....	158
Tabla 63. Tarea de usuario 27 .....	158
Tabla 64. Tarea de usuario 28 .....	159
Tabla 65. Tarea de usuario 29 .....	159

Tabla 66. Tarea de usuario 30.....	159
Tabla 67. Tarea de usuario 31 .....	159
Tabla 68. Tarea de usuario 32.....	159
Tabla 69. Valoración y estimación de tiempo de historias de usuario. ....	160
Tabla 70. Tarjeta CRC Acceso al sistema.....	161
Tabla 71. Tarjeta CRC Control de Usuario. ....	161
Tabla 72. Tarjeta CRC Administración de Terneras.....	161
Tabla 73. Tarjeta CRC Historial Veterinario. ....	162
Tabla 74. Tarjeta CRC Ingreso de animal a la producción lechera.....	162
Tabla 75. Tarjeta CRC Inseminación. ....	162
Tabla 76. Mantenimiento de equipos pecuarios. ....	162
Tabla 77. Tarjeta CRC Inventario veterinario. ....	163
Tabla 78. Tarjeta CRC Potreros.....	163
Tabla 79. Tarjeta CRC Producción lechera.....	163
Tabla 80. Descripción Caso de uso administrador. ....	165
Tabla 81. Descripción Caso de uso Ternero.....	167
Tabla 82. Descripción Caso De Uso Operario Del Ganado. ....	168
Tabla 83. Descripción Caso De Uso Operario Del Ordeño. ....	169
Tabla 84. Descripción Caso De Uso Operario De Potreros. ....	170
Tabla 85. Comparativa modelo conceptual, lógico y físico.....	171
Tabla 86. Descripción de tablas, atributos y claves PK y FK. ....	177
Tabla 87. Descripción de tablas, atributos y claves PK y FK. ....	177
Tabla 88. Descripción de tablas, atributos y claves PK y FK. ....	177
Tabla 89. Descripción de tablas, atributos y claves PK y FK. ....	177
Tabla 90. Descripción de tablas, atributos y claves PK y FK. ....	178
Tabla 91. Descripción de tablas, atributos y claves PK y FK. ....	178
Tabla 92. Descripción de tablas, atributos y claves PK y FK. ....	178
Tabla 93. Descripción de tablas, atributos y claves PK y FK. ....	178
Tabla 94. Descripción de tablas, atributos y claves PK y FK. ....	179
Tabla 95. Seguimiento de ejecución de pruebas.....	235
Tabla 96. Prueba acceso al sistema de Administración. ....	236
Tabla 97. Prueba Acceso incorrecto al sistema.....	236
Tabla 98. Prueba Creación correcta de usuario.....	237
Tabla 99. Prueba Creación incorrecta de usuario.....	237

Tabla 100. Prueba Ingreso de un nuevo animal. ....	237
Tabla 101. Prueba Ingreso incorrecto de un nuevo animal. ....	238
Tabla 102. Prueba Ingreso correcto de la Medición Lechera de un animal. ....	238
Tabla 103. Prueba Ingreso incorrecto de la Medición Lechera de un animal. ....	238
Tabla 104. Prueba Ingreso correcto de una Muerte de un animal. ....	239
Tabla 105. Prueba Ingreso incorrecto de una Muerte de un animal. ....	239
Tabla 106. Prueba Ingreso correcto de Descarte de un animal. ....	239
Tabla 107. Prueba Ingreso incorrecto de Descarte de un animal. ....	240
Tabla 108. Prueba Ingreso correcto del peso de un animal. ....	240
Tabla 109. Prueba Ingreso incorrecto del peso de un animal. ....	240
Tabla 110. Prueba Ingreso correcto de un tratamiento. ....	241
Tabla 111. Prueba Ingreso incorrecto de un tratamiento. ....	241
Tabla 112. Prueba Ingreso correcto de un Traslado. ....	241
Tabla 113. Prueba Ingreso incorrecto de un Traslado. ....	242
Tabla 114. Prueba Ingreso correcto de una Inseminación. ....	242
Tabla 115. Prueba Ingreso incorrecto de una Inseminación. ....	242
Tabla 116. Prueba Ingreso correcto de Chequeo de un animal inseminado. ....	243
Tabla 117. Prueba Ingreso correcto del Parto de un animal. ....	243
Tabla 118. Prueba Ingreso incorrecto del Parto de un animal. ....	243
Tabla 119. Prueba Ingreso correcto del Aborto de un animal. ....	244
Tabla 120. Prueba Ingreso incorrecto del Aborto de un animal. ....	244
Tabla 121. Prueba Ingreso correcto Mantenimiento de equipos e instalaciones. ....	244
Tabla 122. Prueba Ingreso correcto de Productos. ....	245
Tabla 123. Prueba Ingreso incorrecto de Productos. ....	245
Tabla 124. Prueba Ingreso correcto de Pedidos. ....	245
Tabla 125. Prueba Ingreso incorrecto de Pedidos. ....	246
Tabla 126. Prueba Ingreso correcto de un potrero. ....	246
Tabla 127. Prueba Ingreso incorrecto de un potrero. ....	246
Tabla 128. Prueba Ingreso correcto de una Acción realizada a un potrero. ....	247
Tabla 129. Prueba Ingreso incorrecto de una Acción realizada a un potrero. ....	247
Tabla 130. Prueba Registro correcto de la Producción Diaria de Leche. ....	247
Tabla 131. Prueba Registro incorrecto de la Producción Diaria de Leche. ....	248
Tabla 132. Número de módulos ....	248
Tabla 133. Características del software ....	248

Tabla 134. Matriz de prueba de compatibilidad .....	249
Tabla 135. Matriz de uso de memoria, CPU y uso de red .....	251
Tabla 136. Matriz de rendimiento .....	253

## ÍNDICE DE FIGURAS

Figura 1. Sistema Informático de Gestión del Ganado Vacuno.....	30
Figura 2. Software de control ganadero.....	32
Figura 3. Etapas en la metodología de desarrollo ágil XP. ....	41
Figura 4. Esquema de la arquitectura MVC. ....	52
Figura 5. Esquema de la arquitectura Cliente/Servidor.....	53
Figura 6. Esquema de la arquitectura por capas.....	54
Figura 7. Grafico de análisis del cuestionario primera pregunta. ....	78
Figura 8. Grafico de análisis del cuestionario segunda pregunta. ....	78
Figura 9. Grafico de análisis del cuestionario tercer pregunta. ....	79
Figura 10. Grafico de análisis del cuestionario cuarta pregunta.....	80
Figura 11. Grafico de análisis del cuestionario quinta pregunta. ....	80
Figura 12. Grafico de análisis del cuestionario sexta pregunta.....	81
Figura 13. Grafico de análisis del cuestionario séptima pregunta.....	82
Figura 14. Grafico de análisis del cuestionario octava pregunta.....	83
Figura 15. Grafico de análisis del cuestionario novena pregunta.....	83
Figura 16. Grafico de análisis del cuestionario décima pregunta.....	84
Figura 17. Proceso de registro de un nuevo bovino.....	88
Figura 18. Proceso de registro de traslado de un bovino entre predios. ....	89
Figura 19. Proceso de registro de chequeo de bovinos.....	89
Figura 20. Proceso de registro de inventario. ....	90
Figura 21. Proceso de registro de gestión de campo.....	90
Figura 22. Proceso de registro de producción de leche.....	91
Figura 23. Logotipo de la Hacienda ProduAgro.....	92
Figura 24. Estructura organizacional de la hacienda ProduAgro. ....	92
Figura 25. Matriz en Excel que maneja la hacienda referente a la información de los animales bovinos. ....	96
Figura 26. Registro de inseminaciones.....	97
Figura 27. Registro de chequeos.....	98
Figura 28. Registro de productos actualmente. ....	99
Figura 29. Registro de estados de potreros en la hacienda.....	99
Figura 30. Registro de producción lechera. ....	100
Figura 31. Acta de la sustentación de Predefensa del TIC ....	108
Figura 32. Acta de la sustentación de Predefensa del TIC.....	109

Figura 33. Certificado del abstract por parte de idiomas. ....	110
Figura 34. Informe Abstract por parte del CIDEN. ....	111
Figura 35. Oficio de autorización para realizar la investigación. ....	112
Figura 36. Ficha de entrevista. ....	113
Figura 37. Entrevista al administrador encargado de la hacienda ProduAgro. ....	114
Figura 38. Entrevista al administrador encargado de la hacienda ProduAgro. ....	115
Figura 39. Entrevista al administrador encargado de la hacienda ProduAgro. ....	116
Figura 40. Entrevista al administrador encargado de la hacienda ProduAgro. ....	117
Figura 41. Encuesta para los empleados de la hacienda. ....	118
Figura 42. Encuesta para los empleados de la hacienda. ....	119
Figura 43. Encuesta para los empleados de la hacienda. ....	120
Figura 44. P001 Caracterización de proceso de registro bovino. ....	134
Figura 45. P002 Caracterización de proceso de registro de medición lechera. ....	135
Figura 46. P003 Caracterización de proceso de registro de muertes de bovinos. ....	136
Figura 47. P004 Caracterización de proceso de registro de descartes bovinos. ....	137
Figura 48. P005 Caracterización de proceso de registro de pesos bovinos. ....	138
Figura 49. P006 Caracterización de proceso de registro de datos veterinarios. ....	139
Figura 50. P007 Caracterización de proceso de registro de traslados de bovinos. ....	140
Figura 51. P008 Caracterización de proceso de registro de inseminaciones de bovinos. ....	141
Figura 52. P009 Caracterización de proceso de registro de chequeos de bovinos. ....	142
Figura 53. P010 Caracterización de proceso de registro de partos de bovinos. ....	143
Figura 54. P011 Caracterización de proceso de registro de abortos de bovinos. ....	144
Figura 55. P012 Caracterización de proceso de registro de productos. ....	145
Figura 56. P013 Caracterización de proceso de registro de pedidos e ingresos de productos. ....	146
Figura 57. P014 Caracterización de proceso de registro de mantenimientos. ....	147
Figura 58. P015 Caracterización de proceso de registro de producción lechera. ....	148
Figura 59. P016 Caracterización de proceso de registro de despacho de leche. ....	149
Figura 60. Arquitectura de la aplicación. ....	164
Figura 61. Caso de uso: Administrador. ....	165
Figura 62. Caso de Uso: Ternero. ....	167
Figura 63. Caso De Uso: Operario Del Ganado. ....	168
Figura 64. Caso De Uso: Operario Del Ordeño. ....	169

Figura 65. Caso De Uso: Operario De Potreros.....	170
Figura 66. Modelo conceptual de la base de datos de la hacienda ProduAgro. ...	174
Figura 67. Modelo Lógico de la base de datos.....	175
Figura 68. Modelo Físico de la Base de Datos.....	176
Figura 69. Inicio de Sesión. ....	179
Figura 70. Menú de los módulos.....	180
Figura 71. Módulo de Administración de Usuarios.....	180
Figura 72. Modulo de Terneras y Vaconas. ....	181
Figura 73. Módulo de Producción Lechera.....	181
Figura 74. Módulo de Veterinario.....	182
Figura 75. Módulo de Mantenimiento. ....	182
Figura 76. Módulo de Potreros.....	183
Figura 77. Módulo de reporte de producción lechera.....	183
Figura 78. Diseño de la interfaz principal del sistema informático. ....	184
Figura 79. Código fuente de la aplicación. ....	185
Figura 80. Código fuente de la aplicación. ....	186
Figura 81. Diseño de interfaz Animales de la aplicación.....	186
Figura 82. Código fuente de la aplicación. ....	187
Figura 83. Código fuente de la aplicación. ....	187
Figura 84. Código fuente de la aplicación. ....	188
Figura 85. Código fuente de la aplicación. ....	188
Figura 86. Código fuente de la aplicación. ....	189
Figura 87. Código fuente de la aplicación. ....	189
Figura 88. Código fuente de la aplicación. ....	190
Figura 89. Código fuente de la aplicación. ....	191
Figura 90. Código fuente de la aplicación. ....	191
Figura 91. Código fuente de la aplicación. ....	192
Figura 92. Código fuente de la aplicación. ....	193
Figura 93. Código fuente de la aplicación. ....	193
Figura 94. Código fuente de la aplicación. ....	194
Figura 95. Diseño de interfaz de Medición Lechera de la aplicación. ....	195
Figura 96. Diseño de interfaz de Medición Lechera de la aplicación .....	195
Figura 97. Código fuente de la aplicación. ....	196
Figura 98. Código fuente de la aplicación. ....	196

Figura 99. Código fuente de la aplicación.....	197
Figura 100. Código fuente de la aplicación.....	197
Figura 101. Código fuente de la aplicación.....	198
Figura 102. Código fuente de la aplicación.....	198
Figura 103. Diseño de interfaz de la vista de Muertes de la aplicación. ....	199
Figura 104. Código fuente de la aplicación.....	199
Figura 105. Código fuente de la aplicación.....	200
Figura 106. Código fuente de la aplicación.....	200
Figura 107. Código fuente de la aplicación.....	201
Figura 108. Código fuente de la aplicación.....	201
Figura 109. Código fuente de la aplicación.....	202
Figura 110. Código fuente de la aplicación.....	202
Figura 111. Código fuente de la aplicación.....	203
Figura 112. Código fuente de la aplicación.....	203
Figura 113. Código fuente de la aplicación.....	204
Figura 114. Diseño de interfaz de la vista Descartes de la aplicación.....	205
Figura 115. Código fuente de la aplicación.....	205
Figura 116. Código fuente de la aplicación.....	206
Figura 117. Código fuente de la aplicación.....	206
Figura 118. Código fuente de la aplicación.....	207
Figura 119. Código fuente de la aplicación.....	207
Figura 120. Código fuente de la aplicación.....	208
Figura 121. Código fuente de la aplicación.....	208
Figura 122. Código fuente de la aplicación.....	209
Figura 123. Código fuente de la aplicación.....	209
Figura 124. Código fuente de la aplicación.....	210
Figura 125. Código fuente de la aplicación.....	210
Figura 126. Código fuente de la aplicación.....	211
Figura 127. Código fuente de la aplicación.....	211
Figura 128. Código fuente de la aplicación.....	212
Figura 129. Código fuente de la aplicación.....	212
Figura 130. Diseño de interfaz de la vista de Datos Veterinarios de la aplicación. ....	213
Figura 131. Código fuente de la aplicación.....	213
Figura 132. Código fuente de la aplicación.....	214

Figura 133. Código fuente de la aplicación. ....	214
Figura 134. Código fuente de la aplicación. ....	215
Figura 135. Código fuente de la aplicación. ....	215
Figura 136. Código fuente de la aplicación. ....	216
Figura 137. Diseño de interfaz de la vista Traslados de la aplicación. ....	216
Figura 138. Código fuente de la aplicación. ....	217
Figura 139. Código fuente de la aplicación. ....	217
Figura 140. Código fuente de la aplicación. ....	218
Figura 141. Código fuente de la aplicación. ....	218
Figura 142. Código fuente de la aplicación. ....	219
Figura 143. Código fuente de la aplicación. ....	219
Figura 144. Código fuente de la aplicación. ....	220
Figura 145. Diseño de interfaz de la vista Inseminaciones de la aplicación. ....	220
Figura 146. Código fuente de la aplicación. ....	221
Figura 147. Código fuente de la aplicación. ....	221
Figura 148. Código fuente de la aplicación. ....	222
Figura 149. Código fuente de la aplicación. ....	222
Figura 150. Código fuente de la aplicación. ....	223
Figura 151. Código fuente de la aplicación. ....	223
Figura 152. Código fuente de la aplicación. ....	224
Figura 153. Código fuente de la aplicación. ....	224
Figura 154. Diseño de interfaz de la vista Chequeos de la aplicación. ....	225
Figura 155. Código fuente de la aplicación. ....	225
Figura 156. Código fuente de la aplicación. ....	226
Figura 157. Código fuente de la aplicación. ....	226
Figura 158. Código fuente de la aplicación. ....	227
Figura 159. Código fuente de la aplicación. ....	227
Figura 160. Código fuente de la aplicación. ....	228
Figura 161. Diseño de interfaz de la vista Partos de la aplicación. ....	229
Figura 162. Código fuente de la aplicación. ....	229
Figura 163. Código fuente de la aplicación. ....	230
Figura 164. Código fuente de la aplicación. ....	231
Figura 165. Código fuente de la aplicación. ....	231
Figura 166. Código fuente de la aplicación. ....	232

Figura 167. Código fuente de la aplicación.....	232
Figura 168. Diseño de interfaz de la vista Productos de la aplicación. ....	233
Figura 169. Código fuente de la aplicación.....	233
Figura 170. Código fuente de la aplicación.....	234
Figura 171. Código fuente de la aplicación.....	234
Figura 172. Gráfico de prueba de carga .....	254

## ÍNDICE DE ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC.....	109
Anexo 2. Certificado del abstract por parte de idiomas.....	111
Anexo 3. Oficio de autorización para realizar la investigación.....	113
Anexo 4. Ficha de entrevista .....	114
Anexo 5. Entrevista al administrador encargado de la hacienda ProduAgro.....	115
Anexo 6. Encuesta.....	119
Anexo 7. Plan de desarrollo de software .....	122
Anexo 8. Manual de usuario.....	256

## RESUMEN

La ganadería es un sector muy importante en el Ecuador y ha crecido significativamente en los últimos años como parte de la producción agrícola. Las regiones costeras y amazónicas producen principalmente ganado vacuno para la producción de carne, mientras que la región sierra producen vacas con la finalidad de aprovechar su producto de secreción normal de la glándula mamaria denominada coloquialmente leche de ganado vacuno. Por tanto, el objetivo general del trabajo de investigación es el desarrollo de un sistema informático que permita el mejoramiento en el proceso administrar y gestionar la información de la hacienda ProduAgro con el uso de herramientas tecnológicas. El uso de fuentes investigativas válidas y recomendadas fueron leídas e interpretadas para la sustentación teórica del trabajo de investigación teniendo como prioridad las fuentes primarias como es los artículos científicos, libros, trabajos de titulaciones de universidades nacionales e internacionales, páginas web, blogs, etc. Que sirvió para solventar y apoyar el tema de esta investigación con un contenido mucho más científico y al mismo tiempo aumentar el conocimiento y conocer nuevas herramientas que se optó para el desarrollo del sistema informático. El presente trabajo de investigación denominado "Sistema informático para la administración y gestión de la hacienda ProduAgro" está basado en un enfoque mixto como es el enfoque cuantitativo y enfoque cualitativo con el fin de llevar a cabo la obtención y análisis de datos cuantitativos y cualitativos dentro de un mismo estudio con la intención de minimizar el proceso de administración que hace énfasis a la optimización del uso de recursos materiales y humanos de la hacienda ProduAgro y la gestión que ayuda a toma de decisiones para estar preparados ante algún evento de la hacienda y alcanzar metas que son planteadas dentro de la misma.

**Palabras Claves:** Sistema informático, administración, gestión, ganado bovino.

## ABSTRACT

Livestock plays an extremely important role in Ecuador, experiencing significant growth in recent years as an integral part of agricultural production. While the coastal and Amazonian regions are noted for the breeding of cattle mainly for meat production, the highland region focuses on the production of a product of mammary gland secretion called milk from cows. Therefore, the main purpose of this research is to develop a computer system that optimizes the management and administration of information on the ProduAgro farm by technological tools. To theoretically support this research work, several valid and recommended research sources were consulted and analysed. Priority was given to primary sources, such as scientific articles, books, dissertations from national and international universities, websites, and blogs. This approach made it possible to strengthen and support the research topic with scientific content, while facilitating the discovery of new tools for the development of the computer system. The present study, entitled "Computer system for the administration and management of the ProduAgro farm", adopts a mixed approach that combines both quantitative and qualitative elements. This methodology is used for the purpose of obtaining and analysing data of both natures within the same research framework. The objective is to minimize administrative processes, emphasizing the optimization of the material and human resources of the ProduAgro farm, as well as the management that facilitates decision-making in the event of possible eventualities and the achievement of goals established within it.

**Keywords:** Computer system, administration, management, cattle.

## INTRODUCCIÓN

Este trabajo de investigación con su tema: "Sistema informático para la administración y gestión de la hacienda ProduAgro" tiene como estudio analizar la situación actual de los distintos problemas y necesidades que afronta la hacienda ProduAgro en el proceso de administración y gestión de información.

La introducción describe los diversos desafíos y la importancia que enfrenta actualmente el sector ganadero en Ecuador y detalla algunos incidentes que ilustran las restricciones actuales al control y manejo del ganado en la hacienda ProduAgro

En el Capítulo 1 se clasifica principalmente con el planteamiento de la problemática de la investigación indicando su situación real en que se vive en el lugar de estudio que es el bajo conocimiento de tecnologías aptas para una mejor administración de la hacienda, se usó la matriz para la investigación formativa como instrumento para determinar las variables que van hacer estudiadas y concluir con la delimitación del objetivo general y los objetivos específicos.

En el Capítulo 2 se explican los antecedentes a proyectos e investigaciones ya realizadas con algunas similitudes en base a nuestro tema y con resultados favorables, también determinamos los conceptos teóricos, las herramientas utilizadas, la mecánica de cada herramienta y la retroalimentación de haciendas que utilizan sistemas de información para resolver los mismos problemas, que fueron cruciales para el desarrollo de este trabajo de investigación.

El Capítulo 3 reúne la información y requisitos utilizados para el desarrollo de la metodología y establecer el enfoque, tipo de investigación, idea a defender, operalización de variables y finalmente los métodos a utilizar para la obtención de datos e información para un óptimo proceso en la ejecución de este trabajo de investigación.

El capítulo 4 está conformado por los resultados y discusión de la propuesta en base al tema de estudio de esta investigación con las metodologías, conceptos teóricos y herramientas tecnológicas que fueron determinantes para el desarrollo del sistema informático.

## I. EL PROBLEMA

### 1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, los productores ganaderos se han convertido en empresarios muy eficaces, adoptando prácticas de registro como una medida esencial para supervisar la producción. Este enfoque facilita la evaluación de rendimientos, la comparación de diversos indicadores relacionados con la ganadería, tanto en términos de producción, reproducción y aspectos económicos, permitiendo un análisis detallado de su progreso a lo largo del tiempo, ya sea en una dirección favorable o desfavorable.

El ganado bovino aporta un 40 por ciento del valor de producción agrícola mundial y sostiene los medios de vida y la seguridad alimentaria de casi 1 300 millones de personas, que ofrece las oportunidades para el sustento de la economía familiar interna que contribuye a la generación de empleo y reducción de pobreza. (Producción Animal | FAO | Organización de las Naciones Unidas para la Alimentación y la Agricultura, n.d., p. 1)

Según el Banco Central del Ecuador (2020), el sector agrícola, ganadero, forestal, de caza y pesca tiene una significativa importancia en el Producto Interno Bruto (PIB) del país, representando el 10,4%. Además, se destaca que la contribución económica del sector pecuario específicamente al PIB es del 1,4%, incluyendo actividades de transformación (p.3).

Haciendo referencia a estos datos, en la actualidad los ganaderos de medianas y pequeñas haciendas no poseen con información accesible sobre los registros de sus haciendas y, por otra parte, los empleados no dan a esta información un apropiado procedimiento y manejo responsable, por consiguiente, al enfrentarse en situaciones difíciles de las haciendas no contienen argumentación concreta para la toma de decisiones.

Una considerable porción del ámbito ganadero se vale de la tecnología para agilizar la gestión de datos en un período específico. Al contar con sistemas informáticos de

gestión de información, se logra una completa organización de los datos reales relacionados con los distintos procesos llevados a cabo en las explotaciones ganaderas, sin embargo, el uso de la tecnología en haciendas que son conformadas por personas adultas y que se alojan en sectores alejados de las civilizaciones no es tan accesible y por lo tanto limita su uso debido a causas como: el desconocimiento de softwares capaces de automatizar la información y los altos costos para la adquisición de estos software.

Para contrarrestar utilizan métodos convencionales y la información se transcribe en documentos Excel, hojas de cuaderno, carpetas, entre otros, que a futuro esta documentación se deteriora, se pierde, o es manipulada indebidamente.

La ganadería bovina es un pilar fundamental dentro del sector agropecuario del Ecuador debido a que contribuye a la actividad de la economía rural campesina con la oferta de productos cárnicos y leche, que son parte de la canasta básica y la seguridad alimentaria del país. (Ecuador. Ministerio de Agricultura, Ganadería, Acuacultura y Pesca, 2020)

La gestión de las fincas ganaderas en Ecuador involucra diversos aspectos administrativos relacionados con la producción, como el mantenimiento de registros, la supervisión de eventos en la hacienda, la contabilidad y la elaboración de informes e inventarios. Estos elementos desempeñan un papel crucial en la planificación, seguimiento y toma de decisiones dentro de la actividad ganadera.

En la región de la Sierra, específicamente en la Provincia del Carchi al norte del país, se encuentra Tulcán, la capital y la ciudad más grande y poblada de dicha provincia.

En la Provincia del Carchi, conocida por su producción láctea y ganadera, la mayoría de los ganaderos que operan pequeñas y medianas explotaciones en áreas rurales siguen utilizando métodos tradicionales para administrar y gestionar la información sobre su ganado. Se considera común que opten por llevar registros manualmente, utilizando hojas sueltas, libretas o cuadernos, mientras que aquellos con un poco más de experiencia pueden recurrir a herramientas como Microsoft Excel para esta tarea.

En la parroquia San Isidro del cantón Espejo, hay una abundancia de fincas y haciendas enfocadas en la producción de leche. Destacando entre ellas, la hacienda ProduAgro se erige como la más prominente en la región. A pesar de sus progresos en agricultura y ganadería, carece de un sistema informático eficiente

para la gestión de su información. Para superar esta limitación, recurren a métodos tradicionales como el registro manual en papel y cuadernos.

En ocasiones, la falta de registro adecuado o la desordenada gestión de la información puede tener consecuencias significativas en los procedimientos de la hacienda. Esto se refleja en la pérdida de datos, la carencia de una base sólida para la administración financiera, inventarios ilegibles, entre otros problemas.

## **1.2. FORMULACIÓN DEL PROBLEMA**

¿Cómo contribuir para mejorar la gestión y administración de información de la hacienda ProduAgro?

## **1.3. JUSTIFICACIÓN**

En la mayoría de las fincas y haciendas de la región serrana, la gestión administrativa se realiza de manera tradicional, sin la ayuda de sistemas de información para organizar y almacenar los datos generados por las actividades ganaderas.

Hasta hace poco tiempo, las restricciones como la falta de acceso a internet o de tecnología eran notables. Sin embargo, en los últimos años, la tecnología ha alcanzado incluso a las áreas rurales, como la parroquia de San Isidro. A pesar de la relevancia del sector agropecuario y ganadero en nuestra comunidad, aún no ha adoptado plenamente las ventajas que la tecnología puede proporcionar.

Los enfoques tradicionales son caracterizados por su lenta ejecución, tediosidad y falta de fiabilidad. La implementación de un sistema informático administrativo permitirá una gestión integral de los animales y los procesos de la hacienda, brindando una ventaja significativa al proporcionar la información necesaria para tomar decisiones efectivas en relación con las actividades de la finca, al mismo tiempo que se superan las brechas tecnológicas.

Según (Tecnológica et al., 2019) afirma:

Se puede observar en el sector productivo ganadero la falta de herramientas tecnológicas que les permita hacer de su negocio una empresa que no solo ofrezca productos de calidad y uso eficiente de los recursos, sino también una herramienta estratégica para asegurar la continuidad del negocio (p.5).

La introducción del sistema de administración y gestión ganadera en la hacienda ProduAgro resultará en una mejora sustancial en el registro de datos y el control del

ganado. Esto proporcionará beneficios tanto al personal encargado como a los administrativos, al ofrecer información precisa y rápida sobre el ciclo productivo del ganado. Además, el software contará con interfaces intuitivas y fáciles de usar, especialmente diseñadas para usuarios con poca experiencia en el manejo de este tipo de tecnología.

Según (Cajas & Javier, 2019) en su trabajo de investigación afirma:

En el sector agropecuario, específicamente se necesita generar valiosa información que permita a los nuevos administradores, líderes, políticos disponer de información inmediata, indispensable para una mejor administración. Es por tal razón que se necesita de una herramienta valiosa que brinde soluciones prácticas que ayuden a la toma de decisiones en las empresas pecuarias del país (p.30).

Además, esta investigación se compromete con la misión que tenemos como carrera de Computación que, es "Formar Ingenieros en Ciencias de la Computación capaces de proponer y generar soluciones tecnológicas innovadoras que contribuyan al desarrollo de la región, con sólidas bases académicas e investigativas, enmarcados en los aspectos ético, crítico y compromiso social." (UPEC, n.d.)

Como estudiantes formados en la prestigiosa Universidad Politécnica del Carchi y con la obtención de los conocimientos necesarios se puede afirmar que se puede desarrollar un sistema para el mejoramiento de la administración y gestión de la hacienda ProduAgro con el fin de poner énfasis la misión que conlleva estudiar la carrera de Computación brindando una solución tecnológica innovadora enfocada a la región de los sectores ganaderos que están situados en zonas rurales de nuestro País.

## **1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN**

### **1.4.1. Objetivo General**

- Desarrollar un prototipo de sistema informático que permita el mejoramiento de la administración y gestión de información en la hacienda ProduAgro.

### **1.4.2. Objetivos Específicos**

- Fundamentar bibliográficamente las variables de investigación para la administración y gestión de información en la hacienda ProduAgro.

- Analizar la situación actual del uso de la tecnología en la administración y gestión de información en la hacienda ProduAgro.
- Proponer una solución informática para optimizar los procesos manuales que corresponden al manejo adecuado de la información de la hacienda ProduAgro.

#### **1.4.3. Preguntas de Investigación**

- ¿Qué fundamentos teóricos pueden servir de sustento para la administración y gestión de información en la hacienda ProduAgro?
- ¿Cómo es la situación actual del uso de la tecnología en la administración y gestión de información en la hacienda ProduAgro?
- ¿Cuál es la solución informática más favorable para el proceso administrativo y de gestión que se llevan a cabo en la hacienda ProduAgro.

## II. FUNDAMENTACIÓN TEÓRICA

### 2.1. ANTECEDENTES DE LA INVESTIGACIÓN

En las últimas décadas, ha habido un rápido cambio en la industria ganadera debido al aumento en la demanda mundial de alimentos derivados de animales. Este crecimiento se ha visto impulsado por avances tecnológicos importantes y cambios en la estructura de la industria. La demanda en constante aumento está vinculada principalmente a la producción ganadera comercial y a las cadenas alimentarias asociadas.

En el sector agrícola y ganadero, las TIC's están emergiendo como herramientas de impacto horizontal. Aunque su adopción no ha sido tan extensa como en otros sectores, su implementación en la agricultura y ganadería representa un desafío complejo pero prometedor para países como Ecuador, que poseen un vasto potencial en este ámbito. Estas tecnologías facilitan la automatización de procesos, mejoran la toma de decisiones y contribuyen al desarrollo en estas áreas. Según Sánchez (2020, p.19), "las TIC se han vuelto esenciales para el progreso de las empresas agropecuarias"

Según las palabras de Sánchez, en Ecuador, la ganadería es una actividad clave, y los avances tecnológicos ofrecen la oportunidad de optimizar el control del ganado. Esto implica una organización eficiente de la información relacionada con la toma de decisiones y la gestión de la salud de cada animal. Se destaca la relevancia de contar con sistemas de información en el ámbito ganadero, ya que contribuyen a mejorar y resolver los desafíos actuales que enfrenta ProduAgro en su hacienda.

En la Universidad Técnica de Cotopaxi, en el programa de Ingeniería en Informática y Sistema Computacionales, se ha iniciado el proyecto titulado "Sistema Informático de Gestión del Ganado Vacuno de la Hacienda 'La María' ubicada en el barrio Lasso de la parroquia San Lorenzo de Tanicuchi"

“La introducción de un Sistema Informático ha optimizado la administración y supervisión de las operaciones relacionadas con el ganado vacuno en la Hacienda La María, utilizando herramientas de software libre.” (Toapanta Quinatoa & Zhingre Sánchez, 2019, p. 44) Mediante la aplicación, se asistió al dueño y gestor de la finca en la recopilación ordenada de todos los datos generados durante el proceso de gestión del ganado vacuno, facilitando una toma de decisiones ágil y eficiente.



**Figura 1.** Sistema Informático de Gestión del Ganado Vacuno.

**Fuente:** Toapanta y Zhingre 2019. “Sistema informático de gestión del ganado vacuno de la hacienda ‘La María’ ubicada en el barrio Lasso de la parroquia San Lorenzo de Tanicuchi”

En el 2019, Agrocalidad llevó a cabo un estudio con el objetivo de facilitar el proceso de gestión ganadera y la emisión del Certificado Sanitario de Movilización Interna (CMSI) para el ganado bovino. Como parte de este esfuerzo, se implementó una función en el Sistema de Gestión Ganadera que permite a cada ganadero generar electrónicamente el CMSI. Anteriormente, los usuarios debían dirigirse personalmente a las oficinas de Agrocalidad para que el personal correspondiente registrara su información y emitiera el certificado.

Gracias a la introducción del CMSI electrónico, los usuarios tienen la capacidad de llevar a cabo este procedimiento de forma autónoma desde cualquier dispositivo con conexión a internet, asegurándose de obtenerlo en un tiempo eficiente.

Como resultado, se puede concluir que la automatización del proceso llevado a cabo por Agrocalidad en su sistema para la emisión del CMSI representa una

contribución significativa desde la óptica de los sistemas de información. Este avance indica un claro impulso hacia la mejora del sector ganadero, demostrando cómo las TIC's promueven la automatización en los ámbitos agropecuarios y ganaderos.

Con el propósito de promover avances en el control del sector ganadero a través de investigaciones científicas, tecnológicas y pedagógicas, el Máster Universitario en Ingeniería de Software y Sistemas Informáticos de la Universidad Internacional de La Rioja desarrolló un proyecto de investigación denominado "Planificación de LIFTIC Ganadero: aplicación web para la administración de fincas ganaderas".

El propósito de llevar a cabo este proyecto, específicamente el desarrollo de este Trabajo de Fin de Máster (TFM), fue establecer una base sólida para la empresa LIFTIC SAS en la creación de una solución de software llamada LIFTIC Ganadero. Esta aplicación web fue diseñada para facilitar la gestión de fincas de ganadería bovina en la costa norte de Colombia. Durante el proceso, se elaboró detalladamente la planificación del proyecto de desarrollo de software para la implementación exitosa de LIFTIC Ganadero, asegurando además que la aplicación pueda ser accesible desde cualquier dispositivo móvil con conexión a Internet. (Castillo-Álvarez, 2019, p. 95, "Planificación de LIFTIC Ganadero: aplicación web para la administración de fincas ganaderas").

La introducción de este sistema de gestión para el seguimiento del ganado bovino ha resultado en una optimización del manejo de datos y la gestión financiera asociada al ganado. Esto ha permitido una administración más eficiente y organizada, con la conveniencia de acceder a la información de manera remota a través de dispositivos como teléfonos móviles o computadoras conectadas a internet.

Además, en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, en la Carrera de Agropecuaria, se llevó a cabo una investigación titulada "Software de control ganadero en el hato bovino de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López". En ella, Murillo y Zambrano (2019) propusieron: "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."



**Figura 2.** Software de control ganadero.

**Fuente:** Murillo y Zambrano 2019. Software de control ganadero en el hato bovino de la escuela superior politécnica agropecuaria de Manabí Manuel Félix López.

En el estudio mencionado se determinó que el uso de un programa informático para la gestión de la ganadería conlleva una mejora en la eficiencia económica al proporcionar información sobre la producción de leche y la valoración de los activos.

El avance en la ganadería facilita la adopción de tecnología moderna que automatiza el seguimiento del ganado, asegurando así una gestión eficiente del ganado bovino.

## 2.2. MARCO TEÓRICO

### 2.2.1. Sistema Informático

#### 2.2.1.1. Sistemas informáticos

“Los sistemas informáticos desempeñan un papel fundamental en la gestión de información al manipular tanto la entrada como la salida, facilitando el almacenamiento y procesamiento de datos. Además, su utilidad se extiende a la resolución de problemas específicos, empleando controladores o microcontroladores programables” (Guillem, 2019, p.1).

Tanto Senn (2020) como Montilva (2020) concuerdan al definir un sistema como un conjunto de elementos que interactúan para lograr un objetivo común. Esta relación en las definiciones resalta la idea de que un sistema está compuesto por elementos interrelacionados con el propósito de alcanzar un fin común.

En el contexto de la investigación, se entiende un sistema como un conjunto de actividades administrativas y procesos interrelacionados que buscan generar información para respaldar la toma de decisiones.

#### 2.2.1.2. Tipos de sistema

Según las afirmaciones de Senn (2020, p.21) y Stoner (2019, p.51), se aborda la clasificación de sistemas, distinguiendo entre el sistema cerrado, el cual se define como aquel que no tiene interacción con su entorno. No obstante, para el propósito de la presente investigación, no se contempla la utilización de este tipo de sistema.

En contraste, el sistema abierto, según las definiciones de estos autores, se caracteriza por interactuar con su entorno. Además, se destaca que en un sistema abierto se emplean diversas plataformas tecnológicas para mejorar su funcionamiento. En consecuencia, esta investigación se identifica como un sistema abierto, en el cual se busca la interacción con el entorno y se aprovechan diversas plataformas tecnológicas para optimizar su rendimiento.

#### 2.2.1.3. Sistemas de información

Montilva (2020) describe un sistema de información como un sistema hombre-máquina que lleva a cabo el procesamiento de datos para registrar los detalles generados por las transacciones dentro de las entidades que componen una organización. Este sistema tiene la finalidad de proporcionar información que facilite la ejecución de actividades, operaciones y funciones dentro de la organización (p.35).

En concordancia, Senn (2020) establece que un sistema se basa en la computadora para procesar datos de manera que puedan ser utilizados por aquellos que los reciben con el objetivo de respaldar la toma de decisiones (p.628).

Para la presente investigación, se toma en consideración el uso de un tipo de sistema abierto que permite automatizar la ejecución manual de procesos que lo hacen en la Hacienda ProduAgro, tales como el uso de planillas para guardar información del ganado vacuno, generación de listas, reportes, fechas, listado de personal, emisión de informes e inventario, con el fin de optimizar los resultados se opta por el desarrollo de un sistema informático para la administración y gestión de información que tiene como finalidad la toma de decisiones rápidas y fiables que pueden llegar a ser de emergencia o no.

#### 2.2.1.4. Funciones de un sistema de información

Los sistemas de información realizan clasificaciones según la entrada de datos, el procesamiento, la salida y sus objetivos. A pesar de las diferencias, varios autores coinciden en funciones clave: procesar transacciones, crear y mantener archivos y producir informes. Estos informes incluyen periódicos, datos críticos, al instante, de excepción y a medida, cada uno con propósitos específicos. La producción de informes es esencial para mantener a los usuarios informados sobre diversas actividades (Senn, 2020, p.25).

#### 2.2.1.5. Ciclo de vida de un sistema

Según Senn (2020), el método del ciclo de vida para el desarrollo de sistemas abarca un conjunto de actividades que involucra a analistas, diseñadores y usuarios en la creación e implementación de un sistema de información. Las etapas clave incluyen:

##### a) Determinar requerimientos del sistema:

El análisis de sistemas implica comprender las facetas esenciales de la parte empresarial bajo estudio. Se deben responder preguntas cruciales sobre lo que la empresa hace, cómo lo hace, la frecuencia de ciertos procesos, el volumen de transacciones, la eficiencia en las tareas, y la identificación y evaluación de problemas.

##### b) Diseño del sistema:

Esta fase detalla cómo el sistema cumplirá con los requerimientos identificados durante el análisis. Se distingue entre diseño lógico y diseño físico, siendo el primero la estructura conceptual y el segundo la implementación física.

##### c) Desarrollo del software:

Los desarrolladores pueden instalar software de terceros o crear programas personalizados. La elección depende de factores como costos, tiempo disponible y la disponibilidad de programadores.

##### d) Prueba de sistemas:

Durante esta fase, el sistema se utiliza experimentalmente para garantizar que el software funcione sin fallas, procesando conjuntos de datos de prueba y evaluando los resultados.

#### e) Implantación y evaluación:

La implantación implica verificar e instalar nuevo equipo, entrenar a los usuarios y construir archivos de datos necesarios. La evaluación del sistema se realiza en dimensiones como operacional, organizacional, opinión de los administradores y desempeño del desarrollo (p.33).

#### 2.2.2. Generalidades de un sistema informático

Probablemente la definición más formal de software es la atribuida al Institute of Electrical and Electronic Engineers (IEEE) en su estándar 729: "la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo". (Linares, Aleas, Mena, Cruz y Rosales, 2019, p.83).

García (2019) señala que las propiedades del software abarcan una amplia gama de productos informáticos, desde sistemas operativos hasta componentes intangibles que se interrelacionan entre diversos dispositivos dentro de un ordenador, como el monitor, disco duro, procesador y teclado. Estos elementos posibilitan la realización de diversas actividades, como la introducción de datos, la emisión de instrucciones y la realización de operaciones variadas entre distintos dispositivos (hardware).

Un ejemplo práctico de estas funciones es el procesador de textos, que facilita la introducción y edición de información. Es esencial comprender las distintas características del software, tales como la corrección para satisfacer las necesidades del cliente, la usabilidad que garantiza un fácil aprendizaje, la seguridad para mantener protegidos los datos, la flexibilidad que permite la reutilización del código por parte de los desarrolladores, y la portabilidad que asegura su funcionamiento en múltiples plataformas.

##### 2.2.2.1. La internet

Internet es una red global de comunicación que conecta millones de dispositivos y sistemas informáticos en todo el mundo. Utiliza una infraestructura descentralizada y se basa en protocolos estándar, como TCP/IP, para permitir la transferencia de datos de manera eficiente. Facilita el intercambio de información, la comunicación en tiempo real, el acceso a recursos y servicios diversos, así como la colaboración en una escala global. Internet ha evolucionado desde sus inicios, convirtiéndose en un

elemento fundamental para la interconexión y acceso a una amplia gama de contenidos, servicios y aplicaciones (Castells, 2020, p.10).

#### 2.2.2.2. Protocolos de internet

En base al autor Gonzales (2020), los protocolos de Internet constituyen un conjunto de normas que rigen la comunicación entre redes de computadoras, siendo la base fundamental de la infraestructura de Internet. Comúnmente, se hace referencia a ellos como el conjunto de protocolos TCP/IP, destacando dos de los más esenciales: el Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP), los cuales fueron los primeros en definirse y son ampliamente utilizados. Este conjunto de protocolos abarca más de 100 normas diferentes, siendo cruciales para la transmisión eficiente de datos en la red. En esencia, los protocolos de Internet proporcionan las reglas y estándares necesarios para que las computadoras se comuniquen de manera efectiva en el entorno global de Internet (p.1).

Entre los protocolos más principales tenemos los siguientes:

#### TCP-IP

Según Chávez, (2019) el protocolo TCP/IP, que significa Protocolo de Control de Transmisión/Protocolo de Internet, es un conjunto de normas y reglas que establece la forma en que los dispositivos se comunican en redes de computadoras, siendo la columna vertebral de Internet. Consta de dos protocolos principales: el Protocolo de Control de Transmisión (TCP), encargado de asegurar la entrega ordenada y fiable de datos, y el Protocolo de Internet (IP), responsable de asignar direcciones únicas a cada dispositivo conectado a la red y dirigir el tráfico de datos entre ellos (p.1)

#### 2.2.2.3. World Wide Web

En la computación, según el autor Barners, (2019) la World Wide Web (WWW), o simplemente la Web, es un sistema global de información en línea que permite acceder y navegar por una vasta cantidad de contenido a través de la conexión de redes de computadoras. La Web se compone de documentos multimedia, como textos, imágenes y videos, vinculados mediante hipervínculos, que permiten a los usuarios moverse de una página a otra de manera fluida. Desarrollada por Tim Berners-Lee en la década de 1990, la World Wide Web ha evolucionado para

convertirse en una herramienta esencial para la comunicación, el acceso a la información y la colaboración en todo el mundo (p.15).

#### 2.2.2.4. HTML

Según Escobar, (2020) menciona que es HTML, acrónimo de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es un estándar fundamental utilizado en el desarrollo web. Se trata de un lenguaje de marcado que permite la creación y estructuración de contenido para páginas web. En HTML, los elementos se etiquetan con etiquetas específicas que indican su función y presentación. Estas etiquetas definen la estructura básica del contenido, como encabezados, párrafos, enlaces y listas, entre otros. HTML se combina con otras tecnologías web, como CSS y JavaScript, para proporcionar una experiencia de usuario completa y atractiva en la navegación web (p.102).

Larson (2019) sostiene que HTML constituye el medio principal para compartir información en la World Wide Web. En otras palabras, la existencia de la World Wide Web depende esencialmente de HTML, ya que este lenguaje proporciona las instrucciones básicas interpretadas por los navegadores para la visualización de páginas web (p.45).

#### 2.2.2.5. Direcciones de internet (URL)

La URL (Localizador Uniforme de Recursos), es una dirección específica que señala la ubicación única de un recurso en la World Wide Web. Está compuesta por varios elementos, como el protocolo de transferencia HTTP o HTTPS, el dominio (la dirección del servidor), y la ruta al recurso específico. La URL actúa como un identificador único que permite a los usuarios y a los navegadores localizar y acceder a recursos como páginas web, archivos, imágenes o cualquier otro contenido en línea.

#### 2.2.2.6. Navegador web

Según Vasco, (2020) menciona que un navegador web es una aplicación informática diseñada para facilitar la exploración y visualización de contenido en la World Wide Web. Su función principal es permitir a los usuarios acceder a páginas web y navegar a través de ellas de manera intuitiva. Los navegadores interpretan el código HTML de las páginas web y presentan el contenido de forma gráfica, permitiendo la interacción mediante clics, desplazamientos y otras acciones. Además, suelen ofrecer funciones adicionales como marcadores, historial de navegación y

extensiones para personalizar la experiencia del usuario. Ejemplos comunes de navegadores web incluyen Google Chrome, Mozilla Firefox, Microsoft Edge y Safari. (p.1).

### 2.2.3. Metodologías de desarrollo de software

**Tabla 1.** Metodologías de desarrollo más utilizadas en el desarrollo de software.

Metodología	Roles clave	Ciclos de desarrollo	Características	Enfoque	Ventajas	Desventajas
<b>Scrum</b>	Scrum Master, Product Owner, Equipo de Desarrollo.	Sprints, que suelen durar entre 2 y 4 semanas.	Se centra en la gestión de tareas y la mejora continua a través de reuniones regulares (diarias, de planificación del sprint, de revisión y de retrospectiva).	Flexibilidad y adaptabilidad a cambios.	Flexibilidad y adaptabilidad. Transparencia. Fomenta la colaboración.	Dependencia de roles. Puede ser rígido.
<b>Kanban</b>	No define roles específicos.	Continuo, sin sprints fijos.	Visualización del flujo de trabajo a través de un tablero Kanban, limitación del trabajo en progreso.	Eficiencia y minimización del tiempo de ciclo.	Flexibilidad Mejora continua Visualización del trabajo	Menos estructurado Gestión del cambio
<b>Lean</b>	No define roles específicos.	Enfoque continuo en la entrega de valor.	Eliminación de desperdicios (todo lo que no aporte valor al cliente), mejora continua y decisiones basadas en el conocimiento real.	Maximización del valor y minimización de desperdicios.	Eficiencia Valor al cliente Adaptable	Curva de aprendizaje Riesgo de sobrecarga
<b>XP</b>	Programador, Cliente, Coach, Tracker.	Iteraciones cortas.	Programación en parejas, desarrollo orientado a pruebas, diseño simple, integración continua, y refactorización de código.	Mejora de la calidad del software y capacidad de adaptarse a cambios rápidamente.	Alta calidad del software. Respuesta rápida a cambios. Comunicación efectiva. Feedback constante.	Requiere compromiso. Dificultad en proyectos a gran escala. Limitaciones en la documentación. Resistencia al cambio.

Del mismo modo, es importante realizar una comparación entre las metodologías ágiles de desarrollo de software teniendo como base algunos parámetros que ayudarán a determinar cuál de los tipos de metodología es el más adecuado para utilizar en el presente proyecto de investigación. A continuación, se muestra la comparación de las metodologías ágiles de desarrollo de software.

**Tabla 2.** Comparación de metodologías ágiles de desarrollo.

<b>Características</b>	<b>Metodologías de desarrollo</b>			
	<b>Scrum</b>	<b>XP</b>	<b>Kaban</b>	<b>Lean</b>
Tamaño del proyecto pequeños y medianos	✓	✓	✓	✓
Modelo de desarrollo iterativo y rápido.	✓	✓	-	-
Desarrollo orientado a cliente	✓	✓	✓	✓
Implementación de equipos pequeños.	✓	✓	✓	✓
Búsqueda de excelencia técnica.	✓	✓	-	✓
Pruebas TDD.	✓	✓	-	-
Distribución física.	-	✓	-	-

### 2.2.3.1. Metodología XP

Según (Vila, 2016) La Metodología XP (Programación Extrema) es un enfoque ágil para el desarrollo de software que destaca la flexibilidad y la adaptabilidad. Se centra en la colaboración estrecha entre los miembros del equipo, la retroalimentación continua y la entrega frecuente de versiones del producto. XP pone énfasis en la comunicación abierta, la simplicidad en el diseño y la capacidad de respuesta a los cambios en los requisitos del cliente. Además, promueve prácticas como la programación en parejas, las pruebas automáticas y el desarrollo incremental, con el objetivo de mejorar la calidad del software y satisfacer de manera eficiente las necesidades del cliente. (p.1)

Esta metodología Programación Extrema está compuesta por 5 etapas como:



**Figura 3.** Etapas en la metodología de desarrollo ágil XP.

En que consiste cada una se lo describe a continuación según SINNAPS (2021):

**Planificación del proyecto:** En esta fase, se determinan los objetivos y requisitos del proyecto. Se establece un plan que incluye la definición de tareas, asignación de recursos y estimación de tiempos.

**Diseño:** Aquí se crea la estructura y arquitectura del software. Se establecen las pautas para la implementación, considerando la simplicidad y la adaptabilidad a posibles cambios. Como énfasis en esta etapa se crean las tarjetas CRC (Clase-Responsabilidad-Colaboración).

**Codificación:** Durante esta fase, los desarrolladores escriben el código necesario para implementar las funcionalidades definidas en la fase de diseño. Se fomenta la colaboración, y se aplican prácticas como la programación en parejas.

**Pruebas:** En esta etapa, se llevan a cabo pruebas exhaustivas del software para detectar y corregir errores. Las pruebas automatizadas son fundamentales para garantizar la calidad del código.

**Lanzamiento:** La fase final implica la entrega del producto al cliente. La retroalimentación obtenida durante el proceso puede influir en futuras iteraciones del desarrollo. (p.60)

La metodología XP (Programación Extrema) encontramos como características y ventajas las siguientes según Sergio (2020):

- Comunicación Constante
- Flexibilidad
- Desarrollo Incremental.
- Pruebas Automatizadas.
- Programación en Parejas.
- Adaptabilidad.
- Ahorro de tiempo y dinero.
- Se aplica a cualquier lenguaje de programación.
- Cliente Involucrado.
- Mejora Continua (p.12).

Es muy notorio que las ventajas que nos proporciona la metodología ágil XP, está por encima de las desventajas estas ventajas hacen que la metodología XP sea atractiva para proyectos que requieren flexibilidad, rápida adaptación a cambios y una fuerte colaboración entre los miembros del equipo.

#### 2.2.4. Herramientas para el desarrollo de software

##### 2.2.4.1. Lenguaje de programación

Un lenguaje de programación, según lo expresado por Carballo (2019), se refiere a cualquier idioma artificial que posibilita definir una secuencia de instrucciones para ser procesadas por una computadora. Este conjunto de caracteres, reglas y operaciones con significados predefinidos permite la creación de programas. La definición de un lenguaje de programación abarca tres aspectos fundamentales:

- Léxico, que determina los símbolos y reglas para la formación de palabras.
- Sintaxis, que organiza las palabras en frases coherentes
- Semántica, que otorga significado a esas frases.

Carballo también señala distintos tipos de lenguajes de programación que se clasifican en:

- Lenguaje de máquina
- Lenguaje de bajo nivel
- Lenguaje de alto nivel, que incluyen variedad de lenguajes como, C#, C++, Java, PHP, VB.Net, entre otros (Carballo, 2019, p.10).

**Tabla 3.** Cuadro comparativo de los lenguajes de programación más utilizados para aplicaciones web.

Característica	C#	Java	Python
Sistemas operativos	Windows	Ubuntu, Windows	Windows, MacOS y Linux
Web App Framework	Asp.Net Core, Asp.Net MVC, Asp.Net Razor	Angular, Wicket, Grails	Apache, Flask, Django, web2Py, Bottle
Rendimiento	Alto	Alto	Medio
Seguridad	Alto	Moderado	Moderado
Soporte de base de datos	Estable	Estable	Comparativamente débil
Curva de aprendizaje	Simple	Difícil	Simple
Apoyo comunitario	Amplio	Amplio	Amplio

**Fuente:** (Sheldon ,2021)

#### 2.2.4.2. Lenguaje C#

Es un lenguaje muy conocido y popular para la programación moderna, sirve para el desarrollo de aplicaciones seguras y sólidas. Está basado en objetos en C, una de las ventajas de este lenguaje es que los programadores que están familiarizados con lenguajes como C, C++, java y JavaScript resultan fáciles acoplarse.

C# es un lenguaje de programación orientado a componentes, orientado a objetos. C# proporciona construcciones de lenguaje para admitir directamente estos conceptos, por lo que se trata de un lenguaje natural en el que crear y usar componentes de software. (Un Paseo Por C# - Guía de C# | Microsoft Docs, 2021, p.1).

El lenguaje de programación C# fue seleccionado porque presenta características favorables para el desarrollo de un aplicativo web. Algunas de estas características son su baja curva de aprendizaje, sintaxis fácil de comprender, seguridad y multiplataforma, los cuales hacen que sea una buena alternativa de lenguaje de programación para el desarrollo de sistemas web.

#### 2.2.4.3. Variables

Dentro de la programación hay varios términos que se maneja, entre los más principales son las variables que es un dato en el cual se almacena información, para hacer uso de estas variables se necesita varias funciones que pueden ser escritas en cualquier lenguaje de programación.

Las variables en la programación “representan un valor determinante en la operación a programar dentro del sistema de información, para ellos se interactúan letras, estas pueden ser las primeras que integran el nombre de un objeto para poder ser visualizado en el sistema.” (Gustavo Chavana, 2021, p. 1).

#### 2.2.4.4. Tipos de variables

Las variables se pueden encontrar de varios tipos los cuales están clasificados de la siguiente manera:

Int. – Son tipos de variables enteros, ejemplo de ellos, 1,2,3, 4...

Float. – Son tipos de variables que contienen decimales, ejemplo de ellos, 1.33, 2.54, 123.21

Double. – También denominado punto flotante, este contiene valores que se asemejan a fracciones, elevados a exponentes, etc. Ejemplo de ellos, 12.700 es igual a  $1.27 \times 10^4$

Char. – Conocido como carácter, este tipo de variables se utiliza varios tipos de caracteres, ejemplo de ellos, Y, T, %, &, #, 52

Void. – Conocido como variable booleana, se utiliza para el almacenamiento de acuerdo con 2 dualidades como puede ser “verdadero” o “falso” o también se puede escribir un número entero.

#### 2.2.4.5. Objeto

Un objeto viene dado por una identificación similar a la del mundo real de un animal, lugar o cosa, como, por ejemplo: silla, mesa, perro, auto, etc.

Ahora bien, dichos objetos tienen dos características que se denomina estado y comportamiento, por ejemplo: nombre, color, raza, altura, etc. Y el comportamiento que puede ser llorar, aullar, comer, dormir, etc.

En la programación orientada a objetos (POO), su implementación viene dada de la misma manera, en donde se define que un objeto es un conjunto de datos y funciones relacionados, que también se denomina métodos y atributos, ejemplo de ellos:

Objeto: Tarjeta de crédito

Atributo: Nombre, fecha, saldo

Métodos: Transacciones, retiros, depósitos.

#### 2.2.4.6. Clases

Se puede afirmar que una clase es una plantilla donde se definen las variables y sus métodos más utilizados para los objetos de cierto tipo. Si ya se comprendió lo que es un objeto, se puede decir que un objeto es una instancia de una clase.

#### 2.2.4.7. Mensajes

Los mensajes son muy importantes dentro de los objetos debido a que estos interactúan con dicho objeto, un claro ejemplo es: el objeto Julio envía un mensaje "girar a la derecha" al objeto auto.

Los elementos que componen a un mensaje son los siguientes:

- a) Objeto destino (hacia el cual el mensaje es enviado)
- b) Nombre del método a llamar
- c) Parámetros que se solicitan por el método

#### 2.2.4.8. Herencia

La herencia es una estructura jerárquica donde se deriva una clase de otra clase, las clases pueden denominarse hijas y pueden hacer uso de los métodos y propiedades de la clase padre, siempre y cuando su clase sea global, si es una clase privada se utilizan funciones para hacer uso de estos. Ejemplo de ello es:

Clase padre: Mascota

Clase hijos: Perro, gato, loro, etc.

#### 2.2.4.9. Entorno de desarrollo de un sistema informático

Los entornos de desarrollo se clasifican entre los más principales: NetBeans, Atom y Visual Studio.

NetBeans. - Es un entorno de desarrollo integrado principalmente para el lenguaje de programación Java, buen editor de código, gratuito, multilenguaje, es fácil de instalar y utilizar. La herramienta NetBeans soporta el uso de tecnologías de desarrollo tanto para aplicaciones web, móviles y de escritorio Girardi, T. (2015).

Atom. - Morales, R. (2019) dice que es el entorno de desarrollo código abierto que se puede editar en macOS, Linux y Windows, posee plug-ins escritos en Node.js. La mayor parte de las aplicaciones constan con una IDE que puede ser construida utilizando

tecnologías web, son compatible con muchos programas, añaden soporte mediante el sistema de paquetes y mejora mediante intérpretes, debuggers que conecten software de terceros.

Visual Studio. - Es un editor de código fuente que se ejecuta en su escritorio y está disponible para Windows, macOS y linux, es de software libre que se distribuye bajo la licencia MIT es decir que los ejecutables están distribuidos bajo una licencia gratuita no libre. Es de código abierto y proporciona una utilidad para descargar extensiones con las que se puede personalizar esta herramienta Morales, R. (2019).

Una de las características más importantes es que viene con soporte incorporado para JavaScript, TypeScript y Node.js, cuenta con extensiones de lenguaje de programación como por ejemplo para Python, C#, C++ entre otros. Cada versión de Visual Studio Code incorpora nuevas funcionalidades De Luca, D. (2018).

#### 2.2.4.10. Comparativa de entorno de desarrollo de un software.

**Tabla 4.** Cuadro comparativo de entorno de desarrollo de un software.

Nombre	Características	Funciones	Plataformas
NetBeans	Libre y gratuito, soporta JavaScript, soluciones rápidas, menor consumo de memoria	Entorno de desarrollo libre, programación java	Windows, Linux, MacOS
Atom	Desarrollado con C++ y Node.js, utiliza tecnología web.	Código abierto y sistema de autocompletado.	Windows, Linux, MacOS
Visual Studio	Plataforma .NET	Módulo de un programa separado del cuerpo principal, procedimiento que la invoque	Windows, Linux, MacOS

En base a la comparativa como mejor opción en nuestra investigación es el uso del entorno de programación Visual Studio debido a que es de código abierto y se ofrece bajo la licencia MIT, lo cual implica que su uso y distribución, incluso en forma de ejecutables, se hace sin costo, aunque no se clasifique como libre. Además, esta herramienta contiene el framework con el que se va a desarrollar también que es la plataforma .NET.

#### 2.2.4.11. Base de datos

Una base de datos es un sistema organizado para recopilar, almacenar y gestionar datos de manera estructurada y eficiente. En su esencia, actúa como un almacén electrónico que permite guardar información de diversas categorías, como nombres, números, fechas o cualquier otro tipo de datos. Esta información se organiza en tablas interrelacionadas, lo que facilita la consulta y recuperación de datos de manera rápida (Merchán, 2019, p.1).

Una base de datos proporciona un método para ingresar, actualizar y recuperar información de manera sistemática. Además, su diseño permite establecer relaciones entre conjuntos de datos, lo que contribuye a mantener la integridad y consistencia de la información. Las bases de datos son utilizadas en una amplia variedad de contextos, desde aplicaciones empresariales y científicas hasta sistemas de gestión de contenido en línea (Silberschatz, Korth & Sudarshan, 2021, p.1).

#### 2.2.4.12. Estructura de las bases de datos

Según Coutiño (2020), la definición de la base de datos y su estructura se establecen durante el proceso de instalación. En este contexto, la base de datos se concibe como un conjunto de tablas que se pueden manipular siguiendo el modelo relacional de datos. Cada base de datos consta de elementos como:

- Tablas de catálogo de sistema (describen la estructura lógica y física de los datos).
- Archivo de configuración (almacena los valores de parámetros asignados a la base de datos).
- Registro de recuperación (incluye transacciones en curso y archivables).

En el ámbito de la base de datos, los datos relacionados se organizan en tablas, cada una compuesta por filas y columnas, similar a una hoja de cálculo. Para convertir listas de datos en tablas, se crea una tabla para cada tipo de entidad, como productos, ventas, clientes y pedidos. Cada fila en una tabla se denomina "registro" y contiene información específica sobre algo o alguien, como un cliente particular.

Por otro lado, las columnas (también conocidas como "campos" o "atributos") albergan un único tipo de información en cada registro, como las direcciones de todos los clientes enlistados en la tabla.

Además, Coutiño (2020) proporciona ejemplos de tipos de datos utilizados, como CHAR para texto de longitud específica, VARCHAR para texto de longitudes

variables, TEXT para grandes cantidades de texto, INT para números enteros, y otros como FLOAT, DOUBLE y BLOB para diferentes tipos de datos (p.1).

#### 2.2.4.13. Características de las bases de datos

Basado según el autor Morales, (2019) menciona las siguientes características de una base de datos.

- Estructurar la información en tablas: actúan como contenedores lógicos para distintos conjuntos de datos. Además, establece relaciones entre estas tablas, lo que contribuye a mantener la integridad y coherencia de la información.
- Realizar consultas y recuperar datos de manera rápida y precisa: la base de datos también proporciona un medio para modificar y actualizar la información de manera controlada, garantizando la consistencia de los datos.
- La seguridad: una base de datos efectiva incorpora mecanismos para proteger la información, restringiendo el acceso a usuarios autorizados y asegurando la confidencialidad y la integridad de los datos.
- La escalabilidad es fundamental, permitiendo la gestión eficiente de grandes cantidades de datos a medida que la base de datos crece. (p.1).

#### 2.2.4.14. Roles de las bases de datos

Hacer uso de roles dentro de la base de datos ayuda a gestionar con una alta facilidad los privilegios de varios grupos de usuarios, los roles reducen el proceso de gestión de privilegios, debido a que se puede asignar privilegios a un rol y después conceder el rol a los usuarios, y finalmente cuando se requiera eliminar, anular o cancelar los privilegios para un usuario lo que se debe realizar es anular la autorización del rol del usuario, en lugar de cancelar cada privilegio individual.

#### 2.2.4.15. SGBD

Un Sistema Gestor de Base de Datos (SGBD) es una herramienta esencial en el ámbito de la informática encargada de administrar y facilitar la gestión de bases de datos. Su función principal radica en proporcionar una interfaz eficiente entre los usuarios y los datos almacenados.

Este sistema ofrece una serie de funcionalidades clave, como la creación y manipulación de bases de datos, la definición de su estructura, y la capacidad de realizar consultas para recuperar información de manera rápida y precisa. Además, el SGBD se encarga de garantizar la integridad de los datos, implementando

mecanismos para mantener la coherencia y seguridad de la información almacenada. (Merchán, 2019, p.1).

Según Merchán (2019) un aspecto fundamental del SGBD es su capacidad para gestionar la concurrencia, permitiendo que varios usuarios accedan y modifiquen la base de datos simultáneamente de manera controlada y sin conflictos. También incorpora mecanismos de respaldo y recuperación para prevenir la pérdida de datos y asegurar la continuidad operativa en caso de fallos o errores (p.1).

Entendiendo estos conceptos vamos a realizar una comparativa de los sistemas gestores de bases de datos más utilizados en el desarrollo web, recalcando que debe poseer la principal función de estructurar y organizar los datos correspondientes a los registros de cada actividad y proceso inmersos en la hacienda ProduAgro. De esta manera la información que conlleva la hacienda se almacenará en una base de datos donde se podrá acceder de forma más fácil que optando por los métodos convencionales.

**Tabla 5.** Cuadro comparativo de sistemas gestores de bases de datos.

<b>Criterio</b>	<b>SQL Server</b>	<b>MySQL</b>	<b>Oracle</b>
Costo	Bajo	Bajo	Alto
Seguridad	Alto	Medio	Alto
Escalabilidad	Alto	Bajo	Alto
Disponibilidad	Alto	Medio	Alto
Multiplataforma	Alto	Alto	Alto
Autenticación	Alto	Bajo	Alto
Soporte técnico	Alto	Bajo	Alto

**Fuente:** (Florencio, 2017).

Las bases de datos constituyen un elemento crucial para la interacción con sistemas web. Por ello, se llevó a cabo una evaluación de tres gestores de bases de datos potenciales para el diseño físico. Es importante considerar criterios como escalabilidad, seguridad, compatibilidad con diversas plataformas y disponibilidad, ya que estos factores demuestran la capacidad del sistema para adaptarse a variaciones y su fortaleza frente a ataques cibernéticos. En este contexto, se decidió optar por el gestor de bases de datos SQL Server, puesto que ofrece una solución más económica en comparación con Oracle, el cual igualmente cumple con criterios de elevada calidad.

#### 2.2.4.16. Cliente web

Según Gonzáles y Cordero, (2001) afirma que es una aplicación o programa informático que opera en un dispositivo, como una computadora o un dispositivo móvil, y se conecta a servidores remotos a través de Internet para acceder y visualizar contenido. Este tipo de software permite a los usuarios interactuar con servicios en línea, navegar por sitios web, y acceder a información, recursos o aplicaciones alojadas en servidores a través de un navegador web (p.8).

#### 2.2.4.17. Framework Web

Para el desarrollo de aplicaciones web, "un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado" (Villalobos et al., 2019, p. 179).

Herramientas que, mediante una estructura específica, permiten la escalabilidad de aplicaciones web. Están orientadas al desarrollo de sitios con características dinámicas, así como aplicaciones y servicios relacionados con la web (Ilusiónestudio, 2018).

- .NET Framework: Este marco de trabajo es una estructura diseñada en el lenguaje de programación C#, enfocada en el desarrollo ágil de aplicaciones web con funcionalidades de nivel considerable de complejidad. Aglutina una variedad de características que facilitan la incorporación de principios de velocidad, seguridad y escalabilidad en el software producido (Ingenio Learning, 2020).

#### 2.2.4.18. Arquitecturas de desarrollo de software

De acuerdo con Huamán (2018) "es necesario seleccionar una arquitectura adecuada que brinde las funcionalidades esperadas y los atributos de calidad". Por lo tanto, se debe conocer los principales tipos de arquitecturas de desarrollo de software para aplicarlas de manera correcta. Algunos de los principales tipos más conocidos son:

- Arquitectura Modelo – Vista – Controlador
- Arquitectura Cliente – Servidor
- Arquitectura de Capas.

#### a) Modelo MVC (Modelo – Vista - Controlador)

El MVC es un modelo de arquitectónico de software que sirve para estructurar sistemas, es decir, sirve para clasificar la información, la lógica de un sistema y la interfaz, permitiendo realizar aplicativos más robustos, además esta arquitectura nace con la necesidad de crear sistemas escalables y organizados que puedan ser modificados en cada componente sin que se afecte a los demás. (García, 2017).

Esta arquitectura hoy en día es muy utilizada en el mundo del desarrollo web, en efecto, ayuda a los desarrolladores a no crear confusión en las líneas de código gracias a que mantiene una estructura organizada que permite que el desarrollo de aplicaciones web de forma más eficiente.

Los principales componentes del modelo MVC son los siguientes:

- Modelo: Es el componente se encarga de manipular, gestionar y actualizar los datos. En este apartado se realiza las consultas, búsquedas, filtros y actualizaciones.
- Vista: Este componente es la parte visible del aplicativo, es decir, es todo lo que el usuario final podrá ver del sistema como ventanas, páginas y formularios en base a las peticiones. Para un programador este parte es conocida como frontend que contiene código (HTML, CSS, HTML5, JavaScript).
- Controlador: Este componente es el punto de comunicación entre el Modelo y la Vista, se encarga de gestionar instrucciones generadas por el usuario después las procesa y solicita los datos necesarios al Modelo para ser mostrados en la Vista.

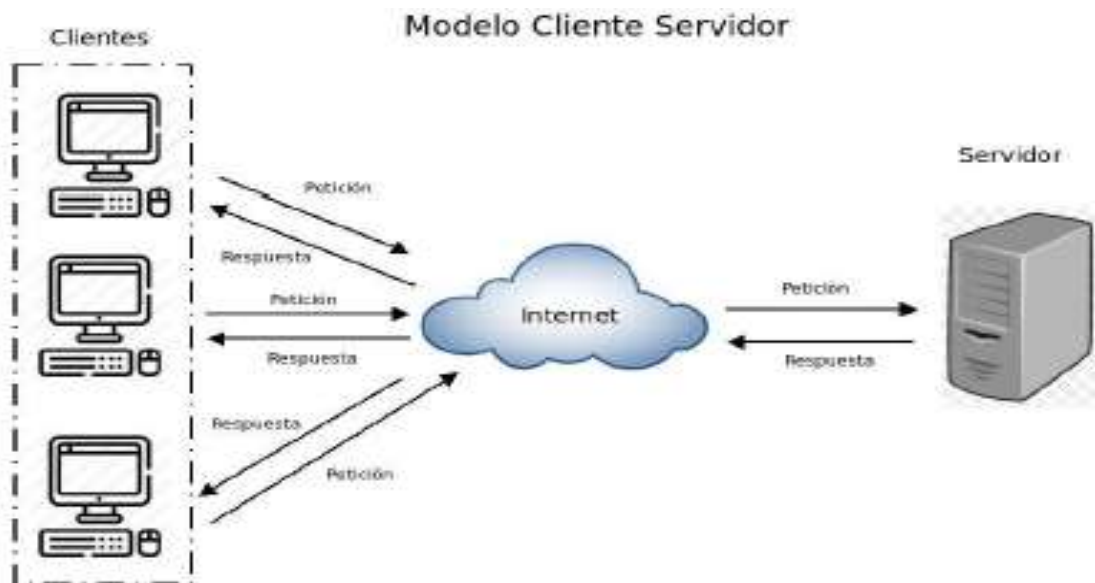


**Figura 4.** Esquema de la arquitectura MVC.  
**Fuente:** (García, 2017).

#### b) Arquitectura cliente – servidor

Es uno de los principales tipos de arquitectura de desarrollo de software y uno de los más utilizados. De acuerdo con Schiaffarino (2019) “la principal característica de esta arquitectura es la comunicación entre el servidor y el o los clientes, donde el servidor actúa como depósito de datos y tiene la función de un sistema gestor de bases de datos”. Esta arquitectura se orienta en brindar servicios a empresas con varios computadores conectados a un servidor para acceder a archivos o recursos. Los principales componentes del modelo cliente – servidor son los siguientes:

- Cliente: Este componente es el encargado de demandar o solicitar servicios. Puede ser un ordenador o una aplicación que requiera información que provenga de la red.
- Servidor: Este componente hace referencia a un proveedor de servicios. Al igual que el cliente puede ser un ordenador o aplicación que envía información al resto de agentes de red.

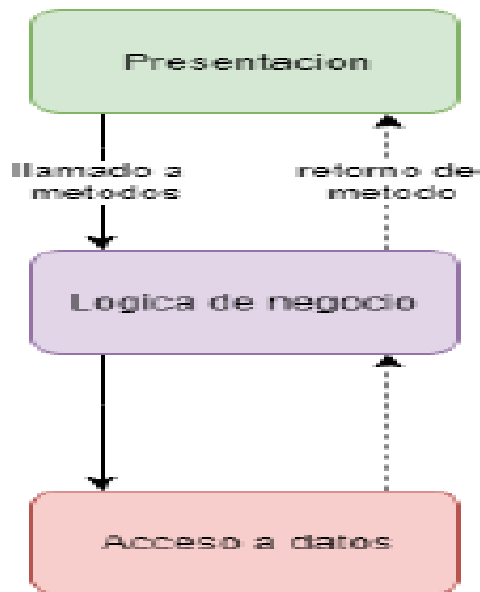


**Figura 5.** Esquema de la arquitectura Cliente/Servidor.  
**Fuente:** (Schiaffarino, 2019).

### c) Arquitectura por Capas

Esta arquitectura se caracteriza por ser sencilla de implementar y de comprender para los desarrolladores. De acuerdo con Bertucelli (2019) este modelo “se basa en idealizar nuestro sistema en capas, las cuales deben especificar las operaciones y funciones que pueden realizar, para lo cual la separación de capas más empleada es la de tres capas que son: presentación, lógica de negocio y acceso”. Los principales componentes de esta arquitectura son:

- Presentación: Representa los datos y atiende las solicitudes de los clientes.
- Lógica de negocio: Contiene los requerimientos funcionales del sistema.
- Acceso a datos: A través de ella se puede obtener o almacenar los datos que empleará el sistema.



**Figura 6.** Esquema de la arquitectura por capas.  
**Fuente:** (Bertucelli, 2019).

Las arquitecturas mencionadas presentan diferentes características y elementos, lo que las hace idóneas para varios escenarios de uso. Por ejemplo, la arquitectura cliente/servidor es frecuentemente elegida por entidades financieras gracias a su estructura en capas, la cual asigna funciones específicas a cada una. Asimismo, la arquitectura MVC se destaca en aplicaciones que gestionan usuarios y permiten interacciones. No obstante, resulta crucial realizar comparativas entre ellas para identificar sus particularidades distintivas y determinar cuál ofrece las mejores prestaciones para el desarrollo de nuestro sistema informático en la hacienda ProduAgro.

**Tabla 6.** Comparación de tipos de arquitecturas de software.

Características	Cliente Servidor	Modelo Controlador Vista	Arquitectura por capas
Escalabilidad	✓	✓	✓
Facilidad de mantenimiento	✓	✓	✓
Flexibilidad	✓	✓	✓
Versatilidad	✓	✓	✓
Control		✓	
Solución		✓	
Reusabilidad		✓	
Seguridad		✓	

En la investigación actual, optamos por utilizar el Framework .NET para gestionar una arquitectura de aplicación que sigue el patrón de tres capas, denominado MVC

(Modelo, Vista, Controlador). Este enfoque se considera esencial, ya que distribuye eficazmente el flujo de la aplicación en línea se divide en tres elementos lógicos, asegurando de esta manera la sostenibilidad y capacidad de escalabilidad a largo plazo. Al desarrollar un sistema informático que implica la creación de objetos y la gestión de información específica de la hacienda ProduAgro, es necesario utilizar un lenguaje de programación. Además, se requiere establecer una conexión con una base de datos para optimizar la eficiencia en el manejo de estos datos.

Información

#### 2.2.5. Ingeniería de Procesos

El área de la ingeniería de procesos se dedica al diseño y desarrollo de procedimientos con el objetivo de crear sistemas de producción eficientes. Cuatrecasas (2020) indica que esta sección de la organización es responsable de desarrollar los procedimientos, operaciones y otras actividades necesarias para la producción de un bien o servicio, que se planea o se realiza simultáneamente con su diseño y finalización (p.18).

Para esta investigación y sistematización de los procesos es necesario estudiar tanto de forma individual como de forma grupal los procesos correspondientes al manejo de información de la Hacienda ProduAgro. Para esto es necesario levantar información mediante fichas técnicas y utilizar flujogramas de procesos para representar las actividades de las actividades que mantiene la hacienda.

##### 2.2.5.1. Proceso

Un proceso se fundamenta en tareas que forman parte del sistema de producción de una empresa. Cuatrecasas (2020) lo describe como una serie de actividades estandarizadas que se llevan a cabo para lograr la creación de un producto o servicio. Estas actividades buscan maximizar la calidad del bien o servicio final, minimizando al mismo tiempo los costos y el tiempo de producción (p.61).

Para esta investigación los procesos permiten los pasos correspondientes a cada actividad administrativa y de gestión que tiene la hacienda ProduAgro, lo que permite comprender el proceso a seguir para cumplir con los objetivos presentes en esta investigación.

#### 2.2.5.2. Gestión de procesos

La administración de procesos implica la organización y sincronización de las tareas realizadas para alcanzar las metas comerciales. Monga (2018) explica que esta gestión se enfoca en identificar y comprender la secuencia de datos de entrada y salida, que se combinan para producir un bien o servicio. Este debe satisfacer las necesidades del cliente (p.9).

La gestión de procesos permite conocer y comprender como las actividades que abarca la administración y gestión de la hacienda ProduAgro y de esta forma entender la secuencia de entradas y salidas de los datos sobre los animales bovinos que forman parte y es principal tema de nuestra investigación la ganadería de la hacienda.

#### 2.2.6. Ingeniería de Software

La ingeniería de software, una disciplina derivada de las Ciencias Computacionales, se orienta hacia la creación de productos de software. En el contexto de la Hacienda ProduAgro y su administración y gestión de información, esta área puede reinterpretarse siguiendo la definición de Moreno, Flores, Rodríguez y Arzola (2019) como un proceso que engloba actividades de modelado para definir los requisitos y representar el software que se desarrollará. La ingeniería de requisitos se convierte así en un método crucial para identificar, analizar, documentar y verificar las necesidades específicas del software destinado a optimizar la gestión de información en ProduAgro (p.161).

La adopción de la ingeniería de software, en particular la metodología XP (eXtreme Programming), facilita un análisis de requisitos eficaz. Esto es especialmente relevante en el desarrollo de sistemas que mejoran la gestión de información ganadera, como los que podrían emplearse en la hacienda ProduAgro. Durante la fase de desarrollo, se recomienda el uso de tecnologías como el framework Asp.Net, conocido por su robusta infraestructura en seguridad, rendimiento, eficiencia y capacidad de escalado. Estas cualidades son esenciales para el desarrollo de aplicaciones web que automatizan y sistematizan procesos críticos, como la administración de recursos y la gestión de datos de los bovinos, ofreciendo interfaces intuitivas y accesibles para los usuarios.

### 2.2.6.1. Modelo de desarrollo de un software

García, F. (2018) manifiesta que un modelo de desarrollo de software es una representación abstracta de las actividades a realizar en el desarrollo. Para esto se han establecido metodologías y modelos para desarrollar un producto (software) en un tiempo determinado y recursos que dispone una empresa o entidad. Estos procesos poseen una fase de requisitos, diseño, implementación, verificación y mantenimiento para estructurar el desarrollo del producto de una forma ordenada y adecuada.

### 2.2.6.2. Etapas del Desarrollo de Software en base a la Norma ISO 12207

Las etapas dentro del desarrollo de software, como se describen a continuación, están alineadas con los procesos definidos por la Norma ISO 12207 - Modelos de Ciclos de Vida del Software, que proporciona un marco integral para los procesos involucrados en el desarrollo y mantenimiento del software.

Planificación (ISO 12207 - Proceso de Adquisición):

La fase inicial del desarrollo según Prieto (2023) se correlaciona con el proceso de adquisición de la ISO 12207. En esta etapa, se definen el alcance del proyecto, se analizan los riesgos, se realiza un estudio de viabilidad y se asignan recursos. La planificación meticulosa sienta las bases para un desarrollo orientado al éxito, alineándose con las mejores prácticas de gestión de proyectos según la norma.

Análisis (ISO 12207 - Proceso de Desarrollo):

El proceso de desarrollo según ISO 12207 incluye actividades de análisis exhaustivo para comprender los requisitos del software, lo cual coincide con la fase de análisis de Prieto (2023). Durante esta etapa, se desentrañan las expectativas del software y se realiza una investigación para obtener una comprensión precisa de los requisitos.

Diseño (ISO 12207 - Proceso de Desarrollo):

La fase de diseño de Prieto (2023) se vincula con el proceso de desarrollo de la ISO 12207. Ambos destacan la importancia de explorar alternativas y considerar aspectos relacionados con la implementación tecnológica, incluyendo hardware, lenguaje y red.

Programación (ISO 12207 - Proceso de Desarrollo):

La etapa de programación en el desarrollo de software, según Prieto (2023), se alinea con el proceso de desarrollo de la ISO 12207, que implica la creación del código utilizando el lenguaje de programación adecuado para producir el software.

Pruebas (ISO 12207 - Proceso de Desarrollo):

La fase de pruebas, según Prieto (2023), coincide con el proceso de desarrollo de la ISO 12207, que adquiere importancia crucial antes de que el software llegue al usuario. La ISO 12207 destaca la necesidad de verificar que el programa ejecute las tareas especificadas y detectar posibles fallos.

Implementación (ISO 12207 - Proceso de Implementación):

La implementación, según Prieto (2023), se relaciona con el proceso de implementación de la ISO 12207. Ambos resaltan la necesidad de habilitar el software para que el usuario lo utilice, resolviendo problemas potenciales y planificando el entorno considerando las interdependencias entre los elementos del programa.

Mantenimiento (ISO 12207 - Proceso de Mantenimiento):

La fase de mantenimiento de Prieto (2023) concuerda con el proceso de mantenimiento de la ISO 12207. Aunque a menudo se subestima, resulta esencial mantener, optimizar y mejorar el programa para corregir errores, adaptarse a nuevas necesidades o añadir funcionalidades adicionales.

Documentación (ISO 12207 - Proceso de Documentación):

La necesidad de documentar cada etapa del proyecto, incluyendo modelos, pruebas, objetivos de correcciones, usabilidad y ajustes, según Prieto (2023), se alinea con el proceso de documentación de la ISO 12207.

### 2.2.6.3. Calidad del Software en base a la Norma ISO/IEC 9126

La Norma ISO/IEC 9126 se centra en la calidad del software, abordando características como funcionalidad, confiabilidad, usabilidad y eficiencia. En el caso de la hacienda ProduAgro, la implementación del sistema informático para la gestión y administración busca respaldar dichos criterios de calidad definidos por esta norma.

Funcionalidad (ISO/IEC 9126 - Característica de Software)

La implementación del sistema busca mejorar la funcionalidad del proceso ganadero, proporcionando un control detallado sobre los animales y procesos que

se ejecutan. Se asegurará de cumplir con los requisitos definidos por el usuario y las necesidades específicas de la hacienda ProduAgro.

#### Confiabilidad (ISO/IEC 9126 - Característica de Software)

El sistema informático para la gestión y administración aborda la necesidad de un sistema confiable que proporcione información veraz y rápida sobre los procesos que se desarrollaran como la gestión reproductiva, gestión animal, inventario y reportes. Se realizarán pruebas exhaustivas para garantizar la confiabilidad de los datos y la consistencia del sistema.

#### Usabilidad (ISO/IEC 9126 - Característica de Software)

Se considerará la usabilidad del sistema, diseñando interfaces intuitivas y fáciles de usar. Esto se alinea con la necesidad de proporcionar herramientas prácticas para los usuarios con poca interacción con una aplicación, como se describe en la justificación.

#### Eficiencia (ISO/IEC 9126 - Característica de Software)

La eficiencia del sistema será un enfoque clave, asegurando que las operaciones sean realizadas de manera rápida y efectiva. Se optimizarán los procesos para mejorar la eficiencia en la gestión de la hacienda ProduAgro.

### 2.2.7. Administrar y Gestionar información

#### 2.2.7.1. Información

La información puede manifestarse en diferentes formatos, como texto, gráficos, números o multimedia, y su importancia radica en su capacidad para apoyar la toma de decisiones, resolver problemas y ofrecer una comprensión más profunda de situaciones o fenómenos. La calidad de la información en informática se mide por su precisión, relevancia, actualidad y presentación clara, contribuyendo así al proceso de toma de decisiones y al logro de objetivos específicos en el contexto tecnológico.

#### 2.2.7.2. Administrar información

Según Quiroa (2021) administrar información implica supervisar y organizar datos de manera sistemática para garantizar su almacenamiento eficiente, acceso oportuno y seguridad. Este proceso incluye la clasificación, actualización y preservación de la información, así como la implementación de políticas y procedimientos para su correcto manejo (p.1).

### 2.2.7.3. Gestionar información

Según Quiroa (2021) "gestionar información en el ámbito informático va más allá de la administración básica, abarcando la planificación estratégica y el uso efectivo de la información como un recurso clave para lograr los objetivos organizativos" (p.1).

Según la Universidad de Ciencias Médicas de Cuba (2017) gestionar implica la toma de decisiones informadas, el análisis de datos para obtener conocimientos significativos y la implementación de estrategias para maximizar el valor de la información. La gestión de información se dedica a obtener beneficios, mejorar la eficiencia y respaldar decisiones estratégicas mediante la aplicación de procesos y tecnologías avanzadas. (p.1).

En base a estos conceptos se puede decir que la diferencia entre dichos términos es que administración se concentra en la logística de los datos, mientras que la gestión se orienta hacia la optimización estratégica de la información para el beneficio de la hacienda ProduAgro en este caso.

### 2.2.7.4. Registro de datos

El registro de datos en un sistema informático implica el acto de almacenar información de manera organizada y estructurada. Este proceso es fundamental para mantener un seguimiento preciso de los datos, permitiendo su posterior consulta, actualización y análisis. El registro asegura que la información se capture de manera sistemática y esté disponible para su uso eficiente en el sistema (Tokola et al. 2019, p.132).

## 2.2.8. Hacienda Ganadera

### 2.2.8.1. Software Ganadero

Un software ganadero es una aplicación optimizada para llevar a cabo una administración de animales por cabeza de manera eficiente, con la finalidad de almacenar la información de todos los procesos que se están realizando en las haciendas de manera clara y comprensible para la toma de decisiones, una manera muy diferente a la que las haciendas están acostumbradas por su uso de técnicas ambiguas como Excel, hojas de cuaderno, etc.

#### 2.2.8.2. Finca Agropecuaria-Ganadera

Según lo expresado por Nieto (2019) las fincas agropecuarias y ganaderas constituyen sistemas complejos en los cuales interactúan diversos elementos para alcanzar sus metas. Se considera que estas fincas representan unidades integradas de producción, comercialización y servicios, con el objetivo fundamental de maximizar sus beneficios. Para lograr esta meta, es esencial la combinación efectiva de los factores de producción, tales como tierra, capital, trabajo y el elemento empresarial (p.71).

En resumen, se destaca la importancia de la interacción equilibrada de estos factores para el éxito y la maximización de beneficios en el contexto de las actividades agropecuarias y ganaderas.

Las fincas al igual que las haciendas ganaderas pertenecen al sector primario que tiene como procesos la crianza, domesticación y manejo de animales bovinos para satisfacer las necesidades de la venta de su mayor producto que es la leche, este es un trabajo realizado por varias personas que son denominados capataz, empleados con sus roles como la crianza de terneros, control de los potreros, veterinarios, etc.

#### 2.2.8.3. Hacienda ProduAgro

En la Hacienda ProduAgro se maneja una variedad de información que se encuentran almacenadas en hojas de papel, y en pocas circunstancias en la herramienta de cálculo Excel, dando como resultado una búsqueda de información deficiente, no transparente, manipulada por terceros, dañada y hasta la pérdida de información, todo esto representa un riesgo del mal uso de la información en la hacienda ProduAgro.

Los procesos que se llevan a cabo en la hacienda son los siguientes de los cuales todos los realizan con uso de técnicas convencionales y el almacenamiento lo hacen en cuadernos, libretas o también Excel.

#### 2.2.8.4. Procesos que hace la Gestión Ganadera

Involucra el manejo integral de los animales, incluyendo seguimiento a la producción lechera, registro de muertes y descartes, monitoreo de pesos, gestión de toros, datos veterinarios, y traslados. Tradicionalmente, esta gestión se realiza con métodos convencionales, anotando la información en libretas o cuadernos y llevando los registros de manera manual sin uso de tecnología.

- Animales: Identificación y seguimiento individual o por grupo.
- Medición Lechera: Registro de la cantidad de leche producida por cada bovino.
- Muertes: Anotación de las incidencias de fallecimientos.
- Descartes: Registro de animales retirados del proceso productivo.
- Pesos: Monitoreo del peso de los animales para su gestión nutricional y de salud.
- Toros: Gestión de la información relacionada con los machos reproductores que usualmente no poseen mucha información ya que no manejan este grupo de ganado y solo lo usan para las inseminaciones naturales.
- Datos Veterinarios: Registro de tratamientos, vacunas.
- Traslados: Anotaciones sobre el movimiento de animales entre diferentes predios que tiene la hacienda ProduAgro.

#### 2.2.8.5. Gestión Reproductiva

Administración de los procesos de reproducción, incluyendo inseminaciones, chequeos, partos y abortos, con un seguimiento detallado y manual en cuadernos o libretas.

- Inseminaciones: Registro de inseminaciones artificiales o naturales realizadas a cada vaca.
- Chequeo: Monitoreo del estado reproductivo de los animales.
- Partos: Anotación de los detalles de cada parto.
- Abortos: Registro de incidentes de abortos en la ganadería.

#### 2.2.8.6. Inventario

Gestión de productos y pedidos, anotando manualmente en libretas y los pedidos realizados o recibidos.

- Productos: Registro de productos disponibles o necesarios para la operación.
- Pedidos: Anotación de pedidos realizados por clientes o para abastecimiento.

#### 2.2.8.7. Gestión de Campo

Supervisión de las actividades agrícolas y ganaderas del campo, incluyendo la asignación de tareas y recursos, llevada a cabo a través de registros manuales.

#### 2.2.8.8. Mantenimientos

Registro y seguimiento de las actividades de mantenimiento de las máquinas y equipos, anotado de manera manual en cuadernos.

#### 2.2.8.9. Producción Lechera

Control de la producción de leche, incluyendo volúmenes producidos, calidad y otros indicadores, gestionados de forma manual.

#### 2.2.8.10. Información General

Incluye el registro de parcelas, grupos de trabajo y trabajadores, manteniendo un seguimiento de la distribución del terreno, la organización del personal y sus responsabilidades, todo registrado manualmente en libretas o cuadernos.

- Predios: Registro de la información detallada de cada predio de terreno.
- Grupos: Gestión de los grupos de animales o de trabajadores asignados a tareas específicas.
- Trabajadores: Registro de la información de cada trabajador, incluyendo roles, tareas asignadas y horarios.

#### 2.2.8.11. Situación inicial y situación final

La situación inicial se refiere a cómo se realizaban todos los procesos involucrados en la administración y gestión de información de la Hacienda ProduAgro. Por otro lado, la situación final aborda los cambios implementados en los procesos de administración y gestión de información de la Hacienda tras la solución informática que fue propuesta inicialmente.

**Tabla 7.** Cuadro comparativo situación inicial y situación final.

<b>Actividades</b>	<b>Situación inicial</b>	<b>Situación final</b>
<b>Gestión Ganadera</b>	Involucra el manejo integral de los animales, incluyendo seguimiento a la producción lechera, registro de muertes y descartes, monitoreo de pesos, gestión de toros, datos veterinarios, y traslados. Actualmente, esta gestión se realiza con métodos convencionales, anotando la información en libretas y Excel.	Uso de base de datos para registrar las actividades realizadas en la hacienda que involucra la producción lechera, registro de muertes y descartes, monitoreo de pesos, gestión de toros, datos veterinarios, y traslados.
<b>Gestión Reproductiva</b>	Administración de los procesos de reproducción, incluyendo inseminaciones, chequeos, partos y abortos, con un seguimiento detallado y manual en cuadernos o libretas.	Almacenamiento de la información de la actividad gestión reproductiva en una base de datos para revisar eficiente y de manera rápida reportes de los chequeos de los animales
<b>Inventario</b>	Gestión de productos y pedidos, anotando	Esta información estará almacenada de igual manera

	manualmente en libretas y los pedidos realizados o recibidos.	en una base de datos y su registro rápido.
<b>Gestión de Campo</b>	Supervisión de las actividades agrícolas y ganaderas del campo, incluyendo la asignación de tareas y recursos, llevada a cabo a través de registros manuales en Excel.	Esta información estará de manera estadística informando que actividad esta cada potrero.
<b>Mantenimientos</b>	Registro y seguimiento de las actividades de mantenimiento de las máquinas de ordeño y equipos, anotado de manera manual en cuadernos.	Registro de mantenimiento realizado en la hacienda.
<b>Producción Lechera</b>	Control de la producción de leche general producida, calidad y otros indicadores, gestionados de forma manual.	Registro de la información y almacenamiento en una base de datos.
<b>Reportes</b>	Este apartado no lo poseen por lo cual es muy importante que lleven un reporte general de todos los animales bovinos.	Esta consulta tendrá a la mano la información de algunas actividades que se llevan en la hacienda de manera rápida.

La tabla 7 indica la comparación entre las características de cada situación respecto a la administración y gestión de información en la hacienda ProduAgro.

### **III. METODOLOGÍA**

#### **3.1. ENFOQUE METODOLÓGICO**

##### 3.1.1. Enfoque

Dentro de la metodología del presente trabajo de integración se optó por un enfoque mixto que combina lo cuantitativo con lo cualitativo:

El enfoque mixto puede ser comprendido como "(...) un proceso que recolecta analiza y vierte datos cuantitativos y cualitativos, en un mismo estudio" (Tashakkori y Teddlie, 2003, citado en Barrantes, 2014, p.100).

Enfoque cuantitativo. - cualitativo se aplica en sus componentes y requerimientos que se conforma como el color, animación, interacción, fondos, interfaz del sistema integrado entre otras características,

Enfoque cualitativo. - se va a hacer uso para la tabulación de datos de las encuestas con un formulario de preguntas organizado mediante análisis estadístico. descriptiva se indagará la información sobre los procesos que se realizan dentro de la hacienda ProduAgro.

##### 3.1.2. Tipo de Investigación

En la presente investigación se concluyó con el uso de 4 tipos de investigación cada una clasificada en el diseño y alcance para facilitar el trabajo en el desarrollo de un sistema informático para la Hacienda ProduAgro que son: Diseño No Experimental Transversal, Alcance Descriptivo e investigación Documental y de Campo.

## Diseño

Según Hernández et al. (2004), se clasifica el diseño de investigación en dos categorías principales: experimental y no experimental. En el diseño experimental, se establece una situación de control donde se manipulan de manera intencionada una o más variables independientes (causas), con el propósito de analizar las consecuencias de dicha manipulación sobre una o más variables dependientes (efectos).

En contraste, el diseño no experimental se describe como una investigación que se lleva a cabo sin la manipulación deliberada de variables, centrándose en la mera observación de los fenómenos en su entorno natural para su posterior análisis (p.145),

Según Sampiere (2004), la categoría de diseño no experimental se desglosa considerando el tiempo durante el cual se recopilan los datos. Estos son el diseño Transversal, donde la recolección de datos ocurre en un solo momento, en un tiempo único, con el propósito de describir variables y su incidencia o interrelación en un momento específico y el diseño Longitudinal, donde la recolección de datos se lleva a cabo a lo largo del tiempo en puntos o períodos, permitiendo realizar inferencias sobre el cambio, sus determinantes y consecuencias (p.214).

En el desarrollo de la investigación, se empleará el diseño no experimental transversal, ya que implica la recopilación de datos en un periodo específico sin intervenir en el entorno en el que residen los habitantes de la hacienda ProduAgro en la parroquia de San Isidro. La investigación se centrará en la observación del problema sin manipulación de variables. Este apartado detalla los tipos de investigación a ser utilizados y describe cómo se llevará a cabo la investigación objeto de este estudio.

## Alcance

Conforme a la explicación de Hernández, Fernández & Baptista (2010), al referirse al alcance de una investigación, no se debe considerar como una tipología, más bien este alcance indica el resultado que se espera obtener del estudio. Estos autores proponen cuatro posibles resultados que se pueden obtener de una investigación:

- Estudio exploratorio: Proporciona información general acerca de un fenómeno o problema poco conocido, incluyendo la identificación de posibles variables a ser estudiadas en el futuro.
- Estudio descriptivo: Ofrece información detallada sobre un fenómeno o problema con el objetivo de describir con precisión sus dimensiones (variables).
- Estudio correlacional: Suministra información sobre la relación actual entre dos o más variables, permitiendo prever su comportamiento futuro.
- Estudio explicativo: Busca entender las causas de los eventos, sucesos o fenómenos estudiados, explicando las condiciones en las que se manifiestan.

#### Investigación-estudio descriptivo

El tipo de investigación a emplear en este estudio es esencialmente de índole descriptiva y correlacional, según Hernández, Fernández & Baptista (2014) en su definición, señalan que este tipo de investigación asocia variables mediante un patrón predecible para un grupo o población, con el propósito de comprender la relación o grado de asociación entre dos o más variables en un contexto particular (p.8).

De esta manera, la investigación se apoya de forma descriptiva, ya que la información recopilada busca explicar los hechos sin alterarlos, permitiendo medir las variables de la investigación para describirlas en los términos deseados. Además, según la afirmación de Sanca (2011), la investigación descriptiva implica la descripción, registro, análisis e interpretación mediante análisis (p.623).

Así, el proyecto de investigación se centra en la obtención, clasificación y agrupación de las funciones principales de la gestión ganadera la cual maneja la hacienda de esta manera nuestra investigación no se enfocará directamente en otros procesos o actividades diferentes a la gestión del ganado bovino sino mantendrá esta información de manera general, en cuanto a la contabilidad de la hacienda ésta no se maneja en la hacienda por lo tanto ese alcance no está tomado en cuenta en nuestra investigación, lo que se busca es trabajar sobre la realidad de los hechos y lograr una interpretación adecuada de los datos de los animales bovinos para optimizar y organizar la información de la Hacienda Produagro, facilitando así una toma de decisiones más acertada.

## Investigación documental

La investigación actual se respalda en distintos tipos de documentos, como describe Sanca (2011) quien identifica los siguientes métodos:

- a) Investigación bibliográfica: Se fundamenta en la investigación y revisión de libros.
- b) Investigación hemerográfica: Se basa en artículos o ensayos de revistas y periódicos.
- c) Investigación archivista: Depende de documentos ubicados en archivos, tales como cartas, oficios, circulares, expedientes (p.623).

Adicionalmente, según Santa Paella & Feliberto Martins (2010) la investigación documental se concentra exclusivamente en la recopilación de información de diversas fuentes, indagando sobre el tema en documentos escritos u orales (p.2).

La elección de la investigación documental para esta presente investigación se justifica por la obtención de información de fuentes primarias como libros, artículos, tesis, así como fuentes secundarias como revistas, registros y ensayos.

## Investigación de campo

De acuerdo con los autores Santa Paella & Feliberto Martins (2010) la investigación de campo se define como la recolección directa de datos desde la realidad donde ocurren los hechos, sin manipular o controlar las variables (p.4).

En esta investigación optamos también por el uso de tipo campo por la necesidad de obtener datos directamente en el lugar de estudio, que en este caso es la hacienda Produagro. Este enfoque permite recopilar información de manera directa sin intervenciones que alteren las condiciones naturales del entorno de estudio.

### **3.2. IDEA A DEFENDER**

El sistema permitirá mejorar la administración y gestión de los procesos con el uso de herramientas tecnológicas en la hacienda ProduAgro.

### **3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES**

El desarrollo de este trabajo de integración curricular se relaciona a dos tipos de variables que son:

- Variable independiente: Sistema informático, seguido de un conjunto de dimensiones como lo son lenguaje de programación, frameworks y base de datos.
- Variable dependiente: Manejo de la información de la hacienda ProduAgro, con sus dimensiones que corresponde al manejo adecuado de la información acerca de la administración y gestión de información en la hacienda ProduAgro

**Tabla 8.** Operacionalización de variables: Variable independiente.

<b>Variable Independiente:</b> Sistema informático				
<b>Conceptualización</b>	<b>Dimensión</b>	<b>Indicadores</b>	<b>Técnicas</b>	<b>Instrumentos</b>
Las herramientas tecnológicas como las aplicaciones informáticas administran y gestionan la información para obtener datos confiables e instantáneos.	Lenguaje programación	de	Velocidad de ejecución	
	Framework		Cantidad de funciones/métodos.	de
			Número de métodos	
			Cantidad de procesamientos.	-Entrevista estructurada.
		Tiempo de ejecución.	-Encuesta	-Ficha
			-Ficha técnica	
	Base de datos	Numero de plataformas compatibles.		

La tabla 3 describe la variable independiente con sus respectivas dimensiones e indicadores para un mejor estudio además de las técnicas e instrumentos a utilizar.

**Tabla 9.** Operacionalización de variables: Variable dependiente.

<b>Variables Dependiente:</b> Manejo de información en la hacienda ProduAgro.				
<b>Conceptualización</b>	<b>Dimensión</b>	<b>Indicadores</b>	<b>Técnicas</b>	<b>Instrumentos</b>
Se refiere a la administración y gestión eficiente de la información en la hacienda ProduAgro.	Planificación	Número de documentos		
		Número de ganado bovino		
	Integración	Número de módulos y sub módulos	-Entrevista	
		Características del software	-Encuesta	
	Casos de prueba	Control de funcionalidades	-Ficha técnica	-Cuestionario
		Pruebas de despliegue	-Pruebas de aceptación	-Ficha
		Pruebas de carga		
	Procesamiento	Nivel de escalabilidad		
		Nivel de aseguramiento de procesos		
		Nivel de rendimiento		

La tabla 4 describe la variable dependiente con sus respectivas dimensiones e indicadores para un mejor estudio además de las técnicas e instrumentos a utilizar.

### **3.4. MÉTODOS UTILIZADOS**

#### 3.4.1. Métodos

No experimental

Se utilizó el método no experimental para evidenciar por medio de la observación como se realiza el trabajo o procesos en la administración y gestión en la hacienda ProduAgro sin intervenir de ninguna forma en el ambiente de estudio para no alterar las variables. Además, se obtendrá información de fuentes primarias, es decir, directamente de trabajadores y administrador para conocer el funcionamiento de los procesos que conlleva la hacienda.

Entrevista estructurada

Se utilizará una entrevista estructurada dirigida al responsable o persona a cargo de la hacienda ProduAgro para tratar acerca de los requerimientos funcionales y detalles del diseño de la aplicación para que se adapte el diseño y sea más amigable para los usuarios.

Encuesta

Se emplea la encuesta como técnica para recopilar información, según la definición de Arias (2006) que la describe como una "técnica que busca obtener información proporcionada por un grupo o muestra de sujetos sobre sí mismos o en relación con un tema específico" (p.72).

#### 3.4.2. Instrumento de recolección de datos

Según Tamayo (1999) las técnicas de recolección de datos se definen como "la expresión operativa del diseño de investigación y especifican concretamente cómo se llevó a cabo la investigación" (p. 126). Además, Bizquera (1990) las define como "los medios técnicos utilizados para registrar observaciones y facilitar su tratamiento" (p. 28).


En esta investigación, la técnica seleccionada es la encuesta, y el instrumento utilizado es el cuestionario. Este cuestionario consta de varias preguntas, que incluyen tanto preguntas cerradas como abiertas.

#### 3.4.3. Cuestionario

Según Tamayo y Tamayo (1998) el cuestionario es de gran utilidad en la investigación científica, ya que representa una forma específica de la técnica de observación. Este instrumento permite que el investigador dirija su atención hacia aspectos particulares y se adhiera a condiciones específicas. El cuestionario aborda los aspectos del fenómeno que se consideran esenciales; además, posibilita el aislamiento de ciertos problemas de interés principal, reduce la complejidad de la realidad a un número concreto de datos esenciales y precisa el objeto de estudio (p. 124).

#### 3.4.4. Validación del Instrumento

Hernández, Fernández y Baptista (1998) definen "la validez en términos generales como el grado en que un instrumento realmente mide la variable que se pretende medir" (p.243). Se considerarán las recomendaciones de expertos en la materia para realizar, si es necesario, ajustes pertinentes al cuestionario que se aplicará a la muestra en estudio.

	<p style="text-align: center;"><b>UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI</b></p> <p style="text-align: center;"><b>FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS</b></p> <p style="text-align: center;"><b>AMBIENTALES.</b></p> <p style="text-align: center;"><b>CARRERA DE COMPUTACIÓN</b></p>
<p><b>Tema:</b> "Sistema informático para la administración y gestión de información de la hacienda ProduAgro"</p> <p><b>O.G:</b> Desarrollar un sistema informático que permita el mejoramiento de la administración y gestión de información de la hacienda ProduAgro.</p> <p>El objetivo de esta encuesta es directamente con fines informativos para obtener los puntos de vista de personas que trabajan en la hacienda ProduAgro, de tal manera comprender sus necesidades y expectativas reales en la que están sobrellevando la información.</p> <p>Antes de iniciar la encuesta debe saber ¿Qué es un sistema informático? un sistema informático es como el cerebro de una computadora. Es un grupo de partes que trabajan juntas para hacer que su computadora haga cosas útiles, como guardar información, mostrar imágenes y ayudarlo a realizar tareas, es como un director que organiza todo para que el computador funcione como tú quieres.</p> <p>LUGAR .....FECHA: .....</p>	

### ENCUESTA

**Instrucciones. -**

Marque con una **X** en la respuesta que usted seleccione.

**1. ¿Cuál es su rango de edad?**

- ( ) 18-30
- ( ) 31-40
- ( ) 41-50
- ( ) 51 o más

**2. ¿Usted maneja información importante relacionada a la hacienda ProduAgro?**

Sí ( )

No( )

**3. ¿Qué método utiliza para el registro y manejo de información de la hacienda ProduAgro?**

( ) Cuadernos

( ) Libros diarios

( ) Hojas

( ) Computadora

( ) Teléfono Inteligente

Otro.....

**4. ¿Cómo considera los métodos de administración y gestión utilizados actualmente en la hacienda?**

( ) Muy buena

( ) Buena

( ) Regular

( ) Mala

( ) Muy mala

**5. ¿Cuáles son los problemas más frecuentes que tiene la hacienda al momento de manejar la información?**

( ) Pérdida de información

( ) Información deteriorada con el tiempo

( ) Falta de integridad en la información

( ) Poca accesibilidad

( ) Falta de organización y presentación de los datos

Otro.....

**6. ¿Cómo considera su capacidad para el uso de la tecnología?**

Ninguno ( )

Bajo ( )

Medio ( )

Alto ( )

**7. ¿Estaría dispuesto a utilizar un sistema informático (es una técnica que permite el almacenamiento y el proceso de información a través del uso de la tecnología) para la administración y gestión de información de la hacienda ProduAgro?**

Sí ( )

No ( )

**8. ¿Usted cree que, con el uso de un sistema informático, se mejore la administración y gestión de la información que usted maneja?**

Sí ( )

No ( )

**9. ¿Qué procesos le gustaría que tuviera el sistema informático?**

( ) Historial clínico

( ) Reportes de producción lechera

( ) Pesos mensuales

( ) Estado del ganado

Otro.....

**10. ¿Le gustaría que el sistema informático, abarque procesos relacionados con el estado de los potreros, fertilizaciones y abonos?**

Sí ( )

No ( )

Porque.....

## **IV. RESULTADOS Y DISCUSIÓN**

### **4.1. RESULTADOS**

#### 4.1.1. ANÁLISIS ESTADÍSTICO

#### 4.1.2. Población

En el ámbito de la investigación, la población a emplear se clasifica como finita, ya que está compuesta por el administrador y los empleados de la hacienda ProduAgro. En consideración al estudio dirigido a los empleados en activo, que suman un total de 8 personas, se ha optado por seleccionar 6 empleados y 1 individuo encargado de la administración, para abordar de manera adecuada los requisitos de la investigación.

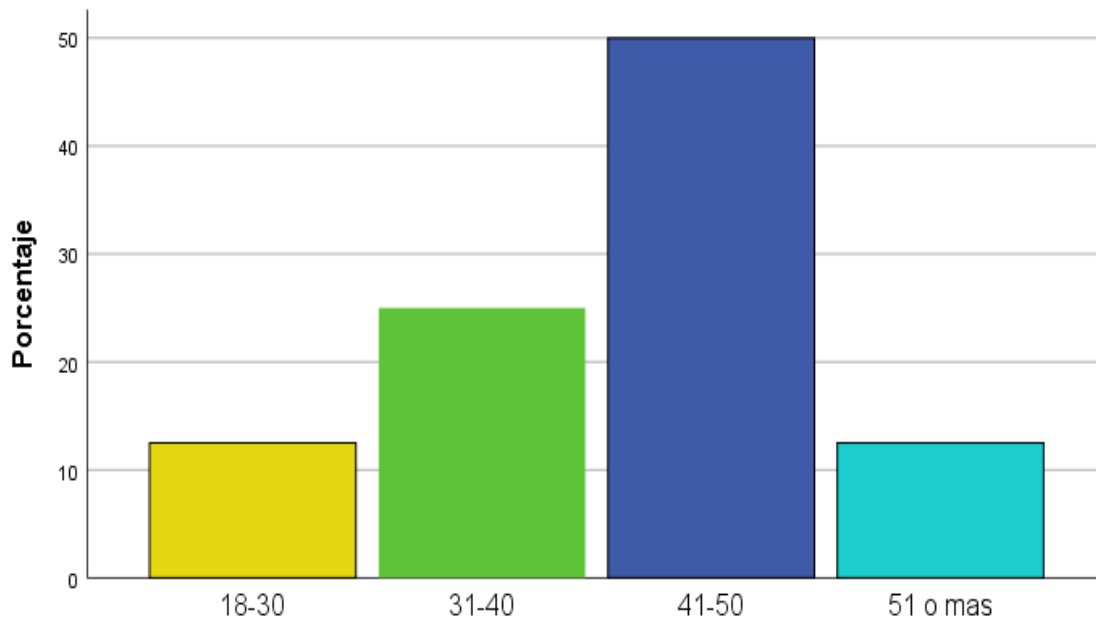
#### 4.1.3. Muestra

La población del proyecto son las personas que realizan la administración y gestión de la hacienda ProduAgro, por lo tanto, al conocer que la población es pequeña su muestreo se basa en un censo que es aplicado a todos los empleados.

#### 4.1.4. Tabulación de la encuesta

La encuesta fue aplicada a ocho trabajadores de la hacienda ProduAgro, entre miembros administrativos y trabajadores agrícolas, con los siguientes resultados.

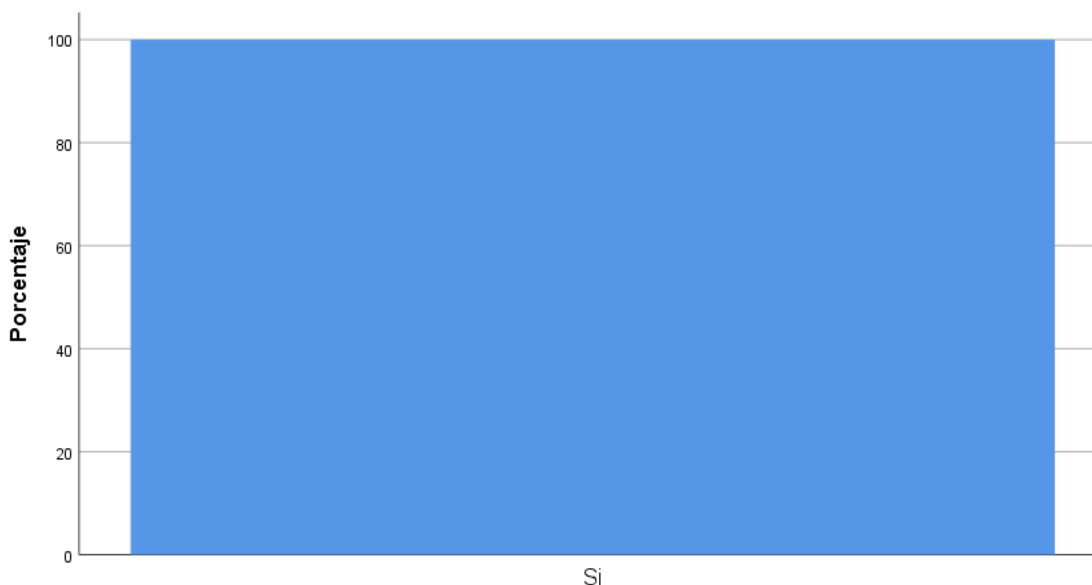
### 1. ¿Cuál es su rango de edad?



**Figura 7.** Grafico de análisis del cuestionario primera pregunta.

Más del 60% de los encuestados tienen una edad mayor a los 40 años y el 40% con una edad entre 18 y 40 años. Con los datos recolectados el sistema informático deberá ser enfocado a personas mayores, con las necesidades que una persona adulta necesita.

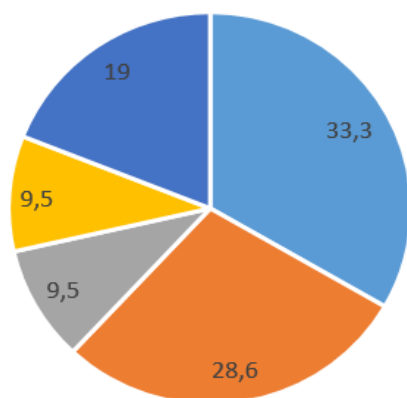
### 2. ¿Usted maneja información importante relacionada a la hacienda ProduAgro?



**Figura 8.** Grafico de análisis del cuestionario segunda pregunta.

Los resultados de la presente pregunta arrojan que todas las personas que fueron encuestadas manejan información importante relacionada a los procesos desarrollados en la hacienda ProduAgro.

### 3. ¿Qué método utiliza para el registro y manejo de información de la hacienda ProduAgro?

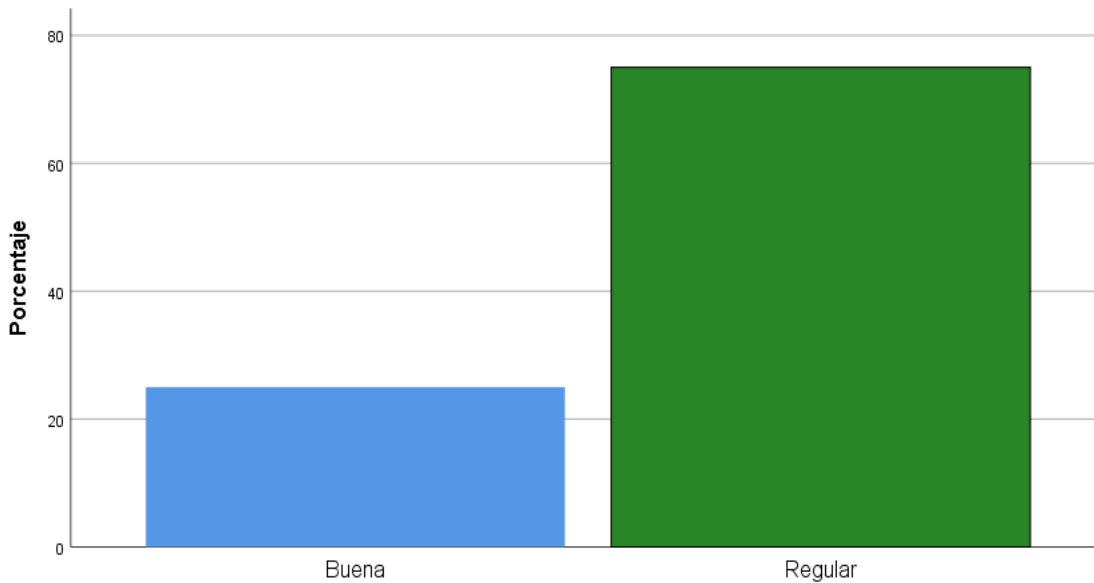


■ Cuaderno ■ Hojas ■ Libro diarios ■ Computadora ■ Telefono inteligente

**Figura 9.** Grafico de análisis del cuestionario tercer pregunta.

Según los resultados, los métodos que la mayoría de las personas encuestadas utiliza son los cuadernos y hojas, con una suma total de 61,9% respectivamente para realizar los registros. Estos son métodos manuales y solo el 30% utiliza métodos tecnológicos, como es la computadora y los teléfonos inteligentes.

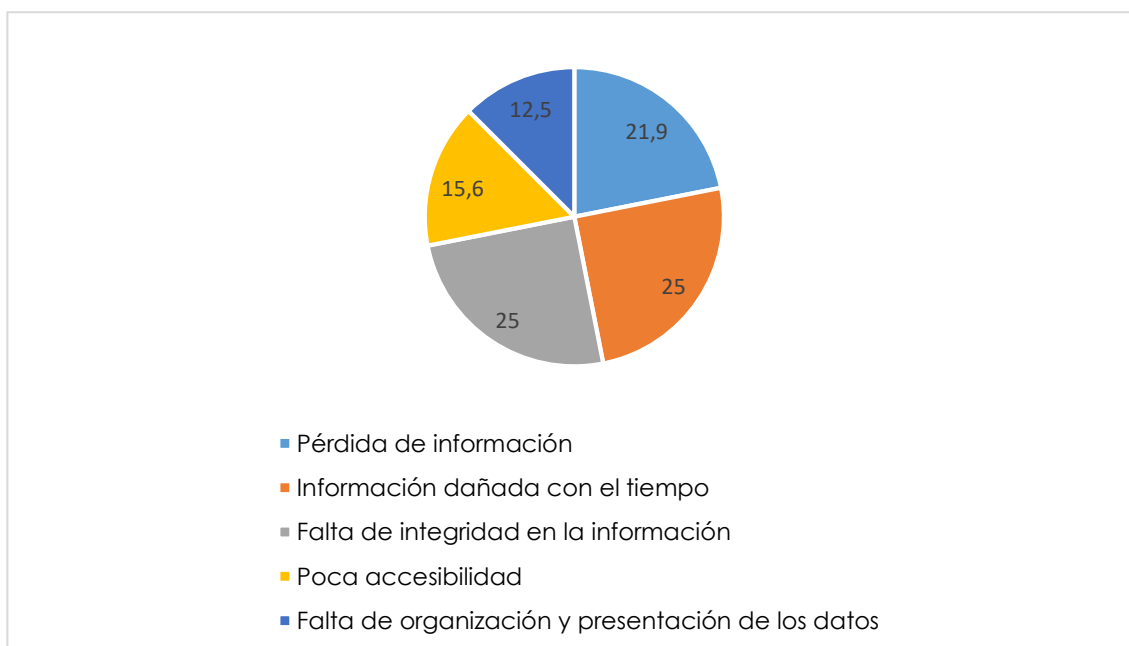
**4. ¿Cómo considera los métodos de administración y gestión utilizados actualmente en la hacienda?**



**Figura 10.** Grafico de análisis del cuestionario cuarta pregunta.

Más del 60% de los encuestados considera que los métodos actuales de administración y gestión utilizados son regulares, es decir, tienen sus deficiencias claras y que existen problemas con los métodos actuales.

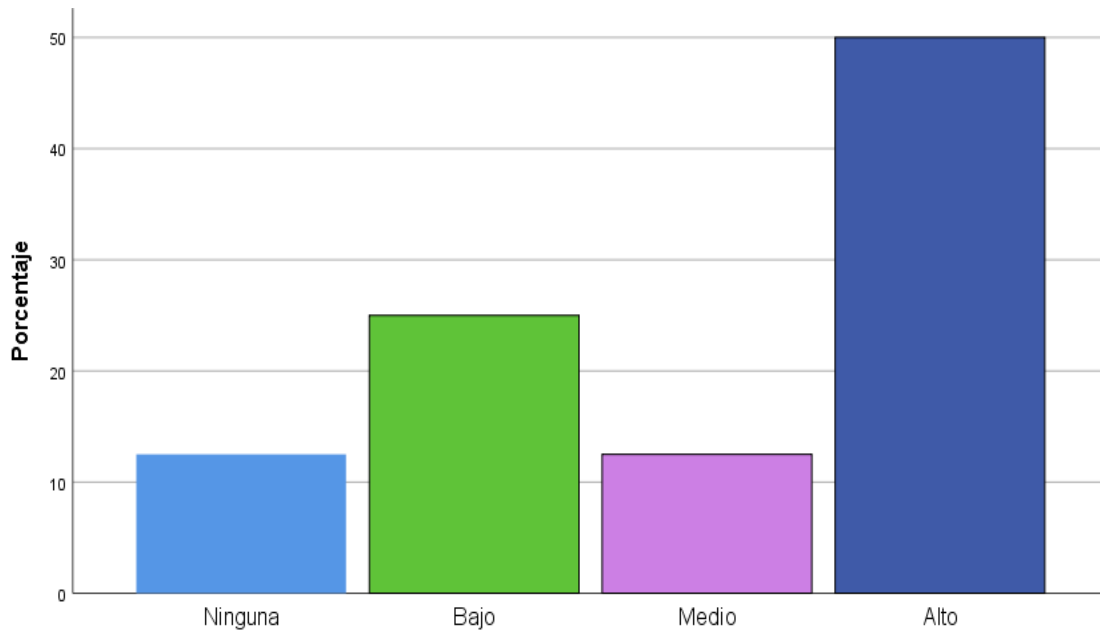
**5. ¿Cuáles son los problemas más frecuentes que tiene la hacienda al momento de manejar la información?**



**Figura 11.** Grafico de análisis del cuestionario quinta pregunta.

Todos los encuestados concuerdan en que el problema más recurrente es la información dañada con el tiempo; esto por las condiciones en las que se almacena o los métodos de registro y la falta de integridad, es decir, no está completa o está expuesta a modificaciones de terceros. Es importante mencionar que todos los encuestados manifiestan tener problemas con el manejo de la información.

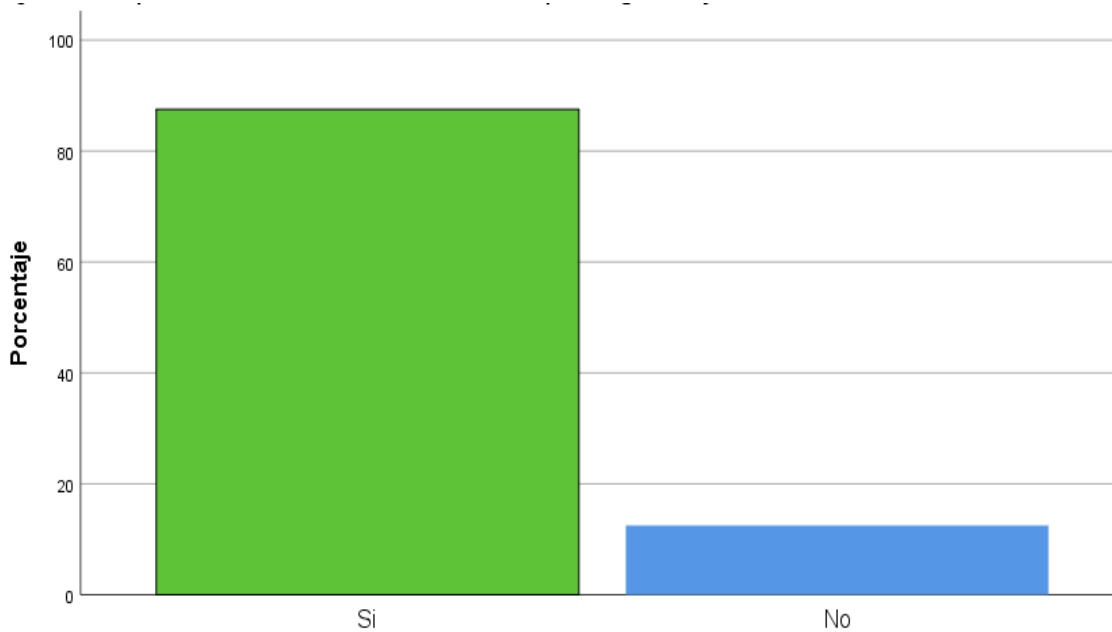
#### 6. ¿Cómo considera su capacidad para el uso de la tecnología?



**Figura 12.** Gráfico de análisis del cuestionario sexta pregunta.

El 60% de los encuestados manifiestan tener contacto con la tecnología, y tener un nivel medio-alto en cuanto al uso de la tecnología. El sistema informático debe tomar en cuenta que existen trabajadores con nivel bajo para el uso de tecnología.

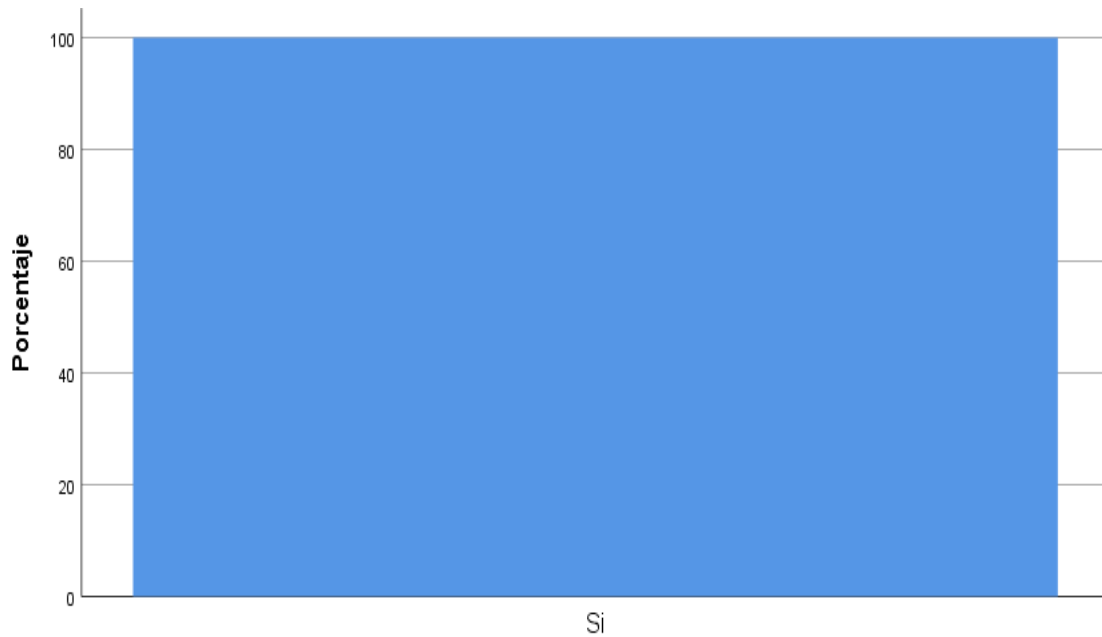
**7. ¿Estaría dispuesto a utilizar un sistema informático (es una técnica que permite el almacenamiento y el proceso de información a través del uso de la tecnología) para la administración y gestión de información de la hacienda ProduAgro?**



**Figura 13.** Gráfico de análisis del cuestionario séptima pregunta.

Siete de los ocho encuestados están dispuestos a utilizar un sistema informático. Es decir que están dispuestos hacer uso de esta tecnología y con esto se concluye que están de acuerdo con el desarrollo de un sistema informático para la hacienda ProduAgro.

**8. ¿Usted cree que, con el uso de un sistema informático, se mejore la administración y gestión de la información que usted maneja?**



**Figura 14.** Gráfico de análisis del cuestionario octava pregunta.

Después de una explicación en lo que consiste el sistema informático, todos los encuestados concuerdas en que, el sistema informático, mejoraría la administración y gestión de la información que ellos manejan en la hacienda ProduAgro.

**9. ¿Qué procesos le gustaría que tuviera el sistema informático?**

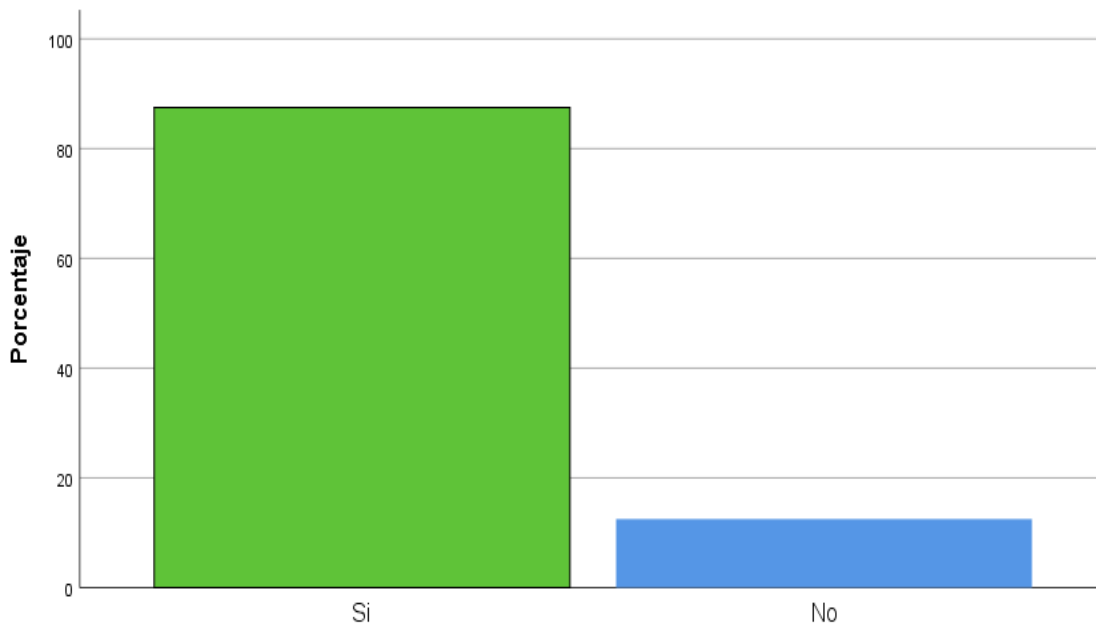


**Figura 15.** Gráfico de análisis del cuestionario novena pregunta.

El 100% de los encuestados concuerda en que el historial clínico, registro de animales y registro de descartes y muerte, son procesos importantes que deben estar en el

sistema informático. La base de los procesos llevados en la hacienda, son el ganado vacuno, y si estos datos tienen deficiencias, los demás también lo tendrán.

**10. ¿Le gustaría que el sistema informático, abarque procesos relacionados con el estado de los potreros, fertilizaciones y abonos?**



**Figura 16.** Gráfico de análisis del cuestionario décima pregunta.

Más del 80% de los encuestados cree que las actividades relacionadas al estado de potreros, fertilizantes y abonos, debe estar en el sistema informático. Solo una persona considero que no debe estar contemplado en el sistema informático.

**4.1.5. Resultados de la entrevista aplicada**

Una vez aplicado el instrumento de recolección de datos, en este caso la entrevista estructurada, al Sr. Administrador responsable de la hacienda ProduAgro se procedió a analizar las respuestas proporcionadas. La información proporcionada sirvió para lograr una mejor comprensión acerca de los procesos que se realizan en la hacienda para la administración y gestión de los animales bovinos que perteneces a la hacienda.

**1) ¿Qué entiende usted por un sistema informático?**

Respuesta: Para mí, un sistema informático es como una herramienta muy avanzada que usa la computadora para guardar y organizar toda la información que necesitamos en el trabajo. Es como tener un gran cuaderno digital donde podemos apuntar todo lo que hacemos con el ganad. No sé mucho de cómo funciona por

dentro, pero entiendo que hace más fácil manejar todo lo que pasa en la hacienda sin tener que usar papel o perder la información.

Análisis: La respuesta del administrador revela una percepción intuitiva sobre los sistemas informáticos, viéndolos como herramientas digitales para organizar y acceder a información, similar a un "cuaderno digital". Este punto de vista destaca la apreciación de la tecnología para mejorar la eficiencia y organización en la hacienda, a pesar de una comprensión limitada subraya la necesidad de desarrollar sistemas accesibles y fáciles de usar para la hacienda.

## **2) ¿Cómo se maneja actualmente la información sobre la administración y gestión de la hacienda ProduAgro?**

Respuesta: Actualmente, se maneja la información a través de registros manuales en papel, cuadernos y en algunos casos en Microsoft Excel.

Análisis: El uso de métodos tradicionales (manual y Excel) para la gestión de información destaca la necesidad de actualización la transición hacia sistemas informáticos permitiría una gestión más eficiente, reduciendo errores y mejorando la accesibilidad y seguridad de los datos.

## **3) ¿Se han implementado soluciones informáticas anteriormente en la hacienda ProduAgro para la administración y gestión de información?**

Respuesta: No, hasta el momento no se han implementado soluciones informáticas para la administración y gestión de la información en la hacienda.

Análisis: La falta de soluciones informáticas previas subraya una oportunidad para la implementación de un sistema específico para las necesidades de la hacienda podría revolucionar la forma en que se manejan los procesos y la información.

## **4) ¿Qué procesos incluye la administración y gestión de la hacienda ProduAgro?**

Respuesta: Incluye el mantenimiento de registros, supervisión de eventos, elaboración de informes e inventarios, gestión de producción de leche y reproducción de ganado, y mantenimiento de las herramientas utilizadas para la producción lechera.

Análisis: La diversidad de procesos mencionados refleja la complejidad de la gestión de una hacienda y subraya la importancia de contar con un sistema informático que pueda integrar todas estas actividades facilitando una mejor planificación y toma de decisiones.

**5) ¿Qué tipo de información se maneja en cada uno de los procesos dentro de la hacienda ProduAgro?**

Respuesta: Se maneja información sobre producción lechera, registros de nacimiento y reproducción, datos veterinarios, inventarios de insumos y productos, registros de mantenimiento de infraestructuras y equipos, y datos de empleados.

Análisis: La variedad de información manejada en los procesos indica la necesidad de un sistema que pueda categorizar, almacenar y procesar diferentes tipos de datos, desde producción lechera hasta registros veterinarios, de manera eficaz.

**6) ¿Quién o quiénes se encargan de gestionar las actividades o procesos relacionados al ganado bovino?**

Respuesta: Los encargados de la gestión de actividades relacionadas con el ganado bovino son: el Capataz que verifica todo y se pone en contacto con el administrador: Jornalero 1 que se encarga de la crianza de terneras: Jornalero 2 que se encarga de la crianza de vaconas, mantenimiento de ordeño, inseminación: Jornalero 3: que se encarga del ganado para el ordeño, ganado preñado, administración de potreros y dos empleados que son remplazos y por lo tanto no tienen trabajos fijos.

Análisis: La gestión por personal encargado de diferentes áreas de trabajo muestra la importancia de tener un sistema informático que sea accesible y fácil de usar para diferentes usuarios, incluyendo aquellos con limitada experiencia tecnológica.

**7) ¿Cómo se llevan a cabo los procesos de traslados del ganado bovino, parcelas, potreros y registros de mantenimientos?**

Respuesta: Estos procesos se realizan de manera manual, registrando en cuadernos o documentos físicos los detalles de los traslados de ganado, uso de parcelas y potreros, así como los mantenimientos realizados.

Análisis: La gestión manual de estos procesos evidencia el riesgo de pérdida o deterioro de la información un sistema informático ayudaría a mantener registros precisos y accesibles, mejorando la eficiencia operativa.

**8) ¿Qué tiempo toma realizar un reporte relacionado a los chequeos del ganado inseminado en la hacienda?**

Respuesta: El tiempo para realizar un reporte varía, pero debido al manejo manual de la información, puede llevar desde unas horas hasta varios días compilar toda la información necesaria.

Análisis: El tiempo variable y a menudo prolongado para generar reportes resalta la ineficiencia de los métodos actuales. La automatización de reportes a través de un sistema informático podría proporcionar información en tiempo real, mejorando la toma de decisiones.

**9) ¿Cuántos trabajadores conforman para la administración y gestión de la hacienda?**

Respuesta: La cantidad de trabajadores dedicados a la administración y gestión de la hacienda son 7 que son el administrador, capataz, 3 jornaleros y 2 empleados de reemplazo, todos ellos están encargados de diferentes aspectos de la gestión ganadera en la hacienda.

Análisis: La implicación de algunos trabajadores en la gestión destaca la necesidad de un sistema que facilite la colaboración y centralice la información mejorando la coordinación y eficiencia del equipo.

**10) ¿Está de acuerdo usted con la implementación de una solución informática para la administración y gestión de la información en la hacienda?**

Respuesta: Sí, estoy de acuerdo una solución informática mejoraría significativamente la eficiencia, precisión y accesibilidad de la gestión de la información en la hacienda.

Análisis: El acuerdo con la implementación sugiere una percepción positiva de los beneficios potenciales que dicha tecnología podría aportar, incluyendo mejoras en eficiencia, precisión y toma de decisiones.

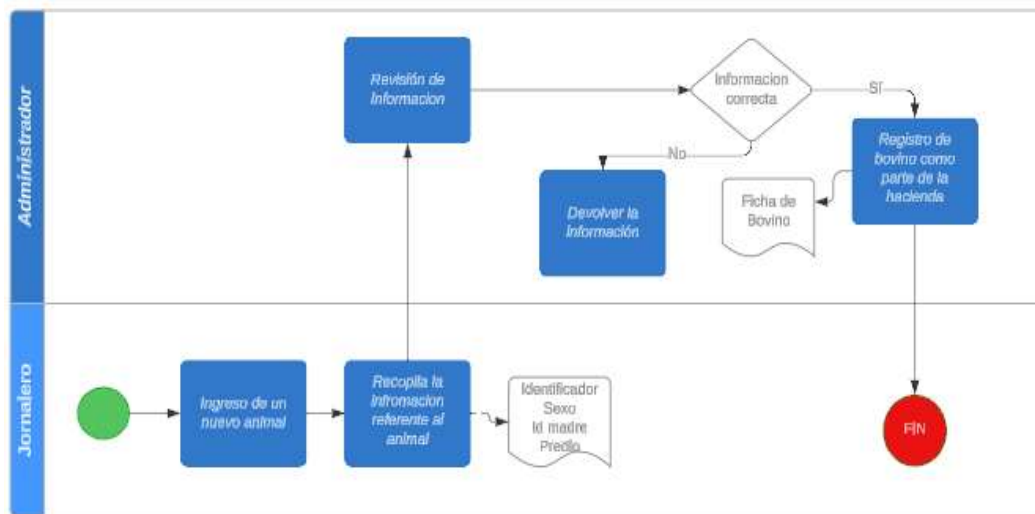
**11) Descripción de los procesos de la hacienda ProduAgro.**

La hacienda ProduAgro gestiona diversos procesos estructurados en módulos específicos dentro de una solución informática propuesta:

Gestión Ganadera: Incluye el manejo de animales, medición lechera, registro de muertes, descartes, pesos, toros, datos veterinarios y traslados.

- Registro de un nuevo bovino

Para el registro de un nuevo bovino se procede con el levantamiento de la información relevante respecto al animal como su identificador, sexo, predio, grupo al que pertenece y fotografía.



**Figura 17.** Proceso de registro de un nuevo bovino.

- Medición Lechera

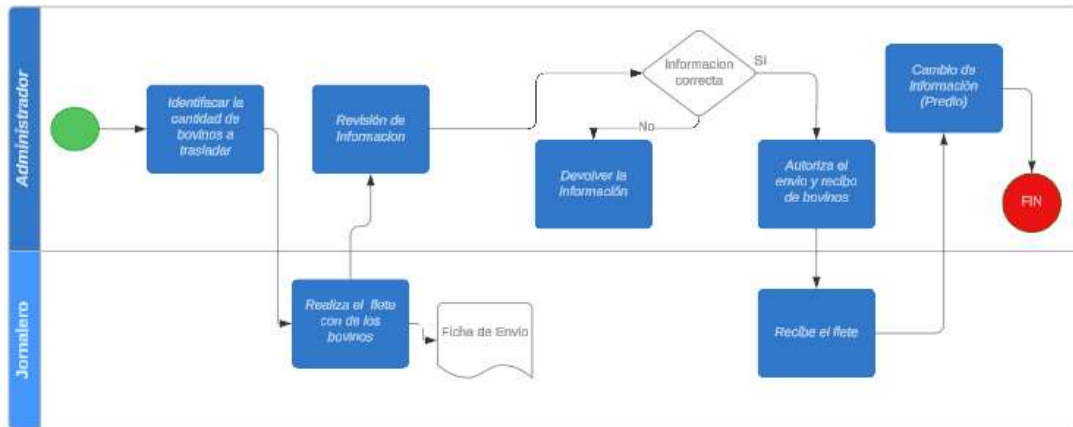
Una vez por semana se registra de producción de cada bovino destinado a producción. Se verifica bovino por bovino su producción en la mañana y en la tarde. Esta labor lo realiza un jornalero y los datos los recibe el administrador.

- Pesos

Los pesos se realizan una vez por semana a todos los bovinos en los grupos de neonatos hasta vaconas. La pesa se realiza a cada uno de los bovinos y se realiza el registro del mismo.

- Traslado de Bovinos

Se realizan movimientos de ganado entre predios. El administrador evalúa cuantos y cuales bovinos va a enviar o recibir y el jornalero es el responsable de enviar y recibir dicho ganado.

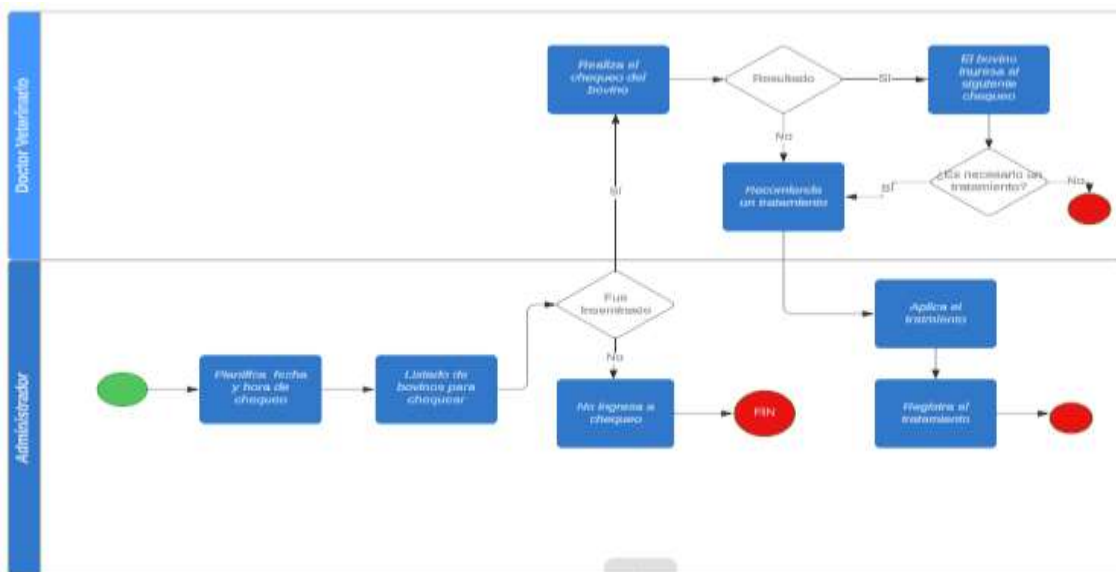


**Figura 18.** Proceso de registro de traslado de un bovino entre predios.

Gestión Reproductiva: Engloba las inseminaciones, chequeos, partos y abortos.

- Chequeos

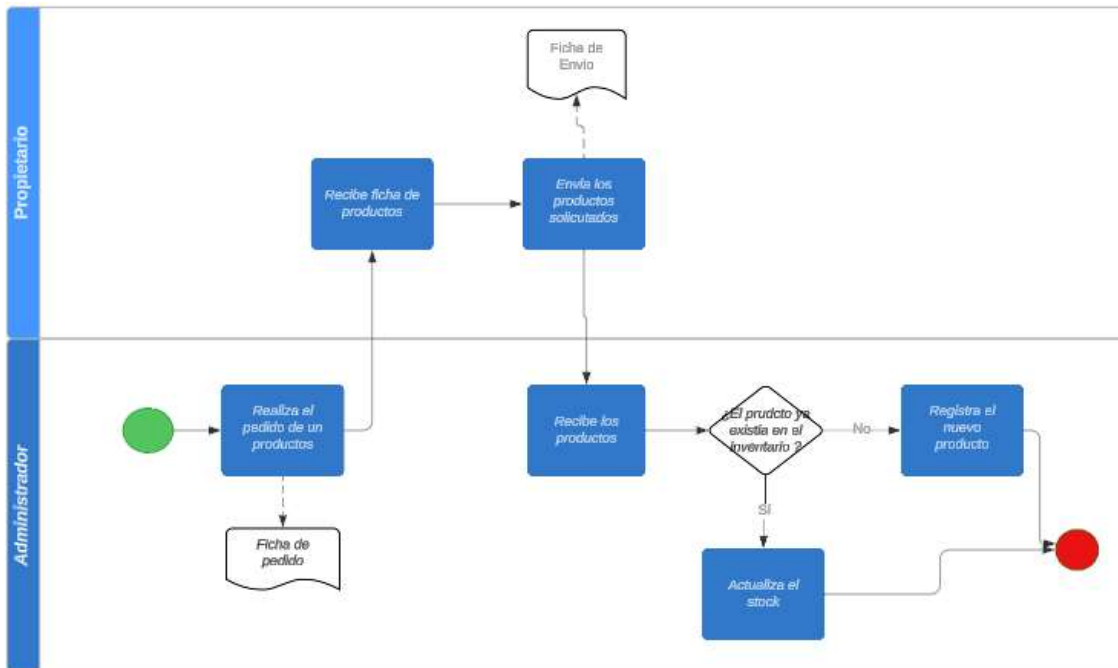
Los chequeos médicos se realizan a los bovinos que hayan sido inseminados o recibida monta por toro. El chequeo lo realiza un doctor veterinario, toda la información resultante la recibe el administrador.



**Figura 19.** Proceso de registro de chequeo de bovinos.

Inventario: Manejo de productos y pedidos.

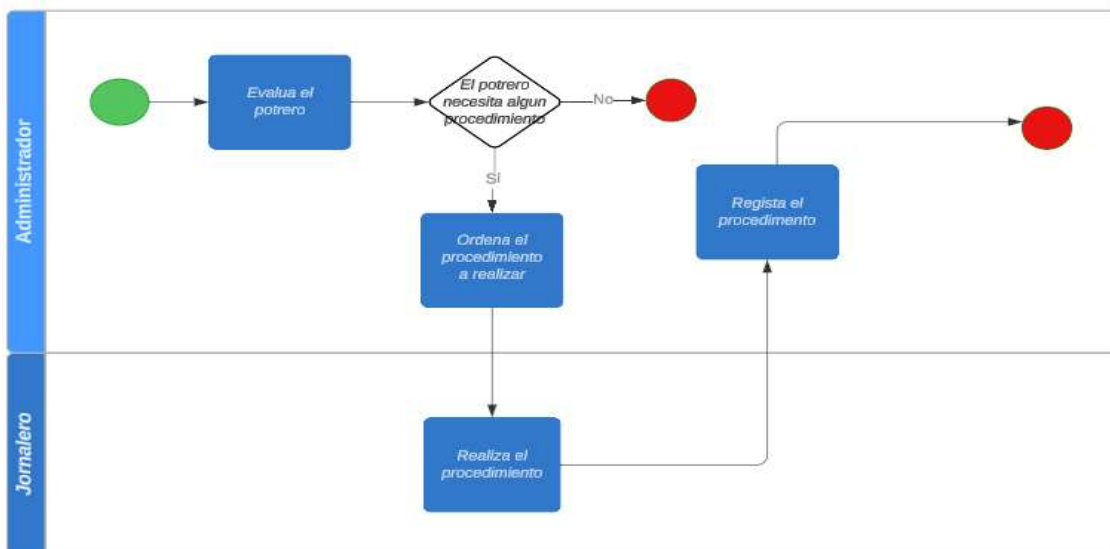
El administrador solicita el ingreso de productos agrícolas y veterinarios para uso de la Hacienda, se realiza el registro de los productos solicitados con su respectiva fecha de pedido. Cuando estos productos ingresan se realiza el registro de los productos recibidos y se actualiza el inventario.



**Figura 20.** Proceso de registro de inventario.

Gestión de Campo: Involucra la administración de las parcelas y potreros.

Los potreros cada cierto tiempo se someten a actividades agrícolas como siembra, fertilización o rotación alimenticia. El administrador evalúa los procedimientos a realizarse y el jornalero se encarga de ejecutarlos.

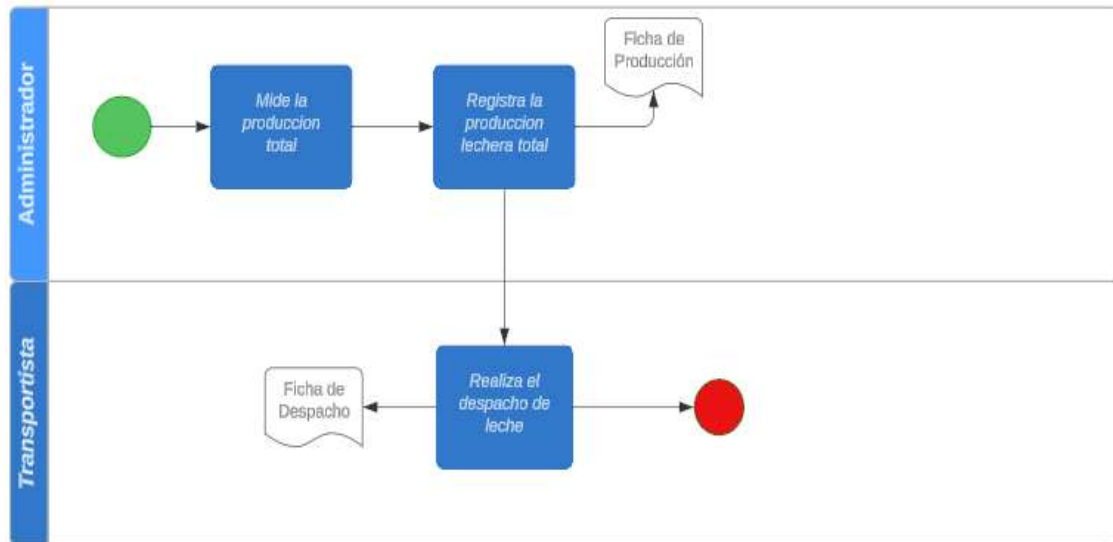


**Figura 21.** Proceso de registro de gestión de campo.

Mantenimientos: Registro de mantenimientos de infraestructuras y equipos.

Producción Lechera: Control y gestión de la producción lechera.

El administrador realiza el registro total de la producción lechera diaria, fecha y medida. Posteriormente el transportista realiza el despacho de la leche, registrando la fecha y demás información.



**Figura 22.** Proceso de registro de producción de leche.

Información General: Datos sobre parcelas, grupos y recursos humanos.

Reportes: Generación de reportes sobre animales, inseminaciones, chequeos, partos y abortos.

La descripción de los módulos para la solución informática propuesta revela una comprensión detallada de las necesidades específicas de la hacienda. La implementación de estos módulos facilitaría una gestión integrada y eficiente, abarcando desde la gestión ganadera hasta los reportes, lo cual es fundamental para el progreso y sostenibilidad de la hacienda.

## 4.2. PROPUESTA

### 4.2.1 Datos Informativos

Hacienda ProduAgro

#### 4.2.1.1 Logotipo

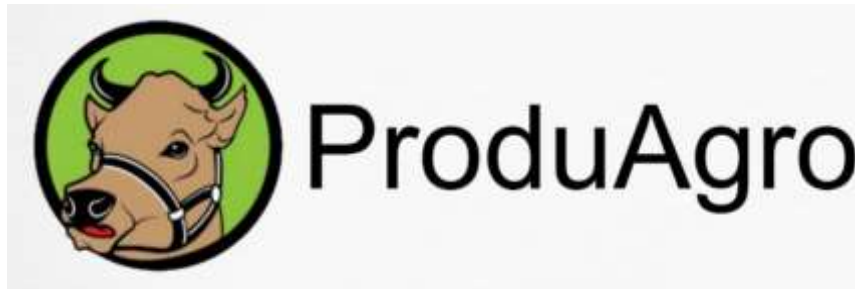


Figura 23. Logotipo de la Hacienda ProduAgro.

#### 4.2.1.2 Ubicación geográfica

San Isidro, Caserío Carlizama, Panamericana norte km 34.

#### 4.2.1.3 Organigrama Hacienda ProduAgro

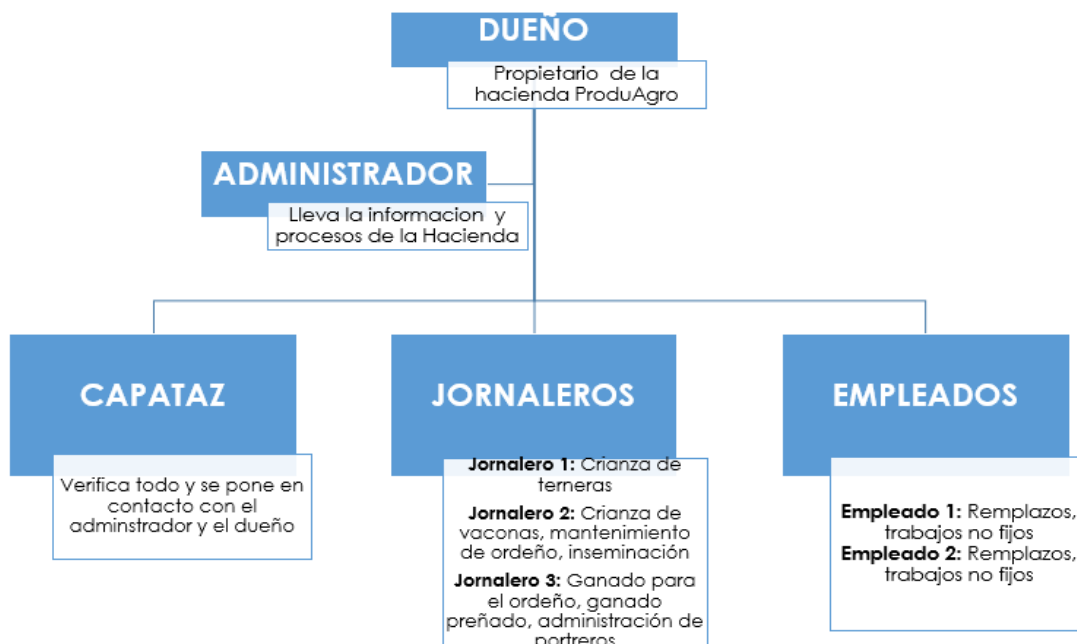


Figura 24. Estructura organizacional de la hacienda ProduAgro.

#### 4.2.1.4 Misión

"Somos una empresa enfocada en la producción y transformación de materia prima, empleando para ello sistemas eficientes e innovadores basados en el desarrollo amigable con el medio ambiente y plenamente sostenible."

#### 4.2.1.5 Visión

“En el 2024, seremos una empresa modelo del sector, reconocida ampliamente a nivel local y nacional por su seriedad, calidad, innovación y compromiso en el desarrollo social de nuestro país.”

#### 4.2.2. Desarrollo de la Propuesta

La propuesta de nuestro proyecto se derivó de un análisis exhaustivo de los resultados de la investigación y de un acercamiento con el administrador de la Hacienda ProduAgro, donde indicó que se desarrolle los módulos más importantes y necesarios para la hacienda, de los cuales se describe a continuación: Administración de usuarios, ingreso del bovino, información del ganado, estado de salud del bovino, medicamentos administración de la leche, reportes de información del ganado (baja, vacía, preñada, enferma, tratamiento, inseminación, cuantos partos tiene).

Analizando las respuestas obtenidas de las encuestas y entrevistas se toma en consideración la necesidad de implementar una técnica de obtención de datos para diferentes consultas que puedan realizar los empleados, de tal manera, se tomó en consideración este módulo referente al requerimiento. En primer lugar, se realizó un estudio de factibilidad que ayuda a determinar si el desarrollo de la aplicación web es realizable. Por otra parte, se optó por usar la metodología ágil XP que facilita el desarrollo de una planificación ordenada haciendo énfasis unos requerimientos del cliente. Con base en esto, se llevan a cabo todas las fases correspondientes a esta metodología, y de manera similar, se realizan las pruebas de aceptación, asegurando así la satisfacción del beneficiario con la aplicación entregada.

En nuestra investigación con el uso de la metodología XP se empieza por la planificación y acorde a eso se describe los beneficiarios directos e indirectos de nuestro trabajo de integración curricular.

##### Beneficiarios Directos

Los beneficiarios directos son los individuos que estarán en contacto inmediato con el sistema y cuyas responsabilidades serán directamente impactadas y asistidas por la solución tecnológica en la Hacienda ProduAgro, esto incluye:

- Dueño: Como propietario de la hacienda, tiene el interés principal en la optimización y en los resultados directos de la gestión.

- Administrador: Responsable de llevar la información y procesos de la Hacienda, quien utilizará el sistema para tareas de administración diaria, mejora en la toma de decisiones y reportes precisos.
- Capataz: Verifica procesos y se coordina con el administrador y el dueño, utilizando el sistema para reportar y optimizar el funcionamiento diario.
- Jornaleros: Cada jornalero, según su área de especialización (cría de terneras, vacas, mantenimiento del ordeño, inseminación, ganado preñado y administración de potreros), se beneficiará de un acceso más eficiente a la información y a la comunicación de tareas relacionadas con sus funciones específicas.
- Empleados: Abarca a aquellos que cubren roles temporales o especializados y que se beneficiarán de la sistematización y automatización de sus tareas diarias.

#### Beneficiarios Indirectos

Los beneficiarios indirectos son aquellos que, sin estar en contacto directo con el sistema, se benefician de sus resultados en la gestión y administración de la hacienda.

Incluyen:

- Trabajadores de la hacienda y sus familias: Que se beneficiarán de un ambiente de trabajo más organizado y posiblemente de mayores rendimientos de la hacienda.
- Clientes y proveedores de la Hacienda ProduAgro: Que experimentarán una mejora en la eficiencia y en la comunicación gracias a los procesos mejor administrados.

La siguiente propuesta corresponde al desarrollo de un aplicativo web prototipo en el cual se evidencia una propuesta de solución a la problemática encontrada y se va a generar un sistema diseñado para desempeñar los procesos de administración y gestión como: ganadería, gestión reproductiva, inventario, gestión de campo, mantenimiento, producción lechera y reportes.

Para el proceso de gestión ganadera se llevan a cabo los siguientes subprocesos:

- Animales
- Medición Lechera
- Muertes
- Descartes

- Pesos
- Toros
- Datos Veterinarios
- Traslados

Para el módulo de gestión reproductiva se llevan a cabo los siguientes subprocesos:

- Inseminaciones
- Chequeo
- Partos
- Abortos

Para el módulo de inventario se llevan a cabo los siguientes subprocesos:

- Productos
- Pedidos

Para el módulo de gestión de campo se llevan a cabo la supervisión y administración de las actividades y recursos del campo, asegurando un uso óptimo del terreno.

Para el módulo de mantenimiento se llevan a cabo el control de las tareas de mantenimiento de equipos y máquina de ordeño, ya sean mantenimientos preventivos o correctivos.

Para el módulo de producción lechera se llevan a cabo la gestión y análisis de la producción lechera total, buscando optimizar el registro de la cantidad y calidad de la leche producida.

Para el módulo de información general se llevan a cabo los siguientes subprocesos:

- Predios
- Grupos
- Recursos Humanos

Para el módulo de reportes se llevan a cabo los siguientes subprocesos:

- Animales
- Inseminaciones
- Chequeos
- Partos
- Abortos

Para alcanzar los objetivos de este estudio y satisfacer las necesidades del cliente, se optó por utilizar el entorno de desarrollo Visual Studio 2019, junto con la metodología de desarrollo ágil eXtreme Programming (XP), que se alinea bien con los objetivos y directrices para la creación del software. Además, se implementó una base de datos relacional utilizando SQL Server y se programó utilizando el lenguaje C#. Para la parte de front-end, se emplearon los lenguajes de marcado HTML y CSS.

La elección de estas tecnologías facilitó un desarrollo más claro y entendible del sistema. También se crearon diagramas de casos de uso para representar las operaciones diarias dentro de la hacienda ProduAgro, garantizando así que el sistema cumple con los requisitos definidos.

#### 4.2.3. Administración y gestión de información en la hacienda ProduAgro

##### 4.2.3.1 Hacienda ProduAgro

La hacienda está conformada por varios procesos que conlleva el registro de la información de la ganadería bovina, si bien actualmente quien hace los registros respectivos de la gestión ganadera es el administrador quien hace uso de un Excel en donde esta descrito el identificador de cada animal su sexo, predio, número de identificación de la madre, fecha de nacimiento y grupo a cuál pertenece.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		Tambo	RP	Nombre	Raza	comp/Racial	Estado	F. nac	Edad	Tipo Reg	Nro.Reg	Cal.FC	Cal.F	Prep	HBA Padre	ApodoPadre RazaF
3		0080-01	J1945	JOYA	HO	HO 300	A	10/11/2019	04.02							
4		0080-01	L1905	LUNA	HO	HO 300	E	20/01/2019	04.12							
5		0080-01	P1943	PERLAIR	HO	HO 300	A	27/08/2019	04.05							
6		0080-01	P2011	PAÑA	HO	HO 300	A	20/02/2020	03.11							
7		0080-01	S1148	SAN-11-48	HO	HO 300	A	27/07/2011	12.06							
8		0080-01	S1247	SAN-12-47	HO	HO 300	A	25/08/2012	13.05							
9		0080-01	S1350	SAN-13-50	HO	HO 300	A	19/07/2013	10.06							
10		0080-01	S1428	SAN-14-28	HO	HO 300	A	28/07/2014	09.06							
11		0080-01	S1458	SAN-14-58	HO	HO 300	A	08/12/2014	09.01							
12		0080-01	S1484	SAN-14-84	HO	HO 300	A	25/12/2014	09.01							
13		0080-01	S1534	SAN-15-34	HO	HO 300	A	05/01/2015	08.10							
14		0080-01	S1535	SAN-15-35	HO	HO 300	A	07/02/2015	08.10							
15		0080-01	S1538	SAN-15-38	HO	HO 300	A	10/03/2015	08.10							
16		0080-01	S1538	SAN-15-38	HO	HO 300	A	26/03/2015	08.10							
17		0080-01	S1529	SAN-15-29	HO	HO 300	A	28/05/2015	08.08							
18		0080-01	S1541	SAN-15-41	HO	HO 300	A	08/07/2015	08.06							
19		0080-01	S1544	SAN-15-44	HO	HO 300	A	11/07/2015	08.06							
20		0080-01	S1549	SAN-15-49	HO	HO 300	A	17/07/2015	06.06							
21		0080-01	S1572	SAN-15-72	HO	HO 300	A	18/11/2015	06.02							
22		0080-01	S1578	SAN-15-78	HO	HO 300	A	02/12/2015	06.02							
23		0080-01	S1578	SAN-15-78	HO	HO 300	A	20/12/2015	06.01							

**Figura 25.** Matriz en Excel que maneja la hacienda referente a la información de los animales bovinos.

#### 4.2.3.2. Gestión Reproductiva

La gestión reproductiva de igual manera esta información la maneja el administrador sin embargo también los jornaleros que trabajan en la hacienda llevan estos registros en los cuales se manejan las fechas de inseminaciones quien está encargado de dicho procedimiento el tipo de inseminación que se realizó como puede ser natural o artificial y alguna nota.

ASOCIACIÓN HOLSTEIN FRIESIAN DEL ECUADOR  
CERTIFICADO DE MONTA  
HOLSTEIN ECUATORIANO Nº 0001901

Nombre de la Vaccona / Vaca	Registro	Nombre del Toro	Registro	Fecha Inseminación
Sun	18-40	Cuñaber	117043	22-09-22
Sun	15-12	Cuñaber	117043	22-09-22
Sun	15-22	Cuñaber	117043	24-09-22
Sun	16-06	Cuñaber	117043	24-09-22
Sun	17-24	Cuñaber	117043	24-09-22

FIRMA  
En caso de inseminación artificial, adjunte pajueta

ASOCIACIÓN HOLSTEIN FRIESIAN DEL ECUADOR  
CERTIFICADO DE MONTA  
HOLSTEIN ECUATORIANO Nº 0001913

Nombre de la Vaccona / Vaca	Registro	Nombre del Toro	Registro	Fecha Inseminación

FIRMA  
En caso de inseminación artificial, adjunte pajueta

Figura 26. Registro de inseminaciones.

De igual manera se registra la información de los chequeos que contiene información del resultado si fue positivo o no, dado el caso que sea positivo se registra la fecha de preñez y en el caso de un resultado negativo se describe el problema y el tratamiento que se sugiere al animal bovino.

Edad	Estado	P	URL parto	URL celo	# ser	URL servicio	URL resultado		
SAN-06-41	13a11m	Desc./Lac	10	03/01/2019	652	05/10/2019175	5	05/10/2019275	TUMOR
SAN-07-28	13a3m	Desc./Lac	7	19/01/2020	211		0		TUMOR
SAN-08-33	12a1m	Desc./Lac	7	17/08/2020	60		0		TUMOR
SAN-08-44	13a11m	Ser/Lac	8	15/12/2019	306	04/10/2020194	7	04/10/2020294	CLD
SAN-08-56	13a9m	Ca/Lac	9	06/01/2020	284	11/10/2020279	6	29/08/2020236	CLI
SAN-09-03	11a9m	Paip./Lac	9	28/06/2020	110		0		GFOL
SAN-10-61	10a1m	Paip./Lac	7	10/04/2020	189	14/09/2020157	3	29/08/2020182	CLD
SAN-10-68	10a0m	PR/Lac	5	14/01/2020	276	27/04/2020104	2	27/04/2020184	PRENADA
SAN-11-07	9a8m	Ser/Lac	5	08/01/2020	282	29/09/2020265	5	29/09/2020265	CLD
SAN-11-16	9a6m	Ser/Lac	7	20/07/2020	88	18/09/2020 58	1	18/09/2020 58	N. INS.
SAN-11-17	9a6m	Ser/Lac	6	04/06/2020	134	03/09/2020 91	2	03/09/2020 91	CLD
SAN-11-24	9a6m	Ca/Lac	6	26/10/2019	356	01/10/2020 841	3	18/09/2020120	GFOL
SAN-11-28	9a5m	Ca/Lac	5	16/06/2020	122	25/07/2020 39	0		CLI
SAN-11-32	9a4m	Paip./Lac	6	09/05/2020	180	30/07/2020 82	1	30/07/2020 82	CLD
SAN-11-48	9a2m	Ser/Lac	7	25/03/2020	205	04/10/2020193	5	04/10/2020193	OVATROF
SAN-11-67	8a10m	PR/Lac	6	06/12/2019	315	10/05/2020156	2	10/05/2020156	PRENADA
SAN-12-16	8a5m	PR/Lac	5	24/11/2019	327	21/06/2020210	4	21/06/2020210	PRENADA
SAN-12-17	8a5m	Ser/Lac	5	31/05/2020	138	12/10/2020134	2	12/10/2020134	N. INS.
SAN-12-48	8a2m	Ser/Lac	6	17/06/2020	121	25/09/2020100	3	25/09/2020100	CLD
SAN-12-55	8a1m	Ser/Lac	5	26/02/2020	233	02/10/2020219	4	02/10/2020219	GFOL
SAN-13-02	7a9m	Nc/Lac	5	25/09/2020	21		0		
SAN-13-04	7a9m	Ca/Lac	5	02/04/2020	197	12/10/2020193	4	03/09/2020154	RCH
SAN-13-33	7a5m	Ca/Lac	3	20/02/2020	239	09/10/2020232	3	30/08/2020192	VACIA
SAN-13-61	7a3m	Paip./Lac	3	23/06/2020	115		0		OVATROF
SAN-13-69	7a0m	Nc/Lac	5	12/10/2020	4		0		
SAN-13-71	7a0m	Ser/Lac	4	28/05/2020	141	08/10/2020133	3	08/10/2020133	CLD
SAN-13-73	7a0m	Ser/Lac	5	27/06/2020	111	15/09/2020 80	2	15/09/2020 80	CLD
SAN-14-02	6a8m	Ca/Lac	5	04/09/2020	42	12/10/2020 38	0		RCH
							1	12/06/2020 32	PRENADA

Figura 27. Registro de chequeos.

También se lleva a cabo el registro de los partos como sus fechas el sexo y estado si nació vivo o muerto y de igual manera con los abortos.

Toda esta información se la maneja actualmente en hojas de cuadernos y libretas por lo que genera que dicha información se deteriore, no sea visible y se pierda.

#### 4.2.3.3. Inventario

Este proceso actualmente la hacienda ProduAgro el registro del inventario se lo realiza en una hoja de cuaderno donde están los productos, sin embargo, esta información no está optimizada y por lo tanto aquí en este módulo se registra los productos y pedidos que realizan en la hacienda.

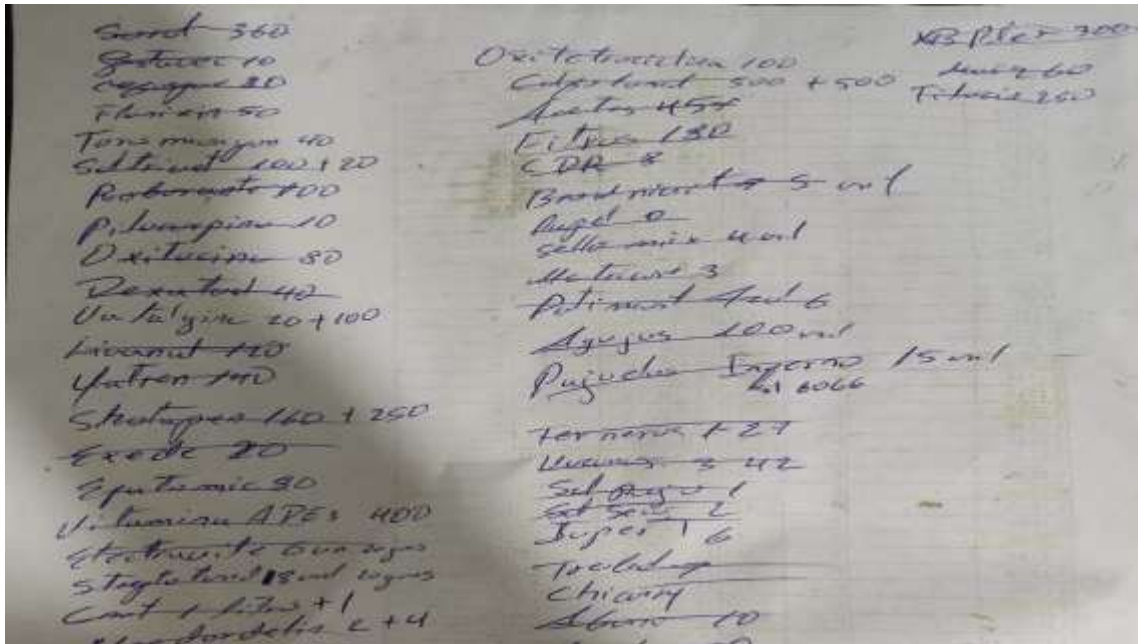


Figura 28. Registro de productos actualmente.

#### 4.2.3.4. Gestión de Campo

La gestión de campo es un módulo muy importante el cual se puede determinar el estado el cual se encuentra un potrero ya sea que este en preparación, siembra o rotación, actualmente de guían de esta gráfica.

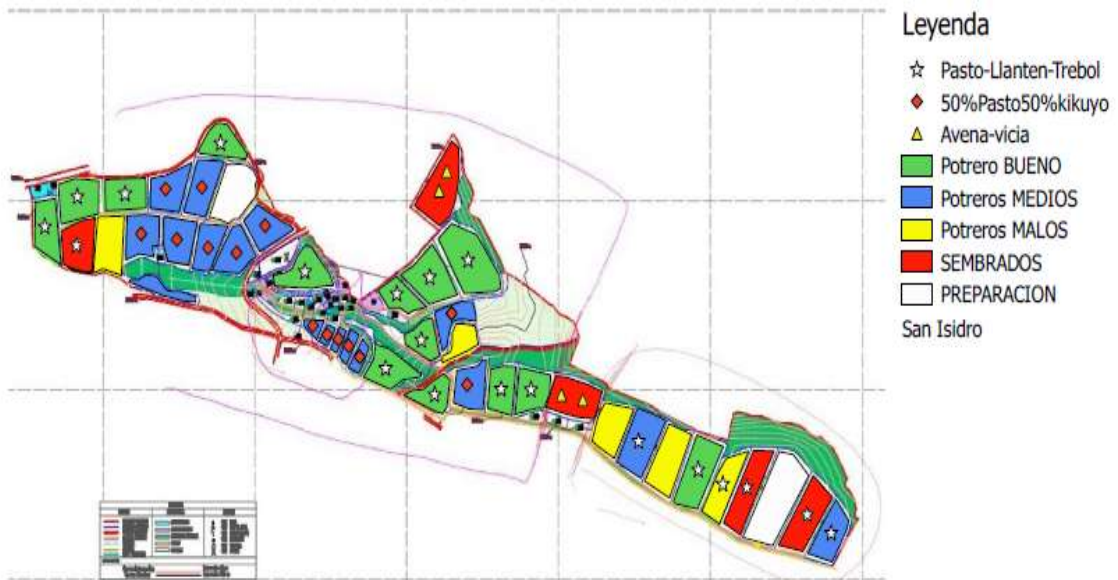


Figura 29. Registro de estados de potreros en la hacienda.

#### 4.2.3.5. Mantenimientos

Este proceso actualmente la hacienda ProduAgro no maneja directamente esta información por lo que algunos desconocen cómo está el estado de la máquina de

ordeño o equipos que usan en la hacienda o si se realizó algún mantenimiento no poseen las fechas de dicha acción y tampoco el costo, por lo que este módulo es un adicional para optimizar el proceso y que la información sea almacenada para determinar fechas y así realizar mantenimientos preventivos o correctivos.

#### 4.2.3.6. Producción Lechera

La producción lechera se la registra de manera diaria y esta información es llevada en un cuaderno de anotaciones de la siguiente manera.



**Figura 30.** Registro de producción lechera.

### 4.3. DISCUSIÓN

Con el objetivo de esta presente investigación denominado, "Desarrollar un prototipo de sistema informático que permita el mejoramiento de la administración y gestión de información en la hacienda ProduAgro se toma en consideración para realizar la siguiente discusión.

En primer lugar, se examinaron investigaciones previas de años anteriores para analizar sus propuestas, problemáticas, herramientas que fueron utilizadas, entre otros aspectos donde se identificaron similitudes con el trabajo de investigación presente, entre ellas concuerdan con la eficiencia de usar herramientas tecnológicas para mejorar la administración y gestión de información de haciendas de ganado bovino, en base a esto, manejan un mismo enfoque los tres antecedentes investigación de la cual hicimos énfasis para el desarrollo de una solución tecnológica.

Los antecedentes proporcionan un marco enriquecedor para la investigación propuesta, destacando la implementación exitosa de sistemas informáticos en la gestión ganadera. La Universidad Técnica de Cotopaxi, mediante el proyecto en la Hacienda 'La María', evidenció cómo el uso de herramientas de software libre optimizó la administración y supervisión de operaciones relacionadas con el ganado vacuno (Toapanta Quinatoa & Zhingre Sánchez, 2019).

Por lo que el sistema informático para la hacienda ProduAgro se ve exitoso para su posterior uso debido a que se optimiza todos los registros que lo realizan manualmente en libretas, Excel, hojas volantes, etc.

El proyecto desarrollado por el Máster Universitario en Ingeniería de Software y Sistemas Informáticos de la Universidad Internacional de La Rioja complementa esta idea, centrándose en la planificación y creación de una aplicación web llamada LIFTIC Ganadero (Castillo-Álvarez, 2019).

La accesibilidad al sistema desde cualquier parte del mundo que contenga conexión a Internet demuestra una ampliación de los horizontes de aplicación de sistemas informáticos en la ganadería donde clientes puedes hacer uso del sistema o revisar información desde cualquier sitio en el que se encuentren de tal manera puedan tomar decisiones acertadas aun sin estar presente en la hacienda.

En cuanto, la investigación realizada en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López resalta la eficiencia económica y la automatización del seguimiento del ganado bovino a través del uso de un software de control ganadero (Murillo & Zambrano, 2019).

La eficiencia en la toma de decisiones son aspectos comunes entre los antecedentes y la investigación presente. El énfasis en la mejora de gestionar y administrar la información a través de un software de control ganadero concuerda en la intención de optimizar la administración y gestión de la hacienda ProduAgro.

La conclusión, podemos determinar que la investigación propuesta destaca la importancia de la ganadería en el Carchi y establece claramente el objetivo de desarrollar un sistema informático para mejorar la administración y gestión de la hacienda ProduAgro. El enfoque mixto, combinando cuantitativo y cualitativo, recalca la integralidad de la investigación buscando no solo la eficiencia en el uso de recursos, sino también la toma de decisiones informada.

## V. CONCLUSIONES Y RECOMENDACIONES

### 5.1. CONCLUSIONES

- La información obtenida teóricamente ofreció una perspectiva detallada sobre las deficiencias en los procesos de administración y gestión de información en la hacienda ProduAgro, especialmente en lo que respecta a al área administrativa del ganado bovino, obteniendo de esta manera referencias confiables para identificar áreas críticas de mejora, también facilitó una comprensión más profunda de los fundamentos teóricos que sustentan los procesos administrativos y de gestión, destacando su relevancia para la eficacia organizacional en haciendas ganaderas.
- El empleo de herramientas especializadas para la recopilación de datos permitió una vinculación efectiva y directa entre los procesos administrativos y de gestión de la hacienda ProduAgro, abarcando los parámetros y actividades clave. Este enfoque clarificó las operaciones y prácticas existentes, facilitando su transformación en módulos y funcionalidades específicas del sistema propuesto.
- Los procesos administrativos analizados, junto con sus necesidades específicas identificadas, demuestran una adaptabilidad fluida al sistema informático gracias a su conexión directa con los procesos funcionales de la hacienda ProduAgro y la información que se generan.
- La metodología ágil XP (Extreme Programming) aporta herramientas valiosas para el desarrollo de sistemas de gestión, destacándose por su capacidad para segmentar los módulos de software según su relevancia. Esta estrategia facilita la organización eficiente de los proyectos mediante la definición de plazos concretos, priorizando simultáneamente la entrega oportuna de las tareas fundamentales.

- El objetivo principal de este estudio consistió en desarrollar un prototipo de sistema informático utilizando las TIC para mejorar la administración y gestión de la información de la hacienda ProduAgro. Este objetivo se alcanzó empleando una metodología ágil de desarrollo de software, específicamente XP, que facilitó la identificación de requisitos a través de historias de usuarios y el desarrollo de prototipos con las funcionalidades requeridas, las cuales luego se implementaron efectivamente.

## **5.2. RECOMENDACIONES**

- La investigación se basó en el desarrollo de un prototipo, por lo cual, se recomienda ampliar el proceso de investigación haciendo uso de la presente documentación generada en este estudio como referencia para analizar la viabilidad de su aplicación en otras haciendas o fincas ganaderas.
- Se recomienda llevar a cabo un análisis detallado de los procesos administrativos y de gestión de las diferentes áreas que conforman la hacienda ProduAgro, con el fin de evaluar la viabilidad de integrar módulos adicionales al sistema existente. Se recomienda hacerlo de forma gradual y metódica, asegurando así una sincronización efectiva y evitando dificultades de integración entre los distintos componentes de software.
- En base al uso de la metodología de desarrollo XP se recomienda tener comunicación muy seguida con el cliente para comprobar que los requerimientos se ejecuten de forma correcta y adecuada, lo cual asegura el proceso de calidad del software cuando se realicen las pruebas de aceptación.
- Se sugiere profundizar en la investigación sobre las diversas herramientas ofrecidas por la metodología ágil XP, teniendo en cuenta que estas pueden variar según la fuente. Es fundamental identificar y adoptar aquellas herramientas y prácticas que mejor se ajusten a las necesidades específicas y a los procesos administrativos relacionados con la hacienda ProduAgro, garantizando así un desarrollo de software eficaz y alineado con los objetivos del proyecto.

## VI. REFERENCIAS BIBLIOGRÁFICAS

- Alvares, A. (2019). Planificación de LIFTIC Ganadero: aplicación web para la administración de fincas ganaderas: Trabajo Fin de Master. Recuperado de <https://reunir.unir.net/bitstream/handle/123456789/9646/Castillo%20%20C3%81lvarez%20%20Anuar%20Farid.pdf?sequence=1&isAllowed=y>
- Agrocalidad. (2019). Agencia Ecuatoriana de Aseguramiento de la Calidad del Agro. Recuperado de <http://www.agrocalidad.gob.ec/agrocalidad-implemento-la-emisionelectronica-de-guias-de-movilizacion-de-bovinos/>
- Asamblea Nacional. (2019). Asamblea Nacional República del Ecuador. Recuperado el 24 de ENERO de 2019, de La Ganadería Bovina: Recuperado de <http://www.asambleanacional.gob.ec/es/contenido/laganaderia-bovina-0>
- Barrantes, R. (2019). Investigación, Un camino al conocimiento, Un Enfoque Cualitativo, Cuantitativo y Mixto. San José, Costa Rica, Editorial EUNED.
- Banco Central del Ecuador. (2019). La economía ecuatoriana creció 14 en 2018. Recuperado de <https://www.bce.fin.ec/index.php/boletines-de-prensa-archivo/item/1158-la-economia-ecuadoriana-crecio-14-en-2018>
- Berners, L. (2019). HyperText and CERN. An explanation of hypertext, and why it is important for CERN. A background document explaining the ideas behind this project.
- Borges. (2021). Servidor Web: ¿Qué es? ¿Para qué sirve? - Infranetworking. <https://blog.infranetworking.com/servidor-web/>
- Cajas, L., & Javier, J. (2020). Sistema de información ganadero para la automatización de la producción lechera y control reproductivo para la Facultad de Veterinaria Hacienda de Uyumbicho. <http://www.dspace.uce.edu.ec/handle/25000/283>Castillo-Álvarez, A. F. (2019). Planificación de LIFTIC Ganadero: aplicación web para la administración de fincas ganaderas. <https://reunir.unir.net/handle/123456789/9646>
- Castells, M. (2020). Lagalaxia Internet- Reflexiones sobre Internet, empresa y sociedad. Barcelona.
- Carballo. Y. (2020) Programación Orientada a Objetos POO. Recuperado de [http://webdelprofesor.ula.ve/ingenieria/hyelitza/materias/programacion2/o/xo/Profayusneyi\\_Tema8\\_POOClasesyObjetos.pdf](http://webdelprofesor.ula.ve/ingenieria/hyelitza/materias/programacion2/o/xo/Profayusneyi_Tema8_POOClasesyObjetos.pdf)
- Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2021). PHP y MYSQL Tecnologías para el desarrollo de aplicaciones web. Fernández: Díaz de Santos.
- Coutiño L. (2021). Estructura de la base de datos. <https://sites.google.com/site/mariluz20188/2-4-estructura-de-la-base-de-datos>

- Crespo, R. (2019). SISTEMA INFORMÁTICO PARA LA GESTIÓN GANADERA BOVINA EN GUATEMALA. Recuperado de <https://core.ac.uk/download/pdf/228879618.pdf>
- Dimas, A y Barroso, B. (2019). IMPLEMENTACION DE UN SISTEMA DE IDENTIFICACION GANADERO DE BAJO COSTO PARA EL CONTROL DE EVENTOS PRODUCTIVOS EN EL SECTOR PECUARIO, MEDIANTE DISPOSITIVOS MÓVILES: TRABAJO DE GRADO PARA OBTENER EL TITULO DE INGENIERO DE SISTEMAS. Recuperado de <https://repositorio.unicordoba.edu.co/bitstream/handle/ucordoba/510/TESIS%20VACKUP%20DOCUMENTACION.pdf?sequence=1&isAllowed=y>
- Ecuador. Ministerio de Agricultura, Ganadería, Acuacultura y Pesca. (2019). El sector agropecuario ecuatoriano: análisis histórico y prospectiva a 2025. MAGAP. Recuperado de <http://www.competencias.gob.ec/wp-content/uploads/2017/05/01PPP2016-POLITICA01.pdf>
- García, F. (2019). Ingeniería del Software. Recuperado de: <https://repositorio.grial.eu/bitstream/grial/1228/1/07-rep.pdf>
- Gillis, A. (2020). Servidor Web. Estados Unidos: TechTarget. Recuperado de <https://searchdatacenter.techtarget.com/es/definicion/Servidor-Web>
- Gustavo Chavana. (2021). TIPOS DE VARIABLES EN PROGRAMACIÓN: CONCEPTOS Y MÁS. <https://notatecnologica.com/programas/1639tipos-de-variables-en-programacion/>
- Hernández R., Fernández C. y Baptista P. (2006) Metodología de la investigación. México. Editorial Mc. Graw Hill
- Hernández, R., Fernández, C., y Baptista, P. (2014). *Metodología de la investigación*. Arturo, C., Álvarez, M. (2011). Cuantitativa y cualitativa: Guía didáctica.
- Hidalgo Cumbicos, M. R., Vargas González, O. N., & Vite Cevallos, H. A. (2020). *Análisis situacional de la actividad ganadera en la parroquia Palmales del cantón Arenillas*. Revista Metropolitana de Ciencias Aplicadas, 3(2), 124-130.
- Ilusionestudio. (2019). ¿Qué es un framework web y qué ventajas aportan? España: Ilusionestudio. Recuperado de <https://www.illusionstudio.es/que-es-un-frameworkweb>
- Ingenio Learning. (24 de julio de 2020). ¿Qué es Django y cómo puede facilitar tus labores al programar? Perú: Ingenio Learning. Recuperado de <https://ingenio.edu.pe/que-esdjango-y-como-puede-facilitar-tus-labores/>
- Kinsta. (8 de noviembre del 2020). ¿Qué Es Nginx y Cómo Funciona? España: Kinsta. Recuperado de <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>
- Marín, R. (16 de abril de 2019). Los gestores de bases de datos más usados en la actualidad España: Revista Digital INESEM. Recuperado de <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-masusados/>

- Merchán, O. (2019). Diseño de Base de Datos. Recuperado de <http://publicaciones.uazuay.edu.ec/index.php/ceuzuay/catalog/view/5/5/37-2>
- Montilva, J. (2019). Desarrollo de sistemas de información. 2da edición. Universidad de los Andes. Mérida. Venezuela
- Morales, J. (2020). *Características y tipos de bases de datos*. IBM Developer. Recuperado de <https://developer.ibm.com/es/articles/tipos-bases-de-datos/>
- Morillo, G y Zambrano, M. (2020). *Software de Control Ganadero en el hato bovino de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López*. [ Tesis de Grado, Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López]. <http://repositorio.espam.edu.ec/handle/42000/90>
- Organización de las Naciones Unidas para Alimentación y la Agricultura. (2018). Producción animal. FAO. Recuperado de <http://www.fao.org/animal-production/es/>
- Prieto, E. (2023). Etapas del Desarrollo de Software. Tiffin University. Recuperado de <https://global.tiffin.edu/noticias/cuales-son-las-etapas-del-desarrollo-de-software>
- Producción animal | FAO | Organización de las Naciones Unidas para la Alimentación y la Agricultura*. (n.d.). Retrieved December 19, 2021, from <https://www.fao.org/animal-production/es/>
- Quiroa, M. (2021). *Diferencia entre administrar y gestionar*. Economipedia. Recuperado de <https://economipedia.com/definiciones/diferencia-entre-administrar-y-gestionar.html#:~:text=Sin%20duda%2C%20la%20diferencia%20entre,poner%20en%20marcha%20lo%20planificado.>
- Sampieri, R., Fernández, C., & Baptista, P. (2014). Metodología de la investigación (6a. ed. --.). México D.F.: McGraw-Hill.
- SINNAPS. (2021). *METODOLOGÍA XP O PROGRAMACIÓN EXTREMA*. Recuperado de <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp#metodologia-xp-fases>
- Sergio, A. (2019). Ventajas de la metodología XP. Recuperado de <https://iswugaps2extremeprogramming.wordpress.com/2015/09/14/ventajas-y-desventajas/>
- Senn, J. (2019). Análisis y diseño de Sistemas de información. 2da. Edición. McGraw-Hill Interamericana
- Stoner, J. (2020). Administración. 6ta. Edición. Prentice-Hall. México.
- Silberschatz, A., Korth, H. y Sudarshan, S. (2019). Fundamentos de bases de datos. 4ta edición. España. Editorial Concepción Fernández Madrid.

- Tamayo y Tamayo, M. (2019) *El proceso de la investigación científica* (3ra ed). México, Limusa. Recuperado de: [https://www.usbcali.edu.co/sites/default/files/documentodeconsultacomplementario-el\\_proyecto\\_de\\_investigacion.pdf](https://www.usbcali.edu.co/sites/default/files/documentodeconsultacomplementario-el_proyecto_de_investigacion.pdf)
- Tecnológica, T., Darío, R., Valenzuela, C., Amílcar, O., & Acosta, R. (2020). *SISTEMA INFORMÁTICO PARA LA GESTIÓN GANADERA BOVINA EN GUATEMALA*. 7(1).
- Torres, A. (2017). *ESTUDIO DE LA PREFACTIBILIDAD PARA LA IMPLEMENTACION DE LA PRODUCCION Y COMERCIALIZACION DE LA PRODUCCION DE LECHE CRUDA EN LA FINCA "LA FLORESTIA"*. Quito: Escuela Politécnica Nacional.
- Toapanta Quinatoa, E. P., & Zhingre Sánchez, M. M. (2017). *Sistema informático de gestión del ganado vacuno de la hacienda "La María" ubicada en el barrio Lasso de la parroquia San Lorenzo de Tanicuchi*. <http://repositorio.utc.edu.ec/handle/27000/4057>
- Tokola, T., Turkia, A., Sarkeala, J. y Soimasuo, J. (2019). An entity-relationship model for forest inventory. *Can. J. For. Res.* 27: 1586-1594.
- Universidad Politécnica Estatal del Carchi. (2022). *Carrera de Computación Misión y Visión*. Recuperado de <https://www.upec.edu.ec/subsitios/computacion/>
- Un paseo por C# - Guía de C# | Microsoft Docs*. (n.d.). Retrieved December 19, 2021, from <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>
- Universidad de Alicante. (2021). *Modelo vista controlador (MVC)*.
- Vasco, G. J. E.-. (2020). *Navegadores web - Servicio Web - Euskadi.eus*. Basque Administration Web Portal. Recuperado de <https://www.euskadi.eus/navegadores-web/web01-a2wz/es/>
- Vila. (2019). *La Metodología XP: la metodología de desarrollo de software más exitosa*. <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/>
- Villalobos, G. M., Darío, G., Sánchez, C., Alberto, D., & Gutiérrez, B. (2020). Abril de 2019. *Scientia et Technica Año XVI, 44*.

## VII. ANEXOS

### Anexo 1. Acta de la sustentación de Predefensa del TIC



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI**

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

**ACTA**

DE LA SUSTENTACIÓN ORAL DE LA PREDENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR



ESTUDIANTE:	MORA HERRERA JEFFERSON FERNANDO	CÉDULA DE IDENTIDAD:	0403120274
PERÍODO ACADÉMICO:	2023B		
PRESIDENTE TRIBUNAL:	MSC. JEFFREY ALEX NARANJO CEBALDO	DOCENTE TUTOR:	MSC. HIDALGO GILMARRO JAIRO VLADIMIR
DOCENTE:	MSC. LASCANO RIVERA SAMUEL BERLUMÍN		
TEMA DEL TIC:	"Sistema informático para la administración y gestión de conocimientos de la Universidad Politécnica"		

No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	8,17	Ajustar los objetivos hacia los contenidos de estudio
2	FUNDAMENTACIÓN TEÓRICA	8,17	
3	METODOLOGÍA	8,50	
4	RESULTADOS	8,17	Añadir el documento de requerimientos de software a los anexos, organizar de mejor forma el levantamiento de requisitos
5	DISCUSIÓN	8,50	
6	CONCLUSIONES Y RECOMENDACIONES	8,50	
7	INFLEXIA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	8,83	
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	8,50	Revisar el formato mejorar la organización y calidad de la información

Obtención de una nota de: **8,43** Por lo tanto: **APRUEBA** el dictamen al o los investigadores doctor al siguiente artículo:

ART. 34.- De los sustentantes que aprueban el informe final del TIC con observaciones.- Los sustentantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firmo en la ciudad de Tulcán el: **jueves, 2 de mayo de 2024**



MSC. JEFFREY ALEX NARANJO CEBALDO  
PRESIDENTE TRIBUNAL



MSC. HIDALGO GILMARRO JAIRO VLADIMIR  
DOCENTE TUTOR



MSC. LASCANO RIVERA SAMUEL BERLUMÍN  
DOCENTE

**Figura 31.** Acta de la sustentación de Predefensa del TIC



## UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES  
CARRERA DE COMPUTACIÓN

### ACTA

#### DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

ESTUDIANTE:	FORTILLA YAMBERLA HÉCTOR ARMANDO	CÉDULA DE IDENTIDAD:	1004820773
PERIODO ACADÉMICO:	2023B		
PRESIDENTE TRIBUNAL:	MSc. JEFFERY ALEX NARANJO CEDAÑO	DOCENTE TUTOR:	MSc. HIDALGO GUILJARRO JAIRO VLADIMIR
DOCENTE:	MSc. LASCANO RIVERA SAMUEL BENJAMÍN		
TEMA DEL TIC:	"Sistema informático para la administración y gestión de información de la hacienda ProduAgro"		
No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
	PROBLEMA - OBJETIVOS	8,17	Ajustar los objetivos hacia las variables de estudio
2	FUNDAMENTACIÓN TEÓRICA	8,17	
3	METODOLOGÍA	8,90	
4	RESULTADOS	8,17	Agregar el documento de requerimientos de software a los anexos, organizar de mejor forma el levantamiento de requisitos
5	DISCUSIÓN	8,90	
6	CONCLUSIONES Y RECOMENDACIONES	8,90	
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	8,83	
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	8,80	Revisar el formato, mejorar la organización y calidad de la información

Obteniendo una nota de: **8,43** Por lo tanto, **APRUEBA** : debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el **Jueves, 2 de mayo de 2024**

MSc. JEFFERY ALEX NARANJO CEDAÑO  
PRESIDENTE TRIBUNAL

MSc. HIDALGO GUILJARRO JAIRO VLADIMIR  
DOCENTE TUTOR

MSc. LASCANO RIVERA SAMUEL BENJAMÍN  
DOCENTE

Figura 32. Acta de la sustentación de Predefensa del TIC.

**Anexo 2.** Certificado del abstract por parte de idiomas



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI**  
**FOREIGN AND NATIVE LANGUAGE CENTER**

<b>ABSTRACT- EVALUATION SHEET</b>				
<b>NAME:</b> Mora Herrera Jefferson Fernando y Portilia Yamberia Héctor Armando				
<b>DATE:</b> 6 de mayo de 2024				
<b>Topic:</b> "Sistema informático para la administración y gestión de información de la hacienda ProduAgro."				
<b>MARKS AWARDED</b>				
<b>QUANTITATIVE AND QUALITATIVE</b>				
<b>VOCABULARY AND WORD USE</b>	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1 Vera Játiva Edwin Andrés, 5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
<b>WRITING COHESION</b>	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
<b>ARGUMENT</b>	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
<b>CREATIVITY</b>	Outstanding flow of ideas and events	Good flow of ideas and events	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
<b>SCIENTIFIC SUSTAINABILITY</b>	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
<b>TOTAL/AVERAGE</b>	<b>TOTAL 9</b>			
	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED			

**Figura 33.** Certificado del abstract por parte de idiomas.



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL  
CARCHI FOREIGN AND NATIVE LANGUAGE  
CENTER**

**Informe sobre el Abstract de Artículo Científico o Investigación.**

**Autor:** Mora Herrera Jefferson Fernando y Portilla Yamberlia Héctor Armando

**Fecha de recepción del abstract:** 6 de mayo de 2024

**Fecha de entrega del informe:** 6 de mayo de 2024

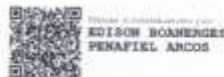
El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

**Observaciones:**

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según los rubrics de evaluación de la traducción en Inglés, ésta alcanza un valor de 9, por lo cual se valida dicho trabajo.

Atentamente



Ing. Edison Peñafiel Arcos MSc  
Coordinador del CIDEN

**Figura 34.** Informe Abstract por parte del CIDEN.

**Anexo 3.** Oficio de autorización para realizar la investigación.

Tulcán, 24 de agosto del 2021

Sr. Mauricio Tirira

Administrador de la Hacienda ProduAgro

Asunto: Permiso para realizar trabajo de investigación.

**Presente. -**

Nosotros, Portilla Yamberla Héctor Armando con cédula de identidad 1004820773 y Mora Herrera Jefferson Fernando con cédula de identidad 0402126296 estudiantes de la Carrera de Computación, Facultad de Industria Agropecuarias y Ciencias Ambientales de la Universidad Politécnica Estatal del Carchi ante usted. Respetuosamente nos presentamos y exponemos: que, previos a culminar la carrera profesional de ingeniería en Computación, solicitamos a usted permiso para realizar trabajo de investigación con el tema: " Sistema informático para la administración y gestión de información de la hacienda ProduAgro" para optar el grado de ingenieros

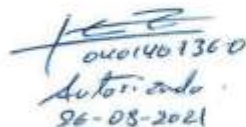
**POR LO EXPUESTO:**

Le agradezco de antemano por su amable atención y ruego a usted acceder a nuestra solicitud.

**Atentamente**

  
Héctor Armando Portilla Yamberla  
hector.portilla@upec.edu.ec  
Cel. 0986243268  
Ci. 100482077-3

  
Jefferson Fernando Mora Herrera  
jefferson.mora@upec.edu.ec  
Cel. 0968602784  
Ci. 040212629-6

  
0401401360  
Autorizado  
26-08-2021

**Figura 35.** Oficio de autorización para realizar la investigación.

**Anexo 4.** Ficha de entrevista

**VALIDEZ DEL INSTRUMENTO DE INVESTIGACIÓN (ENTREVISTA)**  
**FICHA DE ENTREVISTA**

**Datos del entrevistado**

<b>Apellidos:</b>	<b>Nombres:</b>
<b>Edad:</b>	<b>Cédula:</b>
<b>Área:</b>	<b>Cargo:</b>

**Datos del entrevistador**

<b>Apellidos:</b>	<b>Nombre:</b>
-------------------	----------------

**Datos de la entrevista**

<b>Lugar:</b>	<b>Fecha:</b>
<b>Hora de inicio:</b>	<b>Hora de finalización:</b>
<b>Observaciones:</b>	

**Figura 36.** Ficha de entrevista.

**Anexo 5.** Entrevista al administrador encargado de la hacienda ProduAgro

**VALIDEZ DEL INSTRUMENTO DE INVESTIGACIÓN (ENTREVISTA)**  
**FICHA DE ENTREVISTA**

**Datos del entrevistado**

Apellidos: Portilla Zambrano	Nombres: Segunda Leopoldo
Edad: 43	Cédula: 040140873-8
Área: Administrador	Cargo: Administrador

**Datos del entrevistador**

Apellidos: Mora Herrera	Nombre: Jefferson Fernando
-------------------------	----------------------------

**Datos de la entrevista**

Lugar: Parroquia San Isidro - Hacienda ProduAgro	Fecha: 21 - Diciembre - 2021
Hora de inicio: 10:00 am	Hora de finalización: 11:30 am
Observaciones:	

**Figura 37.** Entrevista al administrador encargado de la hacienda ProduAgro.

## GUÍA DE ENTREVISTA

La siguiente información tiene como objetivo obtener información verídica, que aporta significativamente para el desarrollo de la investigación.

### Entrevista

#### 1) ¿Qué entiende usted por un sistema informático?

Para mí es un S.I. es como una herramienta muy avanzada que usa la computadora para guardar y organizar toda la información que necesitamos en el trabajo. Es como tener un gran cuaderno digital donde podemos guardar todo lo que hacemos con el ganado. No sé mucho como funciona pero sé que es más fácil manejar todo lo que pasa en la hacienda sin tener que usar papel o que se pierda la información.

#### 2) ¿Cómo se maneja actualmente la información sobre la administración y gestión de la hacienda ProduAgro?

Actualmente se maneja la información a través de registros manuales en papel, cuadernos y en algunas cosas en Excel.

#### 3) ¿Se han implementado soluciones informáticas anteriormente en la hacienda ProduAgro para la administración y gestión de información?

No, hasta el momento no se han implementado soluciones informáticas para la administración y gestión de la hacienda.

#### 4) ¿Qué procesos incluye la administración y gestión de la hacienda ProduAgro?

Incluye el mantenimiento de registros, supervisión de eventos, elaboración de informes e inventarios, gestión de producción de leche y reproducción de ganado y mantenimiento de las herramientas utilizadas para la producción lechera.

#### 5) ¿Qué tipo de información se maneja en cada uno de los procesos dentro de la hacienda ProduAgro?

Figura 38. Entrevista al administrador encargado de la hacienda ProduAgro

Se maneja información sobre producción lechera, registros de nacimiento y reproducción, datos veterinarios, inventarios de insumos y productos, registros de almacenamiento de infraestructuras y equipos y datos de los campos.....

**6) ¿Quién o quiénes se encargan de gestionar las actividades o procesos relacionados al ganado bovino?**

Los encargados de la gestión de actividades con el ganado bovino son: el capataz que verifica todo y se pone en contacto con el adm. Joraleño que se encarga de la crianza de terneros; Joraleño 2 se encarga de la crianza de vacunas, mantenimiento de ordeño, inseminación, Joraleño 3 se encarga del ganado para el ordeño, ganado preñado administración de potreros y dos empleados que son reemplazos y no tienen trabajo fijo

**7) ¿Cómo se llevan a cabo los procesos de traslados del ganado bovino, parcelas, potreros y registros de mantenimientos?**

Estos procesos se realizan de manera mensual, registrando en modernos o documentos físicos los detalles de los traslados de ganado, uso de parcelas y potreros así como los mantenimientos realizados.....

**8) ¿Qué tiempo toma realizar un reporte relacionado a los chequeos del ganado inseminado en la hacienda?**

El tiempo para realizar un reporte varía por lo debido al manejo manual de la información puede llevar desde unas horas hasta varios días para copiar toda la información almacenada que se requiere.....

**9) ¿Cuántos trabajadores conforman para la administración y gestión de la hacienda?**

La cantidad de trabajadores dedicados a la administración y gestión de la hacienda son 7 que son el administrador, capataz, tres joraleños y dos de la gestión ganadera en la hacienda.....

Figura 39. Entrevista al administrador encargado de la hacienda ProduAgro.

10) ¿Está de acuerdo usted con la implementación de una solución  
Informática para la administración y gestión de la información en  
la hacienda?

*Si estoy de acuerdo en una solución informática que mejoraría.....  
significativamente la eficiencia, precisión y accesibilidad de la.....  
gestión de la información en la hacienda.....*

---

*Leopoldo Portilla*

**Figura 40.** Entrevista al administrador encargado de la hacienda ProduAgro.



- Libros diarios  
 Hojas  
 Computadora  
 Teléfono Inteligente  
 Otro.....
- 4. ¿Cómo considera los métodos de administración y gestión utilizados actualmente en la hacienda?**  
 Muy buena  
 Buena  
 Regular  
 Mala  
 Muy mala
- 5. ¿Cuáles son los problemas más frecuentes que tiene la hacienda al momento de manejar la información?**  
 Pérdida de información  
 Información deteriorada con el tiempo  
 Falta de integridad en la información  
 Poca accesibilidad  
 Falta de organización y presentación de los datos  
 Otro.....
- 6. ¿Cómo considera su capacidad para el uso de la tecnología?**  
 Ninguno       Bajo       Medio       Alto
- 7. ¿Estaría dispuesto a utilizar un sistema informático (es una técnica que permite el almacenamiento y el proceso de información a través del uso de la tecnología) para la administración y gestión de información de la hacienda ProduAgro?**  
 Sí       No
- 8. ¿Usted cree que, con el uso de un sistema informático, se mejore la administración y gestión de la información que usted maneja?**  
 Sí       No
- 9. ¿Qué procesos le gustaría que tuviera el sistema informático?**

**Figura 42.** Encuesta para los empleados de la hacienda.



## **Anexo 7. Plan de desarrollo de software**

### **Plan de Desarrollo de Software – [v1.1]**

**Sistema informático para la administración y gestión de información de la hacienda  
ProduAgro**

**Generado por:**

**Mora Herrera Jefferson Fernando**

**Portilla Yamberla Héctor Armando**

## **Sobre este Documento**

El presente documento detalla toda la información de la cual fue el desarrollo del sistema informático "ProduAgro Sys", diseñado específicamente para la administración y gestión eficiente de la información dentro de la hacienda ProduAgro.

Además, este documento hace énfasis en la utilización de la metodología de desarrollo de software Extreme Programming (XP), adoptada para dirigir el ciclo de vida del proyecto. La metodología XP se destaca por su enfoque ágil y flexible, favoreciendo la adaptabilidad ante cambios y la mejora continua. Las fases de desarrollo de XP implementadas en este proyecto incluyen: planificación, diseño, codificación y pruebas:

Este plan de desarrollo sirve como un recurso integral, proporcionando orientación detallada sobre la infraestructura técnica del sistema "ProduAgro", así como sobre las herramientas y metodologías empleadas durante su desarrollo. Su propósito es asegurar una comprensión clara del enfoque innovador adoptado para la gestión y administración eficaz de la hacienda ProduAgro.

## **Generalidades del Proyecto**

Proporcionar algo de contexto facilita una mayor comprensión de cualquier situación, así que es aconsejable ofrecer una descripción de los objetivos y principios fundamentales del proyecto, además de definir lo que finalmente será reconocido como el resultado del mismo. Es evidente que el proyecto apunta a desarrollar un software o aplicación, pero también se reconoce que este vendrá acompañado de elementos adicionales, como la documentación técnica y los manuales para usuarios.

## **Descripción de Proyecto**

El proyecto "ProduAgro Sys" se concibe como una solución tecnológica integral destinada a la optimización de la administración y gestión de la hacienda ProduAgro. Esta iniciativa surge en respuesta a la necesidad de modernizar y agilizar los procesos internos de la hacienda, desde la gestión de ganado bovino hasta el seguimiento de potreros y la producción lechera. El sistema propuesto promete revolucionar la manera en que se administra la información ganadera, facilitando una toma de decisiones más informada y basada en datos reales y actualizados.

El sistema "ProduAgro Sys" se caracteriza por su enfoque en la eficiencia, garantizando que la gestión y administración de la información se realice de manera segura y sin complicaciones. El uso del patrón de diseño MVC permite una separación clara entre la lógica de negocio, la interfaz de usuario y los datos, lo que facilita el mantenimiento y la escalabilidad del sistema.

## **Propósito**

El propósito principal del proyecto "ProduAgro Sys" es brindar una herramienta robusta y confiable que permita una gestión y administración eficiente de la hacienda ProduAgro, mejorando significativamente la productividad, la seguridad de los datos y la toma de decisiones basada en información precisa y oportuna.

## Historial de revisiones

**Tabla 10.** Historial de Revisiones.

Fecha	Versión	Descripción	Autor	Aprobado por
	1.0	Revisión y aprobación del proyecto	Fernando Mora Héctor Portilla	

## Alcance

El alcance del proyecto de investigación se orienta hacia la recopilación, categorización y sistematización de las funciones primordiales relacionadas con la administración ganadera, específicamente enfocadas en la gestión del ganado bovino dentro de la hacienda. En consecuencia, nuestro estudio no se extenderá a procesos o actividades distintas a la gestión ganadera, sino que mantendrá un enfoque general sobre esta área. Cabe destacar que, dado que la contabilidad de la hacienda se gestiona externamente, este aspecto queda fuera del ámbito de nuestra investigación. El objetivo primordial es analizar de manera exhaustiva la realidad operativa relacionada con el ganado bovino, con el fin de interpretar adecuadamente los datos obtenidos para mejorar y organizar eficazmente la información de la Hacienda ProduAgro. Esto, a su vez, facilitará una toma de decisiones más efectiva y basada en información precisa.

## Objetivos

### Objetivo General:

- Desarrollar un sistema prototipo para la administración y gestión de información en la hacienda ProduAgro con respecto a los procesos y actividades que realizan.

### Objetivos Específicos:

- Recopilar los requerimientos funcionales empleando herramientas de la metodología de desarrollo XP.
- Seleccionar las herramientas de programación adecuadas para el desarrollo del sistema de administración y gestión de información de la hacienda.
- Adoptar las normas establecidas por la metodología de desarrollo de software para la ejecución del proyecto.

## Asunciones y Restricciones

En base a la entrevista estructurada y a los resultados de la encuesta, se recopilaron las siguientes restricciones:

- Se debe analizar a fondo el funcionamiento de los procesos administrativos y toda la información que esté involucrada.
- El sistema de administración y gestión de información de la hacienda debe ser una plataforma web que integre los módulos que el administrador solicite.
- La interfaz de usuario debe ser fácil de usar para el usuario administrador y jornaleros.

## Artículos y Artefactos para entregar

**Tabla 11.** Artículos y artefactos para entregar.

Qué	Propósito
Manual de usuario	Informar a los clientes las funciones principales del sistema mediante capturas de pantalla y explicaciones.
Aplicación	Sistema informático para la administración y gestión de información de la hacienda ProduAgro.
Base de Datos	Almacenamiento de los procesos administrativos de la hacienda ProduAgro.

## Evolución del presente Documento

El plan de desarrollo de software será revisado cada vez que haya una modificación de gran magnitud del sistema de administración y gestión de información de la hacienda ProduAgro.

## ORGANIZACIÓN DEL PROYECTO

### Recursos humanos y profesionales

**Tabla 12.** Recursos humanos y profesionales.

Quien	Información del contacto	Celular
Mora Herrera Jefferson Fernando	Jefferson.mora@upec.edu.ec	0968602784
Portilla Yamberla Héctor Armando	Hector.provilla@upec.edu.ec	0986243268

## Roles y responsabilidades

**Tabla 13.** Roles y responsabilidades.

Nombre	Descripción	Rol XP
Msc. Jairo Hidalgo	Docente Tutor	Consultor
Fernando Mora	Investigador	Programador
Héctor Portilla	Investigador	Programador
Sr. Mauricio Tirira	Administrador de la Hacienda ProduAgro	Cliente

## GESTIÓN DEL PROYECTO

### Plan de proyecto

**Tabla 14.** Plan de proyecto.

Fase	Iteraciones	Duración
Fase de planificación	1	2 meses
Fase de diseño	1	4 meses
Fase de codificación	1	4 meses
Fase de pruebas	1	2 meses

### Fases y líneas base

**Tabla 15.** Fases y líneas base.

Fase	Objetivo asociado	Riesgos contemplados
Fase de planificación	En la primera fase se recopiló información acerca de los procesos operativos y administrativos y las herramientas a utilizar en el desarrollo del sistema.	Recopilar información de los requerimientos del cliente para la planificación de actividades.
Fase de diseño	La fase de diseño se encargó de analizar los componentes que serán utilizados en la distribución del software, lo cual permitirá que la estructura se optimice de manera constante.	Diseñar prototipos en base a los requerimientos obtenidos en la encuesta y la entrevista.
Fase de codificación	Se lleva a cabo la programación lo que permitirá que exista una mutua retroalimentación acerca del desarrollo del software. Además, se identificará de forma más efectiva los errores cometidos.	Desarrollar las funcionalidades de código para integrar la información de los procesos de la hacienda ProduAgro.
Fase de pruebas	Las pruebas unitarias y de aceptación, permitirán crear un ciclo de desarrollo más rápido, lo que permitirá que la aprobación del cliente sea validada.	Realizar pruebas unitarias de las funcionalidades principales del sistema y pruebas de aceptación para garantizar la satisfacción del cliente.

## Objetivos y riesgos de cada fase

Tabla 16. Objetivos y riesgos de cada fase.

Fase	Objetivo asociado	Riesgos contemplados
Fase de planificación	Recopilar información de los requerimientos de los clientes.	Cambio de requerimientos por parte del cliente.
Fase de diseño	Diseñar prototipos en base a los requerimientos.	Fallo en el diseño de las interfaces.
Fase de codificación	Desarrollar las funcionalidades de código para la integración de la información de los procesos de la hacienda ProduAgro.	Pérdida del código fuente. Poca experiencia de los desarrolladores. Inconvenientes con el diseño de la base de datos.
Fase de pruebas	Realizar pruebas unitarias de las funcionalidades principales del sistema y pruebas de aceptación para garantizar la satisfacción del cliente.	Errores y falsos positivos en las pruebas unitarias.

## Diagrama de Gantt

Tabla 17. Diagrama de Gantt.

Fase	Fecha de inicio	Duración en meses	Fecha finalización
Planificación	01/10/2022	2	20/12/2022
Diseño	05/01/2022	4	28/04/2023
Codificación	02/05/2023	4	27/09/2023
Pruebas	03/10/2023	2	20/12/2023

## Calendario del proyecto

Tabla 18. Calendario del proyecto.

Meses	2022							2023							
	OC	NO	DI	EN	FE	MA	AB	MA	JU	JU	AG	SE	OC	NO	DI
Fases	T	V	C	E	B	R	R	Y	N	L	O	P	T	V	C
Planificación	■														
Diseño				■											
Codificación								■							
Pruebas													■		

## Control de desviación a la planificación

Tabla 19. Control de desviación a la planificación.

Ítem	Probabilidad	Impacto
1	Improbable	Bajo
2	Poco probable	Mínimo
3	Probabilidad media	Medio
4	Casi probable	Alto

5	Probable	Muy alto
---	----------	----------

## Matriz de riesgos

Tabla 20. Matriz de riesgo.

Ítem	Riesgo	Probabilidad	Impacto	Valor de riesgo
R1	Falta de compromiso por parte del cliente.	4	4	16
R2	Modificaciones en los requerimientos por parte del cliente.	4	5	20
R3	Errores en el diseño de interfaces.	2	3	6
R4	El diseño del aplicativo no cumple las expectativas del cliente.	4	4	16
R5	Pérdida o eliminación del código fuente del sistema.	3	5	15
R6	Poco nivel de experiencia de los programadores.	4	4	16
R7	Depuración ineficiente de la base de datos del sistema.	3	4	12

## Matriz de calor

	Riesgo alto
	Riesgo medio
	Riesgo bajo

Tabla 21. Matriz de calor.

IMPACTO	5			R5		
	4			R4 R6 R7	R1	
	3		R3	R2		
	2					
	1					
		1	2	3	4	5
	PROBABILIDAD					

## Estudio de factibilidad

### Factibilidad Técnica

Para el desarrollo de este proyecto se elaboró una lista de los recursos que serán utilizados, tales como el hardware y software.

El sistema informático se va a desarrollar con la tecnología C#. Bootstrap con el Framework MVC, MySQL, estos recursos fueron elegidos por ser Open Source, esto es un beneficio al no generar ningún costo en la investigación. Añadiendo a ello se cuenta con los conocimientos necesarios para su desarrollo.

## Hardware

**Tabla 22.** Recursos Hardware.

Tipo de Recurso	Nombre del Recurso	Descripción	Cantidad
Hardware	Equipo de computación	Laptop HP Core i5 Disco duro de 1TB Memoria Ram Kingston 4GB	2
	Servidor Impresora	Epson L3250	1

## Software

**Tabla 23.** Recurso de software.

Tipo de Recurso	Nombre del Recurso	Descripción	Cantidad
Software	C#- HTML 5	Lenguaje de programación	1
		Framework Frontend	1
		Framework de aplicaciones web	1
	SQL Server	Sistema gestor de base de datos	1
	Visual Studio	Entorno de programación	1

Los investigadores y desarrolladores del presente proyecto cuentan con las herramientas y equipos necesarios para la elaboración del proyecto, así mismo se tiene una conexión estable de internet que permite el trabajo y la comunicación en equipo, se concluye que los recursos presentados son idóneos y existe una factibilidad técnica.

## Factibilidad Económica

El presupuesto que abarca el proyecto de investigación está considerado tanto software, hardware, talento humano y materiales de oficina.

**Tabla 24.** Recursos de factibilidad económica.

<b>Costos de Hardware</b>			
<b>Descripción</b>	<b>Cantidad</b>	<b>Costo real</b>	<b>Costo referencial</b>
Equipos de computación	2	00,00	\$ 1600,00
Servidor	1	00,00	
Impresora	1	00,00	\$ 250,00
<b>Total de hardware</b>		<b>00,00</b>	<b>\$ 1850,00</b>
<b>Costos de Software</b>			
<b>Descripción</b>	<b>Cantidad</b>	<b>Costo real</b>	<b>Costo referencial</b>
C#	1	00,00	00,00
SQL Server Desarrollador	1	00,00	3,586
Visual Studio Community	1	00,00	00,00
<b>Total de software</b>		<b>00,00</b>	<b>\$ 3,586</b>
<b>Talento Humano</b>			
<b>Personal</b>	<b>Actividad</b>	<b>Cargo</b>	
Mora Herrera Jefferson Fernando	Desarrollo de la investigación	Estudiante	
Portilla Yamberla Héctor Armando	Desarrollo de la investigación	Estudiante	
Hidalgo Guizarro Jairo Vladimir	Tutor -Lector	Docente	
<b>Materiales de oficina</b>			
<b>Descripción</b>	<b>Cantidad</b>	<b>Costo real</b>	<b>Costo referencial</b>
Internet		\$ 260,00	\$ 360,00
Útiles de oficina		\$ 100,00	\$ 100,00
Impresora	1	\$ 00,00	\$ 250,00
Varios		\$ 150,00	\$ 150,00
<b>Total Materiales de oficina</b>		<b>\$ 410,00</b>	<b>\$ 860,00</b>
<b>Subtotal</b>		<b>\$ 410,00</b>	<b>\$ 7710,00</b>
10% de imprevistos		\$ 61, 00	\$ 771,00
<b>Total</b>		<b>\$ 471,00</b>	<b>\$ 8481,00</b>

En cuanto a los costos de software utilizados para el desarrollo del sistema informático se optó por utilizar software libre, versiones community y versiones de estudiante que no tienen ningún costo adicional, sin embargo, se indica en la tabla los costos referenciales si se llega a contratar dichos softwares.

## Factibilidad Operativa

- **Situación actual**

En la Hacienda ProduAgro se maneja una variedad de información que se encuentran almacenadas en hojas de papel, y en pocas circunstancias en la herramienta de cálculo Excel, dando como resultado una búsqueda de información deficiente, no transparente, manipulada por terceros, dañada y hasta la pérdida de información, todo esto representa un riesgo del mal uso de la información en la hacienda ProduAgro.

- **Situación ideal**

El sistema informático para la administración y gestión pretende mejorar el manejo de la información para un mejor desenvolvimiento en cuanto a su veracidad y transparencia, así mismo al contener un sistema éste genera reportes de varios procesos que se ejecutan en la hacienda ProduAgro, y almacena dicha información como: revisar el estado del ganado, tratamientos veterinarios, revisar sus datos particulares (arte, nombres, edad, partos, estado de gestación, etc.).

**Tabla 25.** Cuadro comparativo entre la Microsoft Excel, cuadernos, libretas físicas y el sistema informático.

Características	Microsoft Excel	Cuadernos/ Libretas	Sistema informático
Acceso mediante Login	X	X	✓
Interfaz amigable e intuitiva	X	X	✓
Almacenamiento en una base de datos	X	X	✓
Base de datos relacional	X	X	✓
Permite la salida de informes en formato PDF y xlsx	✓	X	✓
Permite imprimir informes	X	X	✓
Tablas para almacenar datos	✓	✓	✓
Consultas para encontrar datos	X	X	✓
Plantillas y formularios listos para crear datos		X	✓
Administración de usuarios, roles y operaciones.	X	X	✓
Gestión de procesos como: chequeos, datos veterinarios, etc.	X	X	✓
Gestión de inventarios de productos y pedidos	X	X	✓
Gestión de mantenimientos	X	X	✓
Generar reportes por fechas, identificadores de animales, etc.	X	X	✓
Manipulación indebida de la información	✓	✓	X

## Metodología XP

### Fase de planificación

En esta fase iniciamos estableciendo los roles dentro del proyecto, establecemos una descripción de los módulos de los usuarios del sistema, definimos los módulos presentes en el sistema, así también, con los escenarios de usuario que corresponden a los requisitos funcionales del software.

#### a) Aspectos generales de la organización.

**Empresa:** Hacienda ProduAgro

**Ubicación geográfica:** San Isidro, Caserío Carlizama, Panamericana norte km 34.

**Área:** Administración y gestión de información de la Hacienda ProduAgro.

**Sistema:** Sistema informático para la administración y gestión de información de la hacienda ProduAgro.

**Objeto social:** Servicio Privado.

#### Misión

“Somos una empresa enfocada en la producción y transformación de materia prima, empleando para ello sistemas eficientes e innovadores basados en el desarrollo amigable con el medio ambiente y plenamente sostenible.”

#### Visión

“En el 2024, seremos una empresa modelo del sector, reconocida ampliamente a nivel local y nacional por su seriedad, calidad, innovación y compromiso en el desarrollo social de nuestro país.”

#### b) Asignación de roles

**Tabla 26.** Asignación de roles.

Nombre	Función	Rol
Fernando Mora, Héctor Portilla	Estimar el tiempo de las historias de usuario, desarrollar el código del sistema y realizar las pruebas unitarias.	Programador

Sr. Mauricio Tirira	Establecer las historias de usuario, su importancia y validación de las pruebas de aceptación.	Cliente
Héctor Portilla	Colaborar con el cliente en el desarrollo de las pruebas funcionales, realizar la ejecución y divulgar los resultados.	Encargado de pruebas
Fernando Mora	Supervisar el cumplimiento de las interacciones, retroalimentación al equipo y determinación de los cambios.	Encargado de seguimiento
Fernando Mora, Héctor Portilla	Dirección del cumplimiento de los procesos en XP.	Entrenador
Msc. Jairo Hidalgo	Colaboración de la resolución de problemas.	Consultor
Héctor Portilla	Coordinación de la comunicación con la empresa.	Jefe de proyecto

### **c) Módulos del sistema**

#### **1. Gestión Ganadera**

- Animales
- Medición Lechera
- Muertes
- Descartes
- Pesos
- Toros
- Datos Veterinarios
- Traslados

#### **2. Gestión Reproductiva**

- Inseminaciones
- Chequeo
- Partos
- Abortos

#### **3. Inventario**

- Productos
- Pedidos

#### **4. Gestión de Campo**

#### **5. Mantenimientos**

#### **6. Producción Lechera**

#### **7. Información General**

- Parcelas
- Grupos

- Recursos Humanos

## 8. Reportes

- Animales
- Inseminaciones
- Chequeos
- Partos
- Abortos

### d) Caracterización de procesos

La caracterización de procesos implica comprender y documentar las actividades, roles y recursos involucrados en la ejecución de un proceso específico en una organización. Es fundamental para identificar áreas de mejora y establecer bases para la estandarización y la optimización del proceso. En relación con la norma ISO 9001:2015, la caracterización de procesos es crucial para cumplir con los requisitos de identificación y comprensión de procesos, así como para documentar información sobre su secuencia, interacciones y métodos de aseguramiento de calidad.

P001		Caracterización de Proceso de Registro Bovino		
<b>Proceso:</b> Registro Bovino				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Registrar y mantener actualizada la información de los bovinos, asegurando la integridad y disponibilidad de los datos para una gestión eficiente del ganado.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	Formularios de registro de bovinos  Información sobre el ganado (identificación, edad, raza, estado de salud)  Información del jornalero  Equipos de identificación (marcas, etiquetas electrónicas) Sistema informático para el registro	<b>PLANEAR:</b> -Definir procedimientos de registro de bovinos. -Establecer la frecuencia y los métodos de recolección de datos. -Asignar roles y responsabilidades en el proceso de registro.  <b>HACER:</b> -Realizar la identificación física y digital de cada bovino. -Ingresar la información recolectada en el sistema informático. -Actualizar la base de datos con información relevante sobre cada bovino.  <b>VERIFICAR:</b> -Revisar la consistencia y exactitud de los datos ingresados en el sistema. -Corregir errores y actualizar información cuando sea necesario. -Validar que el proceso  <b>ACTUAR:</b> -Ajustar procedimientos para optimizar el tiempo y los recursos empleados en el registro.	Base de datos de bovinos actualizada  Informes de registro de bovinos  Reportes	Administrador  Jornalero  Dueños del ganado  Veterinarios

Figura 44. P001 Caracterización de proceso de registro bovino.

P002		Caracterización de Proceso de Registro Medición Lechera		
<b>Proceso:</b> Registro Medición Lechera				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Registrar y mantener actualizada la información de la producción lechera de cada bovino, asegurando la precisión y disponibilidad de los datos para optimizar la gestión y producción lechera.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	Formularios de registro de bovinos Información sobre el ganado (identificación, edad, raza, estado de salud) Información del jornalero Equipos de identificación (marcas, etiquetas electrónicas) Sistema informático para el registro	<b>PLANEAR:</b> Establecer la frecuencia de medición y registro (diaria, semanal). <b>HACER:</b> Realizar mediciones de producción lechera de cada bovino. Registrar los datos obtenidos en el sistema informático. Asegurar que los equipos de medición estén calibrados y en buen estado. <b>VERIFICAR:</b> Revisar la exactitud y consistencia de los datos registrados en el sistema. Corregir errores y actualizar la información cuando sea necesario. Validar que el proceso cumpla con los estándares. <b>ACTUAR:</b> Ajustar procedimientos para optimizar el tiempo y los recursos empleados en la medición y registro. Tomar acciones correctivas errores recurrentes en el proceso. Asegurar la capacitación continua de los jornaleros y el administrador sobre nuevas herramientas y técnicas de medición.	Base de datos de bovinos actualizada Informes de registro de bovinos Reportes	Administrador Jornalero Dueños del ganado Veterinarios

**Figura 45.** P002 Caracterización de proceso de registro de medición lechera.

P003	Caracterización de Proceso de Registro de Muertes Bovinos			
<b>Proceso:</b> Registro de Muertes Bovinos				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Registrar y mantener actualizada la información sobre las muertes de bovinos, asegurando la precisión y disponibilidad de los datos para una gestión eficiente del ganado y la toma de decisiones informadas.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	<p>Informes de defunción de bovinos</p> <p>Datos de identificación del bovino fallecido</p> <p>Causas de muerte y condiciones del entorno</p> <p>Información del jornalero que reporta la muerte</p> <p>Sistema informático para el registro de defunciones</p>	<p><b>PLANEAR:</b></p> <ul style="list-style-type: none"> <li>-Definir procedimientos para la recolección y registro de datos sobre muertes de bovinos.</li> <li>-Establecer la forma y el tiempo de reporte de las muertes.</li> <li>-Asignar roles y responsabilidades en el proceso de registro.</li> </ul> <p><b>HACER:</b></p> <ul style="list-style-type: none"> <li>-Recopilar información sobre el bovino fallecido y las circunstancias de su muerte.</li> <li>-Registrar los datos recolectados en el sistema informático.</li> <li>-Asegurar que la causa de muerte esté correctamente documentada.</li> </ul> <p><b>VERIFICAR:</b></p> <ul style="list-style-type: none"> <li>-Revisar la consistencia y exactitud de los datos ingresados en el sistema.</li> <li>-Corregir errores y actualizar información cuando sea necesario.</li> <li>-Validar que el proceso</li> </ul> <p><b>ACTUAR:</b></p> <ul style="list-style-type: none"> <li>-Ajustar procedimientos para optimizar el tiempo y los recursos empleados en el registro.</li> </ul>	<p>Base de datos de muertes de bovinos actualizada</p> <p>Informes detallados de defunciones por período</p>	<p>Administrador</p> <p>Jornalero</p> <p>Dueños del ganado</p> <p>Veterinarios</p>

**Figura 46.** P003 Caracterización de proceso de registro de muertes de bovinos.

P004		Caracterización de Proceso de Descartes Bovinos		
<b>Proceso:</b> Registro de Descartes Bovinos				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Registrar y mantener actualizada la información sobre los bovinos descartados, asegurando la precisión y disponibilidad de los datos para una gestión eficiente del ganado y la toma de decisiones estratégicas.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	<p>Informe de descarte de bovinos</p> <p>Datos de identificación del bovino descartado</p> <p>Razones del descarte (edad, productividad, salud, etc.)</p> <p>Información del jornalero que reporta el descarte</p>	<p><b>PLANEAR:</b></p> <ul style="list-style-type: none"> <li>-Definir procedimientos para la recolección y registro de datos sobre descarte de bovinos.</li> <li>-Establecer la forma y el tiempo de reporte de las muertes.</li> </ul> <p><b>HACER:</b></p> <ul style="list-style-type: none"> <li>-Recopilar información sobre el bovino descartado y las razones del descarte.</li> <li>-Registrar los datos recolectados en el sistema informático.</li> <li>-Asegurar que las razones del descarte estén correctamente documentadas.</li> </ul> <p><b>VERIFICAR:</b></p> <ul style="list-style-type: none"> <li>-Revisar la consistencia y exactitud de los datos ingresados en el sistema.</li> <li>-Corregir errores y actualizar información cuando sea necesario.</li> <li>-Validar que el proceso</li> </ul> <p><b>ACTUAR:</b></p> <ul style="list-style-type: none"> <li>-Ajustar procedimientos para optimizar el tiempo y los recursos empleados en el registro.</li> </ul>	<p>Base de datos de descartes de bovinos actualizada</p> <p>Informes detallados de descartes por período</p>	<p>Administrador</p> <p>Jornalero</p> <p>Dueños del ganado</p> <p>Veterinarios</p>

**Figura 47.** P004 Caracterización de proceso de registro de descartes bovinos.

P005		Caracterización de Proceso de Pesos Bovino		
<b>Proceso:</b> Registro de Pesos Bovino				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Registrar y mantener actualizada la información sobre el peso de los bovinos, asegurando la precisión y disponibilidad de los datos para una gestión eficiente del ganado y optimización de la producción.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	Equipos de pesaje (balanzas, básculas electrónicas)  Datos de identificación de los bovinos  Formularios de registro de peso  Información del jornalero que realiza el pesaje	<p><b>PLANEAR:</b></p> <ul style="list-style-type: none"> <li>-Definir procedimientos para la recolección y registro de datos de peso de los bovinos.</li> <li>-Establecer la frecuencia y los métodos de pesaje (mensual, trimestral, etc.).</li> <li>-Establecer la forma y el tiempo de reporte de las muertes.</li> </ul> <p><b>HACER:</b></p> <ul style="list-style-type: none"> <li>-Realizar el pesaje de los bovinos utilizando equipos calibrados y en buen estado.</li> <li>-Registrar los datos de peso obtenidos en el sistema informático.</li> <li>-Asegurar que el proceso de pesaje se realice de manera consistente y precisa.</li> <li>-Registrar los datos recolectados en el sistema informático.</li> <li>-Asegurar que las razones del descarte estén correctamente documentadas.</li> </ul> <p><b>VERIFICAR:</b></p> <ul style="list-style-type: none"> <li>-Revisar la consistencia y exactitud de los datos ingresados en el sistema.</li> <li>-Corregir errores y actualizar información cuando sea necesario.</li> <li>-Validar que el proceso</li> </ul> <p><b>ACTUAR:</b></p> <ul style="list-style-type: none"> <li>-Ajustar procedimientos para optimizar el tiempo y los recursos empleados en el registro.</li> </ul>	Base de datos de pesos de bovinos actualizada  Informes de peso por bovino y por período	Administrador  Jornalero  Dueños del ganado  Veterinarios

**Figura 48.** P005 Caracterización de proceso de registro de pesos bovinos.

P006		Caracterización de Proceso de Datos Veterinarios Bovinos		
<b>Proceso:</b> RegistroDatos Veterinarios Bovinos				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Registrar y mantener actualizada la información de la producción lechera de cada bovino, asegurando la precisión y disponibilidad de los datos para optimizar la gestión y producción lechera.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Veterinario Jornalero	Formularios de registro de bovinos  Información sobre el ganado (identificación, edad, raza, estado de salud)  Información del jornalero Información del Veterinario	<p><b>PLANEAR:</b> -Definir los campos y parámetros requeridos para la recolección de datos veterinarios.</p> <p><b>HACER:</b> -Recolectar datos veterinarios según los protocolos establecidos -Registrar los datos obtenidos en el sistema informático.</p> <p><b>VERIFICAR:</b> -Revisar la exactitud y consistencia de los datos registrados en el sistema.  -Corregir errores y actualizar la información cuando sea necesario. -Validar que el proceso cumpla con los estándares.</p> <p><b>ACTUAR:</b> -Ajustar procedimientos para optimizar el tiempo y los recursos empleados en la medición y registro. -Tomar acciones correctivas errores recurrentes en el proceso.</p>	<p>Base de datos de bovinos actualizada</p> <p>Base de datos actualizada con información sobre la salud y cuidado de los bovinos.</p> <p>Informes de estado de salud y cuidado individual y colectivo de los bovinos.</p>	<p>Administrador</p> <p>Jornalero</p> <p>Dueños del ganado</p> <p>Veterinarios</p>

**Figura 49.** P006 Caracterización de proceso de registro de datos veterinarios.

P007		Caracterización de Proceso de Traslados de Bovinos		
<b>Proceso:</b> Registro de Traslados de Bovinos				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Gestionar de manera eficiente y segura los traslados de bovinos entre diferentes ubicaciones, asegurando el bienestar animal y cumpliendo con los requisitos legales y de sanidad animal.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Veterinario Jornalero	<p>Solicitud de traslado de bovinos.</p> <p>Documentación sanitaria y legal requerida para el traslado.</p> <p>Información sobre el estado de salud y condiciones de los bovinos a ser trasladados.</p> <p>Registro de inventario de bovinos en las ubicaciones de origen y destino. Información del jornalero encargado del traslado.</p>	<p><b>PLANEAR:</b></p> <p>-Evaluar la viabilidad y necesidad del traslado de bovinos.</p> <p>-Obtener los permisos y autorizaciones necesarias para el traslado, incluyendo certificados sanitarios y documentos de propiedad.</p> <p>-Planificar la logística del traslado, incluyendo la ruta, los medios de transporte adecuados y la preparación de las instalaciones de destino.</p> <p><b>HACER:</b></p> <p>-Preparar a los bovinos para el traslado, asegurando su estado de salud y bienestar durante el proceso.</p> <p>-Cargar los bovinos en los medios de transporte designados de manera segura y conforme a las regulaciones.</p> <p>-Supervisar el traslado de los bovinos, garantizando su integridad física y minimizando el estrés durante el viaje</p> <p><b>VERIFICAR:</b></p> <p>-Verificar que todos los bovinos lleguen a su destino de manera segura y sin incidentes.</p> <p>-Confirmar que se cumplan todos los requisitos legales y sanitarios establecidos para el traslado.</p> <p>-Realizar inspecciones de salud a la llegada para asegurar el bienestar de los bovinos y prevenir la propagación de enfermedades.</p> <p><b>ACTUAR:</b></p> <p>-Registrar y documentar cualquier cambio en la condición de los bovinos durante el traslado.</p>	<p>Base de datos de bovinos actualizada</p> <p>Registro actualizado de traslados de bovinos.</p> <p>Documentación de cumplimiento de requisitos legales y sanitarios.</p> <p>Informes de seguimiento del estado de los bovinos trasladados.</p>	<p>Administrador</p> <p>Jornalero</p> <p>Dueños del ganado</p> <p>Veterinarios</p>

**Figura 50.** P007 Caracterización de proceso de registro de traslados de bovinos.

P008		Caracterización de Inseminaciones de Bovinos		
<b>Proceso:</b> Registro de Inseminaciones de Bovinos				
<b>Responsable:</b> Hector Portilla				
<b>Objetivo:</b> Realizar de manera efectiva y segura el proceso de inseminación artificial o natural en bovinos para mejorar la genética del ganado y aumentar la productividad del ganado.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Veterinario Jornalero	<p>Programación de las inseminaciones, que puede incluir datos como la fecha de inicio del celo de las vacas y la disponibilidad de los toros donantes de semen.</p> <p>Semen congelado de calidad, proveniente de toros seleccionados.</p> <p>Registro de la historia reproductiva de las vacas.</p> <p>Equipo y suministros necesarios para realizar el procedimiento de inseminación.</p>	<p><b>PLANEAR:</b></p> <ul style="list-style-type: none"> <li>-Evaluar la disponibilidad de recursos y la programación de las inseminaciones de acuerdo con el ciclo reproductivo de las vacas.</li> <li>-Preparar el equipo y el área de trabajo para la realización de las inseminaciones.</li> <li>-Coordinar con el veterinario para asegurar la disponibilidad de semen de calidad y determinar el momento óptimo para la inseminación.</li> </ul> <p><b>HACER:</b></p> <ul style="list-style-type: none"> <li>-Preparar el semen congelado y el equipo necesario para la inseminación.</li> <li>-Identificar y seleccionar las vacas elegibles para la inseminación de acuerdo con su historial reproductivo y estado de salud.</li> <li>-Realizar el procedimiento de inseminación artificial de manera cuidadosa y precisa, siguiendo los protocolos establecidos.</li> </ul> <p><b>VERIFICAR:</b></p> <ul style="list-style-type: none"> <li>-Confirmar la correcta realización del procedimiento de inseminación, incluyendo la identificación de la vaca y la ubicación precisa de la inseminación.</li> <li>-Verificar que se utilice el semen adecuado de acuerdo con el plan de inseminación y las características genéticas deseadas.</li> <li>-Supervisar la recuperación y el bienestar de las vacas después del procedimiento.</li> </ul> <p><b>ACTUAR:</b></p> <ul style="list-style-type: none"> <li>-Registrar los detalles de cada inseminación realizada, incluyendo la fecha, la vaca inseminada y el toro donante de semen.</li> <li>-Monitorear el progreso y la respuesta de las vacas inseminadas, y tomar medidas adicionales según sea necesario.</li> </ul>	<p>Base de datos de bovinos actualizada</p> <p>Registro actualizado de inseminaciones de bovinos.</p> <p>Informes de seguimiento del estado reproductivo de las vacas inseminadas.</p>	<p>Administrador</p> <p>Jornalero</p> <p>Dueños del ganado</p> <p>Veterinarios</p>

**Figura 51.** P008 Caracterización de proceso de registro de inseminaciones de bovinos.

P009		Caracterización de Proceso de Chequeos de Bovinos		
<b>Proceso:</b> Registro de Chequeos de Bovinos				
<b>Responsable:</b> Fernando Mora				
<b>Objetivo:</b> Registrar y mantener un historial completo y actualizado de los chequeos de salud realizados a los bovinos, con el fin de monitorear su bienestar, prevenir enfermedades y mejorar la gestión de la salud del ganado.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Veterinario Jornalero	<p>Programación de las inseminaciones, que puede incluir datos como la fecha de inicio del celo de las vacas y la disponibilidad de los toros donantes de semen.</p> <p>Semen congelado de calidad, proveniente de toros seleccionados.</p> <p>Registro de la historia reproductiva de las vacas.</p> <p>Equipo y suministros necesarios para realizar el procedimiento de inseminación.</p>	<p><b>PLANEAR:</b></p> <ul style="list-style-type: none"> <li>-Establecer un calendario para los chequeos de salud, teniendo en cuenta la edad, el estado reproductivo y otros factores relevantes de los bovinos.</li> <li>-Coordinar con el veterinario para determinar los protocolos de examen y los procedimientos a seguir durante los chequeos.</li> <li>-Preparar el equipo y los suministros necesarios para llevar a cabo los chequeos de manera efectiva y segura.</li> </ul> <p><b>HACER:</b></p> <ul style="list-style-type: none"> <li>-Realizar los chequeos de salud de acuerdo con los protocolos establecidos, examinando a cada bovino de manera individual y registrando cualquier hallazgo relevante.</li> <li>-Administrar tratamientos o medicamentos según sea necesario, bajo la supervisión del veterinario.</li> <li>-Proporcionar retroalimentación y orientación al personal de manejo del ganado sobre prácticas de cuidado y prevención de enfermedades.</li> </ul> <p><b>VERIFICAR:</b></p> <ul style="list-style-type: none"> <li>-Revisar y validar la integridad y precisión de la información registrada durante los chequeos de salud.</li> <li>-Confirmar que se hayan seguido los protocolos y procedimientos establecidos durante el proceso de examen.</li> <li>-Monitorear la respuesta de los bovinos a los tratamientos administrados y realizar ajustes si es necesario.</li> </ul> <p><b>ACTUAR:</b></p> <ul style="list-style-type: none"> <li>-Actualizar el historial de salud de cada bovino con los resultados de los chequeos y cualquier intervención médica realizada.</li> <li>-Identificar y abordar cualquier problema de salud emergente o tendencia preocupante en el rebaño.</li> <li>-Analizar los datos recopilados durante los chequeos para identificar áreas de mejora en la gestión de la salud del ganado y tomar medidas correctivas según sea necesario.</li> </ul>	<p>Base de datos de bovinos actualizada</p> <p>Historial completo y actualizado de los chequeos de salud de los bovinos.</p> <p>Registro de tratamientos administrados y medicamentos utilizados durante los chequeos.</p> <p>Informes de seguimiento del estado de salud del rebaño y las tendencias observadas.</p> <p>Recomendaciones para mejorar la gestión de la salud del ganado y prevenir enfermedades.</p>	<p>Administrador</p> <p>Jornalero</p> <p>Dueños del ganado</p> <p>Veterinarios</p>

**Figura 52.** P009 Caracterización de proceso de registro de chequeos de bovinos.

P010		Caracterización de Proceso de Partos de Bovinos		
<b>Proceso:</b> Registro de Partos de Bovinos				
<b>Responsable:</b> Fernando Mora				
<b>Objetivo:</b> Registrar de manera precisa y oportuna los eventos de parto de los bovinos para mantener un historial completo de reproducción, monitorear la salud reproductiva del rebaño y facilitar la gestión del ganado.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Veterinario Jornalero	<p>Programación de los eventos de parto, incluyendo la fecha prevista para los partos y la identificación de las vacas gestantes.</p> <p>Equipamiento médico y suministros necesarios para asistir en el parto si es necesario.</p> <p>Registro del historial reproductivo de las vacas, incluyendo la fecha del último celo y los detalles de cualquier inseminación previa.</p> <p>Información del jornalero encargado de asistir durante el parto.</p>	<p><b>PLANEAR:</b></p> <ul style="list-style-type: none"> <li>-Establecer un calendario para los eventos de parto, considerando la duración de la gestación y otros factores que puedan afectar el momento del parto.</li> <li>-Coordinar con el veterinario para determinar los protocolos de asistencia en el parto y los cuidados posteriores necesarios para la madre y el ternero.</li> <li>-Preparar el equipo y los suministros necesarios para asistir en el parto y garantizar un ambiente seguro y limpio para el proceso.</li> </ul> <p><b>HACER:</b></p> <ul style="list-style-type: none"> <li>-Monitorear de cerca a las vacas gestantes, observando signos de trabajo de parto y preparándose para intervenir si es necesario.</li> <li>-Asistir en el parto según sea necesario, proporcionando apoyo a la madre y al ternero y asegurando un nacimiento seguro y sin complicaciones.</li> </ul> <p><b>VERIFICAR:</b></p> <ul style="list-style-type: none"> <li>-Registrar la fecha, hora y detalles del parto, incluyendo la salud y el peso del ternero, así como cualquier complicación durante el proceso.</li> <li>-Revisar y validar la integridad y precisión de la información registrada sobre los eventos de parto.</li> <li>-Confirmar que se hayan seguido los protocolos y procedimientos establecidos durante el proceso de asistencia en el parto.</li> <li>-Monitorear la recuperación de la madre y el desarrollo del ternero durante las primeras horas y días después del parto.</li> </ul> <p><b>ACTUAR:</b></p> <ul style="list-style-type: none"> <li>-Actualizar el historial reproductivo de cada vaca con los detalles del parto, incluyendo el número de cría y cualquier observación relevante.</li> <li>-Identificar y abordar cualquier problema de salud o comportamiento emergente en la madre o el ternero.</li> </ul>	<p>Base de datos de bovinos actualizada</p> <p>Registro completo y actualizado de los eventos de parto de los bovinos.</p> <p>Informes de seguimiento del estado de salud y el desarrollo de los terneros.</p>	<p>Administrador</p> <p>Jornalero</p> <p>Dueños del ganado</p> <p>Veterinarios</p>

**Figura 53.** P010 Caracterización de proceso de registro de partos de bovinos.

P011		Caracterización de Proceso de Abortos de Bovinos		
<b>Proceso:</b> Registro de Abortos de Bovinos				
<b>Responsable:</b> Fernando Mora				
<b>Objetivo:</b> Registrar de manera precisa y oportuna los casos de aborto en bovinos para monitorear la salud reproductiva del rebaño, identificar posibles problemas de manejo o enfermedades y tomar medidas preventivas para garantizar el bienestar del ganado.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Veterinario Jornalero	Reportes de casos de aborto, que pueden incluir la fecha y hora del aborto, la identificación de la vaca afectada y cualquier síntoma o signo relevante.  Historial de salud reproductiva de las vacas, incluyendo cualquier antecedente de abortos previos o condiciones médicas que puedan afectar la gestación.  Información sobre la alimentación, manejo y condiciones ambientales del ganado durante el período de gestación.  Registro del personal de manejo del ganado que observó o reportó el aborto.	<b>PLANEAR:</b> -Establecer protocolos y procedimientos para la detección, registro y manejo de casos de aborto en el ganado. -Capacitar al personal de manejo del ganado en la identificación de signos de aborto y en el reporte adecuado de casos sospechosos. -Coordinar con el veterinario para establecer criterios de diagnóstico y tratamiento de casos de aborto, así como para implementar medidas preventivas. <b>HACER:</b> -Monitorear de cerca el ganado para detectar signos de aborto, como descargas vaginales anormales, pérdida de apetito o comportamiento anormal. -Investigar y documentar cualquier caso sospechoso de aborto, recopilando información sobre la vaca afectada, condiciones de manejo y cualquier otro factor relevante. <b>VERIFICAR:</b> -Revisar y validar la información registrada sobre casos de aborto para garantizar su integridad y precisión. -Analizar patrones y tendencias en los casos de aborto para identificar posibles factores de riesgo o problemas subyacentes en el manejo del ganado. <b>ACTUAR:</b> -Actualizar el registro de salud reproductiva del ganado con detalles de cada caso de aborto, incluyendo la fecha, la vaca afectada y cualquier diagnóstico o tratamiento realizado. -Implementar medidas correctivas para abordar cualquier problema identificado en el manejo del ganado o en las condiciones ambientales que puedan contribuir a los abortos.	Base de datos de bovinos actualizada  Registro completo y actualizado de casos de aborto en bovinos.  Informes de seguimiento sobre patrones y tendencias en los abortos, así como la efectividad de las medidas preventivas.  Recomendaciones para mejorar la salud reproductiva del ganado y prevenir futuros abortos.	Administrador  Jornalero Dueños del ganado  Veterinarios

Figura 54. P011 Caracterización de proceso de registro de abortos de bovinos.

P012		Caracterización de Proceso de Registro de Productos		
<b>Proceso:</b> Registro de Registro de Productos				
<b>Responsable:</b> Fernando Mora				
<b>Objetivo:</b> Mantener un registro completo y actualizado de los productos utilizados en la operación ganadera, incluyendo insumos veterinarios, alimentación bovina e insumos agrícolas, así como el control del stock de cada producto, para garantizar una gestión eficiente de inventario, un adecuado suministro de productos y la trazabilidad de los mismos en la producción ganadera.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	Facturas y recibos de compra de productos, que incluyen insumos veterinarios, alimentos para el ganado e insumos agrícolas, con detalles como nombre del producto, cantidad adquirida, fecha de compra y proveedor.  Información sobre el uso previsto de los productos, como tratamientos veterinarios, alimentación del ganado o aplicación en actividades agrícolas.  Registro del personal involucrado en la recepción y manejo de los productos. Información sobre el stock inicial de cada producto en el inventario. Registro del personal de manejo del ganado que observó o reportó el aborto.nte el parto.	<b>PLANEAR:</b> -Establecer protocolos y procedimientos para el registro de productos y el control de stock, definiendo cómo se ingresarán los datos y cómo se realizará el seguimiento del inventario. -Definir criterios para la clasificación y etiquetado de los productos, facilitando su identificación y control. <b>HACER:</b> - Registrar los productos recibidos en el inventario, ingresando información detallada sobre cada producto, incluyendo nombre, cantidad, fecha de recepción y proveedor. -Almacenar los productos de manera adecuada, asegurando condiciones de almacenamiento apropiadas para preservar su calidad y eficacia. -Actualizar el stock de cada producto en el inventario después de cada entrada o salida de productos. <b>VERIFICAR:</b> -Revisar y validar la información registrada sobre los productos y el stock para garantizar su integridad y precisión. -Realizar controles periódicos de inventario para verificar la exactitud de los registros y detectar posibles discrepancias o pérdidas de productos. -Auditar el cumplimiento de los procedimientos de registro de productos y control de stock y tomar medidas correctivas si es necesario. <b>ACTUAR:</b> -Actualizar continuamente el registro de productos y el control de stock con información sobre nuevas adquisiciones, uso de productos y cualquier cambio en el inventario. -Analizar los datos de registro de productos y control de stock para identificar oportunidades de optimización en la gestión de inventario y en la selección de proveedores.	Registro completo y actualizado de productos utilizados en la operación ganadera, incluyendo detalles de cada producto y su stock actual.  Informes de inventario que proporcionan una visión general del estado actual del stock de productos y su disponibilidad.	Administrador Jornalero Dueños del ganado Personal de gestión de inventario

**Figura 55.** P012 Caracterización de proceso de registro de productos.

P013		Caracterización de Proceso de Pedidos e Ingreso de Productos		
<b>Proceso:</b> Registro de Pedidos e Ingreso de Productos				
<b>Responsable:</b> Fernando Mora				
<b>Objetivo:</b> Gestionar de manera eficiente el registro de pedidos y el ingreso de productos en el inventario de la operación ganadera, incluyendo insumos veterinarios, alimentación bovina e insumos agrícolas, así como mantener un control actualizado del stock de cada producto para asegurar un suministro adecuado y oportuno de insumos para la producción ganadera.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	Órdenes de compra de productos, incluyendo detalles como nombre del producto, cantidad solicitada, fecha de pedido y proveedor.  Información sobre el stock actual de cada producto en el inventario.  Registro del personal involucrado en la recepción y manejo de los productos.  Facturas y recibos de entrega de productos, que contienen información detallada sobre los productos recibidos, como nombre, cantidad, fecha de entrega y proveedor.	<b>PLANEAR:</b> -Establecer procedimientos para la generación y seguimiento de órdenes de compra de productos, definiendo cómo se generarán y comunicarán los pedidos a los proveedores. -Definir criterios para la recepción y registro de productos, asegurando que se cumplan los requisitos de calidad y cantidad especificados en las órdenes de compra. <b>HACER:</b> - Generar órdenes de compra de productos de acuerdo con las necesidades del ganado y las especificaciones establecidas. -Comunicar las órdenes de compra a los proveedores y hacer seguimiento para asegurar la entrega oportuna de los productos. -Recibir y verificar los productos entregados, comparando la información de las facturas con las órdenes de compra y registrando cualquier discrepancia o problema detectado. <b>VERIFICAR:</b> -Revisar y validar la información registrada sobre los pedidos y el ingreso de productos para garantizar su integridad y precisión. -Realizar controles periódicos de stock para verificar la exactitud de los registros y detectar posibles discrepancias o pérdidas de productos. <b>ACTUAR:</b> - Actualizar continuamente el registro de pedidos e ingreso de productos con información sobre nuevas órdenes de compra, entrega de productos y cualquier cambio en el inventario. -Analizar los datos de registro de pedidos e ingreso de productos para identificar oportunidades de optimización en la gestión de inventario y en la selección de proveedores.	Registro completo y actualizado de pedidos e ingreso de productos en el inventario, incluyendo detalles de cada orden de compra, entrega de productos y stock actual.  Informes de inventario que proporcionan una visión general del estado actual del stock de productos y su disponibilidad.	Administrador  Jornalero  Dueños del ganado  Personal de gestión de inventario

**Figura 56.** P013 Caracterización de proceso de registro de pedidos e ingresos de productos.

P014		Caracterización de Proceso de Registro de Mantenimientos		
<b>Proceso:</b> Registro de Mantenimientos				
<b>Responsable:</b> Fernando Mora				
<b>Objetivo:</b> Registrar y gestionar de manera eficiente los mantenimientos realizados en los equipos e instalaciones de ordeño, con el fin de garantizar su adecuado funcionamiento, prolongar su vida útil y asegurar la calidad e higiene en el proceso de ordeño.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Personal técnico Jornalero	<p>Establecer un plan de mantenimiento preventivo para los equipos e instalaciones de ordeño, definiendo la periodicidad y los procedimientos a seguir.</p> <p>Identificar y programar los mantenimientos necesarios en función del estado y la prioridad de los equipos e instalaciones.</p>	<p><b>PLANEAR:</b></p> <ul style="list-style-type: none"> <li>-Establecer un plan de mantenimiento preventivo para los equipos e instalaciones de ordeño, definiendo la periodicidad y los procedimientos a seguir.</li> <li>-Identificar y programar los mantenimientos necesarios en función del estado y la prioridad de los equipos e instalaciones.</li> </ul> <p><b>HACER:</b></p> <ul style="list-style-type: none"> <li>- Realizar los mantenimientos preventivos según lo programado, siguiendo los procedimientos y estándares establecidos por el fabricante o recomendados por expertos en el área.</li> <li>-Inspeccionar los equipos e instalaciones de ordeño en busca de desgastes, fugas, o cualquier otro problema que requiera atención.</li> <li>-Realizar reparaciones menores o ajustes necesarios para asegurar el funcionamiento óptimo de los equipos e instalaciones.</li> </ul> <p><b>VERIFICAR:</b></p> <ul style="list-style-type: none"> <li>-Revisar y validar la ejecución de los mantenimientos, asegurando que se hayan completado de manera adecuada y satisfactoria.</li> <li>-Verificar el correcto funcionamiento de los equipos e instalaciones después del mantenimiento, realizando pruebas de rendimiento si es necesario.</li> <li>-Documentar cualquier problema encontrado durante el mantenimiento y las acciones tomadas para resolverlo.</li> </ul> <p><b>ACTUAR:</b></p> <ul style="list-style-type: none"> <li>-Actualizar el registro de mantenimientos con detalles de las actividades realizadas, incluyendo fechas, equipos atendidos y trabajos realizados.</li> <li>-Analizar los datos de mantenimiento para identificar tendencias y áreas de mejora en la gestión de mantenimiento preventivo.</li> </ul>	<p>Registro completo y actualizado de los mantenimientos realizados en los equipos e instalaciones de ordeño.</p> <p>Informes de mantenimiento que proporcionan una visión general del estado de los equipos e instalaciones y las actividades realizadas.</p>	Administrador Jornalero Dueños del ganado

**Figura 57.** P014 Caracterización de proceso de registro de mantenimientos.

P015		Caracterización de Proceso de Registro de Producción Lechera		
<b>Proceso:</b> Registro de Producción Lechera				
<b>Responsable:</b> Fernando Mora				
<b>Objetivo:</b> Registrar de manera precisa y oportuna la producción lechera del ganado de ordeño, recopilando información de reportes mensuales y semanales, con el fin de monitorear el rendimiento del ganado, identificar tendencias de producción y tomar decisiones informadas para mejorar la eficiencia y calidad de la producción lechera.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero	Reportes mensuales y semanales de producción lechera, que incluyen datos como la cantidad de leche producida por vaca, la calidad de la leche y cualquier incidencia relevante.  Formulacion de Recoleccion de Infomacion de Produccion Lechera  Registro del personal involucrado en la recolección y reporte de datos de producción lechera.	<b>PLANEAR:</b> -Establecer un plan de mantenimiento preventivo para los equipos e instalaciones de ordeño, definiendo la periodicidad y los procedimientos a seguir. -Identificar y programar los mantenimientos necesarios en función del estado y la prioridad de los equipos e instalaciones.  <b>HACER:</b> -Realizar los mantenimientos preventivos según lo programado, siguiendo los procedimientos y estándares establecidos por el fabricante o recomendados por expertos en el área. -Inspeccionar los equipos e instalaciones de ordeño en busca de desgastes, fugas, o cualquier otro problema que requiera atención. -Realizar reparaciones menores o ajustes necesarios para asegurar el funcionamiento óptimo de los equipos e instalaciones.  <b>VERIFICAR:</b> -Revisar y validar la ejecución de los mantenimientos, asegurando que se hayan completado de manera adecuada y satisfactoria. -Verificar el correcto funcionamiento de los equipos e instalaciones después del mantenimiento, realizando pruebas de rendimiento si es necesario. -Documentar cualquier problema encontrado durante el mantenimiento y las acciones tomadas para resolverlo.  <b>ACTUAR:</b> -Actualizar el registro de mantenimientos con detalles de las actividades realizadas, incluyendo fechas, equipos atendidos y trabajos realizados. -Analizar los datos de mantenimiento para identificar tendencias y áreas de mejora en la gestión de mantenimiento preventivo.	Registro completo y actualizado de la producción lechera del ganado de ordeño.  Informes de producción que proporcionan una visión general del rendimiento del ganado y las tendencias de producción a lo largo del tiempo.	Administrador Jornalero Dueños del ganado

**Figura 58.** P015 Caracterización de proceso de registro de producción lechera.

P016	Caracterización de Proceso de Registro de Despacho de Leche			
Proceso: Registro de Despacho de Leche				
Responsable: Fernando Mora				
Objetivo: Registrar de manera precisa y completa el despacho de leche, incluyendo información detallada con el fin de mantener un control efectivo de la distribución de la leche, garantizar la entrega oportuna y cumplir con los compromisos de los clientes.				
Actores	Entradas	Proceso	Salidas	Receptores
Administrador Jornalero Transportista	Órdenes de despacho de leche, que pueden incluir detalles como la cantidad requerida, la dirección de entrega y la fecha y hora programadas.  Registro del personal encargado del despacho y la entrega de la leche.  Información sobre el stock de leche disponible para despacho.	<b>PLANEAR:</b> -Establecer procedimientos para la gestión y registro de despachos de leche, definiendo la forma en que se generarán los órdenes de despacho y cómo se realizará el seguimiento de las entregas. -Programar los despachos de acuerdo con las órdenes recibidas y la disponibilidad de leche en stock. <b>HACER:</b> -Preparar la leche para su despacho, asegurando que esté correctamente almacenada y envasada según los requisitos de calidad y seguridad. -Coordinar con el personal de transporte para asegurar la disponibilidad de vehículos y choferes para realizar las entregas programadas. -Realizar los despachos de leche según lo programado, registrando información detallada sobre la fecha, hora, cantidad despachada, chofer a cargo y responsable de la entrega. <b>VERIFICAR:</b> -Revisar y validar la información registrada sobre los despachos de leche para garantizar su integridad y precisión. -Comparar los datos de despacho con las órdenes recibidas para verificar que se cumplan los requerimientos de los clientes.  <b>ACTUAR:</b> -Actualizar el registro de despachos de leche con los datos recopilados, incluyendo detalles de cada entrega realizada. -Analizar los datos de despacho para identificar tendencias y oportunidades de mejora en la gestión de despachos y la satisfacción del cliente.	Registro completo y actualizado de los despachos de leche realizados, incluyendo detalles de cada entrega.  Informes de despacho que proporcionan una visión general del volumen de leche despachada, las entregas realizadas y cualquier incidencia registrada.	Administrador Jornalero Dueños del ganado Transportista

Figura 59. P016 Caracterización de proceso de registro de despacho de leche.

### e) Requerimientos funcionales- Historias de usuarios

A través de los métodos de investigación aplicados, la entrevista estructurada realizada al administrador y los cuestionarios realizados a los jornaleros de la Hacienda ProduAgro, se recopiló información crítica que reveló las necesidades específicas en la administración y gestión de la información bovina de la hacienda. Esta valiosa información, basada en experiencias reales sobre la administración de los recursos de la hacienda, fundamenta la estructura del sistema informático propuesto. Para articular y sistematizar adecuadamente los requisitos del sistema, se adoptó el enfoque de historias de usuario, un método eficaz que facilita la definición clara y comprensible de las funcionalidades desde la perspectiva de los usuarios finales. Estas historias se presentan en un lenguaje simple y accesible, permitiendo una interpretación y entendimiento universal de las necesidades y expectativas hacia el sistema informático para la Hacienda ProduAgro.

**Tabla 27.** Gestión de Usuarios.

<b>REQUERIMIENTO 001</b>	
<b>Numero:</b> 001	<b>Usuario:</b> Empleados
<b>Nombre:</b> Acceso al sistema de administración	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Estimación:</b> 1	<b>Iteración:</b> 1
<b>Responsable:</b> Héctor Portilla	
<b>Descripción:</b> Los empleados tendrán facilidad de acceso al sistema a través de su número de cedula y contraseña.	
<b>Detalles:</b> La ventana tendrá una formulación de acceso al sistema donde se ingresará sus credenciales de usuario. El formulario debe tener las siguientes características:	
<ul style="list-style-type: none"> <li>- Dos campos de texto donde se ingresará el número de cedula y la contraseña.</li> <li>- Un botón que muestre "Ingresar"</li> <li>- En caso de dar credenciales invalidas mostrar un mensaje de alerta.</li> </ul>	

**Tabla 28.** Control de usuario.

<b>REQUERIMIENTO 002</b>	
<b>Numero:</b> 002	<b>Usuario:</b> Administrador
<b>Nombre:</b> Control de usuario	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Estimación:</b> 1	<b>Iteración:</b> 1
<b>Responsable:</b> Héctor Portilla	
<b>Descripción:</b> El empleador debe poder observar y realizar acciones sobre los usuarios.	
<b>Detalles:</b> El formulario debe de contener las siguientes características.	
<ul style="list-style-type: none"> <li>- Mostrar una tabla que contenga los registros de los usuarios, nombre, cedula y cargo que desempeña.</li> <li>- Un botón para crear, eliminar y modificar usuarios</li> <li>- Debe de contener dos botones para confirmar el guardado y otro para cancelar la acción.</li> </ul>	

## Módulo 2.- Terneras y Vaconas

**Tabla 29.** Módulo de terneras y vaconas.

<b>REQUERIMIENTO 003</b>	
<b>Numero:</b> 003	<b>Usuario:</b> Empleado / Administrado
<b>Nombre:</b> Administración de terneras	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Alto
<b>Estimación:</b> 3	<b>Iteración:</b> 1
<b>Responsable:</b> Héctor Portilla	
<b>Descripción:</b> El empleador debe ingresar al sistema y registrar los datos referidos al animal durante su proceso de desarrollo.	
<b>Detalles:</b> El formulario debe de contener las siguientes características.	
<ul style="list-style-type: none"> <li>- Campos de texto para ingresar el número de la ternera y el número de la madre, fecha de nacimiento, raza, sexo y peso.</li> <li>- Un botón para guardar, cancelar, modificar registros y eliminar registros.</li> <li>- Visualizar el número de terneras que pertenecen a un grupo, mostrando el número del animal, peso, fecha de nacimiento y número de la madre.</li> </ul>	

**Tabla 30.** Historial veterinario.

<b>REQUERIMIENTO 004</b>	
<b>Numero:</b> 004	<b>Usuario:</b> Empleado / Administrado
<b>Nombre:</b> Historial veterinario	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Estimación:</b> 2	<b>Iteración:</b> 1

**Responsable:** Héctor Portilla

**Descripción:** El empleador debe ingresar al sistema y registrar cuando una ternera se encuentra enferma y necesite medicación, de igual forma los medicamentos que se le administraron.

**Detalles:** El formulario debe de contener las siguientes características.

- Campos de texto para ingresar las dolencias, la fecha que se identificó la enfermedad, la fecha en que se aplicó los medicamentos y los medicamentos administrados.
- Un botón para guardar, cancelar, modificar registros y eliminar registros.
- Visualizar el historial veterinario de la ternera, con los registros de cada enfermedad o dolencia que el animal haya tenido.

### Módulo 3.- Ganado de Producción lechera.

**Tabla 31.** Módulo Ganado de Producción lechera.

REQUERIMIENTO 005	
<b>Numero:</b> 005	<b>Usuario:</b> Empleado / Administrador
<b>Nombre:</b> Ingreso de animal a la producción lechera	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Estimación:</b> 3	<b>Iteración:</b> 1
<b>Responsable:</b> Héctor Portilla	
<b>Descripción:</b> El empleador debe ingresar los registros del animal cuando este esté apto para la producción lechera.	
<b>Detalles:</b> El formulario debe de contener las siguientes características.	
<ul style="list-style-type: none"><li>- Campos de texto para ingresar el número del animal, su genealogía y su estado ya sea de preñez, vacía, seco o descarte.</li><li>- Un botón para guardar, cancelar, modificar registros y eliminar registros.</li><li>- Visualizar los animales que están registrados para la producción lechera</li></ul>	

**Tabla 32.** Inseminación.

REQUERIMIENTO 006	
<b>Numero:</b> 006	<b>Usuario:</b> Empleado / Administrado
<b>Nombre:</b> Inseminación	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Estimación:</b> 1	<b>Iteración:</b> 1
<b>Responsable:</b> Héctor Portilla	
<b>Descripción:</b> El empleador debe ingresar al sistema los datos relacionados al proceso de inseminación del ganado vacuno.	
<b>Detalles:</b> El formulario debe de contener las siguientes características.	
<ul style="list-style-type: none"><li>- Campos de texto para ingresar el código de pajuela, el inseminado, la fecha.</li><li>- Un botón para guardar, cancelar, modificar registros y eliminar registros.</li><li>- Visualizar los registros del ganado inseminado con todos sus datos y el cálculo previsto de preñez.</li></ul>	

### Módulo 4.- Historial de mantenimiento de equipos agropecuarios.

**Tabla 33.** Módulo Historial de mantenimiento de equipos agropecuarios.

REQUERIMIENTO 007	
<b>Numero:</b> 007	<b>Usuario:</b> Empleado / Administrado
<b>Nombre:</b> Mantenimiento de equipos	
<b>Prioridad:</b> Baja	<b>Riesgo:</b> Medio
<b>Estimación:</b> 2	<b>Iteración:</b> 1
<b>Responsable:</b> Héctor Portilla	

**Descripción:** El empleador debe ingresar los datos relacionados al mantenimiento de los equipos de ordeño.

**Detalles:** El formulario debe de contener las siguientes características.

- Campos de texto para ingresar la fecha del mantenimiento, una descripción y quien superviso los cambios realizados a los equipos
- Un botón para guardar, cancelar, modificar registros y eliminar registros.
- Visualizar el historial sobre el mantenimiento del ordeño.

## Módulo 5.- Inventario veterinario.

Tabla 34. Módulo Inventario veterinario.

REQUERIMIENTO 008	
<b>Numero:</b> 008	<b>Usuario:</b> Empleado / Administrado
<b>Nombre:</b> Inventario veterinario	
<b>Prioridad:</b> Media	<b>Riesgo:</b> Medio
<b>Estimación:</b> 2	<b>Iteración:</b> 1
<b>Responsable:</b> Fernando Mora	
<b>Descripción:</b> El empleador debe ingresar los registros de los medicamentos e insumos veterinarios que entren o salgan de la hacienda.	
<b>Detalles:</b> El formulario debe de contener las siguientes características.	
<ul style="list-style-type: none"><li>- Campos de texto para ingresar el producto, descripción, fecha de ingreso, fecha de salida y cantidad.</li><li>- Un botón para guardar, cancelar, modificar registros y eliminar registros.</li><li>- Se debe mostrar una tabla que contenga todos los productos existentes en la hacienda.</li></ul>	

## Módulo 6.- Potreros y abonos.

Tabla 35. Módulo Potreros y abonos.

REQUERIMIENTO 009	
<b>Numero:</b> 009	<b>Usuario:</b> Empleado / Administrado
<b>Nombre:</b> Potreros	
<b>Prioridad:</b> Media	<b>Riesgo:</b> Medio
<b>Estimación:</b> 1	<b>Iteración:</b> 1
<b>Responsable:</b> Fernando Mora	
<b>Descripción:</b> El empleador debe registrar los procesos llevados en los potreros.	
<b>Detalles:</b> El formulario debe de contener las siguientes características.	
<ul style="list-style-type: none"><li>- Campos de texto para ingresar, la fecha, descripción, encargado, numero de potrero.</li><li>- Un botón para guardar, cancelar, modificar registros y eliminar registros.</li><li>- Se debe mostrar una tabla que contenga todos los procedimientos llevados en los potreros.</li></ul>	

## Módulo 7.- Producción lechera.

Tabla 36. Módulo Producción lechera.

REQUERIMIENTO 010	
<b>Numero:</b> 010	<b>Usuario:</b> Empleado / Administrado
<b>Nombre:</b> Producción lechera	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Estimación:</b> 2	<b>Iteración:</b> 1
<b>Responsable:</b> Fernando Mora	

**Descripción:** El empleador debe registrar la salida de leche, cada vez que el tanquero retire la leche de la hacienda

**Detalles:** El formulario debe de contener las siguientes características.

- Campos de texto para ingresar, la fecha, cantidad de leche.
- Un botón para guardar, cancelar, modificar registros y eliminar registros.
- Se debe mostrar una tabla que contenga los registros, semanales, mensuales y anuales de salida de leche.
- Mostrar cuadros estadísticos relacionados a cada periodo.

#### d) Tareas de ingeniería

**Tabla 37.** Tarea de usuario 1

Tarea de usuario	
<b>Número de la tarea:</b> 1	<b>Número de historia:</b> 1
<b>Nombre de la tarea:</b> Creación del modelo de datos de usuario	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 01/09/2022	<b>Fecha fin:</b> 02/09/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Codificar el modelo de datos de usuario en .Net y migración de hacia la base de datos SQL Server.	

**Tabla 38.** Tarea de usuario 2

Tarea de usuario	
<b>Número de la tarea:</b> 2	<b>Número de historia:</b> 1
<b>Nombre de la tarea:</b> Diseño de la ventana del acceso al sistema	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 05/09/2022	<b>Fecha fin:</b> 07/09/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Diseñar la interfaz del acceso al sistema utilizando Bootstrap 5.1, HTML 5 y CSS.	

**Tabla 39.** Tarea de usuario 3

Tarea de usuario	
<b>Número de la tarea:</b> 3	<b>Número de historia:</b> 1
<b>Nombre de la tarea:</b> Validación de credenciales de inicio de sesión	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 08/09/2022	<b>Fecha fin:</b> 09/09/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Codificar la vista en .Net y realizar la validación de que el usuario existe dentro de la base de datos y que sus credenciales de acceso sean las correctas.	

**Tabla 40.** Tarea de usuario 4

Tarea de usuario	
<b>Número de la tarea:</b> 4	<b>Número de historia:</b> 1
<b>Nombre de la tarea:</b> Codificar de la vista de control de usuarios	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 12/09/2022	<b>Fecha fin:</b> 13/09/2022
<b>Programador responsable:</b> Héctor Portilla	

---

**Descripción:** Programación de la vista en .Net y crear el método para enlistar, editar y deshabilitar usuarios.

---

**Tabla 41.** Tarea de usuario 5

---

Tarea de usuario	
<b>Número de la tarea:</b> 5	<b>Número de historia:</b> 1
<b>Nombre de la tarea:</b> Diseño de la interfaz de control de usuarios	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 14/09/2022	<b>Fecha fin:</b> 15/09/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Codificación de la interfaz utilizando Bootstrap 5.1, HTML5, JavaScript y estilos CSS.	

---

**Tabla 42.** Tarea de usuario 6

---

Tarea de usuario	
<b>Número de la tarea:</b> 6	<b>Número de historia:</b> 1
<b>Nombre de la tarea:</b> Diseño de panel principal	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 16/09/2022	<b>Fecha fin:</b> 19/09/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Codificación del panel de control mediante la utilización de Bootstrap 5.1, HTML5, JavaScript y estilos CSS.	

---

**Tabla 43.** Tarea de usuario 7

---

Tarea de usuario	
<b>Número de la tarea:</b> 7	<b>Número de historia:</b> 1
<b>Nombre de la tarea:</b> Crear modelos de datos usuarios y permisos.	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 20/09/2022	<b>Fecha fin:</b> 21/09/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Se debe sobrescribir los modelos correspondientes a usuarios, roles, y asignación de operaciones a roles en .Net y después realizar la migración a SQL Server.	

---

**Tabla 44.** Tarea de usuario 8

---

Tarea de usuario	
<b>Número de la tarea:</b> 8	<b>Número de historia:</b> 2
<b>Nombre de la tarea:</b> Codificación de las vistas de gestión ganadera-bovinos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 22/09/2022	<b>Fecha fin:</b> 23/09/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 45.** Tarea de usuario 9

---

Tarea de usuario	
<b>Número de la tarea:</b> 9	<b>Número de historia:</b> 2

---

---

**Nombre de la tarea:** Codificación de las vistas de gestión ganadera-medición lechera  
**Tipo de tarea:** Desarrollo  
**Fecha inicio:** 26/09/2022 **Fecha fin:** 27/09/2022  
**Programador responsable:** Héctor Portilla  
**Descripción:** Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.

---

**Tabla 46.** Tarea de usuario 10

---

**Tarea de usuario**

---

**Número de la tarea:** 10 **Número de historia:** 2

**Nombre de la tarea:** Codificación de las vistas de gestión ganadera-muertes  
**Tipo de tarea:** Desarrollo  
**Fecha inicio:** 28/09/2022 **Fecha fin:** 29/09/2022  
**Programador responsable:** Héctor Portilla  
**Descripción:** Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.

---

**Tabla 47.** Tarea de usuario 11

---

**Tarea de usuario**

---

**Número de la tarea:** 11 **Número de historia:** 2

**Nombre de la tarea:** Codificación de las vistas de gestión ganadera-descartes  
**Tipo de tarea:** Desarrollo  
**Fecha inicio:** 30/09/2022 **Fecha fin:** 31/09/2022  
**Programador responsable:** Héctor Portilla  
**Descripción:** Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.

---

**Tabla 48.** Tarea de usuario 12

---

**Tarea de usuario**

---

**Número de la tarea:** 12 **Número de historia:** 2

**Nombre de la tarea:** Codificación de las vistas de gestión ganadera-pesos  
**Tipo de tarea:** Desarrollo  
**Fecha inicio:** 01/10/2022 **Fecha fin:** 02/10/2022  
**Programador responsable:** Héctor Portilla  
**Descripción:** Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.

---

**Tabla 49.** Tarea de usuario 13

---

**Tarea de usuario**

---

**Número de la tarea:** 13 **Número de historia:** 2

**Nombre de la tarea:** Codificación de las vistas de gestión ganadera-Toros  
**Tipo de tarea:** Desarrollo  
**Fecha inicio:** 03/10/2022 **Fecha fin:** 04/10/2022  
**Programador responsable:** Héctor Portilla  
**Descripción:** Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.

---

**Tabla 50.** Tarea de usuario 14

---

Tarea de usuario	
<b>Número de la tarea:</b> 14	<b>Número de historia:</b> 2
<b>Nombre de la tarea:</b> Codificación de las vistas de gestión ganadera-Datos veterinarios	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 05/10/2022	<b>Fecha fin:</b> 10/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 51.** Tarea de usuario 15

---

Tarea de usuario	
<b>Número de la tarea:</b> 15	<b>Número de historia:</b> 2
<b>Nombre de la tarea:</b> Codificación de las vistas de gestión ganadera-traslados	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 11/10/2022	<b>Fecha fin:</b> 12/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 52.** Tarea de usuario 16

---

Tarea de usuario	
<b>Número de la tarea:</b> 16	<b>Número de historia:</b> 3
<b>Nombre de la tarea:</b> Codificación de las vistas de gestión reproductiva-inseminaciones	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 13/10/2022	<b>Fecha fin:</b> 14/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 53.** Tarea de usuario 17

---

Tarea de usuario	
<b>Número de la tarea:</b> 17	<b>Número de historia:</b> 3
<b>Nombre de la tarea:</b> Codificación de las vistas de gestión reproductiva-chequeos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 15/10/2022	<b>Fecha fin:</b> 16/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 54.** Tarea de usuario 18

---

Tarea de usuario	
<b>Número de la tarea:</b> 18	<b>Número de historia:</b> 3
<b>Nombre de la tarea:</b> Codificación de las vistas de gestión reproductiva-partos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 17/10/2022	<b>Fecha fin:</b> 18/10/2022
<b>Programador responsable:</b> Héctor Portilla	

---

---

**Descripción:** Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.

---

**Tabla 55.** Tarea de usuario 19

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 19	<b>Número de historia:</b> 3
<b>Nombre de la tarea:</b> Codificación de las vistas de gestión reproductiva-Abortos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 19/10/2022	<b>Fecha fin:</b> 20/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 56.** Tarea de usuario 20

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 20	<b>Número de historia:</b> 4
<b>Nombre de la tarea:</b> Codificación de las vistas de Inventario-productos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 21/10/2022	<b>Fecha fin:</b> 22/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 57.** Tarea de usuario 21

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 21	<b>Número de historia:</b> 4
<b>Nombre de la tarea:</b> Codificación de las vistas de Inventario-pedidos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 23/10/2022	<b>Fecha fin:</b> 24/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 58.** Tarea de usuario 22

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 22	<b>Número de historia:</b> 5
<b>Nombre de la tarea:</b> Codificación de las vistas de Gestión de campo	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 25/10/2022	<b>Fecha fin:</b> 30/10/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 59.** Tarea de usuario 23

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 23	<b>Número de historia:</b> 6
<b>Nombre de la tarea:</b> Codificación de las vistas de Mantenimientos	

---

---

<b>Tipo de tarea:</b> Desarrollo	<b>Fecha fin:</b> 05/11/2022
<b>Fecha inicio:</b> 01/11/2022	
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 60.** Tarea de usuario 24

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 24	<b>Número de historia:</b> 7
<b>Nombre de la tarea:</b> Codificación de las vistas de Producción lechera	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 05/11/2022	<b>Fecha fin:</b> 15/11/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 61.** Tarea de usuario 25

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 25	<b>Número de historia:</b> 8
<b>Nombre de la tarea:</b> Codificación de las vistas de Información general-parcelas	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 16/11/2022	<b>Fecha fin:</b> 20/11/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 62.** Tarea de usuario 26

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 26	<b>Número de historia:</b> 8
<b>Nombre de la tarea:</b> Codificación de las vistas de Información general-grupos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 21/11/2022	<b>Fecha fin:</b> 22/11/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 63.** Tarea de usuario 27

---

<b>Tarea de usuario</b>	
<b>Número de la tarea:</b> 27	<b>Número de historia:</b> 8
<b>Nombre de la tarea:</b> Codificación de las vistas de Información general-Trabajadores	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 23/11/2022	<b>Fecha fin:</b> 25/11/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

---

**Tabla 64.** Tarea de usuario 28

Tarea de usuario	
<b>Número de la tarea:</b> 28	<b>Número de historia:</b> 9
<b>Nombre de la tarea:</b> Codificación de las vistas de Reportes-Animales	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 26/11/2022	<b>Fecha fin:</b> 30/11/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

**Tabla 65.** Tarea de usuario 29

Tarea de usuario	
<b>Número de la tarea:</b> 29	<b>Número de historia:</b> 9
<b>Nombre de la tarea:</b> Codificación de las vistas de Reportes-Inseminaciones	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 01/12/2022	<b>Fecha fin:</b> 03/12/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

**Tabla 66.** Tarea de usuario 30

Tarea de usuario	
<b>Número de la tarea:</b> 30	<b>Número de historia:</b> 9
<b>Nombre de la tarea:</b> Codificación de las vistas de Reportes-cheques	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 04/12/2022	<b>Fecha fin:</b> 07/12/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

**Tabla 67.** Tarea de usuario 31

Tarea de usuario	
<b>Número de la tarea:</b> 31	<b>Número de historia:</b> 9
<b>Nombre de la tarea:</b> Codificación de las vistas de Reportes-Partos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 08/12/2022	<b>Fecha fin:</b> 10/12/2022
<b>Programador responsable:</b> Héctor Portilla	
<b>Descripción:</b> Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.	

**Tabla 68.** Tarea de usuario 32

Tarea de usuario	
<b>Número de la tarea:</b> 32	<b>Número de historia:</b> 9
<b>Nombre de la tarea:</b> Codificación de las vistas de Reportes-Abortos	
<b>Tipo de tarea:</b> Desarrollo	
<b>Fecha inicio:</b> 11/12/2022	<b>Fecha fin:</b> 15/12/2022
<b>Programador responsable:</b> Héctor Portilla	

**Descripción:** Programación de la vista que contiene las clases y métodos de usuarios, roles y permisos que permitan enviar/recibir los datos del modelo y luego mostrarlos en la vista.

### e) Valoración y estimación de tiempo de historias de usuario

El tiempo asignado a cada historia de usuario se calcula en función del tiempo real de desarrollo, alineado con el programa de trabajo del proyecto. La estimación se realiza mediante la técnica de días laborables, contabilizando 5 días por semana y 4 horas diarias por desarrollador. Siguiendo la metodología XP, ninguna historia de usuario debe extenderse por más de tres semanas. También se detalla el tiempo estimado para el desarrollo de las funcionalidades del sistema, expresado en semanas, días y horas.

**Tabla 69.** Valoración y estimación de tiempo de historias de usuario.

N°	Historia de Usuario	Tiempo estimado		
		Semanas	Días	Horas
1	Acceso al sistema	2,6	13	104
2	Control de usuarios	1,6	8	64
3	Administración Gestión bovina	2	10	80
4	Administración Gestión reproductiva	1,6	8	64
5	Gestión de inventario	2,4	12	96
6	Gestión de mantenimiento	2,4	12	96
7	Administración de producción lechera	2,4	12	96
8	Gestión de información general	2,4	12	96
9	Gestión de reportes	1,6	8	64
<b>Tiempo Total Estimado</b>		<b>19</b>	<b>95</b>	<b>760</b>

## Fase de diseño

### a) Tarjetas CRC

En la investigación, las tarjetas CRC se diseñan de acuerdo con cada historia de usuario, lo que facilita la identificación de una o varias clases principales interconectadas. Además, se definen las responsabilidades organizadas como tareas a llevar a cabo, en colaboración con los participantes que integran la clase primaria y contribuyen al desarrollo del software.

**Tabla 70.** Tarjeta CRC Acceso al sistema.

---

<b>Acceso al sistema</b>	
<b>Responsabilidades:</b> Diseñar el formulario para el ingreso de los empleados al sistema Diseñar el proceso de verificación de nombre de usuario y contraseña.	<b>Colaboradores:</b> Administrador
<b>Observaciones.</b> –Solo los usuarios que el administrador haya registrado podrán ingresar al sistema.	

---

**Tabla 71.** Tarjeta CRC Control de Usuario.

---

<b>Control de Usuario</b>	
<b>Responsabilidades:</b> Diseñar el formulario para el ingreso de usuarios. Diseñar proceso de guardar usuarios Diseñar el proceso para edita usuarios Diseñar el proceso para eliminar usuarios Diseño de la tabla con los registros de los usuarios ingresados.	<b>Colaboradores:</b> Administrador
<b>Observaciones.</b> –Solo el administrador podrá registrar, editar y eliminar usuarios.	

---

**Tabla 72.** Tarjeta CRC Administración de Terneras.

---

<b>Administración de Terneras</b>	
<b>Responsabilidades:</b> Diseño del formulario de ingreso de terneras. Diseño de secuencia de guardado de datos Diseño de secuencia para editar datos Diseño de secuencia para eliminar datos Diseño de tabla con los registros de todas las terneras registradas con sus características.	<b>Colaboradores:</b> Usuario Ternero
<b>Observaciones.</b> –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá registrar terneras.	

---

**Tabla 73.** Tarjeta CRC Historial Veterinario.

---

<b>Historial Veterinario</b>	
<b>Responsabilidades:</b> Diseño del formulario de ingreso de dolencias identificadas para cada animal. Diseño de secuencia de guardado de datos Diseño de secuencia para editar datos Diseño de secuencia para eliminar datos Diseño del historial veterinario de cada animal.	<b>Colaboradores:</b> Ternero Operario del ganado.
<b>Observaciones.</b> –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá manipular los datos correspondientes al historial veterinario de cada animal.	

---

**Tabla 74.** Tarjeta CRC Ingreso de animal a la producción lechera.

---

<b>Ingreso de animal a la producción lechera.</b>	
<b>Responsabilidades:</b> Diseño del formulario de ingreso del animal apto para la producción lechera. Diseño de secuencia de guardado de datos Diseño de secuencia para editar datos Diseño de secuencia para eliminar datos Visualizar todos los animales aptos para la producción lechera.	<b>Colaboradores:</b> Operario del ganado.
<b>Observaciones.</b> –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá ingresar animales para la producción lechera.	

---

**Tabla 75.** Tarjeta CRC Inseminación.

---

<b>Inseminación</b>	
<b>Responsabilidades:</b> Diseño del formulario de registro de inseminación. Diseño de secuencia de guardado de datos Diseño de secuencia para editar datos Diseño de secuencia para eliminar datos Visualizar todos los animales inseminados Calculo previsto de preñez.	<b>Colaboradores:</b> Operario del ganado.
<b>Observaciones.</b> –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá registrar inseminaciones y manipular los datos.	

---

**Tabla 76.** Mantenimiento de equipos pecuarios.

---

<b>Mantenimiento de equipos pecuarios</b>	
---	--

---

---

<b>Responsabilidades:</b>	
Diseño del formulario de registro de mantenimiento a los equipos pecuarios.	
Diseño de secuencia de guardado de datos	<b>Colaboradores:</b>
Diseño de secuencia para editar datos	Operario del ordeño.
Diseño de secuencia para eliminar datos	
Visualizar historial de los mantenimientos realizaos.	
<b>Observaciones.</b> –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá registrar el mantenimiento de los equipos pecuarios.	

---

**Tabla 77.** Tarjeta CRC Inventario veterinario.

---

<b>Inventario veterinario</b>	
<hr/>	
<b>Responsabilidades:</b>	
Diseño del formulario de registro de los productos veterinarios existentes	
Diseño de secuencia de guardado de datos	<b>Colaboradores:</b>
Diseño de secuencia para editar datos	Administrador
Diseño de secuencia para eliminar datos	Operario del ganado
Visualizar todos los productos existentes en la hacienda.	Ternero
<b>Observaciones.</b> –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá ingresar los productos existentes en la hacienda.	

---

**Tabla 78.** Tarjeta CRC Potreros.

---

<b>Potreros</b>	
<hr/>	
<b>Responsabilidades:</b>	
Diseño del formulario de registro de los procesos llevados en los potreros.	
Diseño de secuencia de guardado de datos	<b>Colaboradores:</b>
Diseño de secuencia para editar datos	Administrador
Diseño de secuencia para eliminar datos	Operario de los potreros
Visualizar una tabla con todos los potreros y los procesos realizados.	
<b>Observaciones.</b> –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá ingresar procesos de los potreros.	

---

**Tabla 79.** Tarjeta CRC Producción lechera.

---

<b>Producción Lechera</b>	
<hr/>	
<b>Responsabilidades:</b>	
Diseño del formulario de registro de salida de leche.	
Diseño de secuencia de guardado de datos	<b>Colaboradores:</b>
Diseño de secuencia para editar datos	Administrador
Diseño de secuencia para eliminar datos	
Visualizar una tabla con los registros de salida	
Visualizar gráficos estadísticos de la producción lechera.	

---

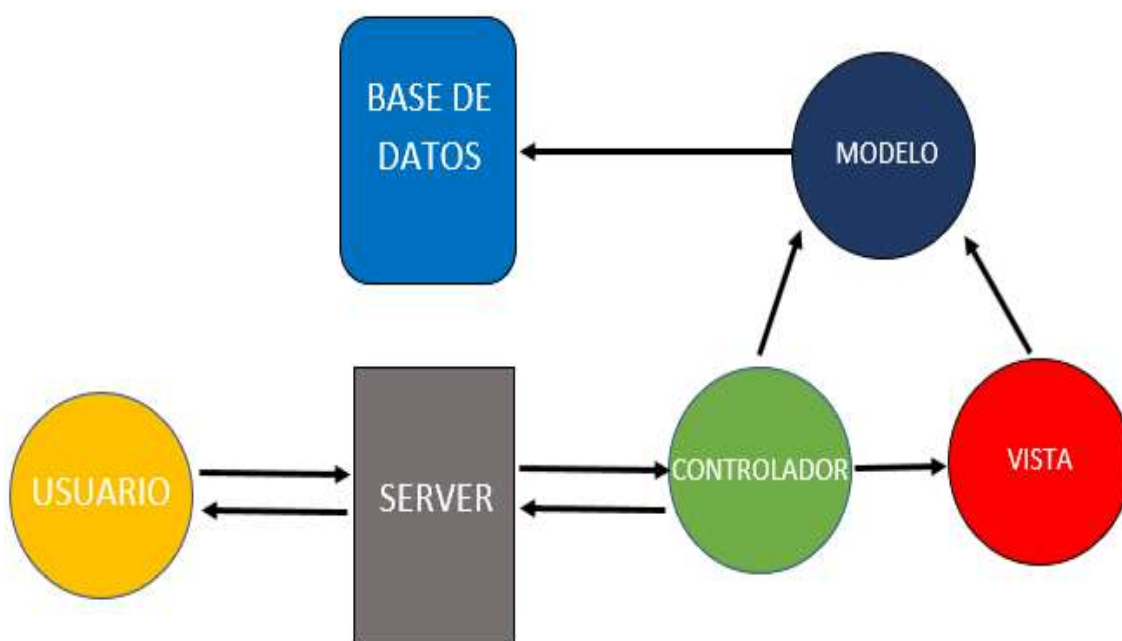
---

**Observaciones.** –Solo el personal autorizado y previamente registrado con los respectivos permisos podrá registrar la producción lechera.

---

### b) Arquitectura de la aplicación

El diseño del sistema a desarrollar se maneja el MVC que se refiere a un patrón dentro del diseño de software que usualmente es utilizado para la implementación de interfaces de usuario, datos y lógica de control. Enfatiza en la división entre la lógica de negocios y su representación visual. A continuación, se indica la arquitectura de la aplicación web con sus 3 componentes principales:

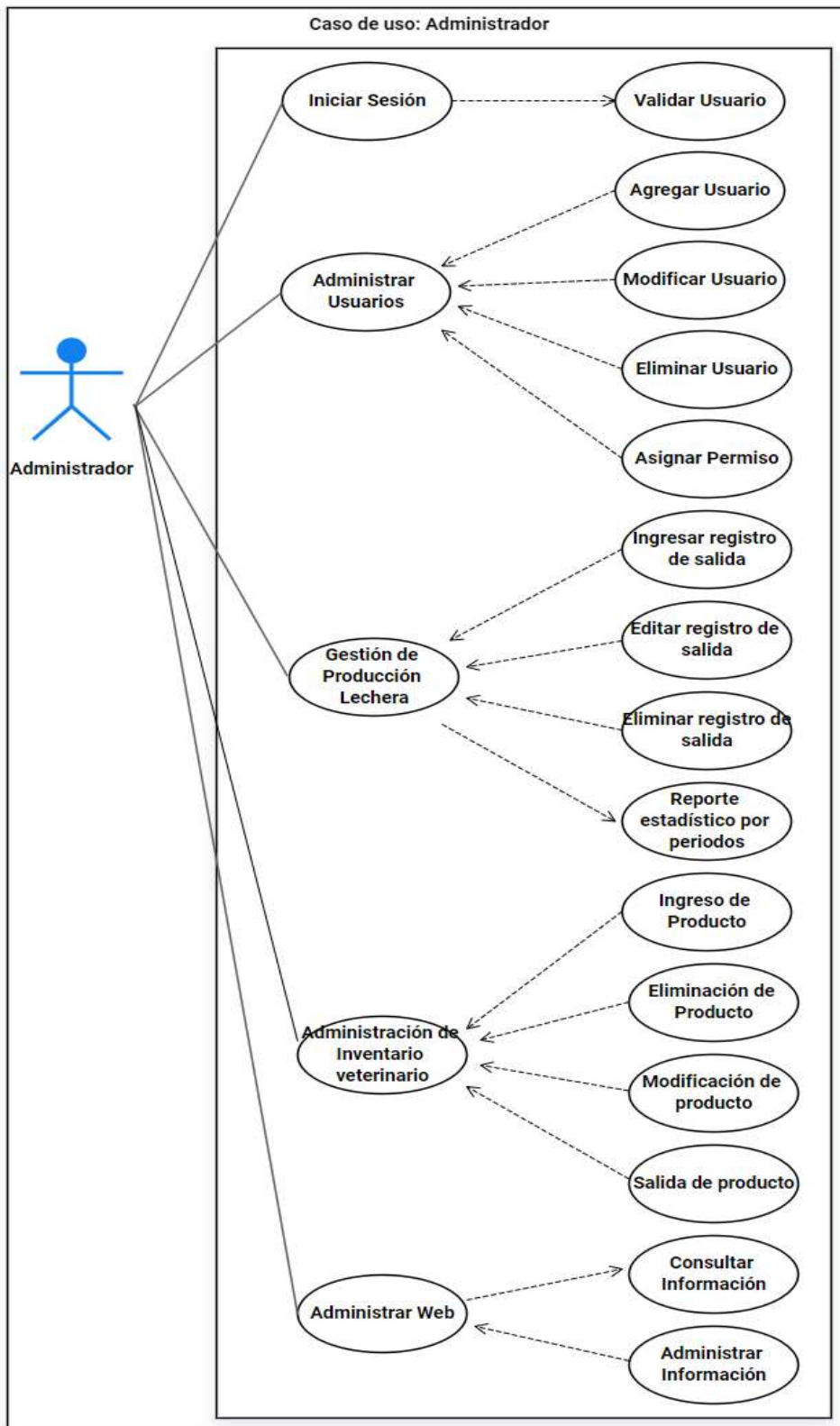


**Figura 60.** Arquitectura de la aplicación.

### c) Diagrama de casos de uso

Los diagramas de casos de uso facilitan la interpretación de los procesos que se llevan a cabo en la Hacienda ProduAgro y a la vez se indica como procedería su demostración en la aplicación web, haciendo énfasis en los procesos ya descritos en la fase de planificación, su finalidad es mostrar claramente cómo actúa el sistema informático con la interacción del usuario, complementando con los requerimientos necesarios con el fin de alcanzar el objetivo, se incorporan los actores, casos de uso y representaciones gráficas.

**Caso de uso: Administrador**



**Figura 61.** Caso de uso: Administrador.

**Tabla 80.** Descripción Caso de uso administrador.

Ítem	Evento
Nombre de caso de uso	Caso de uso administrador.

Actores	Administrador del sistema
Propósito	Administrar los usuarios, gestionar los módulos y controlar los aspectos relacionados con la aplicación web.
Precondiciones	Acceder al sistema mediante un perfil de administrador.
Flujo normal	<p>Iniciar sesión mediante la validación del usuario.</p> <p>Administrar usuarios para realizar acciones como agregar, modificar y eliminar.</p> <p>Gestionar la Producción lechera, ingresar registro de salida, editar registro de salida, eliminar registro de salida, reporte estadístico por periodos (es lo que obtiene de ese modulo).</p> <p>Administración de Inventario veterinario, ingreso de producto, eliminación de producto, modificación de producto, salida de producto.</p> <p>Administrar la web realizando consultas sobre la información referente a la Hacienda ProduAgro.</p>

## Caso de uso: Ternero

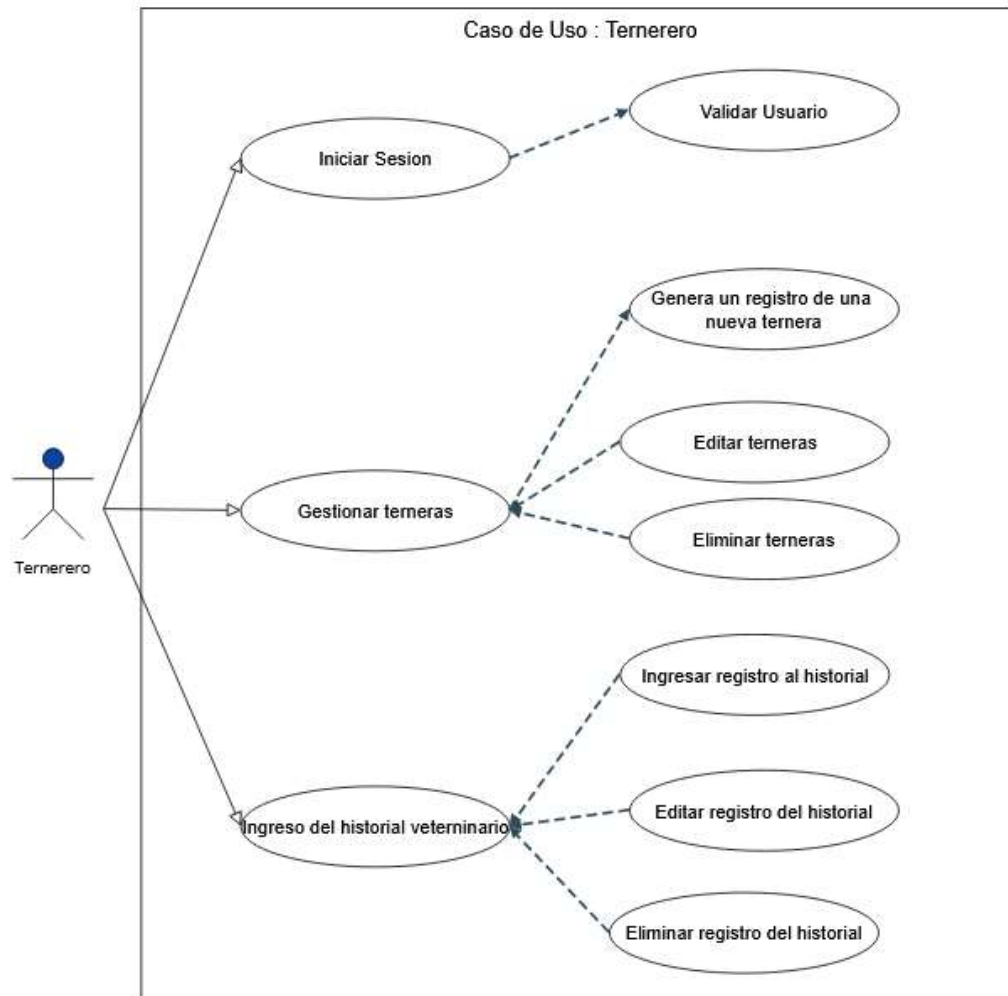
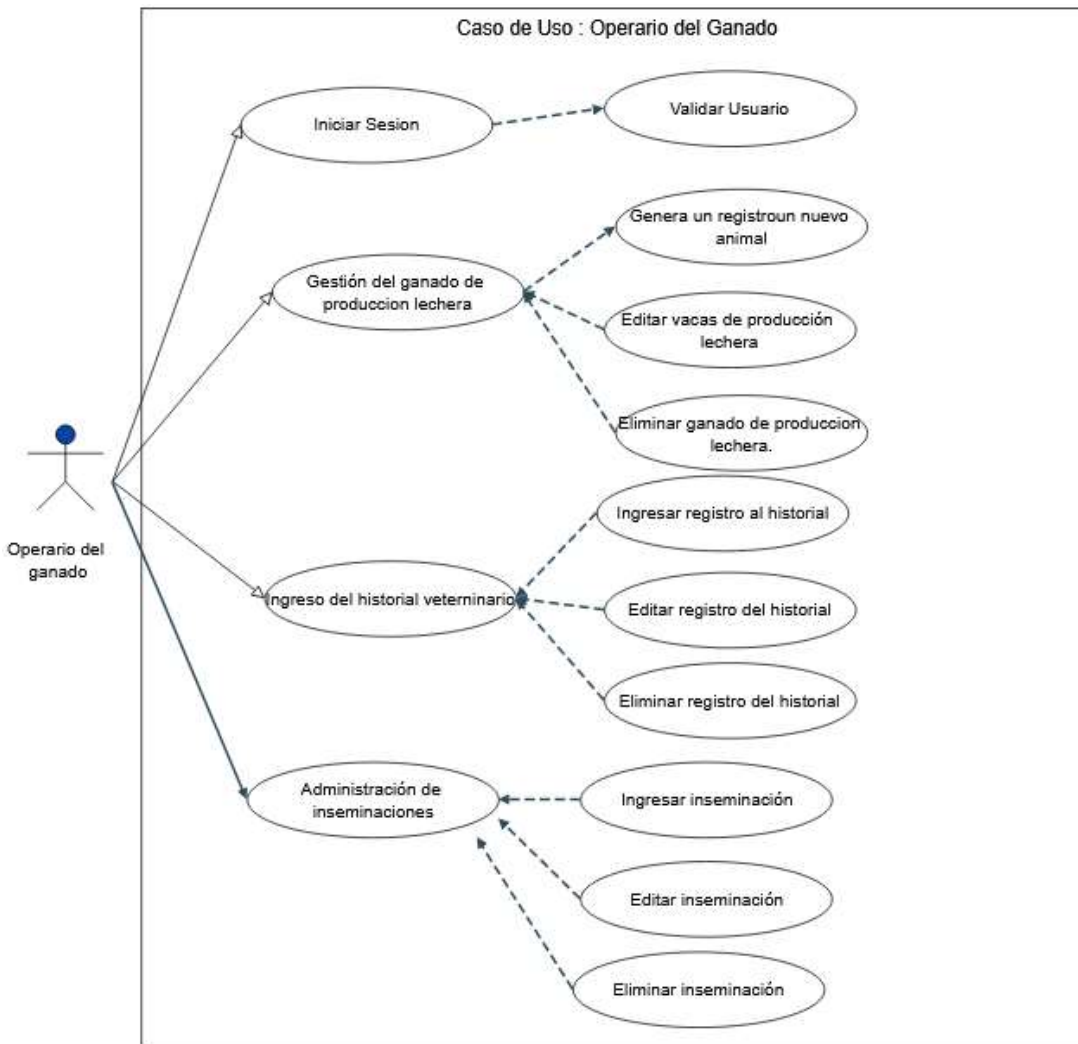


Figura 62. Caso de Uso: Ternero.

Tabla 81. Descripción Caso de uso Ternero.

Ítem	Evento
Nombre de caso de uso	Caso de uso Ternero
Actores	Ternero
Propósito	Gestión de terneras e historial clínico correspondiente.
Precondiciones	Acceder al sistema mediante un perfil de ternero.
Flujo normal	Inicio de sesión mediante la validación del usuario. Administrar las terneras y generar acciones como agregar, eliminar y modificar. Gestionar el módulo de historial veterinario cuando surja la necesidad.

**Caso de uso.** Operario del ganado

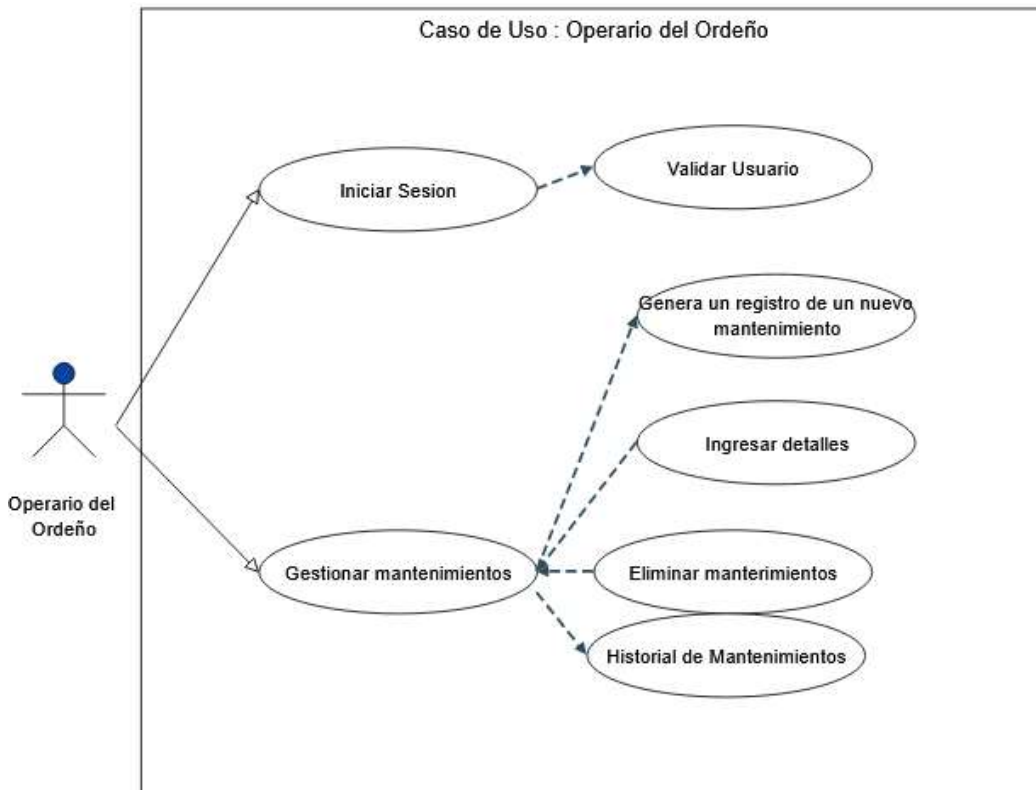


**Figura 63.** Caso De Uso: Operario Del Ganado.

**Tabla 82.** Descripción Caso De Uso Operario Del Ganado.

Ítem	Evento
Nombre de caso de uso	Caso de uso operario del ganado
Actores	Operario del ganado
Propósito	Gestión del ganado de producción lechera, acceso al historial veterinario y gestión de inseminaciones.
Precondiciones	Acceder al sistema mediante un perfil de operario del ganado
Flujo normal	Inicio de sesión mediante la validación del usuario. Administrar el ganado de producción lechera, ingresar a un nuevo animal apto, editar y eliminar. Gestionar el módulo de historial veterinario cuando surja la necesidad. Manejo del proceso de inseminación.

**Caso de uso:** Operario del ordeño

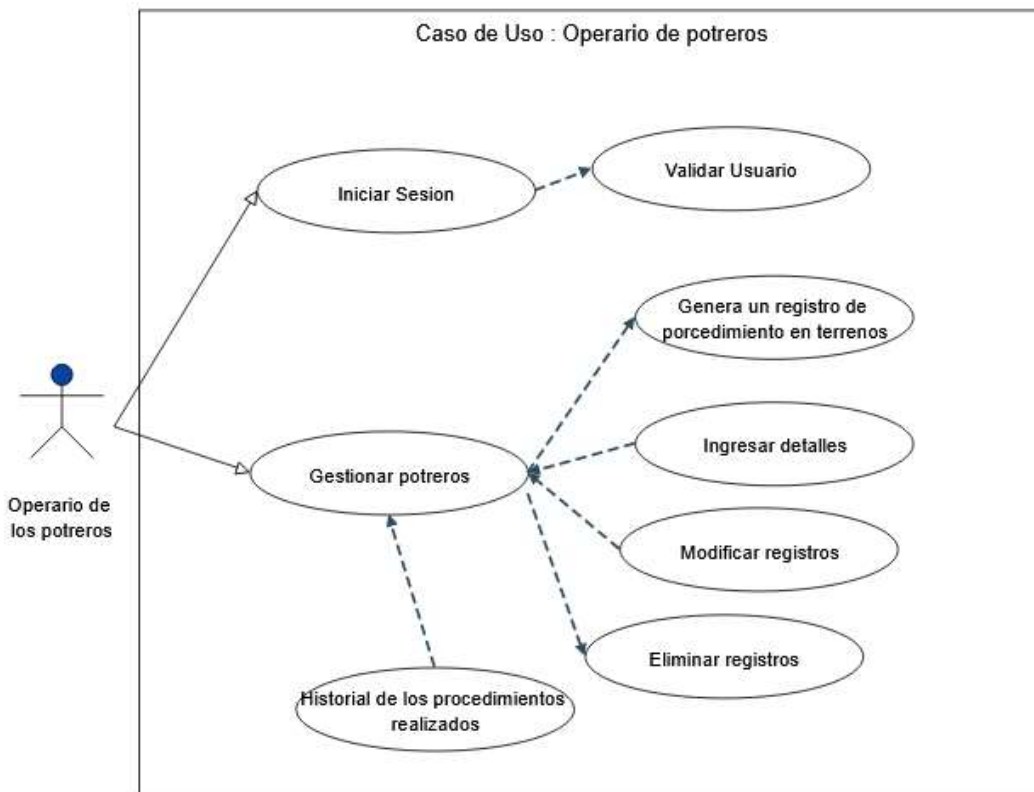


**Figura 64.** Caso De Uso: Operario Del Ordeño.

**Tabla 83.** Descripción Caso De Uso Operario Del Ordeño.

Ítem	Evento
Nombre de caso de uso	Caso de uso operario del ordeño
Actores	Operario del ordeño
Propósito	Gestión de los mantenimientos realizados a los equipos pecuarios.
Precondiciones	Acceder al sistema mediante un perfil operario del ordeño
Flujo normal	Inicio de sesión mediante la validación del usuario. Administrar los mantenimientos que se realizan. Visualizar el historial de mantenimientos realizados.

**Caso de uso:** Operario de potreros



**Figura 65.** Caso De Uso: Operario De Potreros.

**Tabla 84.** Descripción Caso De Uso Operario De Potreros.

Ítem	Evento
Nombre de caso de uso	Caso de uso operario de potreros
Actores	Operario del potrero
Propósito	Gestión de los procesos realizados en los terrenos
Precondiciones	Acceder al sistema mediante un perfil operario de potreros
Flujo normal	Inicio de sesión mediante la validación del usuario. Administrar los procesos realizados en los potreros Visualizar el historial de procedimientos realizados

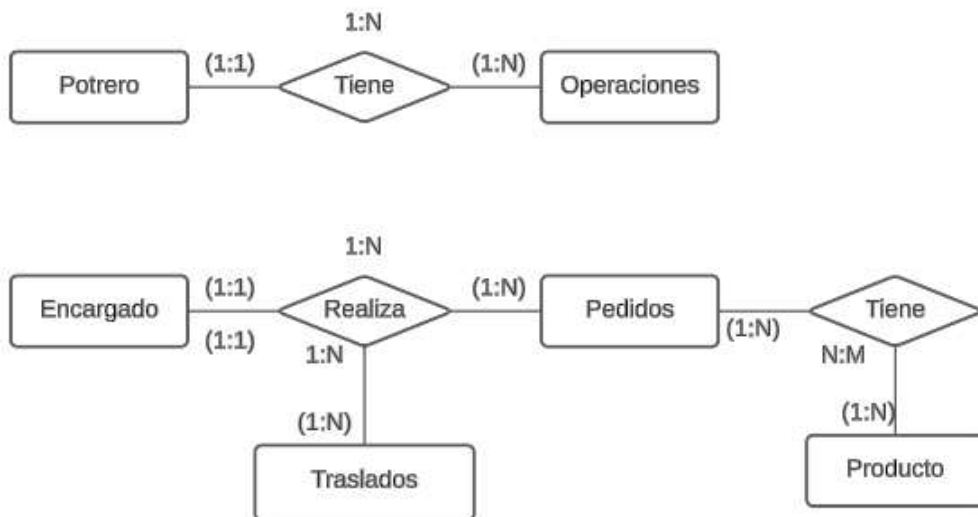
**d) Modelado de la base de datos**

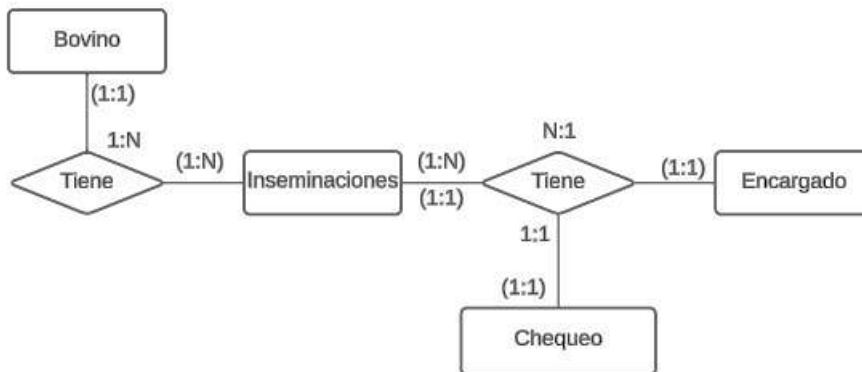
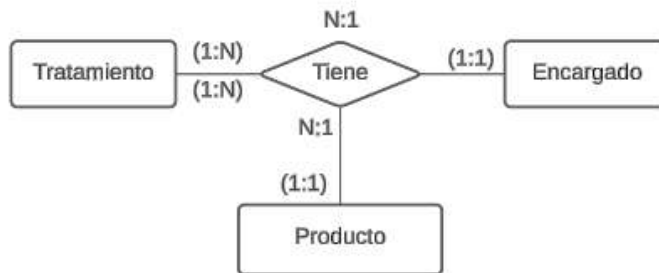
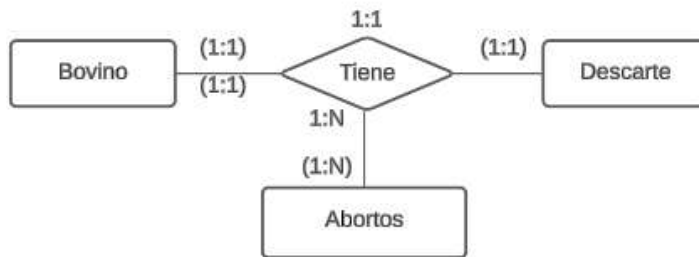
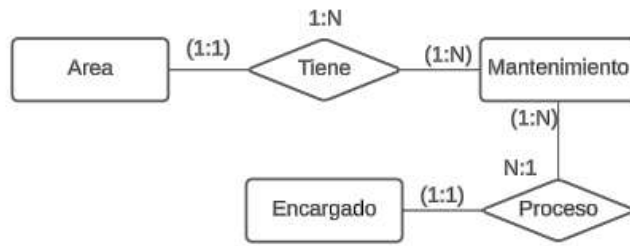
Para el presente proyecto de investigación se ha diseñado una base de datos relacional, la cual está representada por tablas con sus respectivos atributos de datos correspondientes para guardar la información y de esta manera ejecutar los procesos por medio del sistema web. Teniendo en cuenta que se empleó el gestor de base de datos SQL Server, se muestra a continuación la comparativa de los diferentes modelos relacionales como modelo conceptual, lógico y físico:

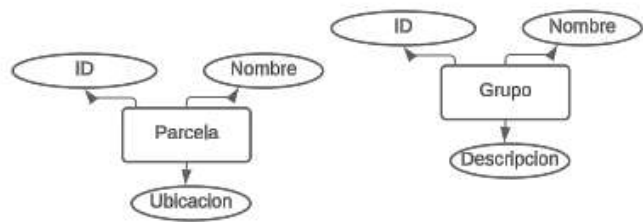
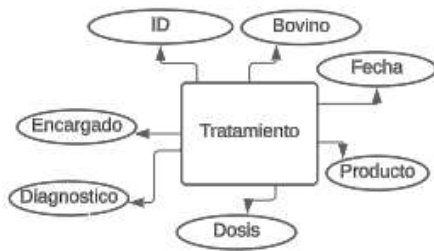
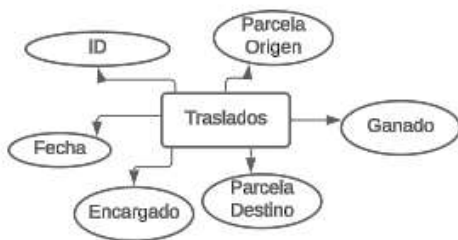
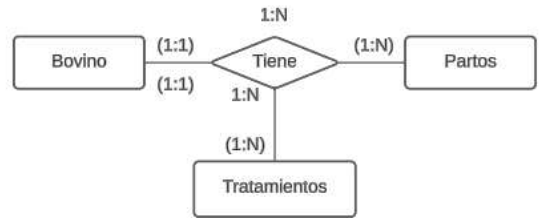
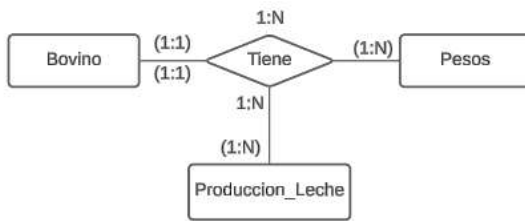
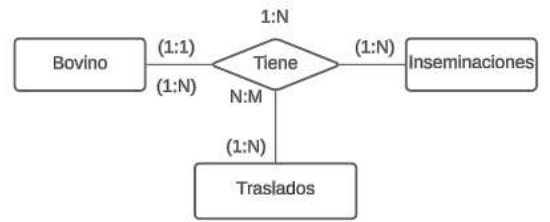
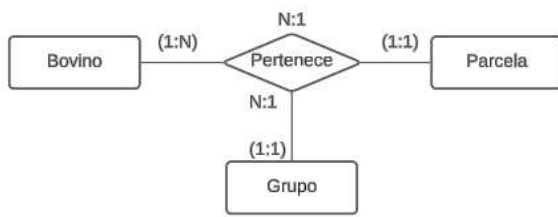
**Tabla 85.** Comparativa modelo conceptual, lógico y físico.

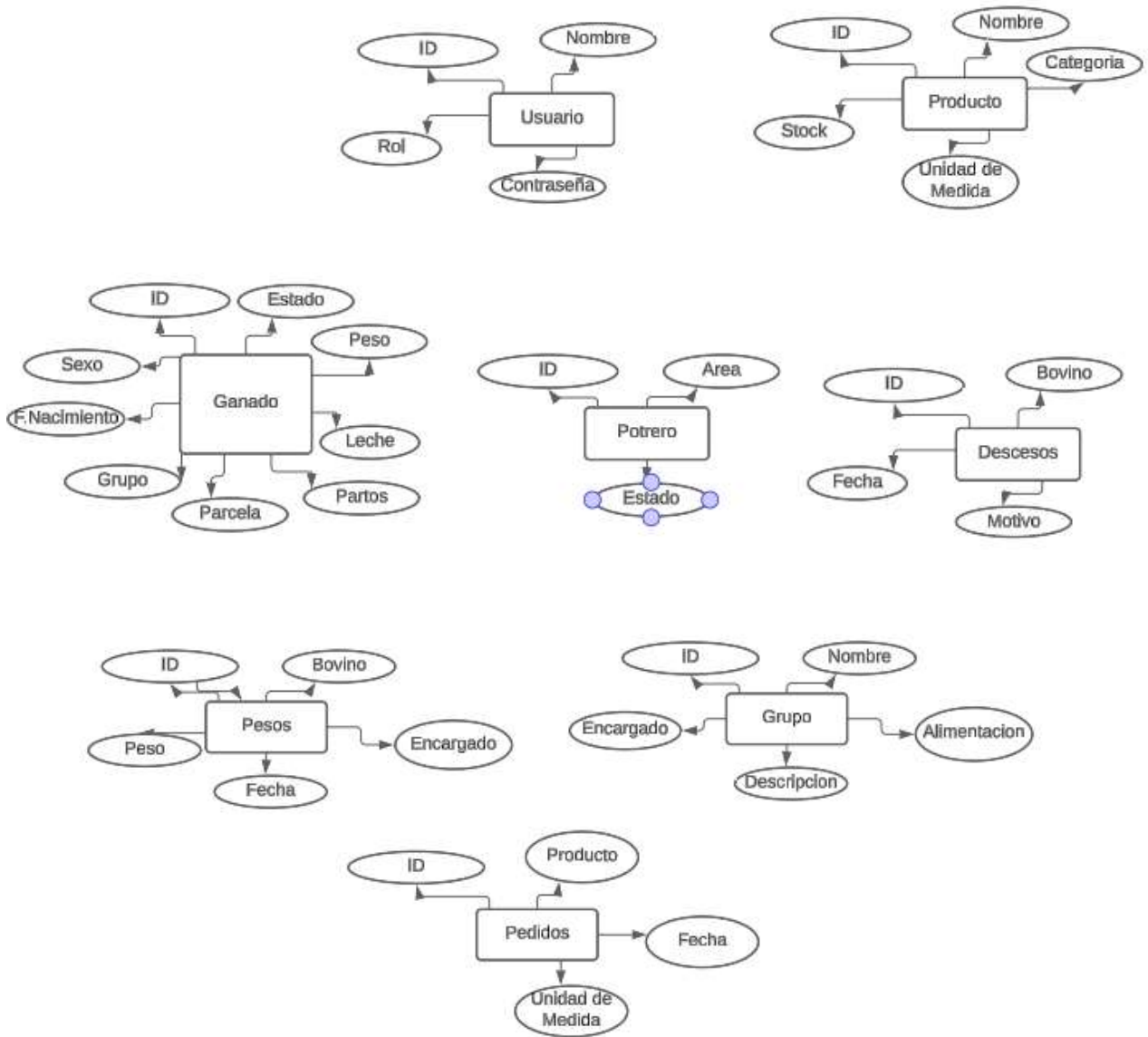
Característica	Conceptual	Lógico	Física
Nombre de entidades	✓	✓	
Relaciones de entidades	✓	✓	
Atributos		✓	
Teclas principales		✓	✓
Foreign keys		✓	✓
Nombres de tabla			✓
Nombres de columna			✓
Tipos de datos de columna			✓

### Modelo Conceptual de la base de datos









**Figura 66.** Modelo conceptual de la base de datos de la hacienda ProDuAgro.



## Modelo físico de la base de datos

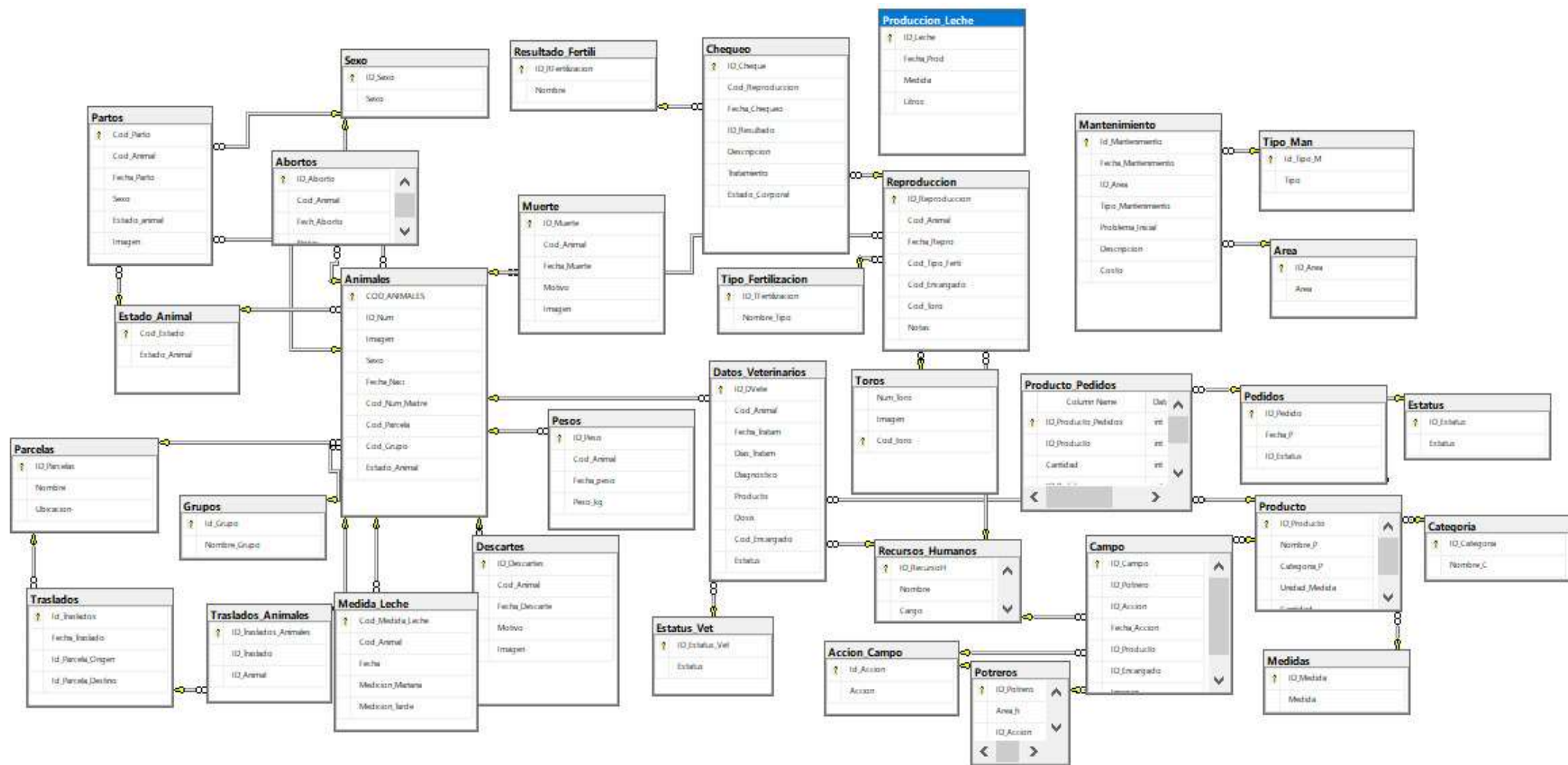


Figura 68. Modelo Físico de la Base de Datos.

La base de datos está conformada por las siguientes tablas y atributos de igual manera de describe sus respectivas claves.

### Descripción de tablas, atributos y claves PK y FK.

**Tabla 86.** Descripción de tablas, atributos y claves PK y FK.

Bovinos	Predio	Grupo	Raza
ID_Bovino	ID_Predio	ID_Grupo	ID_Raza
Identificador	Nombre	Nombre_Grupo	Nombre_Raza
Imagen	Ubicación	Descripcion	
Sexo		Dieta	
Fecha_N		ID_Trabajador	
Identificador_Madre			
Cod_Predio			
Cod_Grupo			
Estado_animal			
Cod_Raza			

**Tabla 87.** Descripción de tablas, atributos y claves PK y FK.

Trabajadores	Pesos	Medicion_Lactea	Decesos
ID_Trabajador	ID_Peso	ID_Medicion	ID_Deceso
Nombre	ID_Bovino	ID_Bovino	ID_Bovino
Cedula	Fecha	Fecha	Fecha_Deceso
Telefono	Peso_kg	Medicion_mañana	Causa
Cargo	Comentario	Medicion_tarde	ID_Trabajador
Responsabilidad	ID_Trabajador	ID_Trabajador	Imagen

**Tabla 88.** Descripción de tablas, atributos y claves PK y FK.

Descartes	Ventas_Descarte	Abortos	Partos
ID_Descarte	ID_Venta	ID_Aborto	ID_Partos
ID_Bovino	ID_Descarte	ID_Bovino	ID_Bovino
Fecha_Descarte	Fecha	Fecha_Aborto	Fecha_Partos
Causa	Precio	Causa	Sexo
ID_Trabajador	ID_Trabajador	ID_Trabajador	Estado_Animal
Imagen			Imagen
			ID_Trabajador

**Tabla 89.** Descripción de tablas, atributos y claves PK y FK.

Categoria	Medidas	Pedidos	Producto_Pedidos
ID_Categoria	ID_Medida	ID_Pedido	ID_Producto_Pedidos
Nombre_Categoria	Medida	Fecha	ID_Producto
		ID_Trabajador	Cantidad

**Tabla 90.** Descripción de tablas, atributos y claves PK y FK.

Traslados	Traslados_Animales	Datos_Veterinarios	Producto
ID_Traslado	ID_Traslado_Animales	ID_Dvete	ID_Producto
Fecha_Traslado	ID_Traslado	ID_Bovino	Nombre_Producto
ID_Predio_Origen	ID_Bovino	Fecha_Tratamiento	ID_Categoria
ID_Trabajador		Dias_Tratamiento	ID_Medida
ID_Predio_Destino		Diagnostico	Cantidad
Chofer		ID_Producto	Stock_Min
Placa		Tipo	
Telefono		Dosis	
Cedula		ID_Trabajador	
		Nota	
		ID_Trabajador	

**Tabla 91.** Descripción de tablas, atributos y claves PK y FK.

Entrada_Producto	Entradas	Reproduccion	Tipo_Fertilizacion
ID_Entrada_Producto	ID_Entrada	ID_Reproduccion	ID_Tipo_Ferti
ID_Producto	Fecha	ID_Bovino	Nombre_Tipo
Cantidad	ID_Trabajador	Fecha_Reproduccion	
ID_Entrada		ID_Tipo_Ferti	
		ID_Trabajador	
		ID_Toro	
		Cod_Lote_Pajuela	
		Raza_Pajuela	

**Tabla 92.** Descripción de tablas, atributos y claves PK y FK.

Toros	Chequeos	Mantenimiento	Tipo_Mantenimiento
ID_Toro	ID_Chequeo	ID_Mantenimiento	ID_Tipo_Mantenimiento
Num_Toro	ID_Reproduccion	Fecha_Mantenimiento	Tipo
Imagen	Fecha_Chequeo	ID_Area	
Raza	Resultado	ID_Tipo_Mantenimiento	
	Descripcion	Problema_Inicial	
	ID_Producto	Descripcion	
	Dosis	Costo	
	Estado_Corporal	ID_Trabajador	
	ID_Trabajador		

**Tabla 93.** Descripción de tablas, atributos y claves PK y FK.

Area	Potreros	Calificacion	Proceso_Potrero
ID_Area	ID_Potrero	ID_Calificacion	ID_Proceso
Area	Area_H	Nombre	Nombre
	ID_Calificacion		

Imagen
ID_Proceso

**Tabla 94.** Descripción de tablas, atributos y claves PK y FK.

Actividad	Actividad_Potrero	Produccion_Lechera	Despacho_Leche
ID_Actividad	ID_Actividad_Potrero	ID_Produccion_Lechera	ID_Despacho_Leche
Nombre	Fecha	Fecha	Fecha_Despacho
	Cantidad	Medida	Transportador
	ID_Trabajador	Litros	Medida
	ID_Potrero	Tanque	Litros
	ID_Actividad	ID_Trabajador	Tanque
			Acidez
			Temperatura
			ID_Trabajador

### e) Diseño de interfaces

**Inicio de Sesión:** Formulario descrito en las historias de usuario, nombre de usuario y contraseña.



**Figura 69.** Inicio de Sesión.

**Menú:** Se muestra el acceso a los distintos modulos del sistema. Carrusel con fines esteticos.



Figura 70. Menú de los módulos.

**Módulo de Administración de Usuarios:** Formulario descrito en historias de usuario correspondiente. Ingreso de datos del usuario. Tabla con usuarios registrados.

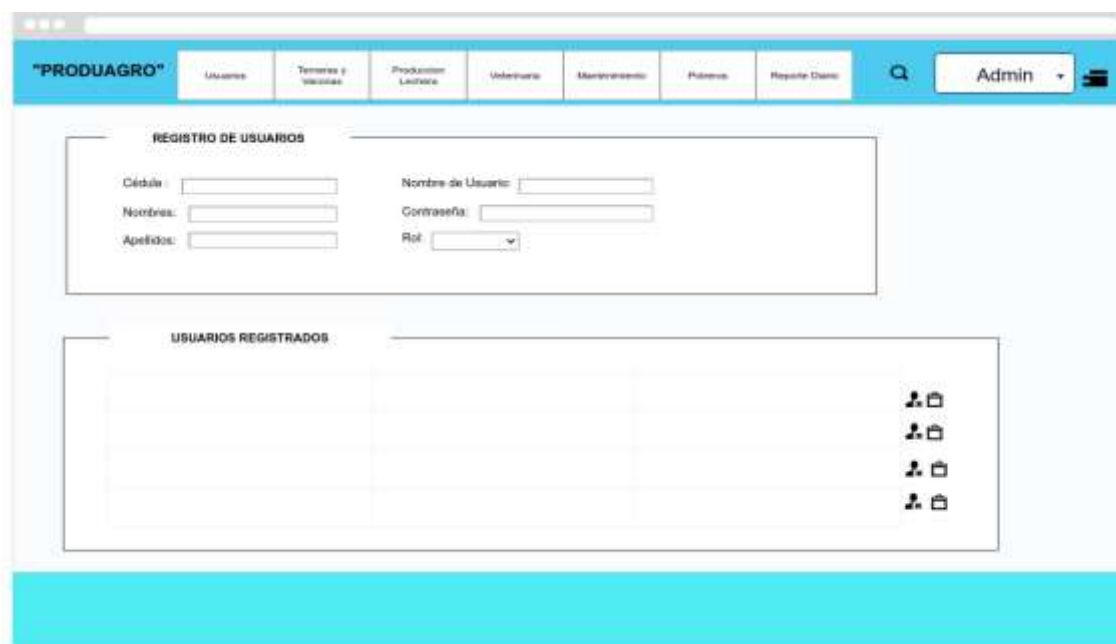


Figura 71. Módulo de Administración de Usuarios.

**Modulo de Terneras y Vaconas:** Formulario de registro, secciones de registro de machos, muertes a edad temprana y registro de pesos.

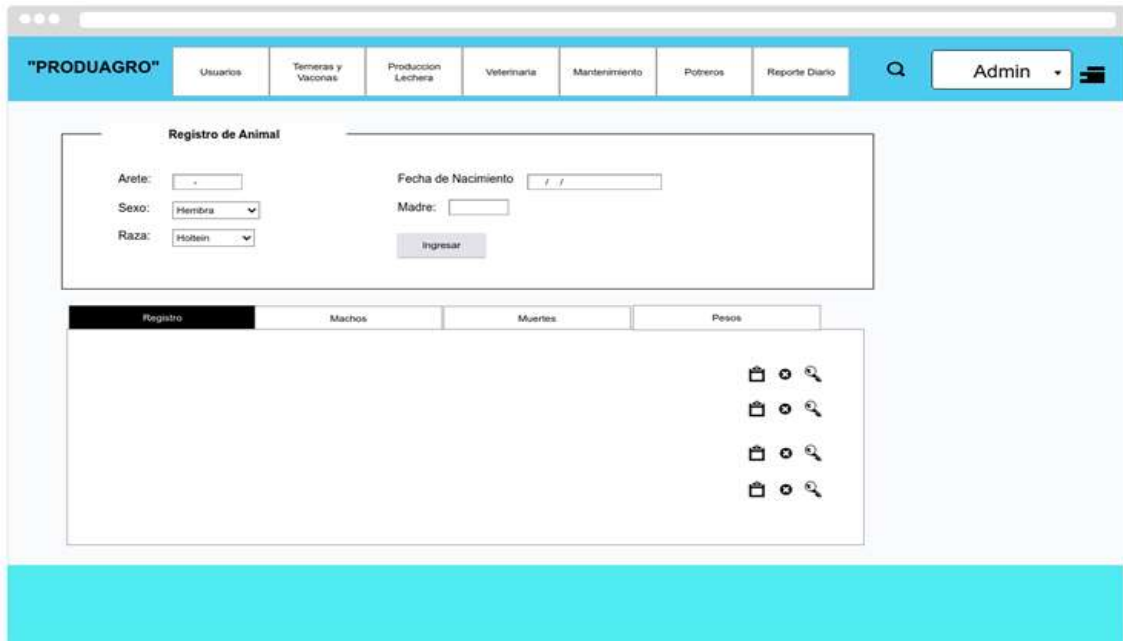


Figura 72. Modulo de Terneras y Vaconas.

**Módulo de Producción Lechera:** Formulario de registro de inseminaciones y cálculo de chequeo próximo. Tabla con datos relacionados al animal.

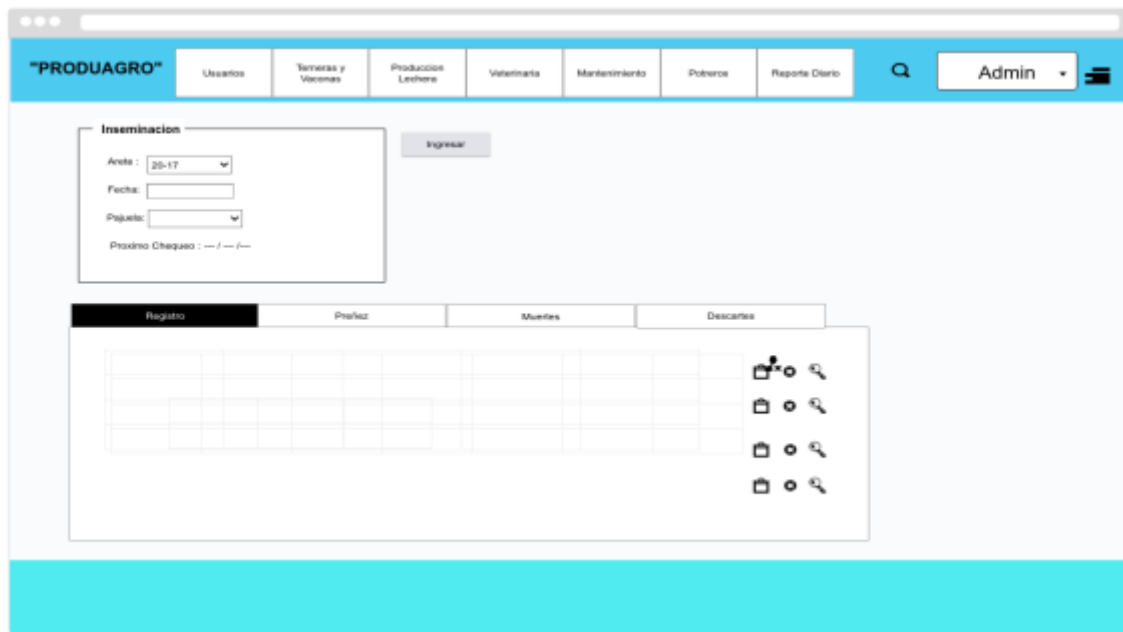


Figura 73. Módulo de Producción Lechera.

**Módulo de Veterinario:** Búsqueda de animal, formulario de registro y calendario del tratamiento. Tabla con historial veterinario.

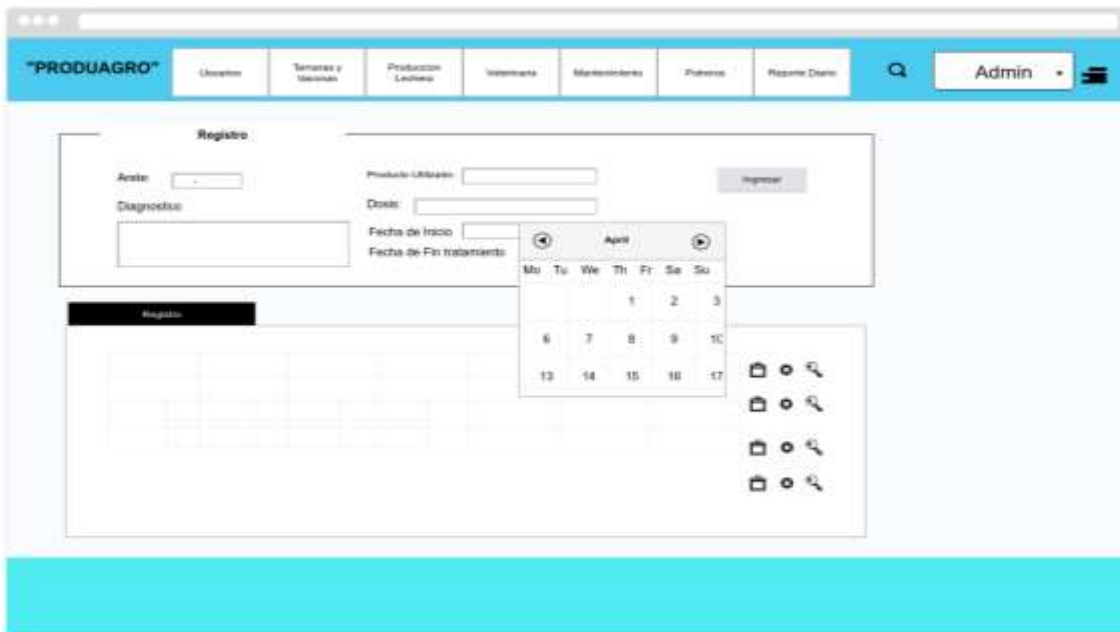


Figura 74. Módulo de Veterinario.

**Módulo de Mantenimiento:** Formulario de registro e historial de mantenimientos realizados.

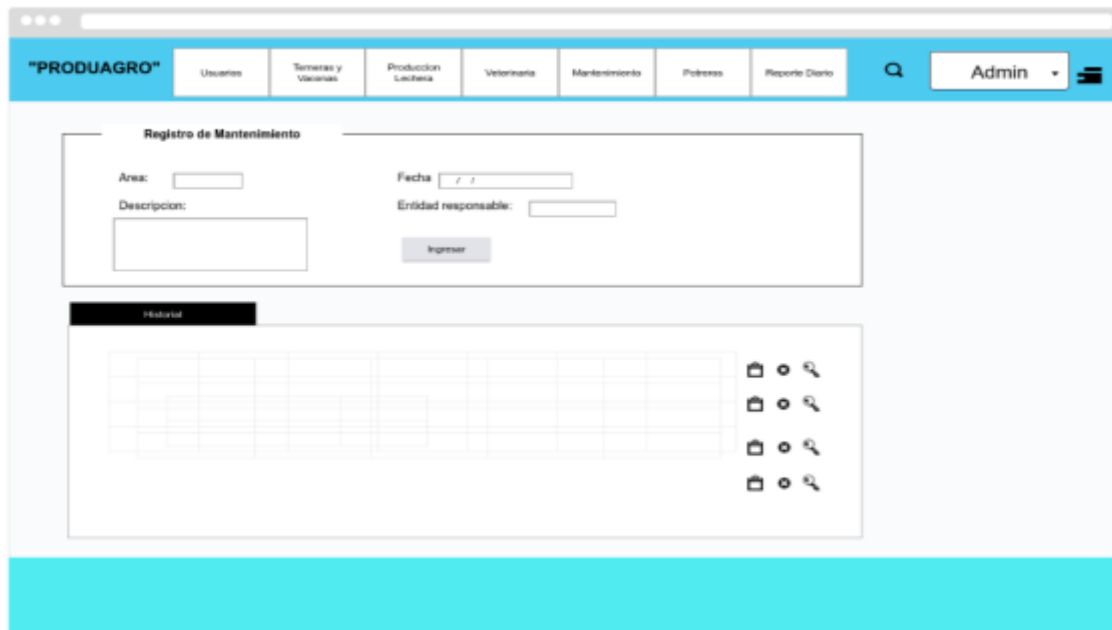


Figura 75. Módulo de Mantenimiento.

**Módulo de Poteros:** Registro del proceso realizado e instrumentos utilizados en los terrenos, historial de registros.

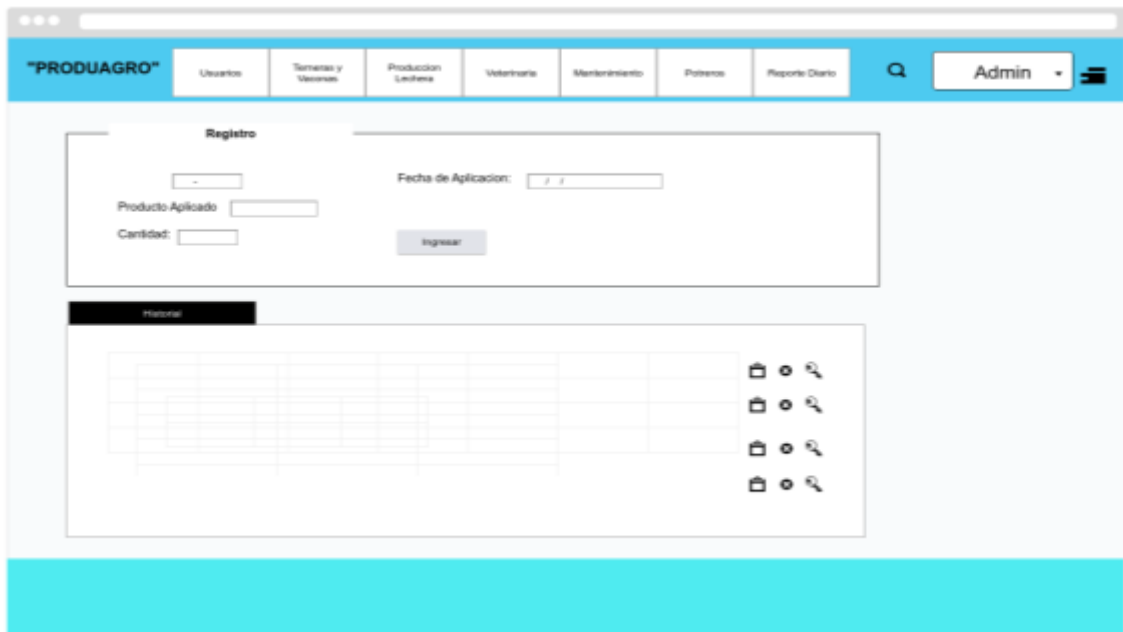


Figura 76. Módulo de Potreros.

**Módulo de reporte de producción lechera:** Formulario de registro diario de salida de leche, historial de registro, cuadros estadísticos generados por periodos.



Figura 77. Módulo de reporte de producción lechera.

### Fase de desarrollo

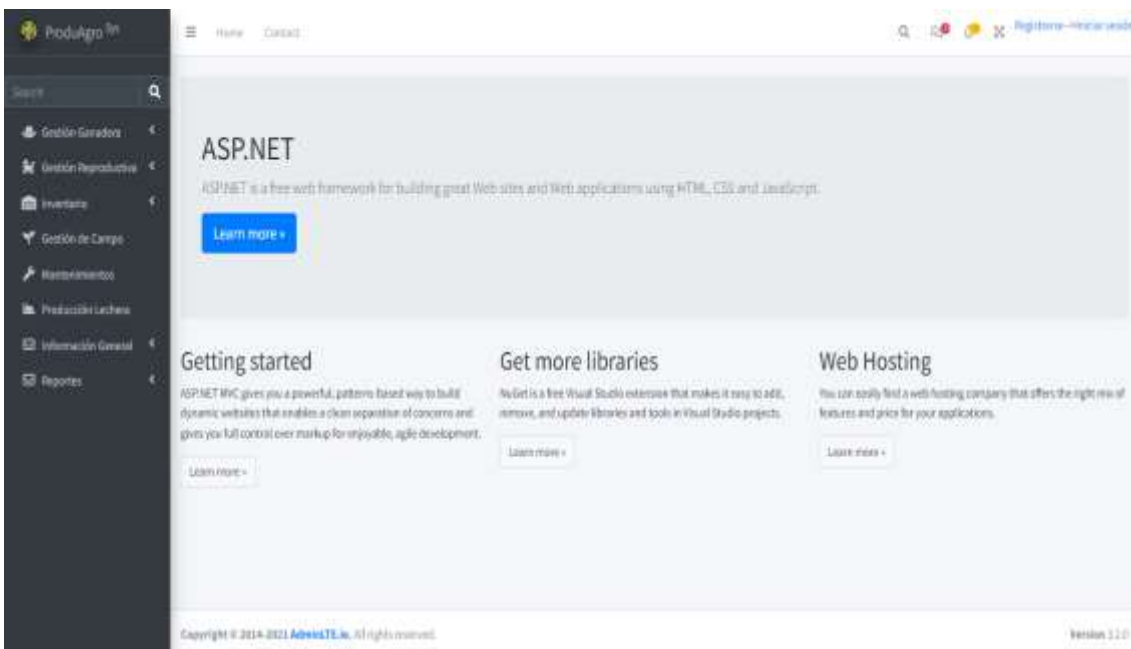
La etapa de codificación en nuestro proyecto implica la implementación práctica de las especificaciones utilizando el lenguaje de programación C# junto con el framework .NET utilizando el patrón de diseño Modelo-Vista-Controlador (MVC) en el

entorno de desarrollo Visual Studio, se escribirán líneas de código para dar forma a la lógica del software.

Además, se integró con una base de datos utilizando SQL Server para almacenar y gestionar la información durante este proceso se prestará especial atención a la calidad del código y se documentará de manera exhaustiva para facilitar futuras modificaciones y mantenimiento del sistema.

Empezamos con la parte del menú en donde van a constar todos los módulos disponibles para el sistema:

### **MVC Gestión Ganadera- Animales**



**Figura 78.** Diseño de la interfaz principal del sistema informático.

```
143 <li class="nav-item">
144 <a href="#" class="nav-link">
145 <i class="nav-icon fas fa-fw fa-hat-cowboy"/>
146 </i>
147 Gestión Ganadera
148 <i class="right fas fa-angle-left"/>
149 </i>
150 </a>
151 <ul class="nav nav-treeview">
152 <li class="nav-item">
153 <a href="/Animales/index" class="nav-link">
154 <i class="far fa-circle nav-icon"/>
155 </i> Animales
156 </a>
157 </li>
158 <li class="nav-item">
159 <a href="/Medida_Leche/index" class="nav-link">
160 <i class="far fa-circle nav-icon"/>
161 </i> Medición Lechera
162 </a>
163 </li>
164 <li class="nav-item">
165 <a href="/Muertes/index" class="nav-link">
166 <i class="far fa-circle nav-icon"/>
167 </i> Muertes
168 </a>
169 </li>
170 <li class="nav-item">
171 <a href="/Descartes/index" class="nav-link">
172 <i class="far fa-circle nav-icon"/>
173 </i> Descartes
174 </a>
175 </li>
176 </ul>
177 </li>
178 <li class="nav-item">
```

**Figura 79.** Código fuente de la aplicación.

Esta parte del código permite mostrar un menú de navegación con la categoría "Gestión Ganadera" al hacer clic en esta categoría se despliegan opciones adicionales de submenú como: Animales, Medición Lechera, Muertes y Descartes.

Cada opción del submenú se vincula a una página específica de la aplicación, lo que permite a los usuarios acceder rápidamente a diferentes funciones relacionadas con la gestión de la hacienda ProduAgro, como el seguimiento de animales, la medición de leche, el registro de muertes y el manejo de descartes y demás módulos.

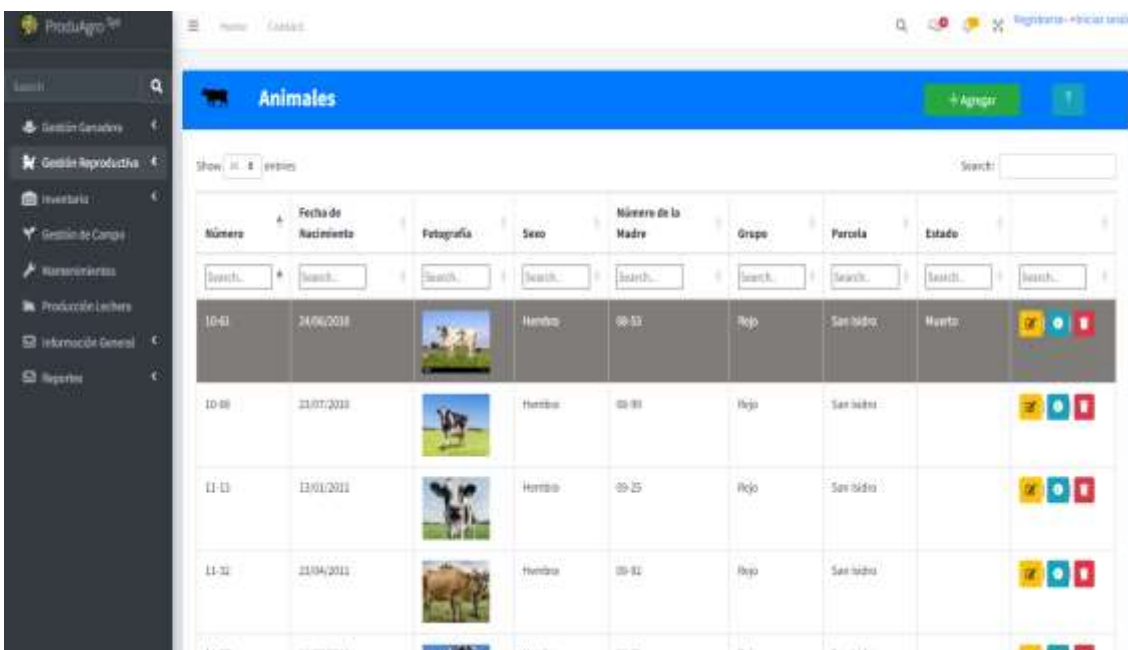
```

343 <script type="text/javascript">
344 $(document).ready(function () {
345     var table = $('#dataTable').DataTable({
346         orderCellsTop: true,
347         fixedHeader: true,
348
349         "createRow": function (row, data, index) {
350             //print row value
351             if (data[7] == 'Muerto' || data[5] == 'Descarte') {
352
353                 $('tr', row).css({
354                     'background-color': '#f7f7f7',
355                     'color': 'white',
356                     'border-style': 'solid',
357                     'border-color': 'red'
358                 });
359             }
360         });
361     });
362     //Creando una fila en el head de la tabla y la clonamos para cada columna
363     $('#dataTable thead tr').clone(false).appendTo('#dataTable thead');
364
365     $('#dataTable thead tr>tr>th').each(function (i) {
366         var title = $(this).text(); //es el nombre de la columna
367         $(this).html('<input type="text" style="width: 100px;" placeholder="Search..." />');
368     });
369
370     $('input', this).on('keyup change', function () {
371         if (table.column(i).search() != this.value) {
372             table
373                 .column(i)
374                 .search(this.value)
375                 .draw();
376         }
377     });
378 });
379 </script>

```

**Figura 80.** Código fuente de la aplicación.

Este script utiliza la biblioteca DataTables de jQuery para mejorar la funcionalidad de una tabla con el id "dataTable" donde configura la tabla con características como la ordenación de celdas en la parte superior y un encabezado fijo. Además, personaliza el formato de las filas según ciertos criterios, como cambiar el fondo y el texto de las celdas cuando el valor en la columna 7 es 'Muerto' o en la columna 5 es 'Descarte'. También añade la capacidad de filtrar los datos de la tabla mediante opciones de búsqueda en el encabezado de cada columna.



**Figura 81.** Diseño de interfaz Animales de la aplicación.

```

17: public partial class Animales
18: {
19:     [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
20:     Animales()
21:     {
22:         Abortos = new HashSet<Abortos>();
23:         Datos_Veterinarias = new HashSet<Datos_Veterinarias>();
24:         Descartes = new HashSet<Descartes>();
25:         Medida_Leche = new HashSet<Medida_Leche>();
26:         Muertes = new HashSet<Muertes>();
27:         Partos = new HashSet<Partos>();
28:         Pesos = new HashSet<Pesos>();
29:         Reproduccion = new HashSet<Reproduccion>();
30:         Traslados_Animales = new HashSet<Traslados_Animales>();
31:     }
32:
33:     [Key]
34:     [Required]
35:     public int COD_ANIMALES { get; set; }
36:
37:     [Required]
38:     [StringLength(10)]
39:     [Display(Name = "Numero")]
40:     public string ID_Madre { get; set; }
41:
42:     [Display(Name = "Fotografia")]
43:     public byte[] Images { get; set; }
44:
45:     [Required]
46:     public int Sexo { get; set; }

```

**Figura 82.** Código fuente de la aplicación.

El modelo representa la entidad "Animales" de la hacienda ProduAgro que contiene propiedades que describen atributos como el código del animal, número de identificación, fotografía, sexo, fecha de nacimiento, número de la madre, parcela y grupo al que pertenece, y estado actual. Además, incluye relaciones con otras entidades, como abortos, datos veterinarios, descartes, mediciones de leche, muertes, partos, pesos, reproducción y traslados de animales. Este diseño permite organizar y gestionar información detallada sobre cada animal, facilitando su seguimiento y control para la hacienda ProduAgro.

```

17: namespace ProduAgro.Controllers.ProduAgroControl
18: {
19:     public class AnimalesController : Controller
20:     {
21:         private ProduAgroEntities db = new ProduAgroEntities();
22:
23:         // GET: Animales
24:         public ActionResult Index()
25:         {
26:             var animales = db.Animales.Include(a => a.Grupos).Include(a => a.Parcelas).Include(a => a.Sexos).Include(a => a.Estados);
27:
28:             return View(animales.ToList());
29:         }
30:
31:         // GET: Animales/Details/5
32:         public ActionResult Details(int? id)
33:         {
34:             if (id == null)
35:             {
36:                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
37:             }
38:
39:             Animales animales = db.Animales.Find(id);
40:
41:             var record = db.Muertes.Where(x => x.Cod_Animal == animales.COD_ANIMALES).FirstOrDefault();
42:
43:             if (record != null)
44:             {
45:                 ViewBag.RecordFound = record.Fecha_Muerte;
46:             }
47:         }
48:     }
49: }

```

**Figura 83.** Código fuente de la aplicación.



```

128 public ActionResult Peso_Estadisticos()
129 {
130
131     return PartialView();
132 }
133 [HttpGet]
134 Referencias
135 public JsonResult Reporte_Medida_Leche_Animal(int id)
136 {
137     DT_Medida_Leche_Animal objReporte = new DT_Medida_Leche_Animal();
138
139     List<Medida_Leche_Animal> objlista = objReporte.RetornarMedidasAnimal(id);
140
141     return Json(objlista, JsonRequestBehavior.AllowGet);
142 }
143
144 [HttpGet]
145 Referencias
146 public JsonResult Reporte_Pesos_Animales(int id)
147 {
148     DT_Peso_Animal objReporte = new DT_Peso_Animal();
149
150     List<Pesos_Animal> objlista = objReporte.RetornarPesoAnimal(id);
151
152     return Json(objlista, JsonRequestBehavior.AllowGet);
153 }

```

Figura 86. Código fuente de la aplicación.

```

159 public ActionResult Imprimir(string parcela, string grupo, string sexo, string estado)
160 {
161     var animalesQuery = db.Animales.Include(a => a.Grupos).Include(a => a.Parcelas).Include(a => a.Sexo1).Include(a => a.E
162
163     // Aplicar filtros si se proporcionan
164     if (!string.IsNullOrEmpty(parcela))
165     {
166         animalesQuery = animalesQuery.Where(a => a.Parcelas.Nombre == parcela);
167     }
168
169     if (!string.IsNullOrEmpty(grupo))
170     {
171         animalesQuery = animalesQuery.Where(a => a.Grupos.Nombre_Grupo == grupo);
172     }
173
174     if (!string.IsNullOrEmpty(sexo))
175     {
176         animalesQuery = animalesQuery.Where(a => a.Sexo1.Sexo1 == sexo);
177     }
178     if (!string.IsNullOrEmpty(estado))
179     {
180         animalesQuery = animalesQuery.Where(a => a.Estado_Animal1.Estado_Animal1 == estado);
181     }
182     var animales = animalesQuery.ToList();
183
184     return View(animales);
185 }

```

Figura 87. Código fuente de la aplicación.

```

186 public ActionResult Create()
187 {
188     var grupos = from c in db.Grupos
189                 where c.Noembre_Grupo != "Descarte"
190                 select c;
191
192     ViewBag.Cod_Grupo = new SelectList(grupos, "Id_Grupo", "Nombre_Grupo");
193     ViewBag.Cod_Parcela = new SelectList(db.Parcelas, "ID_Parcelas", "Nombre");
194     ViewBag.Sexo = new SelectList(db.Sexo, "ID_Sexo", "Sexo1");
195     ViewBag.Estado_Animal = new SelectList(db.Estado_Animal, "Cod_Animal", "Estado_Animal");
196
197     return View();
198 }
199
200 // POST: Animales/Create
201 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que quiere enlazarse. Para
202 // más detalles, vna https://go.microsoft.com/fwlink/?linkid=317598.
203 [HttpPost]
204 [ValidateAntiForgeryToken]
205 public ActionResult Create([Bind(Include = "COD_ANIMALES,ID_Nue,Imagen,Sexo,Fecha_Naci,Cod_Nue_Madre,Cod_Parcela,Cod_Grupo
206
207
208     HttpPostedFileBase FileBase = Request.Files[0];
209
210     if (FileBase.ContentLength == 0)
211     {
212         ModelState.AddModelError("Imagen", "Es necesario seleccionar una imagen.");
213     }
214     else
215     {
216         if (FileBase.FileName.EndsWith(".jpg") || FileBase.FileName.EndsWith(".jpeg"))

```

**Figura 88.** Código fuente de la aplicación.

El controlador "AnimalesController", gestiona las operaciones relacionadas con la entidad "Animales" de la hacienda ProduAgro que Incluye acciones para mostrar la lista de animales, detalles de un animal específico, así como vistas parciales para mostrar datos veterinarios, mediciones de leche, pesos y partos de un animal en particular. Además, proporciona funcionalidades para generar informes y estadísticas relacionadas con los animales, como la cantidad de partos, el número de datos veterinarios registrados, el promedio de leche, entre otros. También permite la creación de nuevos registros de animales, aplicando filtros según la parcela, grupo, sexo y estado del animal.

```

246 public ActionResult Edit(int? id)
247 {
248     if (id == null)
249     {
250         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
251     }
252     Animales animales = db.Animales.Where(p => p.COD_ANIMALES == id).FirstOrDefault();
253     if (animales == null)
254     {
255         return HttpNotFound();
256     }
257     var grupos = from c in db.Grupos
258                 where c.Nombre_Grupo != "Muerto" && c.Nombre_Grupo != "Descarta"
259                 select c;
260
261     ViewBag.Cod_Grupo = new SelectList(grupos, "Id_Grupo", "Nombre_Grupo", animales.Cod_Grupo);
262     ViewBag.Cod_Parcela = new SelectList(db.Parcelas, "ID_Parcels", "Nombre", animales.Cod_Parcela);
263     ViewBag.Sexo = new SelectList(db.Sexo, "ID_Sexo", "Sexo", animales.Sexo);
264     return View(animales);
265 }
266
267 // POST: Animales/Edit/5
268 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que quiere enlazarse. Para
269 // más detalles, vea https://go.microsoft.com/fwlink/?linkid=391809.
270 [HttpPost]
271 [ValidateAntiForgeryToken]
272 public ActionResult Edit([Bind(Include = "COD_ANIMALES,ID_Nue,Imagen,Sexo,Fecha_Naci,Cod_Nue_Madre,Cod_Parcela,Cod_Grupo")
273 {
274     Animales _animales = new Animales();
275
276     HttpPostedFileBase FileBase = Request.Files[0];
277

```

**Figura 89.** Código fuente de la aplicación.

La acción de "Editar" en el controlador "AnimalesController" está conformado por dos métodos que son:

La primera parte (Edit GET) se encarga de cargar la información del animal con el id proporcionado y prepara las listas desplegables (como grupos, parcelas y sexo) para la vista de edición.

La segunda parte (Edit POST) maneja la lógica de actualización después de que se envían los datos editados del formulario.

```

316 public ActionResult Delete(int? id)
317 {
318     if (id == null)
319     {
320         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
321     }
322     Animales animales = db.Animales.Where(p => p.COD_ANIMALES == id).FirstOrDefault();
323     if (animales == null)
324     {
325         return HttpNotFound();
326     }
327     return PartialView("_DeleteAnimalPartialView", animales);
328 }
329
330 // POST: Animales/Delete/5
331
332
333
334 public ActionResult DeleteConfirmed(int id)
335 {
336     Animales animales = db.Animales.Find(id);
337     db.Animales.Remove(animales);
338     db.SaveChanges();
339     return RedirectToAction("Index");
340 }
341
342 protected override void Dispose(bool disposing)
343 {
344     if (disposing)
345     {
346         db.Dispose();
347     }
348 }

```

**Figura 90.** Código fuente de la aplicación.

En esta parte se puede observar cómo está desarrollada la funcionalidad de eliminación de un animal en el controlador "AnimalesController".

La acción Delete GET recibe un identificador (id) como parámetro. Si el identificador es nulo, devuelve un código de estado BadRequest. Luego, busca el animal correspondiente en la base de datos y, si no se encuentra, devuelve un código de estado HttpNotFound. Si encuentra el animal, devuelve una vista parcial llamada "\_DeleteAnimalPartialView" con el modelo del animal.

```
24 </form>
25 <div class="card-body">
26
27 <div class="container">
28 <div class="row">
29 <div class="col">
30
31 <div class="form-group">
32 <html.LabelFor(model => model.ID_Nus, htmlAttributes: new { @class = "control-label col-md-4" })>
33 <input class="form-control col-md-10" id="ID_Nus" name="ID_Nus" placeholder="Número">
34 <html.ValidationMessageFor(model => model.ID_Nus, "", new { @class = "text-danger" })>
35 </div>
36
37 </div>
38 <div class="col">
39
40 <div class="form-group">
41 <html.LabelFor(model => model.Fecha_Naci, htmlAttributes: new { @class = "control-label col-md-18" })>
42 <div class="col-md-10">
43 <html.EditorFor(model => model.Fecha_Naci, new { htmlAttributes = new { @class = "form-control" } })>
44 <html.ValidationMessageFor(model => model.Fecha_Naci, "", new { @class = "text-danger" })>
45 </div>
46 </div>
47
48 </div>
49 <div class="col">
50 <div class="form-group">
51 <html.LabelFor(model => model.Sexo, "Sexo", htmlAttributes: new { @class = "control-label col-md-3" })>
52 <div class="col-md-10">
53 <html.DropDownList("Sexo", null, htmlAttributes: new { @class = "form-control" })>
54 <html.ValidationMessageFor(model => model.Sexo, "", new { @class = "text-danger" })>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
```

Figura 91. Código fuente de la aplicación.

En esta vista de crear se puede observar cómo está estructurado el HTML para el botón de crear nuevo animal con sus respectivos campos para posteriormente esta información sea guardada en la base de datos SQLServer.

```

3
4
5 <div class="table-responsive">
6   <table id="" class="display table table-bordered" width="100%">
7
8     <thead>
9       <tr>
10        <th>
11          <%= Html.DisplayFor(model => model.Animalen.ID_Num) %>
12        </th>
13        <th>
14          <%= Html.DisplayFor(model => model.Estado_Vet.Estado) %>
15        </th>
16        <th>
17          <%= Html.DisplayFor(model => model.Productos.Nombre_P) %>
18        </th>
19        <th>
20          <%= Html.DisplayFor(model => model.Recurso_Humano.Nombre) %>
21        </th>
22        <th>
23          <%= Html.DisplayFor(model => model.Fecha_Tratam) %>
24        </th>
25        <th>
26          <%= Html.DisplayFor(model => model.Dias_Tratam) %>
27        </th>
28        <th>
29          <%= Html.DisplayFor(model => model.Diagnostico) %>
30        </th>
31        <th>
32          <%= Html.DisplayFor(model => model.Dosis) %>
33        </th>
34      </tr>
35    </thead>
36  </table>
37
38

```

**Figura 92.** Código fuente de la aplicación.

En esta vista se puede observar cómo se hace un llamado de los datos para reflejarlo en la tabla la información de datos veterinarios, el estado del animal, los productos, los recursos humanos, fechas, días de tratamiento y su respectivo tratamiento y dosis asignado, con la finalidad de presentar esta información al usuario y esté al tanto de todo lo que se hace con el animal, llevar la información más precisa y clara.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

**Figura 93.** Código fuente de la aplicación.

Esta vista muestra los detalles de un animal en el sistema de gestión ganadera. La vista incluye información general del animal, como su número de identificación, sexo,

fecha de nacimiento, grupo al que pertenece, parcela asignada y número de la madre. Además, se muestra una imagen del animal con la opción de ampliarla en un modal, descargarla y una sección de información adicional que incluye gráficos relacionados con la producción y promedio de leche mensual.

```
91 // Guardar el archivo
92 workbook.xlsx.writeBuffer().then(function (data) {
93     var blob = new Blob([data], { type: "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet" });
94     saveAs(blob, "animales.xlsx");
95 });
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Figura 94. Código fuente de la aplicación.

Este script en JavaScript contiene dos funciones. La primera, `exportarExcel`, utiliza la biblioteca `ExcelJS` para exportar los datos de una tabla HTML, incluyendo imágenes, a un archivo Excel descargable. La segunda, `imprimirTabla`, abre una nueva ventana y la configura para imprimir el contenido de la tabla con estilos específicos. Estas funciones son útiles para generar informes visuales y exportar datos de tablas de manera eficiente.

## MVC Gestión Ganadera- Medición Lechera

Número	Promedio Histórico (Ltrs)	Máximo
10-01	4	13
10-02	5	14
14-03	5	16
14-04	4	10
14-04	5	13
15-14	5	13
15-15	5	12
15-16	5	15

Figura 95. Diseño de interfaz de Medición Lechera de la aplicación.

Fecha	Tarde	Mañana
2021-10-17	4	4
2021-10-24	4	3
2021-10-31	4	4
2021-11-07	4	3
2021-11-04	3	3
2021-11-21	3	3

Figura 96. Diseño de interfaz de Medición Lechera de la aplicación

```

8  <!-- DataTables Example -->
9  <div class="card shadow mb-4">
10 <div class="card-header py-3 d-flex mb-2">
11 <div class="col-lg-4">
12 <div class="row">
13 <a class="d-none d-sm-inline-block card-image col-lg-2">
14 <i class="far fa-clipboard" style="font-size:36px;color:black"/></i>
15 </a>
16 <h2 class="h-0 font-weight-bold col-lg-8 align-middle text-primary">Medida de Leche</h2>
17 </div>
18 </div>
19 </div>
20 <div class="col-lg-1"></div>
21 <div class="col-lg-9">
22 <div class="row">
23 <label class="col-sm-6 col-form-label"></label>
24
25 <a placement="bottom" href="@Url.Action("Create","Medida_Leche")" class="m-0 btn btn-success btn-icon-split col-lg-3">
26 <span class="icon text-white-50">
27 <i class="fas fa-plus"></i>
28 </span>
29 <span class="text">Agregar</span>
30 </a>
31 <label class="col-lg-1"></label>
32
33 <a placement="left" class="d-none d-sm-inline-block btn btn-sm btn-info shadow-sm col-lg-1">
34 <i _ngcontent-tss-c162="" class="fas fa-question fa-sm text-white-50"></i>
35 </a>
36 </div>
37 </div>
38 </div>
39

```

Figura 97. Código fuente de la aplicación.

Este código HTML representa una tarjeta de visualización de datos relacionados con la medida de leche. Incluye un encabezado con opciones de navegación y un cuerpo que contiene una tabla con información detallada sobre el número, el promedio histórico y el máximo de la medida de leche. Además, presenta botones para agregar nuevos datos y acceder a detalles específicos. La tabla se genera dinámicamente a partir de un modelo de datos, y cada fila de la tabla tiene un enlace para ver detalles adicionales.

```

57 [HttpPost]
58 [ValidateAntiForgeryToken]
59 public ActionResult Create([Bind(Include = "Cod_Medida_Leche,Cod_Animal,Fecha,Medicion_Mañana,Medicion_Tarde")] Medida_Leche medida
60 {
61     if (ModelState.IsValid)
62     {
63         db.Medida_Leche.Add(medida_Leche);
64         db.SaveChanges();
65         return RedirectToAction("Index");
66     }
67     var animales = from c in db.Animales
68                   join g in db.Grupos on c.Cod_Grupo equals g.Id_Grupo
69                   where g.Nombre_Grupo == "Rejo"
70                   orderby c.ID_Num descending
71                   select c
72     ;
73     ViewBag.Cod_Animal = new SelectList(animales, "COD_ANIMALES", "ID_Num", medida_Leche.Cod_Animal);
74     return View(medida_Leche);
75 }
76
77 // GET: Medida_Leche/Edit/5

```

Figura 98. Código fuente de la aplicación.

```

public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }

    Medida_Leche medida_Leche = db.Medida_Leche.Find(id);
    if (medida_Leche == null)
    {
        return HttpNotFound();
    }

    ViewBag.Cod_Animal = new SelectList(db.Animales, "COD_ANIMALES", "ID_Num", medida_Leche.Cod_Animal);
    return View(medida_Leche);
}

```

**Figura 99.** Código fuente de la aplicación.

```

96     [HttpPost]
97     [ValidateAntiForgeryToken]
98     public ActionResult Edit([Bind(Include = "Cod_Medida_Leche,Cod_Animal,Fecha,Medicion_Mañana,Medicion_Tarde")] Medida_Leche medida_Leche)
99     {
100         if (ModelState.IsValid)
101         {
102             db.Entry(medida_Leche).State = EntityState.Modified;
103             db.SaveChanges();
104             return RedirectToAction("Index");
105         }
106         var animales = from c in db.Animales
107                       join g in db.Grupos on c.Cod_Grupo equals g.Id_Grupo
108                       where g.Nombre_Grupo == "Rejo"
109                       orderby c.ID_Num descending
110                       select c
111         ;
112         ViewBag.Cod_Animal = new SelectList(animales, "COD_ANIMALES", "ID_Num", medida_Leche.Cod_Animal);
113         return View("Details", medida_Leche);
114     }

```

**Figura 100.** Código fuente de la aplicación.

```

118 public ActionResult Delete(int? id)
119 {
120     if (id == null)
121     {
122         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
123     }
124     Medida_Leche medida_Leche = db.Medida_Leche.Where(p => p.Cod_Medida_Leche == id).FirstOrDefault();
125     if (medida_Leche == null)
126     {
127         return HttpNotFound();
128     }
129     return PartialView("_DeleteMedidaPartialView", medida_Leche);
130 }
131 // POST: Medida_Leche/Delete/
132 public ActionResult DeleteConfirmed(int id)
133 {
134     Medida_Leche medida_Leche = db.Medida_Leche.Find(id);
135     db.Medida_Leche.Remove(medida_Leche);
136     db.SaveChanges();
137     return RedirectToAction("Index");
138 }
139
140 protected override void Dispose(bool disposing)
141 {
142     if (disposing)
143     {
144         db.Dispose();
145     }
146     base.Dispose(disposing);
147 }

```

**Figura 101.** Código fuente de la aplicación.

El controlador Medida\_LecheController gestiona las operaciones CRUD para la entidad Medida\_Leche, incluyendo la visualización de listas, detalles, creación, edición y eliminación de registros.

```

public partial class Medida_Leche
{
    [Key]
    1 referencia
    public int Cod_Medida_Leche { get; set; }

    13 referencias
    public int Cod_Animal { get; set; }

    [Column(TypeName = "date")]
    [Display(Name = "Fecha")]
    [DataType(DataType.Date)]
    [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
    0 referencias
    public DateTime Fecha { get; set; }
    [Required]
    3 referencias
    public int? Medicion_Mañana { get; set; }
    [Required]
    3 referencias
    public int? Medicion_Tarde { get; set; }

    2 referencias
    public virtual Animales Animales { get; set; }
}

```

**Figura 102.** Código fuente de la aplicación.

La clase del Modelo Medida\_Leche representa el modelo de datos para la entidad "Medida de Leche". Incluye atributos como la clave primaria Cod\_Medida\_Leche, información sobre la fecha de medición y las mediciones de leche en la mañana y

tarde. Además, establece una relación con la entidad Animales para asociar cada medida de leche con un animal específico en la base de datos.

## MVC Gestión Ganadera- Muertes

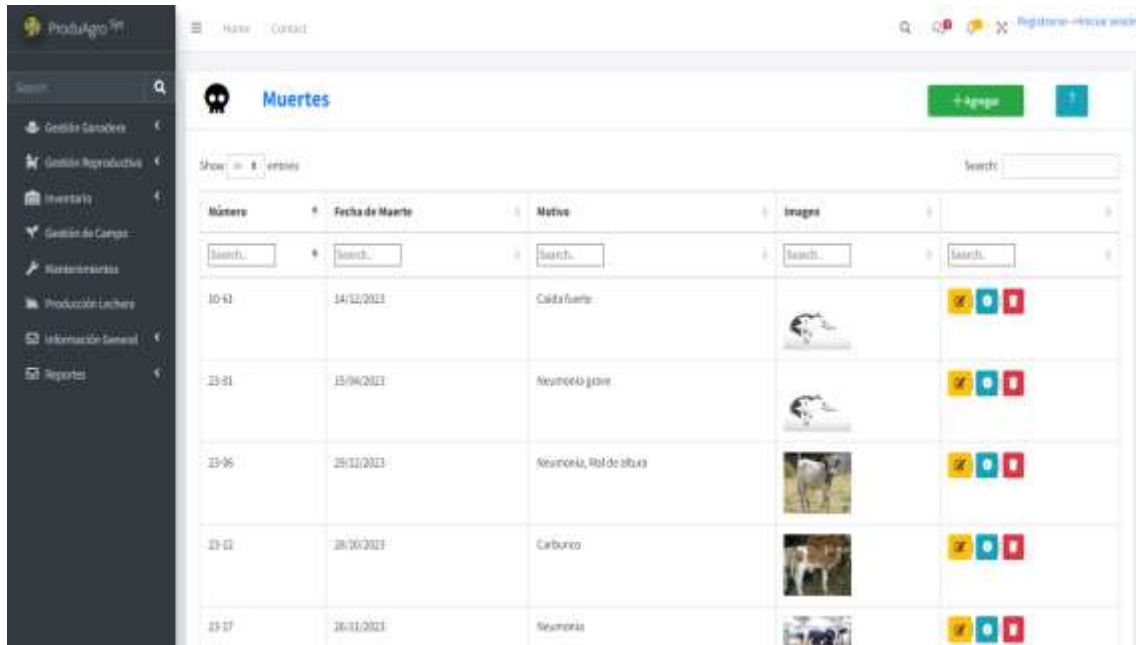


Figura 103. Diseño de interfaz de la vista de Muertes de la aplicación.

```

10 <div class="card shadow mb-4">
11   <div class="card-header py-3 d-flex align-items-center">
12     <div class="col-lg-3">
13       <div class="row">
14         <div class="d-none d-sm-inline-block card-image col-lg-3">
15           <i class="fas fa-skull" style="font-size:36px;color:black;"></i>
16         </div>
17         <div class="m-0 font-weight-bold col-lg-9 align-middle text-primary">Muertes</div>
18       </div>
19     </div>
20   </div>
21   <div class="col-lg-4"></div>
22   <div class="col-lg-5">
23     <div class="row">
24       <div class="row">
25         <div class="col-sm-6 col-form-label"></div>
26       </div>
27       <div class="col-sm-6 col-form-label"></div>
28       <div class="col-sm-6 col-form-label"></div>
29       <a placement="bottom" href="#"> <span class="icon text-white-50">
30         <i class="fas fa-plus"></i>
31       </span>
32       <span class="text">Agregar</span>
33     </a>
34     <div class="col-lg-1"></div>
35     <div class="col-lg-1"></div>
36     <div class="col-lg-1"></div>
37     <div class="col-lg-1"></div>
38     <a placement="left" class="d-none d-sm-inline-block btn btn-sm btn-info shadow-sm col-lg-1">
39       <i _ngcontent-tss-c162="" class="fas fa-question fa-sm text-white-50"></i>
40     </a>
41   </div>

```

Figura 104. Código fuente de la aplicación.

```

45 <div class="card-body">
46 <div class="table-responsive">
47 <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
48 <thead>
49 <tr>
50 <th>
51 @Html.DisplayNameFor(model => model.Animales.ID_Num)
52 </th>
53 <th>
54 @Html.DisplayNameFor(model => model.Fecha_Muerte)
55 </th>
56 <th>
57 @Html.DisplayNameFor(model => model.Motivo)
58 </th>
59 <th>
60 @Html.DisplayNameFor(model => model.Imagen)
61 </th>
62 <th></th>
63 </tr>
64 </thead>
65 </thead>
66 <tbody>
67 @foreach (var item in Model)
68 {
69 <tr>
70 <td>
71 @Html.DisplayFor(modelItem => item.Animales.ID_Num)
72 </td>
73 <td>
74 @item.Fecha_Muerte.ToString("dd/MM/yyyy")
75 </td>
76 <td>
77

```

Figura 105. Código fuente de la aplicación.

```

77 <td>
78 @Html.DisplayFor(modelItem => item.Motivo)
79 </td>
80 <td>
81 
82 </td>
83 <td>
84 <div class="input-group btn">
85 <a href="@Url.Action("Edit", "Muertes", new { id = item.ID_Muerte })" class="btn btn-warning btn-circle btn-sm">
86 <i class="fas fa-edit">
87 </i>
88 </a>
89 <a href="@Url.Action("Details", "Muertes", new { id = item.ID_Muerte })" class="btn btn-info btn-circle btn-sm">
90 <i class="fas fa-info-circle">
91 </i>
92 </a>
93 <button type="button" class="btn btn-danger btn-circle btn-sm" data-toggle="modal" data-target="#DeleteAnimal-@item.ID_Num"
94 href="@Url.Action("Delete", "Muertes", new { id = item.ID_Muerte })">
95 <i class="fas fa-trash">
96 </i>
97 </button>
98 @Html.Partial("DeleteMuertePartialView", item)
99 </div>

```

Figura 106. Código fuente de la aplicación.

La vista Muertes muestra información sobre muertes de animales haciendo uso de Bootstrap para la presentación y una tabla para organizar los datos. Se emplea un bucle para mostrar detalles de cada muerte, incluyendo opciones para editar, ver detalles y eliminar. Se implementa una modal para confirmar las eliminaciones.

```

@section scripts{
<script type="text/javascript">
$(document).ready(function () {
var table = $('#dataTable').DataTable({
orderCellsTop: true,
fixedHeader: true
});

//Creamos una fila en el head de la tabla y lo clonamos para cada columna
$('#dataTable thead tr').clone(true).appendTo('#dataTable thead');

$('#dataTable thead tr:eq(1) th').each(function (i) {
var title = $(this).text(); //es el nombre de la columna
$(this).html('<input type="text" style="width: 100px;" placeholder="Search.." />');

$( 'input', this ).on( 'keyup change', function () {
if ( table.column( i ).search() !== this.value ) {
table
.column( i )
.search( this.value )
.draw();
}
}
});
});
});
</script>

```

Figura 107. Código fuente de la aplicación.

El script utiliza DataTables de jQuery para agregar controles de búsqueda a cada columna de una tabla HTML, mejorando la funcionalidad de búsqueda y filtrado.

```

17 namespace ProduAgra.Controllers.ProduAgraController
18 {
19     //Referencia
20     public class MuertesController : Controller
21     {
22         private ProduAgra_Entities db = new ProduAgra_Entities();
23
24         // GET: Muertes
25         //Referencia
26         public ActionResult Index()
27         {
28             var muerte = db.Muerte.Include(m => m.Animales);
29             return View(muerte.ToList());
30         }
31
32         // GET: Muertes/Details/5
33         //Referencia
34         public ActionResult Datos_Vet(int Id)
35         {
36             Muerte muerte = db.Muerte.Find(Id);
37             DateTime limiteFecha = muerte.Fecha_Muerte.AddDays(-30);
38             db = new ProduAgra_Entities();
39
40             Datos_Veterinarios datos_Veterinarios = db.Datos_Veterinarios.Include(m => m.Animales)
41                 .Where(m => m.Cod_Animal == muerte.Cod_Animal && m.Fecha_Tratam == limiteFecha)
42                 .OrderByDescending(m => m.Fecha_Tratam).FirstOrDefault();
43
44
45             return PartialView("Datos_VetPartial", datos_Veterinarios);
46         }
47     }
48 }

```

Figura 108. Código fuente de la aplicación.

```

99 public ActionResult Details(int? id)
100 {
101     if (id == null)
102     {
103         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
104     }
105
106     Muerte muerte = db.Muerte.Find(id);
107     Pesos pesos = db.Pesos.Include(m => m.Animales).Where(m => m.Cod_Animal == id).OrderByDescending(m => m.Fecha_peso).FirstOrDefault();
108
109     var promedioLeche = (from prod in db.Medida_Leche.Where(x => x.Cod_Animal.Equals(muerte.Cod_Animal))
110                         select prod.Medicion_Mañana)
111                         .Average();
112
113     var promedioLecheT = (from prod in db.Medida_Leche.Where(x => x.Cod_Animal.Equals(muerte.Cod_Animal))
114                          select prod.Medicion_Tarde)
115                          .Average();
116
117     int promedio;
118
119     if (promedioLeche == null || promedioLecheT == null)
120     {
121         promedio = 0;
122     }
123     else
124     {
125         promedio = (int)(promedioLecheT + promedioLeche) / 2;
126     }
127
128     ViewBag.promedio = promedio;
129
130     if (pesos != null)

```

Figura 109. Código fuente de la aplicación.

```

96 public ActionResult DatosVet(int id)
97 {
98     db = new ProduAgro_Entities();
99
100     Datos_Veterinarios datos_Veterinarios = db.Datos_Veterinarios.Include(m => m.Animales).Where(m => m.Cod_Animal == id).OrderByDescending
101
102     return PartialView(datos_Veterinarios);
103 }
104
105 // GET: Muertes/Create
106 // Referencias
107 public ActionResult Create()
108 {
109     var muertes = from c in db.Animales
110                  join g in db.Estado_Animal on c.Estado_animal equals g.Cod_Estado
111                  where g.Cod_Estado == 1
112                  orderby c.ID_Num ascending
113                  select c;
114
115     ViewBag.Cod_Animal = new SelectList(muertes, "COD_ANIMALES", "ID_Num");
116
117     return View();
118 }

```

Figura 110. Código fuente de la aplicación.

```

125 public ActionResult Create([Bind(Include = "ID_Muerte,Cod_Animal,Fecha_Muerte,Motivo,Imagen")] Muerte muerte)
126 {
127     HttpPostedFileBase FileBase = Request.Files[0];
128
129     if (FileBase.ContentLength == 0)
130     {
131         ModelState.AddModelError("Imagen", "Es necesario seleccionar una imagen.");
132     }
133     else
134     {
135         if (FileBase.FileName.EndsWith(".jpg") || FileBase.FileName.EndsWith(".jpeg"))
136         {
137             WebImage image = new WebImage(FileBase.InputStream);
138             muerte.Imagen = image.GetBytes();
139         }
140         else
141         {
142             ModelState.AddModelError("Imagen", "El sistema solo acepta imagenes.");
143         }
144     }
145
146     if (ModelState.IsValid)
147     {
148         db.Muerte.Add(muerte);
149         var query = (from a in db.Animales
150                     where a.COD_ANIMALES == muerte.Cod_Animal
151                        select a).FirstOrDefault();
152         query.Estado_animal = 2;
153         db.SaveChanges();
154         return RedirectToAction("Index");
155     }
156 }
157
158 var muertes = from c in db.Animales

```

Figura 111. Código fuente de la aplicación.

```

168 public ActionResult Edit(int? id)
169 {
170     if (id == null)
171     {
172         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
173     }
174     Muerte muerte = db.Muerte.Find(id);
175
176     if (muerte == null)
177     {
178         return HttpNotFound();
179     }
180     ViewBag.Cod_Animal = new SelectList(db.Animales, "COD_ANIMALES", "ID_Animal", muerte.Cod_Animal);
181     return View(muerte);
182 }
183
184 // POST: Muertes/Edit/5
185 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que quiere enlazarse. Para obtener
186 // más detalles, vea https://go.microsoft.com/fwlink/?linkid=391949.
187 [HttpPost]
188 [ValidateAntiForgeryToken]
189 public ActionResult Edit([Bind(Include = "ID_Muerte,Cod_Animal,Fecha_Muerte,Motivo,Imagen")] Muerte muerte)
190 {
191     Muerte _muerte = new Muerte();
192
193     HttpPostedFileBase FileBase = Request.Files[0];
194
195     if (FileBase.ContentLength == 0)
196     {

```

Figura 112. Código fuente de la aplicación.

```

262 public ActionResult Delete(int? id)
263 {
264     if (id == null)
265     {
266         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
267     }
268     Muerte muerte = db.Muerte.Where(p => p.ID_Muerte == id).FirstOrDefault();
269     if (muerte == null)
270     {
271         return HttpNotFound();
272     }
273     return PartialView("_DeleteMuertePartialView", muerte);
274 }
275
276 // POST: Muertes/Delete/5
277
278
279 public ActionResult DeleteConfirmed(int id)
280 {
281     Muerte muerte = db.Muerte.Find(id);
282     var query = (from a in db.Animales
283                 where a.COD_ANIMALES == muerte.Cod_Animal
284                 select a).FirstOrDefault();
285     query.Estado_animal = 1;
286
287     db.Muerte.Remove(muerte);
288     db.SaveChanges();
289     return RedirectToAction("Index");
290 }

```

**Figura 113.** Código fuente de la aplicación.

El controlador `MuertesController` gestiona las acciones relacionadas con las muertes de animales en el sistema. Incluye operaciones como la visualización de la lista de muertes, la creación, edición, eliminación y detalles de una muerte específica, así como la obtención de la imagen asociada a una muerte. Además, maneja la actualización del estado del animal en función de las muertes y cambios realizados. La lógica incluye la manipulación de imágenes, la validación de datos, la actualización del estado de los animales y la interacción con la base de datos para realizar las operaciones CRUD.

## MVC Gestión Ganadera- Descartes

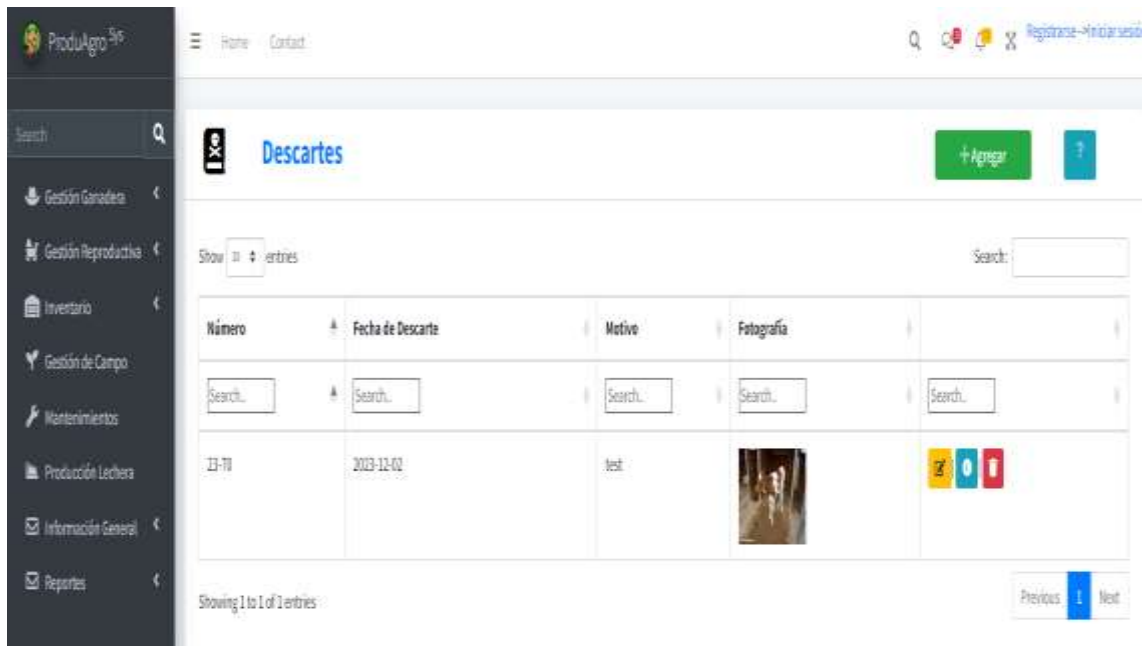


Figura 114. Diseño de interfaz de la vista Descartes de la aplicación.

```
11 <div class="card shadow mb-4">
12 <div class="card-header py-3 d-flex mb-2">
13 <div class="col-lg-3">
14 <div class="row">
15 <a class="d-none d-sm-inline-block card-image col-lg-3">
16 <i class="fas fa-book-dead" style="font-size:36px;color:black"></i>
17 </a>
18 <h3 class="m-0 font-weight-bold col-lg-9 align-middle text-primary">Descartes</h3>
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 <div class="col-lg-4"></div>
25 <div class="col-lg-5">
26 <div class="row">
27 <div class="row">
28 <div class="row">
29 <label class="col-sm-6 col-form-label"></label>
30 <a placement="bottom" href="url.Action("create","descartes")" class="m-0 btn btn-success btn-icon-split col-lg-1">
31 <span class="icon text-white-50">
32 <i class="fas fa-plus"></i>
33 </span>
34 <span class="text">Agregar</span>
35 </a>
36 <label class="col-lg-1"></label>
37 </div>
38 </div>
39 </div>
40 <a placement="left" class="d-none d-sm-inline-block btn btn-sm btn-info shadow-sm col-lg-1">
41 <i _ngcontent-tss-cl62="" class="fas fa-question fa-sm text-white-50"></i>
42 </a>
43 </div>
```

Figura 115. Código fuente de la aplicación.

```

48 <div class="card-body">
49 <div class="table-responsive">
50 <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
51 <thead>
52 <tr>
53 <th>
54 <@Html.DisplayNameFor(model => model.Animales.ID_Num)
55 </th>
56 <th>
57 <@Html.DisplayNameFor(model => model.Fecha_Descarte)
58 </th>
59 <th>
60 <@Html.DisplayNameFor(model => model.Motivo)
61 </th>
62 <th>
63 <@Html.DisplayNameFor(model => model.Imagen)
64 </th>
65 </tr>
66 </thead>
67 </table>
68 </div>
69 <tbody>
70 <@foreach (var item in Model)
71 <tr>
72 <td>
73 <@Html.DisplayFor(modelItem => item.Animales.ID_Num)
74 </td>
75 <td>
76 <@Html.DisplayFor(modelItem => item.Fecha_Descarte)
77 </td>
78 <td>
79 <@Html.DisplayFor(modelItem => item.Motivo)
80 </td>
81 <td>
82 <@Html.DisplayFor(modelItem => item.Imagen)
83 </td>
84 </tr>
85 </tbody>
86 </table>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>

```

Figura 116. Código fuente de la aplicación.

```

75 <td>
76 <@Html.DisplayFor(modelItem => item.Fecha_Descarte)
77 </td>
78 <td>
79 <@Html.DisplayFor(modelItem => item.Motivo)
80 </td>
81 <td>
82 <@Html.DisplayFor(modelItem => item.Imagen)
83 </td>
84 </tr>
85 </tbody>
86 </table>
87 </div>
88 <div class="input-group-btn">
89 <a href="@Url.Action("Edit", "Descartes", new { id = item.ID_Descartes })" class="btn btn-warning btn-circle btn-sm">
90 <i class="fas fa-edit">
91 </i>
92 </a>
93 <a href="@Url.Action("Details", "Descartes", new { id = item.ID_Descartes })" class="btn btn-info btn-circle btn-sm">
94 <i class="fas fa-info-circle">
95 </i>
96 </a>
97 <button type="button" class="btn btn-danger btn-circle btn-sm" data-toggle="modal" data-target="#DeleteAnimal-@item.ID_D
98 href="@Url.Action("Delete", "Descartes", new { id = item.ID_Descartes })">
99 <i class="fas fa-trash">
100 </i>
101 </button>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>

```

Figura 117. Código fuente de la aplicación.

La vista descartes muestra registros de animales descartados con opciones para agregar, editar, ver detalles y eliminar. Incluye un diseño con tarjeta, botones de acción y una tabla de datos. Las imágenes asociadas a cada registro se cargan desde el controlador, y se utilizan vistas parciales para la lógica de eliminación.

```

131 @section scripts{
132
133 <script type="text/javascript">
134
135 $(document).ready(function () {
136
137     var table = $('#dataTable').DataTable({
138         orderCellsTop: true,
139         fixedHeader: true
140     });
141
142
143
144     //Creamos una fila en el head de la tabla y lo clonamos para cada columna
145     $('#dataTable thead tr').clone(true).appendTo('#dataTable thead');
146
147     $('#dataTable thead tr:eq(1) th').each(function (i) {
148         var title = $(this).text(); //es el nombre de la columna
149         $(this).html('<input type="text" style="width: 100px;" placeholder="Search." />');
150
151         $('input', this).on('keyup change', function () {
152             if (table.column(i).search() !== this.value) {
153                 table
154                     .column(i)
155                     .search(this.value)
156                     .draw();
157             }
158         });
159     });
160 });
161
162
163 </script>

```

Figura 118. Código fuente de la aplicación.

Este script utiliza DataTables, una biblioteca de jQuery, para agregar funcionalidades de búsqueda a una tabla HTML. Clona la fila del encabezado para agregar celdas de búsqueda a cada columna y permite filtrar los datos de la tabla en función del texto ingresado en los campos de búsqueda.

```

7 namespace ProduAgro.Controllers.ProduAgroController
8 {
9     0 referencias
10     public class DescartesController : Controller
11     {
12         private ProduAgro_Entities db = new ProduAgro_Entities();
13
14         // GET: Descartes
15         0 referencias
16         public ActionResult Index()
17         {
18             var descartes = db.Descartes.Include(d => d.Animales);
19             return View(descartes.ToList());
20         }
21
22         // GET: Descartes/Details/5
23
24         0 referencias
25         public ActionResult Datos_Vet(int id)
26         {
27             Descartes descartes = db.Descartes.Find(id);
28             DateTime limiteFecha = descartes.Fecha_Descarte.AddDays(-30);
29             db = new ProduAgro_Entities();
30
31             Datos_Veterinarios datos_Veterinarios = db.Datos_Veterinarios.Include(m => m.Animales)
32                 .Where(m => m.Cod_Animal == descartes.Cod_Animal && m.Fecha_Tratam >= limiteFecha).
33                 OrderByDescending(m => m.Fecha_Tratam).FirstOrDefault();
34
35             return PartialView("Datos_VetPartial", datos_Veterinarios);
36         }
37     }
38 }

```

Figura 119. Código fuente de la aplicación.

```

47 public ActionResult Details(int? id)
48 {
49     if (id == null)
50     {
51         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
52     }
53     Descartes descartes = db.Descartes.Find(id);
54     Pesos pesos = db.Pesos.Include(m => m.Animales).where(m => m.Cod_Animal == id).OrderByDescending(m => m.Fecha_peso).FirstOrDefault();
55
56     var promedioLeche = (from prod in db.Medida_Leche.Where(x => x.Cod_Animal.Equals(descartes.Cod_Animal))
57                         select prod.Medicion_Mañana)
58                         .Average();
59
60     var promedioLecheT = (from prod in db.Medida_Leche.Where(x => x.Cod_Animal.Equals(descartes.Cod_Animal))
61                          select prod.Medicion_Tarde)
62                          .Average();
63
64     int promedioI;
65
66     if (promedioLeche == null || promedioLecheT == null)
67     {
68         promedioI = 0;
69     }
70     else
71     {
72         promedioI = (int)(promedioLecheT + promedioLeche) / 2;
73     }
74
75     ViewBag.promedio = promedioI;
76
77     if (pesos != null)
78     {

```

Figura 120. Código fuente de la aplicación.

```

94 public ActionResult Create()
95 {
96     var animales = from c in db.Animales
97                   where c.Estado_animal == 1
98                   orderby c.ID_Num descending
99                   select c
100
101     ViewBag.Cod_Animal = new SelectList(animales, "COD_ANIMALES", "ID_Num");
102     return View();
103 }
104
105 // POST: Descartes/Create
106 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que quiere enlazarse. Para obtener
107 // más detalles, vea https://go.microsoft.com/fwlink/?linkid=317598.
108 [HttpPost]
109 [ValidateAntiForgeryToken]
110 public ActionResult Create([Bind(Include = "ID_Descartes,Cod_Animal,Fecha_Descarte,Partivo,Imagen")] Descartes descartes)
111 {
112     HttpPostedFileBase fileBase = Request.Files[0];
113
114     if (fileBase.ContentLength == 0)
115     {
116         ModelState.AddModelError("Imagen", "Es necesario seleccionar una imagen.");
117     }
118     else
119     {
120         if (fileBase.FileName.EndsWith(".jpg") || fileBase.FileName.EndsWith(".jpeg"))
121         {
122             WebImage image = new WebImage(fileBase.InputStream);
123             descartes.Imagen = image.GetBytes();
124         }

```

Figura 121. Código fuente de la aplicación.

```

191 public ActionResult Edit([Bind(Include = "ID_Descartes,Cod_Animal,Fecha_Descarte,Motivo,Imagen")] Descartes descartes)
192
193
194 Descartes _descartes = new Descartes();
195
196 HttpPostedFileBase FileBase = Request.Files[0];
197
198 if (FileBase.ContentLength == 0)
199 {
200     _descartes = db.Descartes.Find(descartes.ID_Descartes);
201     descartes.Imagen = _descartes.Imagen;
202 }
203 else
204 {
205     if (FileBase.ContentType.StartsWith("image"))
206     {
207         WebImage image = new WebImage(FileBase.InputStream);
208         descartes.Imagen = image.GetBytes();
209     }
210     else
211     {
212         ModelState.AddModelError("Imagen", "El sistema solo acepta imágenes en formato JPEG.");
213     }
214 }
215
216 if (ModelState.IsValid)
217 {
218     db.Entry(_descartes).State = EntityState.Detached;
219     db.Entry(descartes).State = EntityState.Modified;
220
221     var query = (from a in db.Animales
222

```

Figura 122. Código fuente de la aplicación.

```

235
236 public ActionResult Delete(int? id)
237 {
238     if (id == null)
239     {
240         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
241     }
242     Descartes descartes = db.Descartes.Find(id);
243     if (descartes == null)
244     {
245         return HttpNotFound();
246     }
247     return View(descartes);
248 }
249
250 // POST: Descartes/Delete/5.
251
252 public ActionResult DeleteConfirmed(int id)
253 {
254     Descartes descartes = db.Descartes.Find(id);
255     db.Descartes.Remove(descartes);
256     db.SaveChanges();
257     return RedirectToAction("Index");
258 }
259
260 // Referencias
261 public ActionResult getImage(int id)
262 {
263     Descartes DescartesK = db.Descartes.Find(id);
264     byte[] byteImage = DescartesK.Imagen;
265
266     MemoryStream memoryStream = new MemoryStream(byteImage);
267     Image image = Image.FromStream(memoryStream);
268

```

Figura 123. Código fuente de la aplicación.

El controlador DescartesController gestiona operaciones CRUD para la entidad "Descartes" sus funciones principales incluyen mostrar, crear, editar y eliminar registros de descartes, así como gestionar datos veterinarios asociados, detalles de animales y operaciones con imágenes. Además, realiza validaciones y actualizaciones en la base de datos.



```

106 <script type="text/javascript">
107
108     $(document).ready(function () {
109
110         var table = $('#dataTable').DataTable({
111             orderCellsTop: true,
112             fixedHeader: true
113         });
114
115         //Creamos una fila en el head de la tabla y lo clonamos para cada columna
116         $('#dataTable thead tr').clone(true).appendTo('#dataTable thead');
117
118         $('#dataTable thead tr:eq(1) th').each(function (i) {
119             var title = $(this).text(); //es el nombre de la columna
120             $(this).html('<input type="text" style="width: 100px;" placeholder="Search.." />');
121
122             $('input', this).on('keyup change', function () {
123                 if (table.column(i).search() !== this.value) {
124                     table
125                         .column(i)
126                         .search(this.value)
127                         .draw();
128                 }
129             });
130         });
131     });
132
133 </script>
134
135
136
137

```

Figura 126. Código fuente de la aplicación.

La vista index de pesos nos ayudará a visualizar y gestionar registros de pesos de los animales de la hacienda ProduAgro, incluye botones para agregar y acceder a detalles, y una tabla que muestra información sobre los pesos de los animales, con la capacidad de ver detalles adicionales.

```

32 //Referencias
33 public ActionResult Create()
34 {
35     var animales_peso = from c in db.Animales
36                       join g in db.Grupos on c.Cod_grupo equals g.Id_grupo
37                       where g.Nombre_grupo == "Crianza" || g.Nombre_grupo == "Crianza 2" ||
38                             g.Nombre_grupo == "Destete" || g.Nombre_grupo == "Vacunas" ||
39                             g.Nombre_grupo == "Levante"
40                       orderby c.ID_Mas ascending
41                       select c;
42
43     ViewBag.Cod_Animal = new SelectList(animales_peso, "COD_ANIMALES", "ID_Rua");
44     return View();
45 }
46
47 // POST: Pesos/Create
48 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que quiere enlazarse. Para más
49 // detalles, vea http://go.microsoft.com/fwlink/?linkid=317398.
50 [HttpPost]
51 [ValidateAntiForgeryToken]
52 //Referencias
53 public ActionResult Create([Bind(Include = "ID_Peso,Cod_Animal,Fecha_peso,Peso_kg")] Pesos pesos)
54 {
55     if (ModelState.IsValid)
56     {
57         db.Pesos.Add(pesos);
58         db.SaveChanges();
59         return RedirectToAction("Index");
60     }
61
62     var animales_peso = from c in db.Animales
63                       join g in db.Grupos on c.Cod_grupo equals g.Id_grupo
64                       where g.Nombre_grupo == "Crianza" || g.Nombre_grupo == "Crianza 2" ||
65                             g.Nombre_grupo == "Destete" || g.Nombre_grupo == "Vacunas" ||

```

Figura 127. Código fuente de la aplicación.

```

120 public ActionResult Edit([Bind(Include = "ID_Peso,Cod_Animal,Fecha_peso,Peso_kg")] Pesos pesos)
121 {
122     if (ModelState.IsValid)
123     {
124         db.Entry(pesos).State = EntityState.Modified;
125         db.SaveChanges();
126         return RedirectToAction("Index");
127     }
128     var animales_peso = from c in db.Animales
129                        join g in db.Grupos on c.Cod_Grupo equals g.Id_Grupo
130                        where g.Nombre_Grupo == "Crianza" || g.Nombre_Grupo == "Crianza 2" ||
131                               g.Nombre_Grupo == "Destete" || g.Nombre_Grupo == "Vaconas" ||
132                               g.Nombre_Grupo == "Levante"
133                        orderby c.ID_Num ascending
134                        select c;
135
136     ViewBag.Cod_Animal = new SelectList(animales_peso, "COD_ANIMALES", "ID_Num", pesos.Cod_Animal);
137     return View(pesos);
138 }

```

**Figura 128.** Código fuente de la aplicación.

```

141 public ActionResult Delete(int? id)
142 {
143     if (id == null)
144     {
145         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
146     }
147     Pesos pesos = db.Pesos.Where(p => p.ID_Peso == id).FirstOrDefault();
148     if (pesos == null)
149     {
150         return HttpNotFound();
151     }
152     return PartialView("_DeletePesoPartialView", pesos);
153 }
154
155 // POST: Pesos/Delete/5
156
157 Referencias
158 public ActionResult DeleteConfirmed(int id)
159 {
160     int cod = db.Pesos.Where(p => p.ID_Peso == id).FirstOrDefault().Cod_Animal;
161     Pesos pesos = db.Pesos.Find(id);
162
163     db.Pesos.Remove(pesos);
164     db.SaveChanges();
165     return RedirectToAction("Details", new { id = cod});
166 }
167
168 Referencias
169 protected override void Dispose(bool disposing)
170 {
171     if (disposing)
172     {
173         db.Dispose();
174     }
175 }

```

**Figura 129.** Código fuente de la aplicación.

El controlador PesosController maneja las operaciones relacionadas con el registro de pesos de animales de la hacienda ProduAgro tomando en cuenta las acciones de mostrar una lista de pesos, detalles específicos de los pesos de un animal, crear nuevos registros de pesos, editar y eliminar registros existentes.

## MVC Gestión Ganadera- Datos Veterinarios

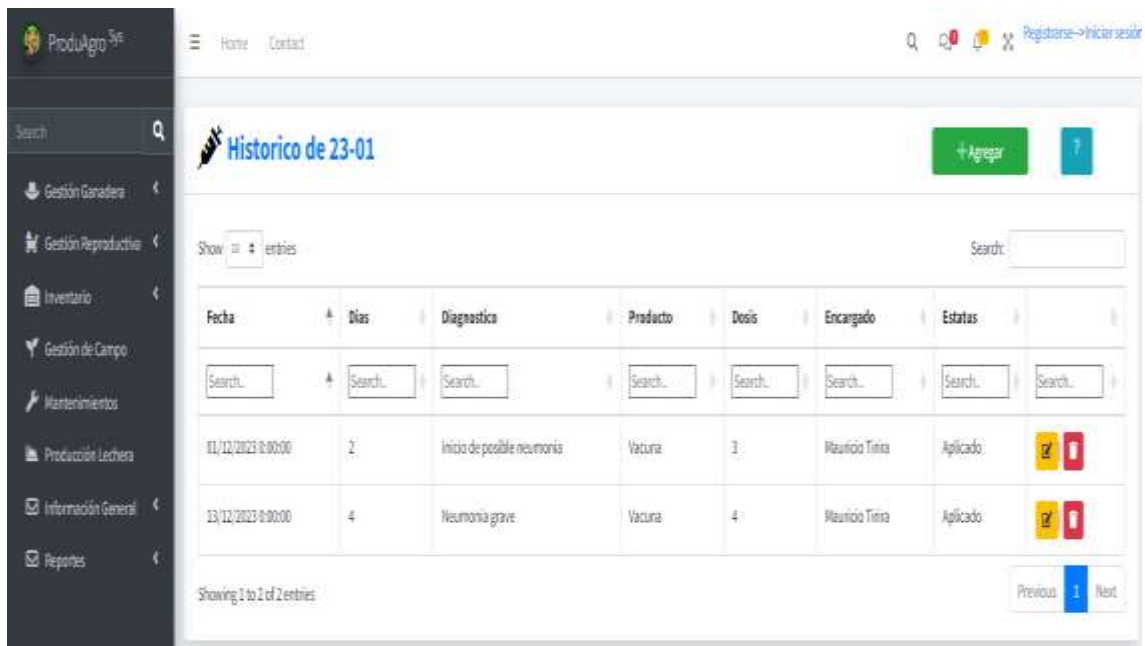


Figura 130. Diseño de interfaz de la vista de Datos Veterinarios de la aplicación.



Figura 131. Código fuente de la aplicación.

```

47 <div class="card-body">
48 <div class="table-responsive">
49 <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
50 <thead>
51 <tr>
52 <th>
53     Numero
54 </th>
55 <th>
56     Ultimo Tratamiento
57 </th>
58 <th>
59     Producto
60 </th>
61 <th>
62     Diagnostico
63 </th>
64 <th></th>
65 </tr>
66 </thead>
67 </thead>
68 <tbody>
69 <@foreach (var item in Model)
70 <tr>
71 <td>
72 <@Html.DisplayFor(modelItem => item.Num)
73 </td>
74 <td>
75 <@Html.DisplayFor(modelItem => item.Fecha_max)
76 </td>
77 <td>
78 <@Html.DisplayFor(modelItem => item.Producto)
79 </td>
80 </tr>
81 </tbody>
82 </table>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>

```

Figura 132. Código fuente de la aplicación.

```

115 @section scripts{
116 <script type="text/javascript">
117
118     $(document).ready(function () {
119
120         var table = $('#dataTable').DataTable({
121             orderCellsTop: true,
122             fixedHeader: true
123         });
124
125         //Creamos una fila en el head de la tabla y lo clonamos para cada columna
126         $('#dataTable thead tr').clone(true).appendTo('#dataTable thead');
127
128         $('#dataTable thead tr:eq(1) th').each(function (i) {
129             var title = $(this).text(); //es el nombre de la columna
130             $(this).html('<input type="text" style="width: 100px;" placeholder="Search.." />');
131
132             $('input', this).on('keyup change', function () {
133                 if (table.column(i).search() !== this.value) {
134                     table
135                         .column(i)
136                         .search(this.value)
137                         .draw();
138                 }
139             });
140         });
141     });
142 </script>
143 </section>
144 </div>
145 </div>

```

Figura 133. Código fuente de la aplicación.

La vista DatosVeterinarios muestra información sobre datos veterinarios con botones de acción. Además, presenta una tabla que contiene detalles específicos de los datos veterinarios, como el número, la fecha del último tratamiento, el producto utilizado y el diagnóstico. También, se incluyen botones para agregar nuevos registros y ver detalles adicionales.

```

32 public ActionResult Details(int? id)
33 {
34     if (id == null)
35     {
36         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
37     }
38     db = new ProduAgro_Entities();
39     List<Datos_Veterinarios> datos_Veterinarios = db.Datos_Veterinarios.Where(m => m.Cod_Animal == id).ToList();
40
41     var ext = (from ex in db.Datos_Veterinarios where ex.Cod_Animal == id select ex).First().Animales.ID_Num;
42     ViewBag.Nombre = ext;
43     if (datos_Veterinarios == null)
44     {
45         return HttpNotFound();
46     }
47     return View(datos_Veterinarios);
48 }
49
50 // GET: Datos_Veterinarios/Create
51 public ActionResult Create()
52 {
53     ViewBag.Cod_Animal = new SelectList(db.Animales, "COD_ANIMALES", "ID_Num");
54     ViewBag.Estatus = new SelectList(db.Estatus_Vet, "ID_Estatus_Vet", "Estatus");
55     ViewBag.Producto = new SelectList(db.Producto, "ID_Producto", "Nombre_P");
56     ViewBag.Cod_Encargado = new SelectList(db.Recursos_Humanos, "ID_RecursoH", "Nombre");
57     return View();
58 }

```

Figura 134. Código fuente de la aplicación.

```

82 public ActionResult Edit(int? id)
83 {
84     if (id == null)
85     {
86         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
87     }
88     Datos_Veterinarios datos_Veterinarios = db.Datos_Veterinarios.Find(id);
89     if (datos_Veterinarios == null)
90     {
91         return HttpNotFound();
92     }
93     ViewBag.Cod_Animal = new SelectList(db.Animales, "COD_ANIMALES", "ID_Num", datos_Veterinarios.Cod_Animal);
94     ViewBag.Estatus = new SelectList(db.Estatus_Vet, "ID_Estatus_Vet", "Estatus", datos_Veterinarios.Estatus);
95     ViewBag.Producto = new SelectList(db.Producto, "ID_Producto", "Nombre_P", datos_Veterinarios.Producto);
96     ViewBag.Cod_Encargado = new SelectList(db.Recursos_Humanos, "ID_RecursoH", "Nombre", datos_Veterinarios.Cod_Encargado);
97     return View(datos_Veterinarios);
98 }
99
100 // POST: Datos_Veterinarios/Edit/5
101 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que quiere enlazarse. Para obtener
102 // más detalles, vea https://go.microsoft.com/fwlink/?linkid=117396.
103 [HttpPost]
104 [ValidateAntiForgeryToken]
105 public ActionResult Edit([Bind(Include = "ID_Vete,Cod_Animal,Fecha_Tratam,Dias_Tratam,Diagnostico,Producto,Dosis,Cod_Encargado,Est
106 {
107     if (ModelState.IsValid)
108     {
109         db.Entry(datos_Veterinarios).State = EntityState.Modified;
110         db.SaveChanges();
111         return RedirectToAction("Index");
112     }
113 }

```

Figura 135. Código fuente de la aplicación.

```

121 public ActionResult Delete(int? id)
122 {
123     if (id == null)
124     {
125         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
126     }
127     Datos_Veterinarios datos_Veterinarios = db.Datos_Veterinarios.Where(p => p.ID_DVete == id).FirstOrDefault();
128
129     if (datos_Veterinarios == null)
130     {
131         return HttpNotFound();
132     }
133     return PartialView("_DeleteTratamientoPartial", datos_Veterinarios);
134 }
135
136 // POST: Datos_Veterinarios/Delete/5
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figura 136. Código fuente de la aplicación.

Aquí en el controlador de los datos veterinarios maneja las acciones que involucra el CRUD como agregar, eliminar, detalles de la información ingresada, de tal manera sea visible en la vista del mismo.

**MVC Gestión Ganadera- Traslados**

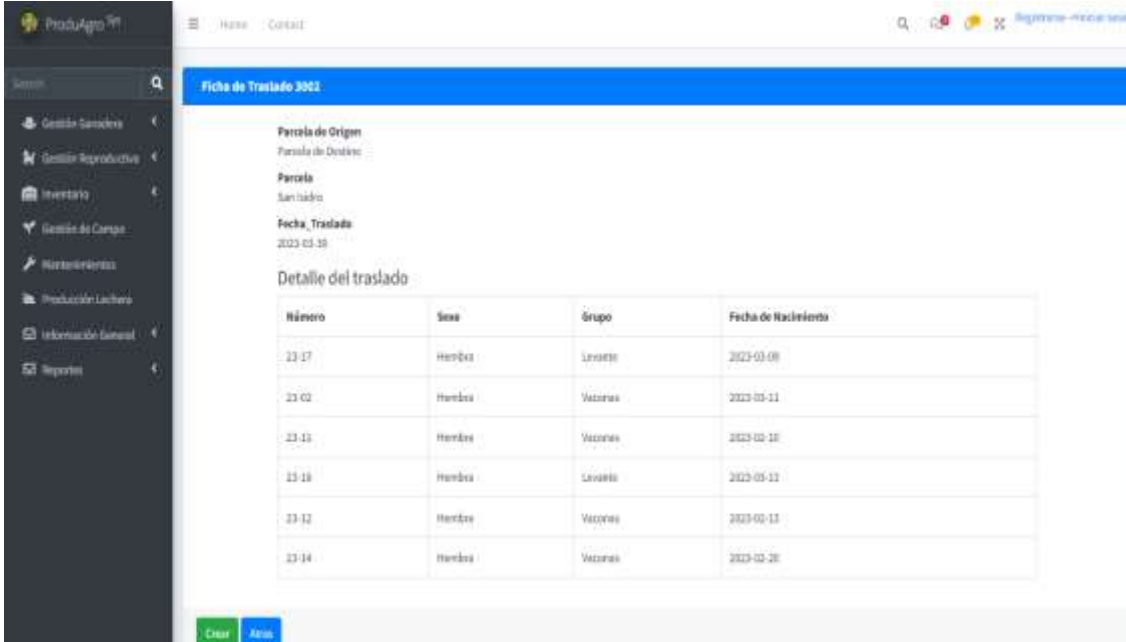


Figura 137. Diseño de interfaz de la vista Traslados de la aplicación.

```

8 <div class="card shadow mb-4">
9 <div class="card-header py-3 d-flex mb-2">
10 <div class="col-lg-3">
11 <div class="row">
12 <a class="d-none d-sm-inline-block card-image col-lg-3">
13 <i class="fas fa-truck" style="font-size:36px;color:black"></i>
14 </a>
15 <h3 class="m-0 font-weight-bold col-lg-9 align-middle text-primary">Traslados</h3>
16 </div>
17 </div>
18 </div>
19 </div>
20
21 <div class="col-lg-4"></div>
22 <div class="col-lg-5">
23 <div class="row">
24
25 <label class="col-sm-6 col-form-label"> </label>
26
27 <a placement="bottom" href="{url.Action("NewTraslado", "Traslados")}" class="m-0 btn btn-success btn-icon-split col-lg-3">
28 <span class="icon text-white-50">
29 <i class="fas fa-plus"></i>
30 </span>
31 <span class="text">Agregar</span>
32 </a>
33
34 <label class="col-lg-1"></label>
35
36 <a placement="left" class="d-none d-sm-inline-block btn btn-sm btn-info shadow-sm col-lg-1">
37 <i _ngcontent-tss-c162"" class="fas fa-question fa-sm text-white-50"></i>
38 </a>
39
40

```

Figura 138. Código fuente de la aplicación.

```

76 <td style="text-align:center">
77 
78 </td>
79 <td>
80 <@Html.DisplayFor(modelItem => item.Id_Traslados)
81 </td>
82
83 <td>
84 <@Html.DisplayFor(modelItem => item.Fecha_Traslado)
85 </td>
86
87 <td>
88 <@Html.DisplayFor(modelItem => item.Parcelas.Nombre)
89 </td>
90
91 <td>
92 <@Html.DisplayFor(modelItem => item.Parcelas1.Nombre)
93 </td>
94 <td>
95 <div class="input-group btn">
96 <a href="{url.Action("Edit", "Traslados", new { id = item.Id_Traslados })}" class="btn btn-warning btn-circle btn-sm">
97 <i class="fas fa-edit">
98 </i>
99 </a>
100
101 <a href="{url.Action("Detalle", "Traslados", new { id = item.Id_Traslados })}" class="btn btn-info btn-circle btn-sm">
102 <i class="fas fa-info-circle">
103 </i>
104 </a>
105
106 <button type="button" class="btn btn-danger btn-circle btn-sm" data-toggle="modal" data-target="{url.Action("DetalleModal", "Traslados", new { id = item.Id_Traslados })}">
107 </button>
108

```

Figura 139. Código fuente de la aplicación.

```

142 @section scripts{
143 <script type="text/javascript">
144
145     $(document).ready(function () {
146
147         $(function () {
148             $("#dataTable tbody img").click(function () {
149                 var id_pedido = $(this).attr("id_pedido");
150                 var fila = $("#fila-" + id_pedido);
151                 if (fila.css("display") == "none") {
152                     fila.css("display", "table-row");
153                     $(this).attr("src", "@Url.Content("~/Content/Theme/img/minus.png")");
154                 } else {
155                     fila.css("display", "none");
156                     $(this).attr("src", "@Url.Content("~/Content/Theme/img/plus.png")");
157                 }
158             });
159
160         });
161     });
162
163
164
165 });

```

Figura 140. Código fuente de la aplicación.

La vista traslados muestra información sobre traslados y botones de acción. Además, presenta una tabla que contiene detalles específicos de los traslados, como el código, la fecha, la parcela de origen y la parcela de destino. También, se incluyen botones para agregar nuevos traslados, editar, ver detalles adicionales y eliminar.

```

26 public ActionResult DetalleTraslado(int id_traslado)
27 {
28     List<Traslados_Animales> traslados_Animales = db.Traslados_Animales.Where(m => m.ID_Traslado == id_traslado).ToList();
29     return PartialView(traslados_Animales);
30 }
31
32 // Referencias
33 public ActionResult Detalles(int? id)
34 {
35     if (id == null)
36     {
37         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
38     }
39     Traslados traslados = db.Traslados.Find(id);
40     if (traslados == null)
41     {
42         return HttpNotFound();
43     }
44     return View(traslados);
45 }
46
47 // Referencias
48 public ActionResult Detalles(int id)
49 {
50     db = new ProduAgro_Entities();
51     List<Traslados_Animales> traslados_Animales = db.Traslados_Animales.Where(m => m.ID_Traslado == id).ToList();
52
53     return PartialView(traslados_Animales);
54 }
55
56 // GET: Traslados/Create

```

Figura 141. Código fuente de la aplicación.

```

54 public ActionResult Create()
55 {
56     ViewBag.Id_Parcela_Origen = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre");
57     ViewBag.Id_Parcela_Destino = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre");
58     ViewBag.Id_Parcela_Destino = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre");
59     return View();
60 }
61
62 // POST: Traslados/Create
63 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que quiere enlazarse. Para obtener
64 // más detalles, vea https://go.microsoft.com/fwlink/?LinkId=317598.
65 [HttpPost]
66 [ValidateAntiForgeryToken]
67 public ActionResult Create([Bind(Include = "Id_Traslados,fecha_Traslado,Id_Parcela_Origen,Id_Parcela_Destino")] Traslados traslados
68 {
69     if (ModelState.IsValid)
70     {
71         db.Traslados.Add(traslados);
72         db.SaveChanges();
73         return RedirectToAction("Index");
74     }
75
76     ViewBag.Id_Parcela_Origen = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre", traslados.Id_Parcela_Origen);
77     ViewBag.Id_Parcela_Destino = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre", traslados.Id_Parcela_Destino);
78     ViewBag.Id_Parcela_Destino = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre", traslados.Id_Parcela_Destino);
79     return View(traslados);
80 }

```

Figura 142. Código fuente de la aplicación.

```

83 public ActionResult Edit(int? id)
84 {
85     if (id == null)
86     {
87         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
88     }
89     Traslados traslados = db.Traslados.Find(id);
90     if (traslados == null)
91     {
92         return HttpNotFound();
93     }
94     ViewBag.Id_Parcela_Origen = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre", traslados.Id_Parcela_Origen);
95     ViewBag.Id_Parcela_Destino = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre", traslados.Id_Parcela_Destino);
96     ViewBag.Id_Parcela_Destino = new SelectList(db.Parcelas, "ID_Parcelsas", "Nombre", traslados.Id_Parcela_Destino);
97     return View(traslados);
98 }
99
100 // Referencias
101 public ActionResult DetalleT(int id)
102 {
103     db = new ProduAgro_Entities();
104     List<Traslados_Animales> traslados_Animales = db.Traslados_Animales.Where(m => m.ID_Traslado == id).ToList();
105
106     return PartialView(traslados_Animales);
107 }
108 }

```

Figura 143. Código fuente de la aplicación.

```

129 public ActionResult Delete(int? id)
130 {
131     if (id == null)
132     {
133         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
134     }
135     Traslados traslados = db.Traslados.Find(id);
136     if (traslados == null)
137     {
138         return HttpNotFound();
139     }
140     return PartialView("DeletePartialTraslado", traslados);
141 }
142
143 // POST: Traslados/Delete/5
144
145 @referencias
146 public ActionResult DeleteConfirmed(int id)
147 {
148     Traslados traslados = db.Traslados.Find(id);
149
150     if (traslados != null)
151     {
152         var trasladoDetalles = db.Traslados_Animales.Where(td => td.ID_Traslado == traslados.Id_Traslados);
153         foreach (var detalle in trasladoDetalles)
154         {
155             db.Traslados_Animales.Remove(detalle);
156         }
157     }
158 }

```

Figura 144. Código fuente de la aplicación.

El controlador traslados maneja operaciones CRUD para traslados y detalles de animales. Incluye funciones para mostrar, crear, editar y eliminar traslados, así como detalles específicos de animales trasladados. La lógica está diseñada para interactuar con una base de datos y proporcionar vistas correspondientes en una aplicación web MVC.

## MVC Gestión Reproductiva- Inseminaciones

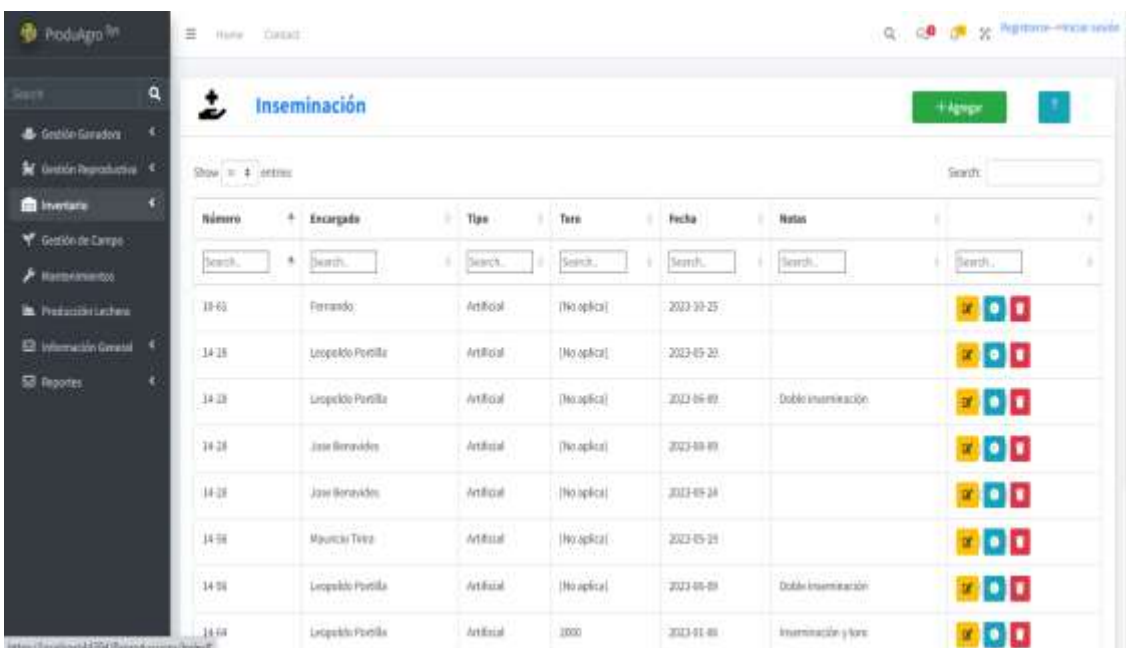


Figura 145. Diseño de interfaz de la vista Inseminaciones de la aplicación.

```

6  <div class="card shadow mb-4">
7  <div class="card-header py-3 d-flex mb-2">
8  <div class="col-lg-3">
9  <div class="row">
10 <a class="d-none d-sm-inline-block card-image col-lg-3">
11 <i class="fas fa-hand-holding-medical" style="font-size:36px;color:black;"></i>
12 </a>
13 <h3 class="h-0 font-weight-bold col-lg-9 align-middle text-primary">Inseminación</h3>
14 </div>
15 </div>
16 </div>
17 <div class="col-lg-4"></div>
18 <div class="col-lg-3">
19 <div class="row">
20 <label class="col-sm-6 col-form-label"></label>
21 <a placement="bottom" href="@Url.Action("Create","Reproducciones")" class="w-8 btn btn-success btn-icon-split col-lg-3">
22 <span class="icon text-white-50">
23 <i class="fas fa-plus"></i>
24 </span>
25 <span class="text">Agregar</span>
26 </a>
27 <label class="col-lg-1"></label>
28 <a placement="left" class="d-none d-sm-inline-block btn btn-sm btn-info shadow-sm col-lg-1">
29 <i _ngcontent-tss-c162="" class="fas fa-question fa-sm text-white-50"></i>
30 </a>
31 </div>
32 </div>
33 </div>
34 </div>
35 <div class="card-body">
36 <div class="table-responsive">
37 <table class="table table-bordered" id="dataTable" width="100%" cellpadding="0">
38 <thead>
39 <tr>
40 </tr>
41 </thead>
42 </table>
43 </div>
44 </div>

```

Figura 146. Código fuente de la aplicación.

```

62 <tbody>
63 <@foreach (var item in Model)
64 <tr>
65 <td>
66 <@Html.DisplayFor(modelItem => item.Animales.ID_Num)
67 </td>
68 <td>
69 <@Html.DisplayFor(modelItem => item.Recursos_Humanos.Nombre)
70 </td>
71 <td>
72 <@Html.DisplayFor(modelItem => item.Tipo_Fertilizacion.Nombre_Tipo)
73 </td>
74 <td>
75 <@Html.DisplayFor(modelItem => item.Toros.Num_Toro)
76 </td>
77 <td>
78 <@Html.DisplayFor(modelItem => item.Fecha_Repro)
79 </td>
80 <td>
81 <@Html.DisplayFor(modelItem => item.Notas)
82 </td>
83 </tr>
84 </tbody>

```

Figura 147. Código fuente de la aplicación.

```

<script type="text/javascript">
    $(document).ready(function () {
        var table = $('#dataTable').DataTable({
            orderCellsTop: true,
            fixedHeader: true
        });
        //Creamos una fila en el head de la tabla y lo clonamos para cada columna
        $('#dataTable thead tr').clone(true).appendTo('#dataTable thead');

        $('#dataTable thead tr:eq(1) th').each(function (i) {
            var title = $(this).text(); //es el nombre de la columna
            $(this).html('<input type="text" style="width: 100px;" placeholder="Search.." />');

            $('input', this).on('keyup change', function () {
                if (table.column(i).search() !== this.value) {
                    table
                        .column(i)
                        .search(this.value)
                        .draw();
                }
            });
        });
    });
});

```

Figura 148. Código fuente de la aplicación.

Este código de la vista Inseminaciones incluye opciones para agregar registros nuevos, y una tabla que muestra información detallada sobre inseminaciones existentes, como el número del animal, el encargado, el tipo de fertilización, el toro, la fecha, las notas, y acciones como editar, ver detalles y eliminar registros.

```

28 public ActionResult Details(int? id)
29 {
30     if (id == null)
31     {
32         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
33     }
34     Reproduccion reproduccion = db.Reproduccion.Find(id);
35     if (reproduccion == null)
36     {
37         return HttpNotFound();
38     }
39     return View(reproduccion);
40 }
41
42 // GET: Reproductions/Create
43 // Referencias
44 public ActionResult Create()
45 {
46     var animales = (from a in db.Animales
47                     join g in db.Grupos on a.Cod_grupo equals g.Id_grupo
48                     where g.Nombre_grupo == "Rejo" || g.Nombre_grupo == "Vaconas"
49                     || g.Nombre_grupo == "Seco"
50                     select a).ToList();
51
52     ViewBag.Cod_Animal = new SelectList(animales.OrderBy(p => p.ID_Num), "COD_ANIMALES", "ID_Num");
53     ViewBag.Cod_Encargado = new SelectList(db.Recursos_Humanos, "ID_RecursoH", "Nombre");
54     ViewBag.Cod_Tipo_Fertil = new SelectList(db.Tipo_Fertilizacion, "ID_Fertilizacion", "Nombre_Tipo");
55
56     ViewBag.Cod_Toro = new SelectList(db.Toros.OrderBy(p => p.Num_Toro), "Cod_toro", "Num_Toro");
57     return View();
58 }

```

Figura 149. Código fuente de la aplicación.

```

85 public ActionResult Imprimir(string encargado, string tipo)
86 {
87     var reproduccionQuery = db.Reproduccion.Include(r => r.Animales).Include(r => r.Recursos_Humanos).Include(r => r.Tipo_Fertiliza
88
89     // Aplicar filtros si se proporcionan
90     if (!string.IsNullOrEmpty(encargado))
91     {
92         reproduccionQuery = reproduccionQuery.Where(a => a.Recursos_Humanos.Nombre == encargado);
93     }
94
95     if (!string.IsNullOrEmpty(tipo))
96     {
97         reproduccionQuery = reproduccionQuery.Where(a => a.Tipo_Fertilizacion.Nombre_Tipo == tipo);
98     }
99
100     var reproduccion = reproduccionQuery.ToList();
101
102     return View(reproduccion);
103 }
104 // GET: Reproduccion/Imprimir

```

Figura 150. Código fuente de la aplicación.

```

85 public ActionResult Create([Bind(Include = "ID_Reproduccion,Cod_Animal,Fecha_Repro,Cod_Tipo_Ferti,Cod_Encargado,Cod_Toro,Notas")] R
86 {
87     if (ModelState.IsValid)
88     {
89         db.Reproduccion.Add(reproduccion);
90         db.SaveChanges();
91         return RedirectToAction("Index");
92     }
93
94     var animales = (from a in db.Animales
95                     join g in db.Grupos on a.Cod_Grupo equals g.Id_Grupo
96                     where g.Nombre_Grupo == "Rejo" || g.Nombre_Grupo == "Vacunas"
97                        || g.Nombre_Grupo == "Seco"
98                     select a).ToList();
99
100     ViewBag.Cod_Animal = new SelectList(animales.OrderBy(p => p.ID_Num), "COD_ANIMALES", "ID_Num", reproduccion.Cod_Animal);
101     ViewBag.Cod_Encargado = new SelectList(db.Recursos_Humanos, "ID_RecursosH", "Nombre", reproduccion.Cod_Encargado);
102     ViewBag.Cod_Tipo_Ferti = new SelectList(db.Tipo_Fertilizacion, "ID_TFertilizacion", "Nombre_Tipo", reproduccion.Cod_Tipo_Ferti);
103     ViewBag.Cod_Toro = new SelectList(db.Toros.OrderBy(p => p.Num_Toro), "Cod_toro", "Num_Toro");
104     return View(reproduccion);
105 }
106
107 // GET: Reproduccion/Edit/5
108
109 Referencias

```

Figura 151. Código fuente de la aplicación.

```

109 public ActionResult Edit(int? id)
110 {
111     if (id == null)
112     {
113         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
114     }
115     Reproduccion reproduccion = db.Reproduccion.Find(id);
116     if (reproduccion == null)
117     {
118         return HttpNotFound();
119     }
120     var animales = (from a in db.Animales
121                    join g in db.Grupos on a.Cod_Grupo equals g.Id_Grupo
122                    where g.Nombre_Grupo == "Rejo" || g.Nombre_Grupo == "Vaconas"
123                    || g.Nombre_Grupo == "Seco"
124                    select a).ToList();
125
126
127     ViewBag.Cod_Animal = new SelectList(animales.OrderBy(p => p.ID_Num), "COD_ANIMALES", "ID_Num", reproduccion.Cod_Animal);
128     ViewBag.Cod_Encargado = new SelectList(db.Recursos_Humanos, "ID_Recursos", "Nombre", reproduccion.Cod_Encargado);
129     ViewBag.Cod_Tipo_Ferti = new SelectList(db.Tipo_Fertilizacion, "ID_TFertilizacion", "Nombre_Tipo", reproduccion.Cod_Tipo_Ferti);
130
131     ViewBag.Cod_Toro = new SelectList(db.Toros.OrderBy(p => p.Num_Toro), "Cod_toro", "Num_Toro", reproduccion.Cod_Toro);
132     return View(reproduccion);
133 }

```

Figura 152. Código fuente de la aplicación.

```

164 public ActionResult Delete(int? id)
165 {
166     if (id == null)
167     {
168         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
169     }
170     Reproduccion reproduccion = db.Reproduccion.Find(id);
171     if (reproduccion == null)
172     {
173         return HttpNotFound();
174     }
175     return View(reproduccion);
176 }
177
178 // POST: Reproducciones/Delete/5
179 [HttpPost, ActionName("Delete")]
180 [ValidateAntiForgeryToken]
181 public ActionResult DeleteConfirmed(int id)
182 {
183     Reproduccion reproduccion = db.Reproduccion.Find(id);
184     db.Reproduccion.Remove(reproduccion);
185     db.SaveChanges();
186     return RedirectToAction("Index");
187 }
188

```

Figura 153. Código fuente de la aplicación.

El controlador ReproduccionesController maneja las operaciones CRUD para las reproducciones de animales, incluye acciones para mostrar, crear, editar y eliminar reproducciones, así como una acción para imprimir un informe filtrado. Utiliza un contexto de base de datos y modelos para interactuar con los datos relacionados con la reproducción de animales de la hacienda ProduAgro.



```

<div class="card-body">
  <div class="table-responsive">
    <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
      <thead>
        <tr>
          <th>
            @Html.DisplayNameFor(model => model.Reproduccion.Animales.ID_Num)
          </th>
          <th>
            Resultado
          </th>
          <th>
            @Html.DisplayNameFor(model => model.Fecha_Chequeo)
          </th>
          <th>
            Descripción
          </th>
          <th>
            @Html.DisplayNameFor(model => model.Tratamiento)
          </th>
          <th>
            Estado Corporal
          </th>
        </tr>
      </thead>
    </table>
  </div>
</div>

```

Figura 156. Código fuente de la aplicación.

```

<tbody>
  @foreach (var item in Model)
  {
    <tr>
      <td>
        @Html.DisplayFor(modelItem => item.Reproduccion.Animales.ID_Num)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.Resultado_Fertili.Nombre)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.Fecha_Chequeo)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.Descripcion)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.Tratamiento)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.Estado_Corporal)
      </td>
      <td>
        <div class="input-group btn">
          <a href="@Url.Action("Edit", "Cheques", new { id = item.ID_Cheque })" class="btn btn-warning btn-circle bt
            <i class="fas fa-edit">
          </i>
        </a>
          <a href="@Url.Action("Details", "Cheques", new { id = item.ID_Cheque })" class="btn btn-info btn-circle bt
            <i class="fas fa-info-circle">
          </i>
        </a>
        </div>
      </td>
    </tr>
  }
</tbody>

```

Figura 157. Código fuente de la aplicación.

La vista Cheques muestra información sobre chequeos veterinarios realizados a los animales de la hacienda ProduAgro, además, permite agregar nuevos chequeos, editar o ver detalles de los existentes, y eliminar registros. La tabla presenta detalles como el número de identificación del animal, resultado del chequeo, fecha, descripción, tratamiento y estado corporal. Además, se utiliza botones para las acciones de editar, ver detalles y eliminar.

```

40 public ActionResult Create()
41 {
42
43     var result = from r in db.Reproduccion
44                  join a in db.Animales on r.Cod_Animal equals a.COD_ANIMALES
45                  group r by a.ID_Num into g
46                  select new { ID_Num = g.Key, Cod_Reproduccion = g.Max(r => r.ID_Reproduccion) };
47
48     ViewBag.Cod_Reproduccion = new SelectList(result, "ID_Num", "ID_Num");
49     ViewBag.ID_Resultado = new SelectList(db.Resultado_Fertili, "ID_RFertilizacion", "Nombre");
50     return View();
51 }
52
53 O referencias
54 public ActionResult Imprimir(string resultado)
55 {
56     var chequeoQuery = db.Chequeo.Include(c => c.Reproduccion).Include(c => c.Resultado_Fertili);
57
58     if (!string.IsNullOrEmpty(resultado))
59     {
60         chequeoQuery = chequeoQuery.Where(a => a.Resultado_Fertili.Nombre == resultado);
61     }
62
63     var resultados = chequeoQuery.ToList();
64
65     return View(resultados);
66 }

```

Figura 158. Código fuente de la aplicación.

```

126 [HttpPost]
127 [ValidateAntiForgeryToken]
128 O referencias
129 public ActionResult Edit([Bind(Include = "ID_Chequeo,Cod_Reproduccion,Fecha_Chequeo,ID_Resultado,Descripcion,Tratamiento,Estado_Corporal",
130
131 {
132     if (ModelState.IsValid)
133     {
134         db.Entry(chequeo).State = EntityState.Modified;
135         db.SaveChanges();
136         return RedirectToAction("Index");
137     }
138     ViewBag.Cod_Reproduccion = new SelectList(db.Reproduccion, "ID_Reproduccion", "ID_Reproduccion", chequeo.Cod_Reproduccion);
139     ViewBag.ID_Resultado = new SelectList(db.Resultado_Fertili, "ID_RFertilizacion", "Nombre", chequeo.ID_Resultado);
140     return View(chequeo);
141 }
142
143 // GET: Chequeos/Delete/5
144 O referencias

```

Figura 159. Código fuente de la aplicación.

```

142 public ActionResult Delete(int? id)
143 {
144     if (id == null)
145     {
146         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
147     }
148     Chequeo chequeo = db.Chequeo.Find(id);
149     if (chequeo == null)
150     {
151         return HttpNotFound();
152     }
153     return View(chequeo);
154 }
155
156 // POST: Chequeos/Delete/5
157 [HttpPost, ActionName("Delete")]
158 [ValidateAntiForgeryToken]
159 public ActionResult DeleteConfirmed(int id)
160 {
161     Chequeo chequeo = db.Chequeo.Find(id);
162     db.Chequeo.Remove(chequeo);
163     db.SaveChanges();
164     return RedirectToAction("Index");
165 }
166

```

**Figura 160.** Código fuente de la aplicación.

El controlador de chequeos nos ayuda a gestionar información relacionada con chequeos veterinarios realizados a los animales de la hacienda ProduAgro también contiene las acciones de visualización de la lista de chequeos, detalles de un chequeo específico, creación de nuevos chequeos, edición y eliminación de registros existentes. Además, se proporciona una acción para imprimir los resultados de los chequeos filtrados por un resultado específico.

## MVC Gestión Reproductiva- Partos

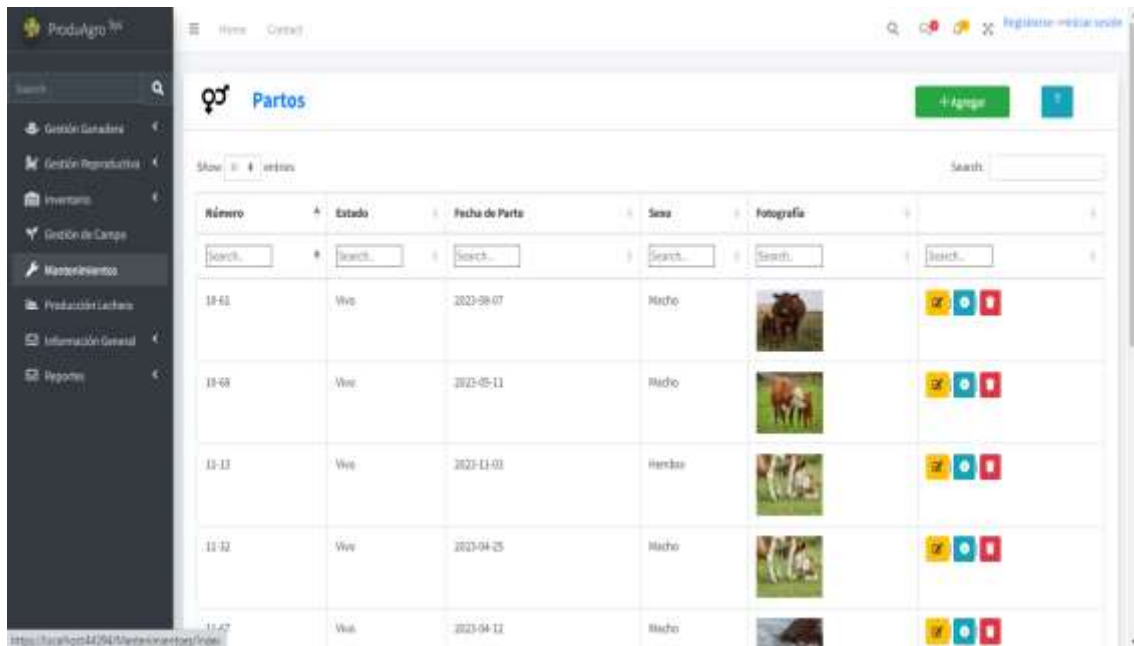


Figura 161. Diseño de interfaz de la vista Partos de la aplicación.

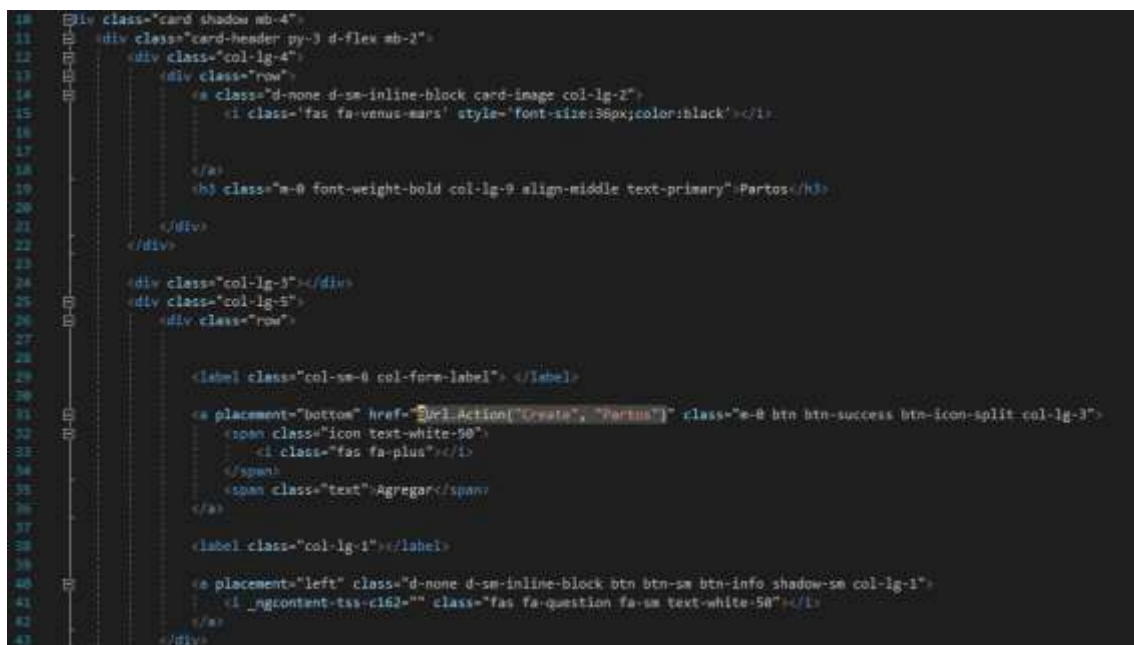


Figura 162. Código fuente de la aplicación.

```

73 | <tbody>
74 |   <tr>
75 |     <td>
76 |       <div class="input-group">
77 |         <input type="text" value="@Html.DisplayFor(modelItem => item.Animales.ID_Ana)" />
78 |       </div>
79 |     </td>
80 |     <td>
81 |       <div class="input-group">
82 |         <input type="text" value="@Html.DisplayFor(modelItem => item.Estado_Animall.Estado_Animall)" />
83 |       </div>
84 |     </td>
85 |     <td>
86 |       <div class="input-group">
87 |         <input type="text" value="@Html.DisplayFor(modelItem => item.Fecha_Part)" />
88 |       </div>
89 |     </td>
90 |     <td>
91 |       <div class="input-group">
92 |         <input type="text" value="@Html.DisplayFor(modelItem => item.Sexo1.Sexo1)" />
93 |       </div>
94 |     </td>
95 |     <td>
96 |       
97 |     </td>
98 |     <td>
99 |       <div class="input-group-btn">
100 |         <a href="@Url.Action("Edit", "Partos", new { id = item.Cod_Part })" class="btn btn-warning btn-circle btn-sm">
101 |           <i class="fas fa-edit">
102 |         </a>
103 |         <a href="@Url.Action("Details", "Partos", new { id = item.Cod_Part })" class="btn btn-info btn-circle btn-sm">
104 |           <i class="fas fa-info-circle">

```

**Figura 163.** Código fuente de la aplicación.

La vista del index de Partos ayuda a gestionar información relacionada con partos de los animales de la hacienda ProduAgro, se puede observar una lista de partos, detalles de un parto específico, creación de nuevos registros de partos, edición y eliminación de registros existentes. Además, se proporciona una acción para mostrar detalles específicos de un animal, incluyendo su identificación, estado, fecha de parto, sexo e imagen asociada. También se incluyen opciones para agregar, editar, ver detalles y eliminar registros de partos. La información de los partos se almacena en una base de datos, y se utiliza una imagen asociada a cada registro de parto.

```

47 public ActionResult Create()
48 {
49     var animales_p = from c in db.Animales
50                     join g in db.Grupos on c.Cod_Grupo equals g.Id_Grupo
51                     where g.Nombre_Grupo != "Muerto" && g.Nombre_Grupo != "Descartes"
52                        && g.Nombre_Grupo != "Crianza" && g.Nombre_Grupo != "Crianza"
53                        && g.Nombre_Grupo != "Levante"
54                     select c;
55
56     ViewBag.Cod_Animal = new SelectList(animales_p, "COD_ANIMALES", "ID_Num");
57     ViewBag.Estado_animal = new SelectList(db.Estado_Animal, "Cod_Estado", "Estado_Animal");
58     ViewBag.Sexo = new SelectList(db.Sexo, "ID_Sexo", "Sexo");
59     return View();
60 }
61
62 // Referencias
63 public ActionResult Imprimir(string estado, string sexo)
64 {
65     var partosQuery = db.Partos.Include(p => p.Animales).Include(p => p.Estado_Animal).Include(p => p.Sexo);
66
67     // Aplicar filtros si se proporcionan
68     if (!string.IsNullOrEmpty(estado))
69     {
70         partosQuery = partosQuery.Where(a => a.Estado_Animal.Estado_Animal == estado);
71     }
72
73     if (!string.IsNullOrEmpty(sexo))
74     {
75         partosQuery = partosQuery.Where(a => a.Sexo.Sexo == sexo);
76     }
77
78     var partos = partosQuery.ToList();
79 }

```

Figura 164. Código fuente de la aplicación.

```

130 public ActionResult Edit(int? id)
131 {
132     if (id == null)
133     {
134         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
135     }
136     Partos partos = db.Partos.Find(id);
137     if (partos == null)
138     {
139         return HttpNotFound();
140     }
141
142     var animales_p = from c in db.Animales
143                     join g in db.Grupos on c.Cod_Grupo equals g.Id_Grupo
144                     where g.Nombre_Grupo != "Muerto" && g.Nombre_Grupo != "Descartes"
145                        && g.Nombre_Grupo != "Crianza" && g.Nombre_Grupo != "Crianza"
146                        && g.Nombre_Grupo != "Levante"
147                     select c;
148
149     ViewBag.Cod_Animal = new SelectList(animales_p, "COD_ANIMALES", "ID_Num", partos.Cod_Animal);
150     ViewBag.Estado_animal = new SelectList(db.Estado_Animal, "Cod_Estado", "Estado_Animal", partos.Estado_animal);
151     ViewBag.Sexo = new SelectList(db.Sexo, "ID_Sexo", "Sexo", partos.Sexo);
152     return View(partos);
153 }

```

Figura 165. Código fuente de la aplicación.

```

159 public ActionResult Edit([Bind(Include = "Cod_Partos,Cod_Animal,Fecha_Partos,Estado_animal,Sexo,Imagen")] Partos partos)
160 {
161     Partos _partos = new Partos();
162
163     HttpPostedFileBase FileBase = Request.Files[0];
164
165     if (FileBase.ContentLength == 0)
166     {
167         _partos = db.Partos.Find(partos.Cod_Partos);
168         partos.Imagen = _partos.Imagen;
169     }
170     else
171     {
172         if (FileBase.ContentType.StartsWith("image"))
173         {
174             WebImage image = new WebImage(FileBase.InputStream);
175             partos.Imagen = image.GetBytes();
176         }
177         else
178         {
179             ModelState.AddModelError("Imagen", "El sistema solo acepta imágenes en formato JPEG.");
180         }
181     }
182 }

```

**Figura 166.** Código fuente de la aplicación.

```

206 public ActionResult Delete(int? id)
207 {
208     if (id == null)
209     {
210         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
211     }
212     Partos partos = db.Partos.Find(id);
213     if (partos == null)
214     {
215         return HttpNotFound();
216     }
217     return View(partos);
218 }
219
220 // POST: Partos/Delete/5
221 [HttpPost, ActionName("Delete")]
222 [ValidateAntiForgeryToken]
223 public ActionResult DeleteConfirmed(int id)
224 {
225     Partos partos = db.Partos.Find(id);
226     db.Partos.Remove(partos);
227     db.SaveChanges();
228     return RedirectToAction("Index");
229 }
230
231

```

**Figura 167.** Código fuente de la aplicación.

El controlador de Partos gestiona información relacionada con partos de animales en la hacienda ProduAgro así mismo proporciona acciones para mostrar la lista de partos, detalles de un parto específico, crear nuevos registros de partos, editar y eliminar registros existentes. Además, se incluye una acción para imprimir la lista de partos con filtros opcionales por estado y sexo.



```

47: <div class="card-body">
48:   <div class="table-responsive">
49:     <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
50:       <thead>
51:         <tr>
52:           <th>
53:             Categoria
54:           </th>
55:           <th>
56:             @Html.DisplayNameFor(model => model.Medidas.Medida)
57:           </th>
58:           <th>
59:             @Html.DisplayNameFor(model => model.Nombre_P)
60:           </th>
61:           <th>
62:             @Html.DisplayNameFor(model => model.Cantidad)
63:           </th>
64:         </tr>
65:       </thead>
66:       <tbody>
67:         <tr>
68:           <td>
69:             @foreach (var item in Model)
70:             <tr>
71:               <td>
72:                 @Html.DisplayFor(modelItem => item.Categoria.Nombre_C)
73:               </td>
74:               <td>
75:                 @Html.DisplayFor(modelItem => item.Medidas.Medida)
76:               </td>
77:               <td>
78:                 @Html.DisplayNameFor(modelItem => item.Nombre_P)
79:               </td>
80:               <td>
81:                 @Html.DisplayNameFor(modelItem => item.Cantidad)
82:               </td>
83:             </tr>
84:           </td>
85:           <td>
86:             <div class="input-group-btn">
87:               <a href="@Url.Action("Edit", "Productos", new { id = item.ID_Producto })" class="btn btn-warning btn-circle btn-sm">
88:                 <i class="fas fa-edit">
89:                 </i>
90:               </a>
91:               <a href="@Url.Action("Edit", "Productos", new { id = item.ID_Producto })" class="btn btn-info btn-circle btn-sm">
92:                 <i class="fas fa-edit">
93:                 </i>
94:               </a>
95:               <button type="button" class="btn btn-danger btn-circle btn-sm" data-toggle="modal" data-target="#DeleteAnimal-+item.ID" href="@Url.Action("Delete", "Productos", new { id = item.ID_Producto })">
96:                 <i class="fas fa-trash">
97:                 </i>
98:               </button>
99:             </div>
100:             @Html.Partial("Delete", item)
101:           </td>
102:         </tr>
103:       </tbody>
104:     </table>
105:   </div>
106: </div>

```

Figura 170. Código fuente de la aplicación.

```

80:   @Html.DisplayNameFor(modelItem => item.Nombre_P)
81: </td>
82: </td>
83:   @Html.DisplayNameFor(modelItem => item.Cantidad)
84: </td>
85: </td>
86: </td>
87: </td>
88: <div class="input-group-btn">
89:   <a href="@Url.Action("Edit", "Productos", new { id = item.ID_Producto })" class="btn btn-warning btn-circle btn-sm">
90:     <i class="fas fa-edit">
91:     </i>
92:   </a>
93:   <a href="@Url.Action("Edit", "Productos", new { id = item.ID_Producto })" class="btn btn-info btn-circle btn-sm">
94:     <i class="fas fa-edit">
95:     </i>
96:   </a>
97:   <button type="button" class="btn btn-danger btn-circle btn-sm" data-toggle="modal" data-target="#DeleteAnimal-+item.ID" href="@Url.Action("Delete", "Productos", new { id = item.ID_Producto })">
98:     <i class="fas fa-trash">
99:     </i>
100:   </button>
101: </div>
102: @Html.Partial("Delete", item)
103: </td>
104: </tr>
105: </tbody>
106: </table>
107: </div>

```

Figura 171. Código fuente de la aplicación.

La vista de Productos y pedidos funcionan de igual manera con la misma estructura donde en la vista usa una interfaz realizada en HTML y Razor para gestionar productos. También incluye botones para agregar nuevos productos y un área para mostrar una tabla de productos con información detallada, como categoría, medidas, nombre, cantidad, y opciones para editar o eliminar cada producto. Además, se usa una vista parcial llamada "Delete" que se utiliza para confirmar la eliminación de un producto.

La funcionalidad principal incluye la visualización, creación, edición y eliminación de productos.

### Fase de pruebas

Esta etapa se incorporó en el estudio debido a su importancia dentro de la metodología XP para asegurar un avance sólido del sistema. Durante este periodo, se ejecutaron evaluaciones tales como la inserción de información a través del sitio web en un entorno de servidor local, la verificación de los procedimientos y capacidades clave del sistema, y la confirmación de que su operatividad coincidiera con las expectativas previstas. Asimismo, se efectuaron pruebas de aceptación con el administrador de la hacienda ProduAgro, los cuales culminaron satisfactoriamente en todas las situaciones planteadas.

### Historial de seguimiento de ejecución de pruebas

**Tabla 95.** Seguimiento de ejecución de pruebas.

Escenario	Nro.	Resultado espedo	Resultado de la Prueba	
<b>Acceso al sistema</b>	1	Ingreso correcto al sistema	Exitosa	
	2	Ingreso incorrecto al sistema	Exitosa	
<b>Control de Usuario</b>	3	Creación correcta de usuario	Exitosa	
	4	Creación incorrecta de Usuario	Exitosa	
<b>Administración de ganado</b>	5	Ingreso de un nuevo animal	Exitosa	
	6	Ingreso incorrecto de un nuevo animal	Exitosa	
	7	Ingreso correcto de la Medición Lechera de un animal	Exitosa	
	8	Ingreso incorrecto de la Medición Lechera de un animal	Exitosa	
	9	Ingreso correcto de una Muerte de un animal	Exitosa	
	10	Ingreso incorrecto de una Muerte de un animal	Exitosa	
	11	Ingreso correcto de Descarte de un animal	Exitosa	
	12	Ingreso incorrecto de Descarte de un animal	Exitosa	
	13	Ingreso correcto del peso de un animal	Exitosa	
	14	Ingreso incorrecto del peso de un animal	Exitosa	
	15	Ingreso correcto de un tratamiento	Exitosa	
	16	Ingreso incorrecto de un tratamiento	Exitosa	
	17	Ingreso correcto de un Traslado	Exitosa	
	18	Ingreso incorrecto de un Traslado	Exitosa	
	<b>Administración Reproductiva</b>	19	Ingreso correcto de una Inseminación	Exitosa
		20	Ingreso incorrecto de una Inseminación	Exitosa
		21	Ingreso correcto de Chequeo de un animal inseminado	Exitosa
		22	Ingreso correcto del Parto de un animal	Exitosa

	23	Ingreso incorrecto del Parto de un animal	Exitosa
	24	Ingreso correcto del Aborto de un animal	Exitosa
	25	Ingreso incorrecto del Aborto de un animal	Exitosa
<b>Mantenimiento</b>	26	Ingreso correcto del Mantenimiento a los equipos e instalaciones	Exitosa
	27	Ingreso correcto de Productos	Exitosa
<b>Inventario</b>	28	Ingreso incorrecto de Productos	Exitosa
	29	Ingreso correcto de Pedidos	Exitosa
	30	Ingreso incorrecto de Pedidos	Exitosa
	31	Ingreso correcto de un potrero	Exitosa
<b>Gestión de Campo</b>	32	Ingreso incorrecto de un potrero	Exitosa
	33	Ingreso correcto de una Acción realizada a un potrero	Exitosa
	34	Ingreso correcto de una Acción realizada a un potrero	Exitosa
<b>Producción de Leche</b>	35	Registro correcto de la Producción Diaria de Leche	Exitosa
	36	Registro incorrecto de la Producción Diaria de Leche	Exitosa

## PREUBAS DE ACEPTACION

**Tabla 96.** Prueba acceso al sistema de Administración.

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 1	<b>Nº Historia de Usuario:</b> 1
<b>Historia de Usuario:</b> Acceso al sistema de Administración	
<b>Nombre caso de prueba:</b> Ingreso correcto al sistema	
<b>Descripción:</b> Ingreso al sistema a través de un usuario y contraseña	
<b>Condiciones de ejecución:</b> El usuario debe tener credenciales correspondientes al sistema.	
<b>Entrada:</b>	
<ol style="list-style-type: none"> <li>1. El usuario debe ingresar su correo en el primer campo</li> <li>2. El usuario debe ingresar su contraseña en el segundo campo.</li> <li>3. El usuario debe dar clic en el botón de 'Login'</li> </ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>• Se muestra la página de inicio</li> </ul>	
<b>Evaluación:</b>	
El ingreso al sistema es correcto	

**Tabla 97.** Prueba Acceso incorrecto al sistema.

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 2	<b>Nº Historia de Usuario:</b> 1
<b>Historia de Usuario:</b> Acceso al sistema de Administración	
<b>Nombre caso de prueba:</b> Acceso incorrecto al sistema	
<b>Descripción:</b> Ingreso al sistema a través de un usuario y contraseña	
<b>Condiciones de ejecución:</b> El usuario debe tener credenciales correspondientes al sistema.	
<b>Entrada:</b>	
<ol style="list-style-type: none"> <li>1. El usuario no ingresa su correo electrónico</li> <li>2. El usuario no ingresa su contraseña</li> <li>3. El usuario ingresa correo electrónico o contraseña incorrectos</li> <li>4. El usuario no ingresa datos en ninguno de los campos</li> <li>5. El usuario debe dar clic en 'Login'</li> </ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>• El sistema vuelve a cargar con los campos de texto vacíos</li> <li>• El sistema le informa los datos incorrectos o faltantes</li> </ul>	

- 
- No se realiza en ingreso al sistema
- Evaluación:** El ingreso al sistema de con credenciales incorrectas no se realiza.
- 

**Tabla 98.** Prueba Creación correcta de usuario.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 3	<b>Nº Historia de Usuario:</b> 2
<b>Historia de Usuario:</b> Control de Usuario	
<b>Nombre caso de prueba:</b> Creación correcta de usuario	
<b>Descripción:</b> Para la creación de usuarios es necesario completar los campos de correo electrónico y contraseña	
<b>Condiciones de ejecución:</b> El administrador puede crear usuarios.	
<b>Entrada:</b>	
1. El administrador debe ingresar a 'Añadir Usuario'	
2. El administrador debe ingresar los datos necesarios para la creación de un usuario	
3. El usuario debe dar clic en 'Crear'	
<b>Resultado esperado:</b>	
• El usuario aparece en el Listado de Usuarios	
<b>Evaluación:</b> La creación de usuarios es correcta.	

---

**Tabla 99.** Prueba Creación incorrecta de usuario.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 4	<b>Nº Historia de Usuario:</b> 2
<b>Historia de Usuario:</b> Control de Usuario	
<b>Nombre caso de prueba:</b> Creación incorrecta de usuario	
<b>Descripción:</b> Para la creación de usuarios es necesario completar los campos de correo electrónico y contraseña	
<b>Condiciones de ejecución:</b> El administrador puede crear usuarios.	
<b>Entrada:</b>	
4. El administrador debe ingresar a 'Añadir Usuario'	
5. El administrador debe ingresar los datos necesarios para la creación de un usuario	
6. El usuario debe dar clic en 'Crear'	
<b>Resultado esperado:</b>	
• Se mostrarán los errores existentes en los campos de texto	
<b>Evaluación:</b> La creación de usuarios es incorrecta.	

---

**Tabla 100.** Prueba Ingreso de un nuevo animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 5	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración de ganado	
<b>Nombre caso de prueba:</b> Ingreso de un nuevo animal	
<b>Descripción:</b> Se realiza el ingreso de un nuevo animal con todos los datos correspondientes	
<b>Condiciones de ejecución:</b> El usuario es capaz de ingresar un nuevo animal a la nómina de animales.	
<b>Entrada:</b>	
1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Animales' y luego al botón 'Agregar'	
2. El usuario debe ingresar todos los datos correspondientes al animal	
3. El usuario debe dar clic en 'Crear'	
<b>Resultado esperado:</b>	
• Aparece una notificación de éxito en la creación.	
• El animal aparece en la Lista de Animales	
<b>Evaluación:</b> La creación de un animal es correcta	

---

**Tabla 101.** Prueba Ingreso incorrecto de un nuevo animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 6	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración de ganado	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de un nuevo animal	
<b>Descripción:</b> Se realiza el ingreso de un nuevo animal con todos los datos correspondientes	
<b>Condiciones de ejecución:</b> El usuario es capaz de ingresar un nuevo animal a la nómina de animales.	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Animales' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes al animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se señalan los campos donde existen errores.</li></ul>	
<b>Evaluación:</b> La creación de un animal de forma incorrecta no se realiza.	

---

**Tabla 102.** Prueba Ingreso correcto de la Medición Lechera de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 7	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración de ganado	
<b>Nombre caso de prueba:</b> Ingreso correcto de la Medición Lechera de un animal	
<b>Descripción:</b> Se realiza el ingreso de la medición lechera de un animal	
<b>Condiciones de ejecución:</b> El usuario es capaz de ingresar la medición de leche de un animal que está en producción.	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Medición Lechera' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes a la medición de leche</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• La medición aparece en el listado de medición del animal.</li></ul>	
<b>Evaluación:</b> El registro de la medición es correcta.	

---

**Tabla 103.** Prueba Ingreso incorrecto de la Medición Lechera de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 8	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración de ganado	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de la Medición Lechera de un animal	
<b>Descripción:</b> Se realiza el ingreso de la medición lechera de un animal	
<b>Condiciones de ejecución:</b> El usuario es capaz de ingresar la medición de leche de un animal que está en producción.	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Medición Lechera' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes a la medición de leche</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se señalan los campos donde existen errores.</li></ul>	
<b>Evaluación:</b> El registro de la medición de forma incorrecta no se realiza.	

---

**Tabla 104.** Prueba Ingreso correcto de una Muerte de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 9	<b>N° Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso correcto de una Muerte de un animal	
<b>Descripción:</b> Se realiza el ingreso a la muerte correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar la muerte de un animal con pruebas fotográficas.	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Muertes' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes a la Muerte del animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• El animal aparece en el listado Muertes.</li><li>• Se actualiza su estado en la nomina</li></ul>	
<b>Evaluación:</b> El registro de la muerte es correcto	

---

**Tabla 105.** Prueba Ingreso incorrecto de una Muerte de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 10	<b>N° Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de una Muerte de un animal	
<b>Descripción:</b> Se realiza el ingreso a la muerte correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar la muerte de un animal con pruebas fotográficas.	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Muertes' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes a la Muerte del animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se muestran los campos en los cuales existen errores</li></ul>	
<b>Evaluación:</b> El registro de la muerte es incorrecto	

---

**Tabla 106.** Prueba Ingreso correcto de Descarte de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 11	<b>N° Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso correcto de Descarte de un animal	
<b>Descripción:</b> Se realiza el ingreso de descarte correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar el descarte de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Descartes' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes al Descarte del animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• El animal aparece en el listado de Descartes</li><li>• Se actualiza su estado en la nomina</li></ul>	
<b>Evaluación:</b> El registro de la Descarte es correcto	

---

**Tabla 107.** Prueba Ingreso incorrecto de Descarte de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 12	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de Descarte de un animal	
<b>Descripción:</b> Se realiza el ingreso de descarte correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar el descarte de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Descartes' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes al Descarte del animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se muestran los campos en los cuales existen errores</li></ul>	
<b>Evaluación:</b> El registro de la Descarte es incorrecto	

---

**Tabla 108.** Prueba Ingreso correcto del peso de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 13	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso correcto del peso de un animal	
<b>Descripción:</b> Se realiza el ingreso del peso correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar el peso de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Pesos y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes al peso del animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• El peso aparece en el listado de Pesos correspondientes al animal</li></ul>	
<b>Evaluación:</b> El registro del peso es correcto	

---

**Tabla 109.** Prueba Ingreso incorrecto del peso de un animal.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 14	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso incorrecto del peso de un animal	
<b>Descripción:</b> Se realiza el ingreso del peso correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar el peso de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Pesos y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes al peso del animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se muestran los campos en los cuales existen errores.</li></ul>	
<b>Evaluación:</b> El registro del peso es incorrecto	

---

**Tabla 110.** Prueba Ingreso correcto de un tratamiento.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 15	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso correcto de un tratamiento	
<b>Descripción:</b> Se realiza el ingreso de un tratamiento correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar in tratamiento de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Datos Veterinarios' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes al tratamiento realizado en un animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• El tratamiento realizado aparece en el listado de tratamientos correspondiente al animal.</li></ul>	
<b>Evaluación:</b> El registro del Tratamiento es correcto	

---

**Tabla 111.** Prueba Ingreso incorrecto de un tratamiento.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 16	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de un tratamiento	
<b>Descripción:</b> Se realiza el ingreso de un tratamiento correspondiente de un animal.	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar in tratamiento de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Datos Veterinarios' y luego al botón 'Agregar'</li><li>2. El usuario no ingresa todos los datos correspondientes al tratamiento realizado en un animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se marcan los campos en los cuales existen errores.</li></ul>	
<b>Evaluación:</b> El registro del Tratamiento es incorrecto	

---

**Tabla 112.** Prueba Ingreso correcto de un Traslado.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 17	<b>Nº Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso correcto de un Traslado	
<b>Descripción:</b> Se realiza el ingreso de un traslado correspondiente de un animal o animales	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar un Traslado de animales entre parcelas.	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Traslados' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes al tratamiento realizado en un animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• Aparece el Traslado en la lista de Traslados</li><li>• El animal cambia de parcela en la nómina.</li></ul>	

---

---

**Evaluación:** El registro del Traslado es correcto

---

**Tabla 113.** Prueba Ingreso incorrecto de un Traslado.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 18	<b>N° Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración ganadera	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de un Traslado	
<b>Descripción:</b> Se realiza el ingreso de un traslado correspondiente de un animal o animales	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar un Traslado de animales entre parcelas.	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Ganadera' y dar clic en 'Traslados' y luego al botón 'Agregar'</li><li>2. El usuario no ingresa todos los datos correspondientes al tratamiento realizado en un animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se señalan los campos en los cuales existen errores.</li></ul>	
<b>Evaluación:</b> El registro del Traslado es incorrecto	

---

**Tabla 114.** Prueba Ingreso correcto de una Inseminación.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 19	<b>N° Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración Reproductiva	
<b>Nombre caso de prueba:</b> Ingreso correcto de una Inseminación	
<b>Descripción:</b> Se realiza el registro de la inseminación correspondiente a un animal	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar un la Inseminación de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Reproductiva' y dar clic en 'Inseminaciones' y luego al botón 'Agregar'</li><li>2. El usuario debe ingresar todos los datos correspondientes a la Inseminación realizado en un animal</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• El registro aparece en el listado de inseminaciones.</li></ul>	
<b>Evaluación:</b> El registro del Inseminación es correcto	

---

**Tabla 115.** Prueba Ingreso incorrecto de una Inseminación.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 20	<b>N° Historia de Usuario:</b> 3
<b>Historia de Usuario:</b> Administración Reproductiva	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de una Inseminación	
<b>Descripción:</b> Se realiza el registro de la inseminación correspondiente a un animal	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar un la Inseminación de un animal	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Reproductiva' y dar clic en 'Inseminaciones' y luego al botón 'Agregar'</li><li>2. El usuario no ingresa todos los datos correspondientes a la Inseminación realizado en un animal</li></ol>	

---

- 
3. El usuario debe dar clic en 'Crear'

**Resultado esperado:**

- Aparece una notificación de error en la creación.
- Se marcan los campos en los cuales existen errores.

**Evaluación:** El registro del Inseminación es incorrecto

---

**Tabla 116.** Prueba Ingreso correcto de Chequeo de un animal inseminado.

---

**PRUEBA DE ACEPTACIÓN**

---

**Número:** 21

**N° Historia de Usuario:** 3

**Historia de Usuario:** Administración Reproductiva

**Nombre caso de prueba:** Ingreso correcto de Chequeo de un animal inseminado

**Descripción:** Se realiza el registro del Chequeo correspondiente a un animal inseminado

**Condiciones de ejecución:** El usuario es capaz de registrar el Chequeo de un animal

**Entrada:**

1. El usuario debe ingresar a 'Gestión Reproductiva y dar clic en Chequeo y luego al botón 'Agregar'
2. El usuario ingresa todos los datos correspondientes al Chequeo realizado en un animal
3. El usuario debe dar clic en 'Crear'

**Resultado esperado:**

- Aparece una notificación de éxito en la creación.
- El registro aparece en el listado de chequeos

**Evaluación:** El registro del Chequeo es correcto

---

**Tabla 117.** Prueba Ingreso correcto del Parto de un animal.

---

**PRUEBA DE ACEPTACIÓN**

---

**Número:** 22

**N° Historia de Usuario:** 3

**Historia de Usuario:** Administración Reproductiva

**Nombre caso de prueba:** Ingreso correcto del Parto de un animal

**Descripción:** Se realiza el registro del Parto correspondiente a un animal

**Condiciones de ejecución:** El usuario es capaz de registrar el Parto de un animal

**Entrada:**

1. El usuario debe ingresar a 'Gestión Reproductiva y dar clic en 'Partos' y luego al botón 'Agregar'
2. El usuario ingresa todos los datos correspondientes al Parto realizado en un animal
3. El usuario debe dar clic en 'Crear'

**Resultado esperado:**

- Aparece una notificación de éxito en la creación.
- El registro aparece en el listado de Partos correspondiente al animal

**Evaluación:** El registro del Parto es correcto

---

**Tabla 118.** Prueba Ingreso incorrecto del Parto de un animal.

---

**PRUEBA DE ACEPTACIÓN**

---

**Número:** 23

**N° Historia de Usuario:** 3

**Historia de Usuario:** Administración Reproductiva

**Nombre caso de prueba:** Ingreso incorrecto del Parto de un animal

**Descripción:** Se realiza el registro del Parto correspondiente a un animal

**Condiciones de ejecución:** El usuario es capaz de registrar el Parto de un animal

**Entrada:**

1. El usuario debe ingresar a 'Gestión Reproductiva y dar clic en 'Partos' y luego al botón 'Agregar'
  2. El usuario no ingresa todos los datos correspondientes al Parto realizado en un animal
  3. El usuario debe dar clic en 'Crear'
-

---

**Resultado esperado:**

- Aparece una notificación de error en la creación.
- Los campos con errores son marcados

**Evaluación:** El registro del Parto es incorrecto

---

**Tabla 119.** Prueba Ingreso correcto del Aborto de un animal.

---

**PRUEBA DE ACEPTACIÓN**

---

**Número:** 24

**N° Historia de Usuario:** 3

**Historia de Usuario:** Administración Reproductiva

**Nombre caso de prueba:** Ingreso correcto del Aborto de un animal

**Descripción:** Se realiza el registro del Aborto correspondiente a un animal

**Condiciones de ejecución:** El usuario es capaz de registrar el Aborto de un animal

**Entrada:**

1. El usuario debe ingresar a 'Gestión Reproductiva y dar clic en 'Abortos' y luego al botón 'Agregar'
2. El usuario ingresa todos los datos correspondientes al Parto realizado en un animal
3. El usuario debe dar clic en 'Crear'

**Resultado esperado:**

- Aparece una notificación de éxito en la creación.
- El aborto aparece en el listado de abortos correspondiente al animal.

**Evaluación:** El registro del Aborto es correcto

---

**Tabla 120.** Prueba Ingreso incorrecto del Aborto de un animal.

---

**PRUEBA DE ACEPTACIÓN**

---

**Número:** 25

**N° Historia de Usuario:** 3

**Historia de Usuario:** Administración Reproductiva

**Nombre caso de prueba:** Ingreso incorrecto del Aborto de un animal

**Descripción:** Se realiza el registro del Aborto correspondiente a un animal

**Condiciones de ejecución:** El usuario es capaz de registrar el Aborto de un animal

**Entrada:**

1. El usuario debe ingresar a 'Gestión Reproductiva y dar clic en 'Abortos' y luego al botón 'Agregar'
2. El usuario no ingresa todos los datos correspondientes al Parto realizado en un animal
3. El usuario debe dar clic en 'Crear'

**Resultado esperado:**

- Aparece una notificación de error en la creación.
- Se marcan los campos en los cuales existen errores.

**Evaluación:** El registro del Aborto es incorrecto

---

**Tabla 121.** Prueba Ingreso correcto Mantenimiento de equipos e instalaciones.

---

**PRUEBA DE ACEPTACIÓN**

---

**Número:** 26

**N° Historia de Usuario:** 4

**Historia de Usuario:** Mantenimiento

**Nombre caso de prueba:** Ingreso correcto del Mantenimiento a los equipos e instalaciones

**Descripción:** Se realiza el registro del Mantenimiento correspondiente a un equipo o instalación

**Condiciones de ejecución:** El usuario es capaz de registrar el Mantenimiento realizado

**Entrada:**

1. El usuario debe ingresar a 'Mantenimiento' y dar clic al botón 'Agregar'
2. El usuario ingresa todos los datos correspondientes al Mantenimiento realizado
3. El usuario debe dar clic en 'Crear'

**Resultado esperado:**

---

- Aparece una notificación de éxito en la creación.
- El registro aparece en el listado de Mantenimientos

**Evaluación:** El registro del Mantenimiento es correcto

**Tabla 122.** Prueba Ingreso correcto de Productos.

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 27	<b>N° Historia de Usuario:</b> 5
<b>Historia de Usuario:</b> Inventario	
<b>Nombre caso de prueba:</b> Ingreso correcto de Productos	
<b>Descripción:</b> Se realiza el registro de los productos existentes en la Hacienda	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar los productos que existen en la hacienda.	
<b>Entrada:</b>	
<ol style="list-style-type: none"> <li>1. El usuario debe ingresar a 'Inventario y dar clic al botón 'Productos'</li> <li>2. El usuario ingresa todos los datos correspondientes al Producto</li> <li>3. El usuario debe dar clic en 'Crear'</li> </ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>• Aparece una notificación de éxito en la creación.</li> <li>• El registro aparece en el listado de Mantenimientos</li> </ul>	
<b>Evaluación:</b> El registro del Producto es correcto	

**Tabla 123.** Prueba Ingreso incorrecto de Productos.

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 28	<b>N° Historia de Usuario:</b> 5
<b>Historia de Usuario:</b> Inventario	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de Productos	
<b>Descripción:</b> Se realiza el registro de los productos existentes en la Hacienda	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar los productos que existen en la hacienda.	
<b>Entrada:</b>	
<ol style="list-style-type: none"> <li>1. El usuario debe ingresar a 'Inventario y dar clic al botón 'Productos'</li> <li>2. El usuario no ingresa todos los datos correspondientes al Producto</li> <li>3. El usuario debe dar clic en 'Crear'</li> </ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>• Aparece una notificación de error en la creación.</li> <li>• Se marcan los campos en los cuales existen errores.</li> </ul>	
<b>Evaluación:</b> El registro del Producto es incorrecto	

**Tabla 124.** Prueba Ingreso correcto de Pedidos.

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 29	<b>N° Historia de Usuario:</b> 5
<b>Historia de Usuario:</b> Inventario	
<b>Nombre caso de prueba:</b> Ingreso correcto de Pedidos	
<b>Descripción:</b> Se realiza el registro Pedidos para la hacienda	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar los pedidos que realiza la hacienda	
<b>Entrada:</b>	
<ol style="list-style-type: none"> <li>1. El usuario debe ingresar a 'Inventario y dar clic al botón 'Pedidos'</li> <li>2. El usuario ingresa todos los datos correspondientes al Pedido</li> <li>3. El usuario debe dar clic en 'Crear'</li> </ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>• Aparece una notificación de éxito en la creación.</li> <li>• El pedido aparece en el listado de Pedidos.</li> </ul>	

---

**Evaluación:** El registro del Pedido es correcto

---

**Tabla 125.** Prueba Ingreso incorrecto de Pedidos.

---

PRUEBA DE ACEPTACIÓN	
<b>Número:</b> 30	<b>N° Historia de Usuario:</b> 5
<b>Historia de Usuario:</b> Inventario	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de Pedidos	
<b>Descripción:</b> Se realiza el registro Pedidos para la hacienda	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar los pedidos que realiza la hacienda	
<b>Entrada:</b>	
4. El usuario debe ingresar a 'Inventario y dar clic al botón 'Pedidos'	
5. El usuario no ingresa todos los datos correspondientes al Pedido	
6. El usuario debe dar clic en 'Crear'	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se marcan los campos en los cuales existen errores.</li></ul>	
<b>Evaluación:</b> El registro del Pedido es incorrecto	

---

**Tabla 126.** Prueba Ingreso correcto de un potrero.

---

PRUEBA DE ACEPTACIÓN	
<b>Número:</b> 31	<b>N° Historia de Usuario:</b> 6
<b>Historia de Usuario:</b> Gestión de Campo	
<b>Nombre caso de prueba:</b> Ingreso correcto de un potrero	
<b>Descripción:</b> Se realiza la creación de un potrero	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar los potreros que conforman la hacienda	
<b>Entrada:</b>	
1. El usuario debe ingresar a 'Gestión Campo' y dar clic al botón 'Agregar'	
2. El usuario ingresa todos los datos correspondientes al Potrero	
3. El usuario debe dar clic en 'Crear'	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• El potrero aparece en el listado de Potreros.</li></ul>	
<b>Evaluación:</b> El registro del Potrero es correcto	

---

**Tabla 127.** Prueba Ingreso incorrecto de un potrero.

---

PRUEBA DE ACEPTACIÓN	
<b>Número:</b> 32	<b>N° Historia de Usuario:</b> 6
<b>Historia de Usuario:</b> Gestión de Campo	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de un potrero	
<b>Descripción:</b> Se realiza la creación de un potrero	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar los potreros que conforman la hacienda	
<b>Entrada:</b>	
1. El usuario debe ingresar a 'Gestión Campo' y dar clic al botón 'Agregar'	
2. El usuario no ingresa todos los datos correspondientes al Potrero	
3. El usuario debe dar clic en 'Crear'	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se marcan los campos en los cuales existen errores.</li></ul>	

---

---

**Evaluación:** El registro del Potrero es incorrecto

---

**Tabla 128.** Prueba Ingreso correcto de una Acción realizada a un potrero.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 33	<b>N° Historia de Usuario:</b> 6
<b>Historia de Usuario:</b> Gestión de Campo	
<b>Nombre caso de prueba:</b> Ingreso correcto de una Acción realizada a un potrero	
<b>Descripción:</b> Se realiza el registro de una acción realizada a un potrero	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar las acciones realizadas en un potrero	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Campo', seleccionar el Potrero, clic en Información y luego dar clic al botón Agregar</li><li>2. El usuario ingresa todos los datos correspondientes a la Acción</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• La acción aparece en el listado de Acciones realizadas al potrero.</li></ul>	
<b>Evaluación:</b> El registro de la Acción es correcto	

---

**Tabla 129.** Prueba Ingreso incorrecto de una Acción realizada a un potrero.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 34	<b>N° Historia de Usuario:</b> 6
<b>Historia de Usuario:</b> Gestión de Campo	
<b>Nombre caso de prueba:</b> Ingreso incorrecto de una Acción realizada a un potrero	
<b>Descripción:</b> Se realiza el registro de una acción realizada a un potrero	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar las acciones realizadas en un potrero	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Gestión Campo', seleccionar el Potrero, clic en Información y luego dar clic al botón Agregar</li><li>2. El usuario no ingresa todos los datos correspondientes a la Acción</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de error en la creación.</li><li>• Se muestran los campos en los cuales existen errores.</li></ul>	
<b>Evaluación:</b> El registro de la Acción es incorrecto	

---

**Tabla 130.** Prueba Registro correcto de la Producción Diaria de Leche.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 35	<b>N° Historia de Usuario:</b> 7
<b>Historia de Usuario:</b> Producción Lechera	
<b>Nombre caso de prueba:</b> Registro correcto de la Producción Diaria de Leche	
<b>Descripción:</b> Se realiza el registro diario de la Producción de Leche	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar la producción diaria de leche	
<b>Entrada:</b>	
<ol style="list-style-type: none"><li>1. El usuario debe ingresar a 'Producción Lechera 'clic en el botón de Agregar</li><li>2. El usuario ingresa todos los datos correspondientes a la Producción Diaria de Leche</li><li>3. El usuario debe dar clic en 'Crear'</li></ol>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"><li>• Aparece una notificación de éxito en la creación.</li><li>• El registro aparece en el listado de Producción de Leche</li></ul>	

---

---

**Evaluación:** El registro de la Producción de Leche es correcto

---

**Tabla 131.** Prueba Registro incorrecto de la Producción Diaria de Leche.

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b> 36	<b>N° Historia de Usuario:</b> 7
<b>Historia de Usuario:</b> Producción Lechera	
<b>Nombre caso de prueba:</b> Registro incorrecto de la Producción Diaria de Leche	
<b>Descripción:</b> Se realiza el registro diario de la Producción de Leche	
<b>Condiciones de ejecución:</b> El usuario es capaz de registrar la producción diaria de leche	
<b>Entrada:</b>	
1. El usuario debe ingresar a 'Producción Lechera' clic en el botón de Agregar	
2. El usuario no ingresa todos los datos correspondientes a la Producción Diaria de Leche	
3. El usuario debe dar clic en 'Crear'	
<b>Resultado esperado:</b>	
• Aparece una notificación de error en la creación.	
• Se marcan los campos con errores.	

---

**Evaluación:** El registro de la Producción de Leche es incorrecto

---

## Fichas técnicas

**Tabla 132.** Número de módulos

---

<b>N°</b>	<b>Denominación de módulo</b>	<b>Sub módulos</b>
1	Gestión Ganadera	8
2	Gestión Reproductiva	4
3	Inventario	2
4	Gestión de Campo	0
5	Mantenimientos	0
6	Producción Lechera	0
7	Información General	3
8	Reportes	5
	<b>Total</b>	<b>22</b>

---

Los valores mostrados en la tabla anterior muestran la cantidad de módulos que tiene el aplicativo informático desarrollado como parte de la investigación. De la misma forma se indica la cantidad de sub módulos que contiene cada módulo del sistema.

**Tabla 133.** Características del software

---

<b>Característica</b>	<b>Muy de acuerdo</b>	<b>De acuerdo</b>	<b>Ni de acuerdo ni en desacuerdo</b>	<b>En desacuerdo</b>	<b>Muy en desacuerdo</b>
-----------------------	-----------------------	-------------------	---------------------------------------	----------------------	--------------------------

---

Funcionabilidad	✓
Usabilidad	✓
Rendimiento	✓
Mantenibilidad	✓
Seguridad	✓
Fiabilidad	✓
Interoperabilidad	✓

### Matrices técnicas de funcionamiento de la aplicación

**Tabla 134.** Matriz de prueba de compatibilidad

Área de proceso	PC				
	Chrome	Firefox	Opera	Brave	Edge
Inicio de sesión	✓	✓	✓	✓	✓
Resumen estadístico general de la hacienda	✓	✓	✓	✓	✓
Creación de animal	✓	✓	✓	✓	✓
Consulta de información animal	✓	✓	✓	✓	✓
Registro de medición lechera	✓	✓	✓	✓	✓
Consulta de información medición lechera	✓	✓	✓	✓	✓
Registro de animal muerto	✓	✓	✓	✓	✓
Consulta de información de animal muerto	✓	✓	✓	✓	✓
Registro de animal descartado	✓	✓	✓	✓	✓
Consulta de información de animal descartado	✓	✓	✓	✓	✓
Registro de pesos de animales	✓	✓	✓	✓	✓
Consulta de información de pesos de animales	✓	✓	✓	✓	✓
Registro de toro	✓	✓	✓	✓	✓
Consulta de información de toro	✓	✓	✓	✓	✓
Registro de dato veterinario	✓	✓	✓	✓	✓
Consulta de información de datos veterinarios	✓	✓	✓	✓	✓
Registro de traslados	✓	✓	✓	✓	✓

Consulta de información de traslados	✓	✓	✓	✓	✓
Registro de inseminaciones	✓	✓	✓	✓	✓
Consulta de información de inseminaciones	✓	✓	✓	✓	✓
Registro de chequeos	✓	✓	✓	✓	✓
Consulta de información de chequeos	✓	✓	✓	✓	✓
Registro de partos	✓	✓	✓	✓	✓
Consulta de información de partos	✓	✓	✓	✓	✓
Registro de abortos	✓	✓	✓	✓	✓
Consulta de información de abortos	✓	✓	✓	✓	✓
Registro de producto	✓	✓	✓	✓	✓
Consulta de información de producto	✓	✓	✓	✓	✓
Registro de pedido	✓	✓	✓	✓	✓
Consulta de información de pedido	✓	✓	✓	✓	✓
Registro de potreros	✓	✓	✓	✓	✓
Consulta de estado de potreros	✓	✓	✓	✓	✓
Registro de mantenimiento	✓	✓	✓	✓	✓
Consulta de información de mantenimiento	✓	✓	✓	✓	✓
Registro de producción lechera	✓	✓	✓	✓	✓
Consulta estadística de producción lechera	✓	✓	✓	✓	✓
Registro de parcelas	✓	✓	✓	✓	✓
Registro de grupos	✓	✓	✓	✓	✓
Registro de recursos humanos	✓	✓	✓	✓	✓
Reporte animales	✓	✓	✓	✓	✓
Reporte inseminaciones	✓	✓	✓	✓	✓
Reporte chequeos	✓	✓	✓	✓	✓
Reporte partos	✓	✓	✓	✓	✓
Reporte abortos	✓	✓	✓	✓	✓

Los resultados que arrojó la matriz de despliegue indican que el uso del nuevo aplicativo con sus funciones, módulos y procesos es favorable. Esto se debe a que la

aplicación puede ser ejecutada desde cualquier computadora con conexión a internet, al igual que en cualquier navegador web que tenga instalado el respectivo dispositivo.

**Tabla 135.** Matriz de uso de memoria, CPU y uso de red

Área de proceso	Uso de memoria	CPU	Uso de red
Inicio de sesión	76,076 k	3,1	1,89 KB/s
Resumen estadístico general de la hacienda	79,61 k	3,1	1,37 KB/s
Creación de animal	78,43 k	4,1	4,5 KB/s
Consulta de información animal	84,18 k	1,2	5,9 KB/s
Registro de medición lechera	80,42 k	1,5	16,1 KB/S
Consulta de información medición lechera	80,72 k	1,6	5,9 KB/s
Registro de animal muerto	84,50 k	1,6	8,1 KB/s
Consulta de información de animal muerto	82,93 k	3,1	5,5 KB/s
Registro de animal descartado	82,42 k	1,6	6,4 KB/s
Consulta de información de animal descartado	82,28 k	1,5	2,24 KB/s
Registro de pesos de animales	82,70 k	1,5	5,6 KB/s
Consulta de información de pesos de animales	85,30 k	1,6	2,24 KB/s
Registro de toro	84,97 k	1,6	5,7 KB/s
Consulta de información de toro	98,52 k	1,5	7,7 KB/s
Registro de dato veterinario	95,68 k	3,1	5,1 KB/s
Consulta de información de datos veterinarios	86,56 k	1,6	6,5 KB/s
Registro de traslados	84,04 k	3,1	2,98 KB/s
Consulta de información de traslados	85,97 k	1,6	4,6 KB/s
Registro de inseminaciones	87,29 k	1,5	4,5 KB/s
Consulta de información de inseminaciones	85,22 k	1,6	4,4 KB/s
Registro de chequeos	76,076 k	3,1	1,89 KB/s
Consulta de información de chequeos	79,61 k	3,1	1,37 KB/s
Registro de partos	78,43 k	4,1	4,5 KB/s
Consulta de información de partos	84,18 k	1,2	5,9 KB/s
Registro de abortos	80,42 k	1,5	16,1 KB/S
Consulta de información de abortos	80,72 k	1,6	5,9 KB/s
Registro de producto	84,50 k	1,6	8,1 KB/s
Consulta de información de producto	82,93 k	3,1	5,5 KB/s
Registro de pedido	82,42 k	1,6	6,4 KB/s
Consulta de información de pedido	82,28 k	1,5	2,24 KB/s

Registro de potreros	82,70 k	1,5	5,6 KB/s
Consulta de estado de potreros	85,30 k	1,6	2,24 KB/s
Registro de mantenimiento	84,97 k	1,6	5,7 KB/s
Consulta de información de mantenimiento	98,52 k	1,5	7,7 KB/s
Registro de producción lechera	95,68 k	3,1	5,1 KB/s
Consulta estadística de producción lechera	86,56 k	1,6	6,5 KB/s
Registro de parcelas	84,04 k	3,1	2,98 KB/s
Registro de grupos	85,97 k	1,6	4,6 KB/s
Registro de recursos humanos	87,29 k	1,5	4,5 KB/s
Reporte animales	85,22 k	1,6	4,4 KB/s
Reporte inseminaciones	84,50 k	1,6	8,1 KB/s
Reporte chequeos	82,93 k	3,1	5,5 KB/s
Reporte partos	82,42 k	1,6	6,4 KB/s
Reporte abortos	82,28 k	1,5	2,24 KB/s

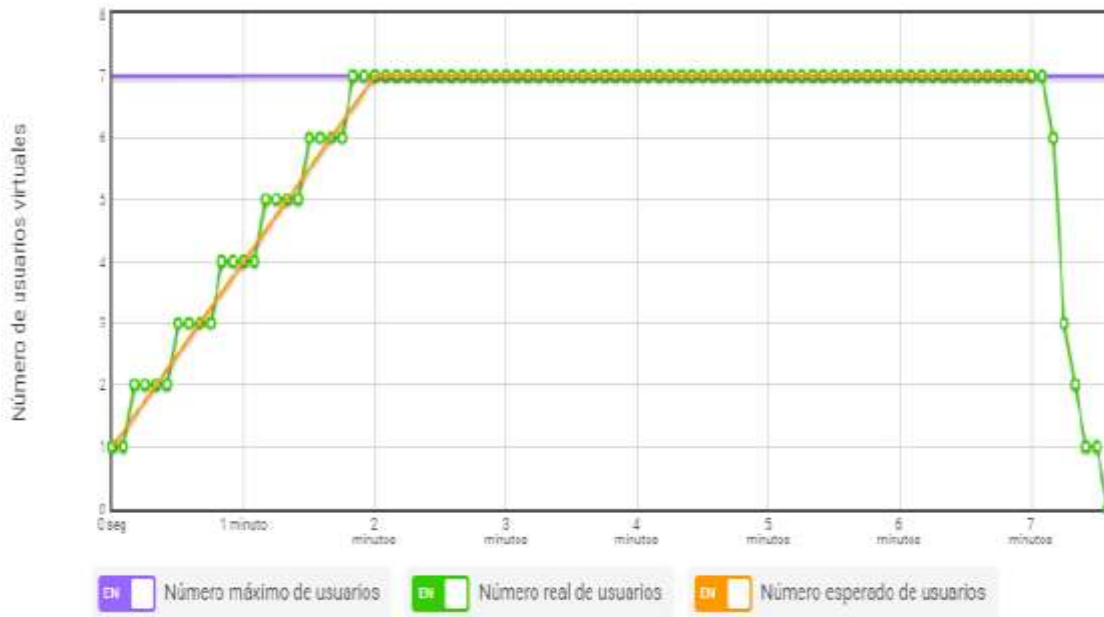
Los valores obtenidos en esta matriz indican que los recursos del dispositivo (uso de la memoria RAM y CPU), que necesita la aplicación para ejecutar sus procesos son adecuados. Además, el ancho de banda que emplea la aplicación es baja, lo cual evitaría la congestión de red. Por lo tanto, esto indica que el nuevo sistema informático está listo para poder usarse dentro de la hacienda ProduAgro si se lo requiere.

**Tabla 136.** Matriz de rendimiento

Área del proceso	Primera imagen con información	Índice de velocidad	Renderizado del mayor elemento con contenido	Tiempo para interacción	Tiempo total de bloqueo	Cambio de diseño acumulativo
Inicio de sesión	0,5 s	0,7 s	0,6 s	2,7 s	800 ms	0
Resumen estadístico	0,6 s	0,6 s	0,7 s	2,6 s	720 ms	0
Animales	0,6 s	0,6 s	0,6 s	2,5 s	610 ms	0
Medición lechera	0,6 s	0,8 s	0,7 s	2,6 s	720 ms	0
Muertes	0,6 s	0,7 s	0,7 s	2,5 s	680 ms	0
Descartes	0,5 s	0,5 s	0,7 s	2,7 s	870 ms	0
Pesos	0,5 s	0,8 s	0,7 s	2,8 s	830 ms	0
Toros	0,6 s	1,1 s	0,7 s	2,7 s	930 ms	0
Datos veterinarios	0,6 s	1,1 s	0,6 s	2,6 s	750 ms	0
Traslados	0,6 s	0,8 s	0,7 s	2,4 s	550 ms	0
Inseminaciones	0,6 s	0,6 s	0,7 s	2,5 s	620 ms	0
Chequeos	0,6 s	0,6 s	0,7 s	2,6 s	730 ms	0
Partos	0,6 s	0,9 s	0,7 s	3,1 s	1250 ms	0
Abortos	0,5 s	0,5 s	0,6 s	2,3 s	540 ms	0
Productos	0,5 s	0,5 s	0,7 s	2,6 s	680 ms	0
Pedidos	0,6 s	0,8 s	0,6 s	2,4 s	530 ms	0
Potreros	0,5 s	0,5 s	0,7 s	2,5 s	570 ms	0
Mantenimiento	0,6 s	0,6 s	0,6 s	2,7 s	770 ms	0
Producción Lechera	0,6 s	1,0 s	0,7 s	2,9 s	980 ms	0
Parcelas	0,6 s	0,6 s	0,7 s	2,6 s	720 ms	0
Grupos	0,6 s	0,6 s	0,6 s	2,5 s	610 ms	0
Recursos Humanos	0,6 s	0,8 s	0,7 s	2,6 s	720 ms	0

Los valores obtenidos por la matriz de rendimiento muestran el tiempo de respuesta entre la solicitud de usuario y la respuesta de la aplicación. De acuerdo con los resultados obtenidos del portal PageSpeed Insights, el rendimiento que ofrece la nueva aplicación con cada uno de sus módulos y procesos es favorable y óptimo para su uso dentro de la hacienda ProduAgro.

## Prueba de carga



**Figura 172.** Gráfico de prueba de carga

De acuerdo con el portal LoadView, la aplicación puede tener hasta siete usuarios como máximo, el cual es un número adecuado para su uso dentro de nuestro ambiente de estudio donde se está realizando la investigación que es en la hacienda ProduAgro. Además, esto indicaría que el nivel de escalabilidad se puede mejorar con el objetivo de que el número de usuarios máximo aumente.

**Anexo 8.** Manual de Usuario

# UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

## MANUAL DE USUARIO



AUTORES: Mora Herrera Jefferson Fernando  
Portilla Yamberla Héctor Armando

Tulcán, 2024

## INTRODUCCIÓN

El sistema informático denominado ProduAgro Sys tiene como finalidad administrar y gestionar la información de la hacienda y almacenar dichos datos para la toma de decisiones que se hagan posteriormente, posee una interfaz amigable con el usuario y tomando en cuenta sus requisitos se logró concluir con dicha aplicación.

La aplicación web puede ser ejecutada desde cualquier parte del mundo, ya que está alojada en un host y puede ser visible digitando su dirección URL, con esto crea una facilidad para poder administrar y gestionar desde cualquier sitio que se encuentre.

A continuación, se presenta un manual de usuario general de su funcionamiento.

## INICIO DE SESIÓN

Para iniciar sesión basta con tener el link de la aplicación, en este caso el link que se usa es <http://produagro-001-site1.ctempurl.com/> donde se ejecuta en cualquier navegador Chrome, Mozilla, Edge, etc.

Al abrir el link se abrirá esta ventana de inicio de sesión:



Una vez entrado a esta ventana hay que digitar las credenciales de inicio:

E-mail: produAgro@gmail.com

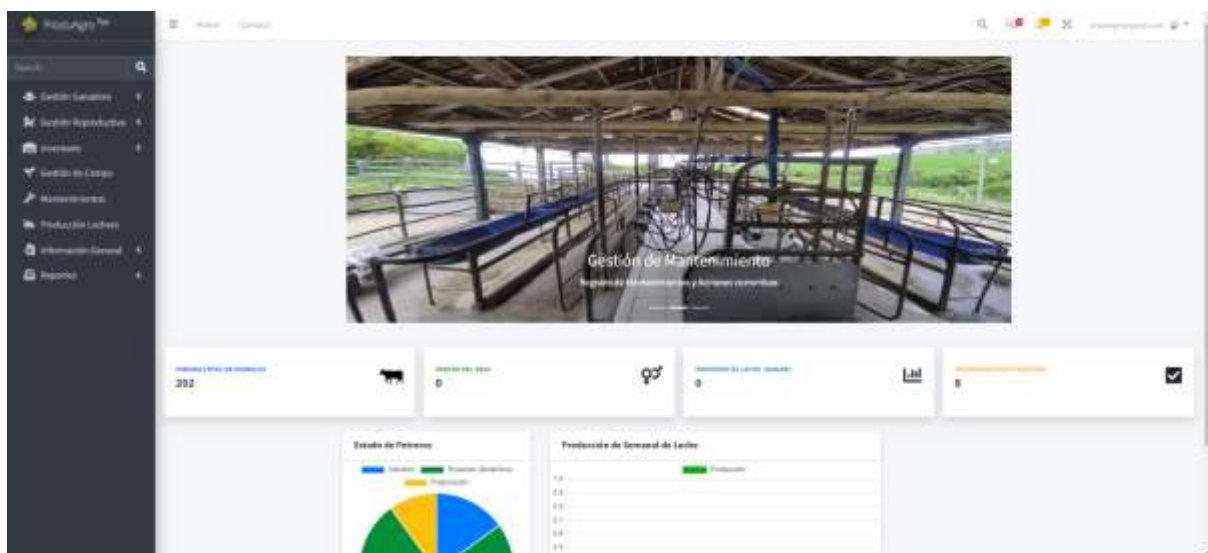
Contraseña: ProduAgro.2023

Ingresado las credenciales nos va dirigir al menú principal que se mira a continuación:

## MENU PRINCIPAL

EL menú principal está conformado por información actualizada como en el que caso del nominal total de animales, partos que se hicieron en el año, promedio de leche que se hace mensual y las inseminaciones positivas que conlleva la hacienda.

De igual manera se puede observar gráficos estadísticos que brindan la información del estado de potreros y la producción semanal de leche.



En el menú principal también se alojan los módulos que posee el sistema y los cuales se van a describir a continuación:

## GESTIÓN GANADERA

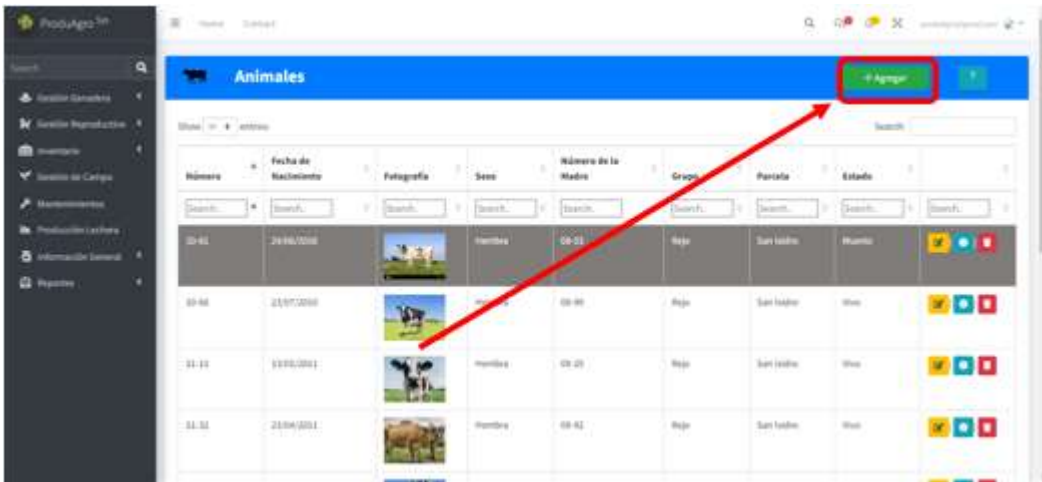
### Animales

En este apartado de animales se puede observar la información de cada animal que tiene la hacienda que están identificados con un código único, fecha de nacimiento, fotografía, sexo, numero de la madre, grupo, parcela, estado y los botones de acción como editar, detalles y eliminar.

The screenshot shows the 'Animales' management interface. It features a table with the following columns: Número, Fecha de Nacimiento, Fotografía, Sexo, Número de la Madre, Grupo, Parcela, and Estado. There are also search filters for each column and an 'Agregar' button at the top right.

Número	Fecha de Nacimiento	Fotografía	Sexo	Número de la Madre	Grupo	Parcela	Estado
10-41	30/04/2010		Hembra	00-43	Hijo	San Isidro	Muerto
10-88	23/07/2010		Hembra	00-25	Hijo	San Isidro	Vivo
11-13	13/01/2011		Hembra	00-25	Hijo	San Isidro	Vivo
11-32	23/04/2011		Hembra	00-42	Hijo	San Isidro	Vivo

Si queremos agregar un animal de damos clic en Agregar y llenamos los campos correspondientes:



Los campos a llenar son Numero (código único del animal), fecha de nacimiento, sexo, grupo, numero de la madre, parcela y fotografía del animal registrado y presionamos en el botón crear.

**Registro de un nuevo animal :**

Número:

Fecha de Nacimiento:

Sexo:

Grupo:

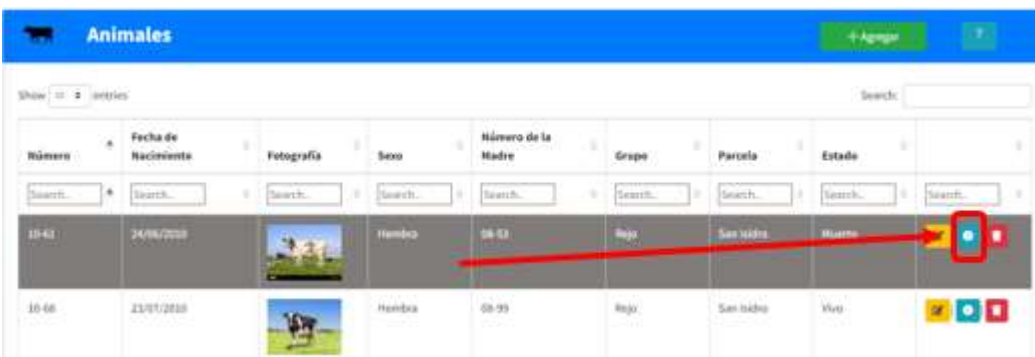
Número de la Madre:

Parcela:

Fotografía:  Ninguno archivo selec.

Ahora si queremos saber los detalles que posee un animal damos clic en el botón

**Detalles:**



Al hacer clic en el botón detalles se nos despliega esta información del animal.

**Detalles :** 🏠

---

**ProduAgro - 10-61**



Fecha: 31/1/2024

**Información General :**

Número : 10-61	Grupo : Rojo
Sexo : Hembra	Parcela : San Isidro
Fecha de Nacimiento : 2010-06-24	Número de la Madre : 08-53
	Fecha de Muerte : 12/14/2023 12:00:00 AM

TRATAMIENTOS

0

PRÓMEDIO DE LECHE

4

N° DE PARTOS

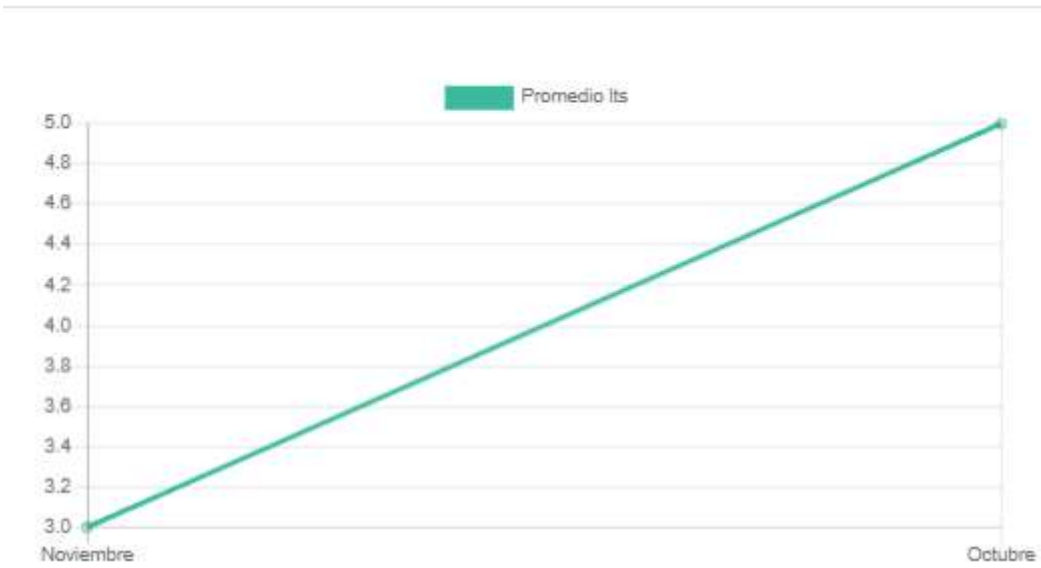
1

FECHAS DE MUERTE

18

Nos indica de manera gráfica un reporte promedio de la producción de leche mensual.

**Reporte de promedio de producción de leche mensual**



## Reporte de producción de leche mensual



Un breve historial de la fecha y los litros que da en animal en la mañana y en la tarde.

### Historial de Medida de Leche

Show  entries Search:

Fecha	Medicion_Mañana	Medicion_Tarde
2023-10-17	6	4
2023-10-24	4	9
2023-10-31	4	4
2023-11-07	4	3
2023-11-14	3	3
2023-11-21	3	3

Showing 1 to 6 of 6 entries Previous 1 Next

Un breve historial de los pesos que tiene el animal.

### Historial de Pesos

Show  entries Search:

Fecha	Peso_kg
No data available in table	

Showing 0 to 0 of 0 entries Previous Next

Los partos que tiene o ha tenido el animal.

**Historial de Partos**

Show 10 entries Search:

Fecha de Parto	Sexo	Estado_Animal1	
2023-08-07	2	Vivo	  

Showing 1 to 1 of 1 entries Previous 1 Next

Y datos veterinarios para saber si esta con un tratamiento y la descripción del tratamiento y la medicación que se le está suministrando, toda esta información se la puede imprimir para tener un reporte visual.

**Datos Veterinarios**

Show 10 entries Search:

Número	Estatus	Nombre de Producto	Nombre	Fecha de Tratamiento	Dias_Tratam	Diagnostico	Dosis
No data available in table							

Showing 0 to 0 of 0 entries Previous Next

[Editar](#) [Atras](#)

## Medición lechera

La medición lechera está conformada por el código único del animal, promedio histórico de litros y el número máximo de litros del animal. Posee los botones de agregar y detalles para ingresar nuevo y visualizar la información.

**Medida de Leche** [+ Agregar](#) 

Show 10 entries Search:

Número	Promedio Historico (Ltrs)	Maximo	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
10-62	4	13	
13-90	5	14	
14-28	5	16	
14-58	4	10	
14-64	5	13	

Información para agregar una medida de leche de un animal se presiona clic en el botón **Agregar** y se llenan los respectivos campos.

**Registro de medición :**

Número: 21-56

Fecha: dd/mm/aaaa

Medición Mañana:

Medición de Tarde:

[Crear](#) [Atras](#)

Si deseamos visualizar los detalles históricos de leche de un animal damos clic en el botón de información y se muestra la fecha y los litros ingresados en la tarde y mañana.

**Medida de Leche** [Agregar](#)

Show 10 entries Search

Número	Promedio Historico (Ltrs)	Maximo	
10-61	4	11	<a href="#">i</a>
13-50	5	14	<a href="#">i</a>
14-20	3	10	<a href="#">i</a>
14-55	4	10	<a href="#">i</a>
16-24	5	11	<a href="#">i</a>

**Historico de 10-61** [?](#)

Show 6 entries Search

Fecha	Tarde	Mañana	
2023/10/17	6	4	<a href="#">i</a> <a href="#">e</a>
2023/10/24	4	9	<a href="#">i</a> <a href="#">e</a>
2023/10/31	4	4	<a href="#">i</a> <a href="#">e</a>
2023/11/07	4	3	<a href="#">i</a> <a href="#">e</a>
2023/11/24	3	3	<a href="#">i</a> <a href="#">e</a>
2023/12/21	3	3	<a href="#">i</a> <a href="#">e</a>





Showing 1 to 6 of 6 entries [Previous](#) [Next](#)

## Muertes

Las muertes indica como datos el código único del animal (número), fecha de muerte, motivo y la imagen del animal muerto para que quede como evidencia del animal. También están los botones de agregar, editar, detalles y eliminar.

**Muertes** + Agregar

Show 10 entries Search:

Número	Fecha de Muerte	Motivo	Imagen	
<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>
10-68	14/12/2023	Caída fuerte		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>
23-01	15/04/2023	Neumonía grave		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>
23-06	26/12/2023	Neumonía, Mal de altura		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>
23-12	26/10/2023	Carbunco		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>

Si queremos agregar un animal damos clic en **Agregar** y llenamos los siguientes campos: número del animal, fecha, fotografía y una descripción o razón de muerte del animal.

**Registro de Muerte :**

**Número**

**Fecha de Muerte**





**Imagen**  
 Ninguno archivo selec.

**Motivo**

Para visualizar los detalles del animal presionamos en el botón de información de cada animal.

**Muertes** + Agregar

Show 10 entries Search:

Número	Fecha de Muerte	Motivo	Imagen	
<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>
10-68	14/12/2023	Caída fuerte		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>
23-01	15/04/2023	Neumonía grave		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>
23-06	26/12/2023	Neumonía, Mal de altura		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>
23-12	26/10/2023	Carbunco		<input type="button" value="Info"/> <input type="button" value="Share"/> <input type="button" value="Delete"/>

En dichos datos que se genera una ficha se puede observar la información del animal como sexo, grupo, promedio de leche, fecha de nacimiento, fecha de muerte, motivo y como parte adicional se indica los datos veterinarios por sí estuvo en un tratamiento antes de morir.

La ficha puede ser impresa para llevar un registro físico.

## Descartes

Los descartes son animales que no son aptos para la producción o al nacer tuvieron algún defecto o enfermedad, en esta ventana se visualiza el número la fecha de cuando fue descartado el motivo y su fotografía, también tiene como botones agregar, editar, detalles y eliminar.

Para añadir un nuevo descarte le damos clic en **Agregar** y llenamos los campos correspondientes como número, fecha de descarte, motivo y su fotografía:

**Registro de Descarte :**

**Número**

**Fecha de Descarte**


**Motivo**

**Fotografía**  
 Ninguno archivo selec.

Para visualizar los detalles del animal descartado presionamos en el botón de **información** y se muestra una ficha de la información detallada del animal y también los datos veterinarios para estar al margen de cuál fue su diagnóstico o motivo si fue por enfermedad o alguna lesión, de igual manera se puede imprimir dicha ficha para un reporte físico.

**Descartes**

Show: 1 entries Search:

Número	Fecha de Descarte	Motivo	Fotografía	
<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>
23-70	2023-12-02	test		<input type="button" value="Info"/> <input type="button" value="Print"/>

Showing 1 to 1 of 1 entries

**Ficha de Descarte**

**Fotografía**



**Actualidad**



**Número**  
23-70

**Sexo**  
Hembra

**Grupo**  
Neonato

**Fecha de Descarte**  
2023-12-02

**Motivo de Descarte**  
test

Tratamiento del Último mes

**Producto Utilizado:** Vitamina

**Responsable:** Mauricio Tirira

**Fecha:** 2023-12-11

**Duración:** 5 días

**Diagnostico:** Fuerte neumonia

**Dosis:** 5

## Pesos

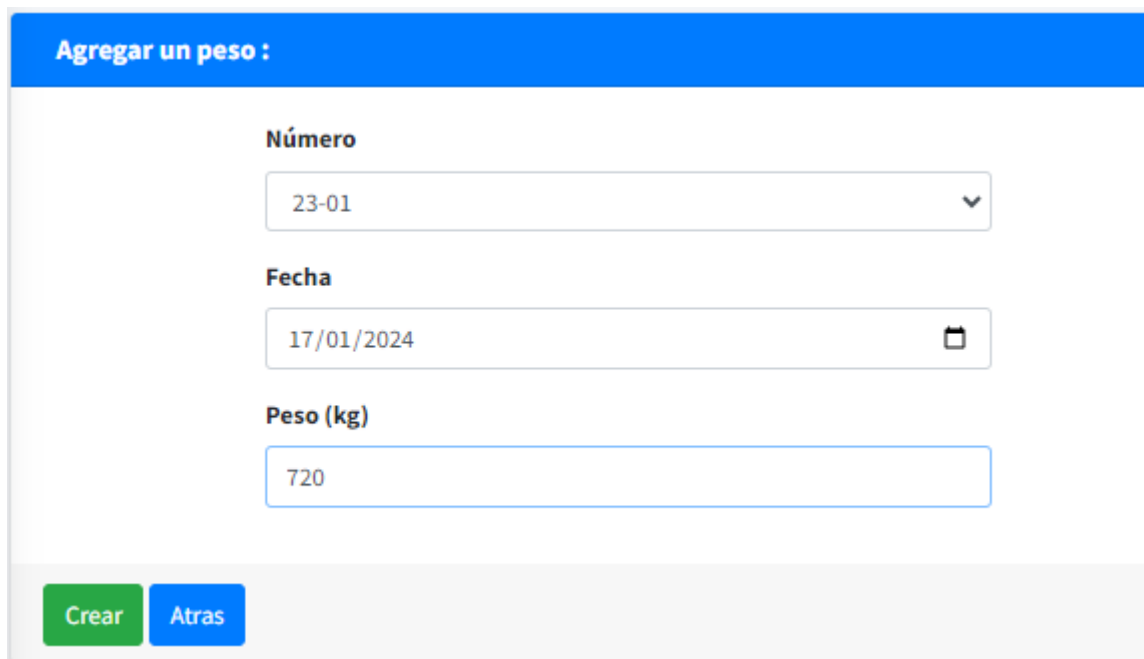
En los pesos se visualiza y se añaden los kilos de cada animal de la hacienda ProduAgro con los campos Numero del animal, ultimo pesaje en kg y la fecha del ultimo pesaje, también tenemos los botones de agregar y detalles.



The screenshot shows the 'Pesos' application interface. At the top, there is a header with a refresh icon, the title 'Pesos', and two buttons: '+ Agregar' (green) and 'i' (blue). Below the header, there is a search bar and a table. The table has three columns: 'Numero', 'Ultimo peso kg', and 'Fecha ultimo peso'. The first row contains the values '23-01', '720', and '2024-01-17'. To the right of the table, there is a search bar and a button with a plus sign. At the bottom, there is a pagination bar with 'Showing 1 to 1 of 1 entries', 'Previous', '1', and 'Next'.

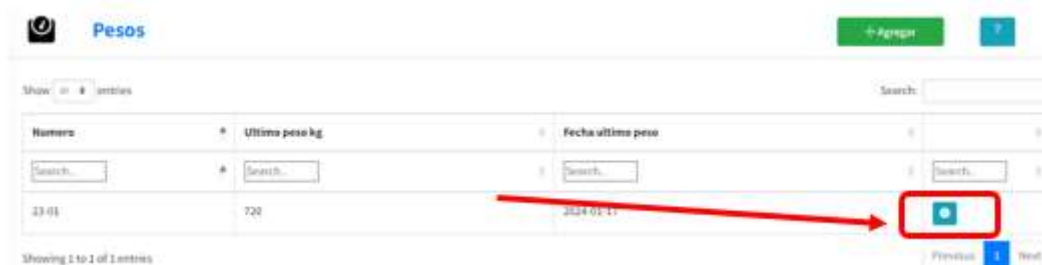
Numero	Ultimo peso kg	Fecha ultimo peso
23-01	720	2024-01-17

Para agregar un nuevo pesaje debe presionar en el botón Agregar y llenar los campos correspondientes del animal.



The screenshot shows the 'Agregar un peso' form. It has a blue header with the title 'Agregar un peso :'. Below the header, there are three input fields: 'Número' with a dropdown menu showing '23-01', 'Fecha' with a date picker showing '17/01/2024', and 'Peso (kg)' with a text input field showing '720'. At the bottom, there are two buttons: 'Crear' (green) and 'Atras' (blue).

Para visualizar los detalles del peso del animal presionamos en el botón de **Información** y se nos despliega la ficha con la información del histórico de peso del animal.







The screenshot shows the 'Pesos' application interface. At the top, there is a header with a refresh icon, the title 'Pesos', and two buttons: '+ Agregar' (green) and 'i' (blue). Below the header, there is a search bar and a table. The table has three columns: 'Numero', 'Ultimo peso kg', and 'Fecha ultimo peso'. The first row contains the values '23-01', '720', and '2024-01-17'. To the right of the table, there is a search bar and a button with a plus sign. At the bottom, there is a pagination bar with 'Showing 1 to 1 of 1 entries', 'Previous', '1', and 'Next'. A red arrow points from the table row to the plus sign button.

Numero	Ultimo peso kg	Fecha ultimo peso
23-01	720	2024-01-17

Historico de 23-01				+ Agregar	
Show: 1 of 1 entries		Search:			
Numero	Fecha	Peso (kg)			
23-01	2024/01/17	720.00	 		
Showing 1 to 1 of 1 entries				Previous   Next	

## Toros

Aquí en la ventana de toros se lleva el registro de todos los toros que existen en la hacienda, de igual manera con su código único y fotografía y sus respectivos botones de agregar, editar, detalles y eliminar.

Toros				+ Agregar	
Show: 1 of 1 entries		Search:			
Número	Imagen				
2000		  			
Showing 1 to 1 of 1 entries				Previous   Next	

Para añadir un nuevo toro presionamos el botón de **Agregar** y llenamos los respectivos campos.

**Toros**

Num\_Toro

Imagen

Seleccionar archivo | Ninguno archivo selec.

Create

## Datos veterinarios

En esta ventana se puede visualizar los datos veterinarios de todos los animales en general como su número, ultimo tratamiento, producto y diagnóstico. También sus respectivos botones de Agregar nuevo animal con su dato y el botón detalles.

Numero	Ultimo Tratamiento	Producto	Diagnostico
23-06	2023-12-13	Vacuna	Neumonia grave
23-70	2023-12-11	Vitamina	Falta memoria

Para agregar un nuevo animal damos clic en Agregar y llenamos los campos correspondientes como número, fecha de tratamiento, días que debe llevar el tratamiento, estado, producto, dosis, persona encargado de aplicar y diagnóstico, todo esto se llena cuando se ingresa un nuevo animal.







Para visualizar la información de un animal en específico vamos a dar clic en el botón de **Información** y se nos despliega dicha información.

Numero	Ultimo Tratamiento	Producto	Diagnostico
23-06	2023-12-13	Vacuna	Neumonia grave
23-70	2023-12-11	Vitamina	Falta memoria

Aquí indica el historial veterinario del animal seleccionado con sus respectivas fechas, días, diagnóstico, producto, dosis, encargado y estado, de igual manera se puede añadir, editar y eliminar la información.

**Historico de 23-01** + Agregar

Show 2 entries Search:

Fecha	Dias	Diagnostico	Producto	Dosis	Encargado	Estatus	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2023-12-01	2	Inicio de posible neumonia	Vacuna	1	Mauricio Torres	Aplicado	  
2023-12-13	4	Neumonia grave	Vacuna	4	Mauricio Torres	Aplicado	  

Showing 1 to 2 of 2 entries Previous **1** Next

## Traslados

Los traslados hacen referencia a los animales que fueron reubicados a otra parcela para lo cual se muestra la información general de todos los traslados ejecutados con su número del animal, fecha, parcela de origen y parcela destino. También tenemos los botones para agregar, editar, detalles y eliminar.

**Traslados** + Agregar

	Codigo	Fecha	Parcela Origen	Parcela Destino	
	3032	2023-05-30	Cayambe	San Isidro	  
	3033	2023-05-30	Cayambe	San Isidro	  
	3004	2023-06-29	Cayambe	San Isidro	  

Para agregar un nuevo traslado de un animal damos clic en **Agregar** y llenamos los respectivos campos como fecha de traslado, parcela de origen y destino al tener los datos registrados vamos agregar el animal en el botón **Agregar Animal** ahí se

selecciona el código del animal.

**Nuevo Traslado**

**Fecha de Traslado**  
10/01/2024

**Parcela de Origen**  
Ingueza

**Parcela de Destino**  
San Isidro

**Agregar Animal** | **Crear**

**Agregar un animal :**

**ID\_Animal**

- 15-14
- 15-14**
- 17-30
- 18-53
- 17-24
- 19-14
- 17-22
- 17-63
- 20-82
- 21-11
- 18-12
- 17-82
- 17-56
- 18-29
- 14-64
- 16-85
- 18-44
- 18-20
- 21-10
- 16-84
- 17-13

Y al agregar nos indica un breve detalle del animal seleccionado como su sexo y fecha de nacimiento.

**Detalle del traslado**

Número	Sexo	Fecha de Nacimiento

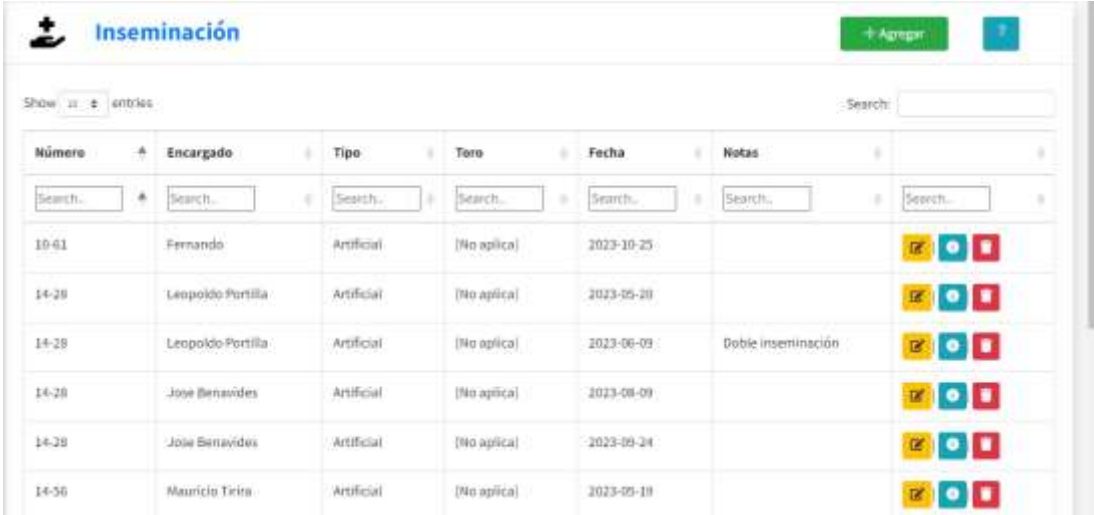
**Abrir**

## GESTIÓN REPRODUCTIVA

El módulo de gestión reproductiva podemos añadir y visualizar la información de los siguientes sub módulos:

## Inseminaciones


En la pestaña de inseminaciones se visualiza la información de todos los animales inseminados.



The screenshot shows a web interface for 'Inseminación' (Insemination). At the top right, there is a green '+ Agregar' button. Below the header, there is a search bar and a 'Show 11 entries' indicator. The main content is a table with the following columns: Número, Encargado, Tipo, Toro, Fecha, and Notas. Each row represents an insemination record. The first row has the number 10-61, assigned to Fernando, with an artificial insemination type on 2023-10-25. The second row has 14-28, assigned to Leopoldo Portilla, with an artificial insemination type on 2023-05-20. The third row has 14-28, assigned to Leopoldo Portilla, with an artificial insemination type on 2023-06-09 and a note 'Doble inseminación'. The fourth row has 14-28, assigned to Jose Benavides, with an artificial insemination type on 2023-08-09. The fifth row has 14-28, assigned to Jose Benavides, with an artificial insemination type on 2023-09-24. The sixth row has 14-56, assigned to Mauricio Tirira, with an artificial insemination type on 2023-09-10. Each row has three action buttons: a yellow checkmark, a blue plus, and a red minus.

Número	Encargado	Tipo	Toro	Fecha	Notas
10-61	Fernando	Artificial	[No aplica]	2023-10-25	
14-28	Leopoldo Portilla	Artificial	[No aplica]	2023-05-20	
14-28	Leopoldo Portilla	Artificial	[No aplica]	2023-06-09	Doble inseminación
14-28	Jose Benavides	Artificial	[No aplica]	2023-08-09	
14-28	Jose Benavides	Artificial	[No aplica]	2023-09-24	
14-56	Mauricio Tirira	Artificial	[No aplica]	2023-09-10	

Para añadir un nuevo animal inseminado damos clic en el botón de **Agregar** y llenamos los respectivos campos como código del animal, fecha de inseminación, tipo de inseminación, personal encargado en aplicarla y en el caso de ser una inseminación natural se agrega el código del toro y una breve nota.



The screenshot shows the 'Agregar' form for adding a new insemination record. The form has the following fields: 'Cod\_Animal' (dropdown menu with '10-61'), 'Fecha de Inseminación' (calendar icon), 'Cod\_Tipo\_Ferti' (dropdown menu with 'Artificial'), 'Cod\_Encargado' (dropdown menu with 'Mauricio Tirira'), 'Cod\_Toro' (dropdown menu with '[No aplica]'), and 'Notas' (text input field). At the bottom, there is a 'Create' button.

Para ver los detalles de una vaca en particular se da clic en detalles en el número del animal a consultar y se visualiza la información.

## Reproduccion

---

**Número**  
10-61

**Nombre**  
Fernando

**Nombre\_Tipo**  
Artificial

**Num\_Toro**  
[No aplica]

**Fecha de Inseminación**  
2023-10-25

**Notas**

### Chequeos

En la pestaña de chequeos se puede visualizar la información de los resultados de los animales inseminados, fecha del chequeo, descripción, tratamiento y estado corporal. También tenemos los botones de agregar un nuevo animal a chequear, editar, detalles y eliminar.

Número	Resultado	Fecha de Chequeo	Descripción	Tratamiento	Estado Corporal	
10-61	Negativo	2023-11-15	MUERTE EMBRIONARIA, CELD, MASTITIS	OPTIMIZADOR	2.50	[Iconos]
10-61	Negativo	2023-11-15	QUISTE FOLICULAR	GoRH+9dPG, OPTIMIZADOR	2.50	[Iconos]
14-28	Positivo	2023-11-15	Preñada		3.00	[Iconos]
14-64	Negativo	2023-11-15	PP NORMAL UN-FN	OPTIMIZADOR	2.70	[Iconos]
15-15	Positivo	2023-11-15	Preñada		3.00	[Iconos]
15-41	Negativo	2023-11-15	CLI	PROSTAGLANDINA	2.00	[Iconos]

Para agregar un nuevo animal al chequeo se llenan los respectivos campos como código del animal inseminado, fecha del chequeo, resultado, descripción (dado el caso del resultado sea negativo), tratamiento (dado el caso del resultado sea negativo) y estado corporal.

**Chequeo**

**Cod\_Reproduccion**

**Fecha de Chequeo**

**ID\_Resultado**

**Descripcion**

**Tratamiento**

**Estado\_Corporal**

## Partos

En la pestaña de partos se puede visualizar una información general de todos los partos obtenidos con su respectivo Numero, estado, fecha de parto, sexo, fotografía (debe incluir la madre y cría). También posee los botones principales como agregar, editar, detalles y eliminar.

**Partos** + Agregar

Show 10 entries Search:

Número	Estado	Fecha de Parto	Sexo	Fotografía	
<input type="text" value="Search.."/>	<input type="text" value="Search.."/>	<input type="text" value="Search.."/>	<input type="text" value="Search.."/>	<input type="text" value="Search.."/>	<input type="text" value="Search.."/>
10-61	Vivo	2023-08-07	Macho		<input type="checkbox"/> <input type="info"/> <input type="delete"/>
10-68	Vivo	2023-05-11	Macho		<input type="checkbox"/> <input type="info"/> <input type="delete"/>
11-13	Vivo	2023-11-03	Hembra		<input type="checkbox"/> <input type="info"/> <input type="delete"/>
11-32	Vivo	2023-04-25	Macho		<input type="checkbox"/> <input type="info"/> <input type="delete"/>

Para añadir un nuevo parto damos clic en el botón **Agregar** e ingresamos los datos respectivos del formulario como número del animal, fecha de parto, el estado de la cría, sexo y fotografía.

**Partos**

**Cod\_Animal**

**Fecha de Parto**


**Estado\_animal**

**Sexo**

**Fotografía**  
 Ninguno archivo selec.

## Abortos

Los abortos esta constituidos por todos los animales que fueron inseminados y en el proceso de embarazo abortó, se puede visualizar el número de la madre, número de abortos y su ultimo aborto. También posee el botón de agregar y detalles.

 **Abortos**

Show  entries Search:

Número	N° de Abortos	Ultimo aborto	
<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>
15-36	1	2023-03-19	<input type="button" value="i"/>
15-71	1	2023-06-25	<input type="button" value="i"/>
18-48	1	2023-07-01	<input type="button" value="i"/>
19-03	1	2023-01-08	<input type="button" value="i"/>
19-30	1	2023-07-06	<input type="button" value="i"/>
19-42	1	2023-05-02	<input type="button" value="i"/>

Para añadir un nuevo aborto damos clic en **Agregar** y llenamos los respectivos campos como el número del animal, fecha de aborto y una nota con una descripción.

**Abortos**

**Cod\_Animal**

15-14

**Fecha de Aborto**

dd/mm/aaaa

**Notas**

Para ver los detalles del animal que abortó damos clic en el botón de **Información** para visualizar.

**Abortos** + Agregar

Show 10 entries Search:

Número	N° de Abortos	Ultimo aborto	
Search...	Search...	Search...	Search...
15-36	1	2023-03-19	<b>i</b>
15-71	1	2023-06-25	<b>i</b>
18-48	1	2023-07-01	<b>i</b>
19-03	1	2023-01-08	<b>i</b>
19-38	1	2023-07-06	<b>i</b>
19-42	1	2023-05-02	<b>i</b>

Y observamos un breve histórico de los abortos y sus respectivas notas o descripción de la razón del aborto para tomar en consideración en su próxima inseminación.

**Historico de 15-36**

Show 10 entries Search:

Fecha	Notas	
Search...	Search...	Search...
2023/03/19	Deficiencias Nutricionales	<b>i</b> <b>r</b>

Showing 1 to 1 of 1 entries Previous 1 Next

## INVENTARIO

En la sección de Inventario de este manual se encuentra información detallada sobre la gestión de productos y pedidos, elementos esenciales para mantener un control eficiente de las existencias y asegurar un proceso de gestión de inventario fluido.

### Productos

En la sección de productos se puede actualizar y gestionar productos en el inventario con la información de categoría, medida, nombre del producto y cantidad. También el botón creación de nuevos elementos hasta la modificación de detalles existentes.

Categoría	Medida	Nombre de Producto	Cantidad	
Search..	Search..	Search..	Search..	Search..
Agrícola	Costales	Cal	100	
Alimento	Cajas	Vitamina	2	
Fertilizantes	Galones	Biol	4	
Veterinaria	Cajas	Vacuna	4	

Para añadir un nuevo producto damos clic en Agregar y llenamos los respectivos campos que nos indican como el nombre del producto, categoría, medida y cantidad.

Producto

**Nombre de Producto**

**Categoría\_P**









Alimento

**Unidad\_Medida**

Cajas

**Cantidad**

Si deseamos agregar más cantidad al stock damos clic en editar para aumentar la cantidad o bien corregir algún error ingresado.

Categoria	Medida	Nombre de Producto	Cantidad	
Search...	Search...	Search...	Search...	Search...
Agricola	Costales	Cal	100	 
Alimento	Cajas	Vitamina	3	 
Fertilizantes	Galones	Biol	4	 
Veterinaria	Cajas	Vacuna	4	 

Y se nos despliega el formulario para realizar la respectiva modificación de la información.

Producto

**Nombre de Producto**

**Categoria\_P**













**Unidad\_Medida**

**Cantidad**

Save

## Pedidos

Los pedidos están conformados por el proceso de los productos a recibir en el cual se identifica por un código de pedido, fecha de pedido y el estado. De igual manera se tiene los botones de agregar, editar, detalles y eliminar pedidos.

	Código	Fecha de Pedido	Estatus							
⊖	5	2023-10-04	Completada	  						
<p><b>Detalle</b></p> <table border="1"> <thead> <tr> <th>Nombre de Producto</th> <th>Medida</th> <th>Cantidad</th> </tr> </thead> <tbody> <tr> <td>Vacuna</td> <td>Cajas</td> <td>0</td> </tr> </tbody> </table>					Nombre de Producto	Medida	Cantidad	Vacuna	Cajas	0
Nombre de Producto	Medida	Cantidad								
Vacuna	Cajas	0								
⊕	6	2023-10-04	Enviada	  						
⊕	1002	2023-10-05	Ordenada	  						
⊕	1003	2023-10-05	Completada	  						

Para agregar un nuevo pedido damos clic en **Agregar** y llenamos los respectivos campos y damos clic en **Agregar Producto**:

**Nuevo Pedido**

Estado: [Seleccione un estatus.]

AddProduct Guardar Pedido

Detalle de la Orden:

Nombre de Producto	Medida	Cantidad	Categoria_P

Atras

Agregamos el producto y cantidad luego guardamos el pedido.

**Agregue un producto:**

**Producto**

Vacuna

**Cantidad**

5

Guardar Pedido

Al guardar el pedido este se va reflejar en el apartado de **Detalle de la Orden**.

**Nuevo Pedido**

Estado  
[Seleccione un estatus...]

AddProduct Guardar Pedido

**Detalle de la Orden**

Nombre de Producto	Medida	Cantidad	Categoría_P
Vacuna	1	5	

Editar | Delete

Atras

Y tenemos los botones de editar el detalle de orden o eliminar, dado esto el próximo paso es guardar el pedido.

**Nuevo Pedido**

Estado  
[Seleccione un estatus...]

AddProduct **Guardar Pedido**

**Detalle de la Orden**

Nombre de Producto	Medida	Cantidad	Categoría_P
Vacuna	1	5	

Atras

Al guardar el pedido este se actualizará de manera automática en el inicio de los pedidos con su respectiva fecha y estado.

3003 2024-01-31 Completada

**Detalle**

Nombre de Producto	Medida	Cantidad
Vacuna	Cajas	10

## GESTION DE CAMPO

La gestión de campo hace referencia a los estados de los potreros este se visualiza con su número de potrero, el área, estado y la imagen del potrero. Posee sus botones de agregar nuevo potrero, editar, detalles y eliminar, adicionalmente se puede tener un reporte estadístico de todos los potreros que existen.

Potrero	Area	Estado	Imagen	
1	1.42	Preparacion		
2	1.45	Siembra		
3	1.72	Siembra		

Para agregar un nuevo potrero damos clic en **Agregar** e ingresamos los respectivos datos del potrero como el código del potrero, área y su estado y adicional su imagen.

Potrerros

ID\_Potrero

Area\_h

ID\_Accion

Siembra

Fotografía

Seleccionar archivo Ninguno archivo selec.

Create

Para visualizar el detalle de un potrero en específico damos clic en el botón de **información**.

Potrero	Area	Estado	Imagen	
1	1.42	Preparacion		
2	1.45	Siembra		
3	1.72	Siembra		

Y se nos abrirá esta ventana con la información del estado, producto, encargado, fecha y la imagen, la cual puede agregar, modificar y detalles del potrero.



The screenshot shows the 'Campo' application interface. At the top left is the 'Campo' logo. On the top right, there is a green '+ Agregar' button and a blue button with the number '7'. Below the header, there is a search bar and a 'Show 10 entries' dropdown. The main content is a table with the following data:

Acción	Producto	Encargado	Fecha	Imagen	
Preparacion	Cal	Mauricio Tirira	05/12/2023		

At the bottom left, it says 'Showing 1 to 1 of 1 entries'. At the bottom right, there are 'Previous' and 'Next' navigation buttons.

Para añadir damos clic en **Agregar** y llenamos los respectivos campos como el código del potrero, su estado, fecha, producto utilizado en el potrero, la persona que estuvo a cargo y su imagen.



The screenshot shows the 'Campo' application form for adding a new entry. The form has the following fields:




- ID\_Potrero**: A dropdown menu with the value '1' selected.
- ID\_Accion**: A dropdown menu with the value 'Siembra' selected.
- Fecha**: A date input field with the placeholder 'dd/mm/aaaa' and a calendar icon.
- ID\_Producto**: A dropdown menu with the value 'Biol' selected.
- ID\_Encargado**: A dropdown menu with the value 'Mauricio Tirira' selected.
- Imagen**: A file upload field with a 'Seleccionar archivo' button and the text 'Ninguno archivo selec.'

Si deseamos saber el estado de todos los potreros y su estado, se puede generar un reporte estadístico muy fácil de comprender, haciendo clic en Datos Estadísticos.

Potreros

Show 10 entries

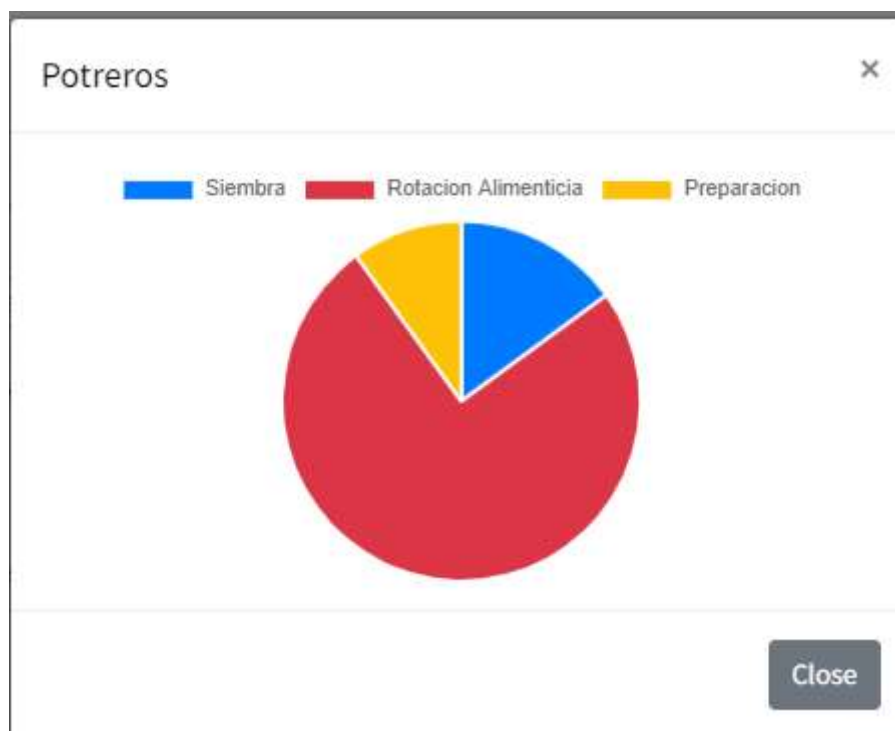
Search:

Potrero	Area	Estado	Imagen
1	1.42	Preparacion	
2	1.45	Siembra	
3	1.72	Siembra	

Mi Datos Estadísticos

+ Agregar

Y obtendremos la siguiente imagen con la información detallada gráficamente.



## MANTENIMIENTOS

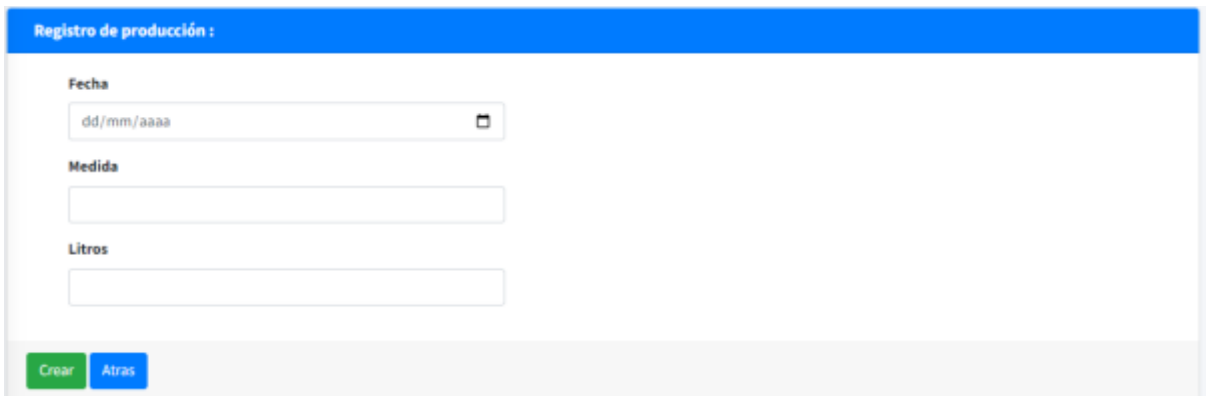
En esta sección de mantenimiento nos brinda información de alguna falla que se encuentre en la hacienda detalladamente por su área, tipo, fecha, problema inicial, descripción y costo del mantenimiento. También posee los botones de agregar nuevos mantenimientos, editar, detalles y eliminar.

Para agregar un nuevo mantenimiento damos clic en Agregar y llenamos los respectivos campos con información veraz.

## PRODUCCION LECHERA

En la sección de producción lechera se visualiza los litros generados con la información de fecha, medida y litros. Posee de igual manera los botones de agregar, editar y eliminar los datos. Adicionalmente se implementa un botón con los datos estadísticos.

Para añadir un nuevo reporte de producción lechera damos clic en **Agregar** y llenamos los campos respectivos fecha, medida y litros.



Registro de producción :

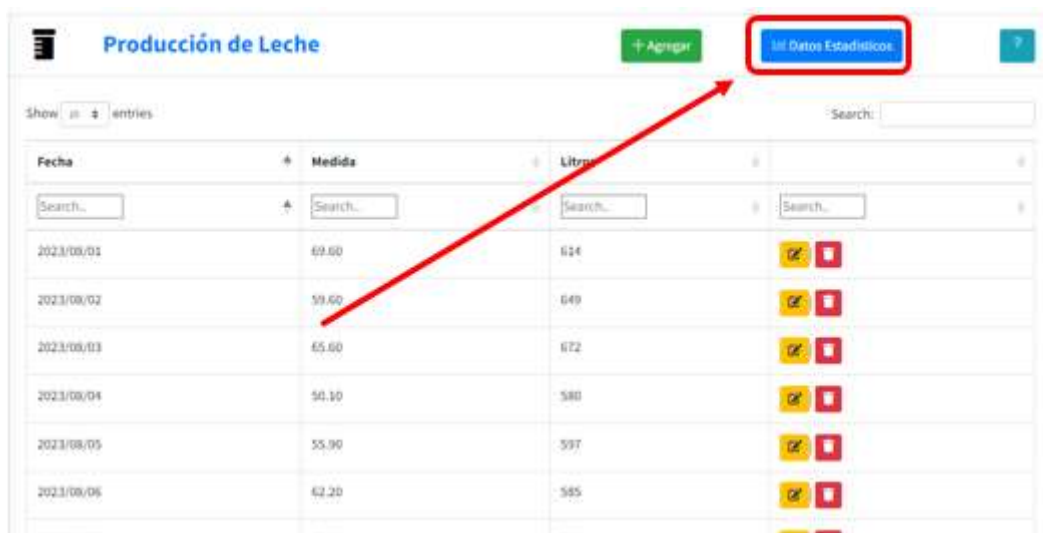
Fecha  
dd/mm/aaaa

Medida

Litros

Crear Atras

Para visualizar los datos estadísticos de la producción de leche damos clic en **Datos Estadísticos** y visualizamos de manera gráfica dichos reportes.



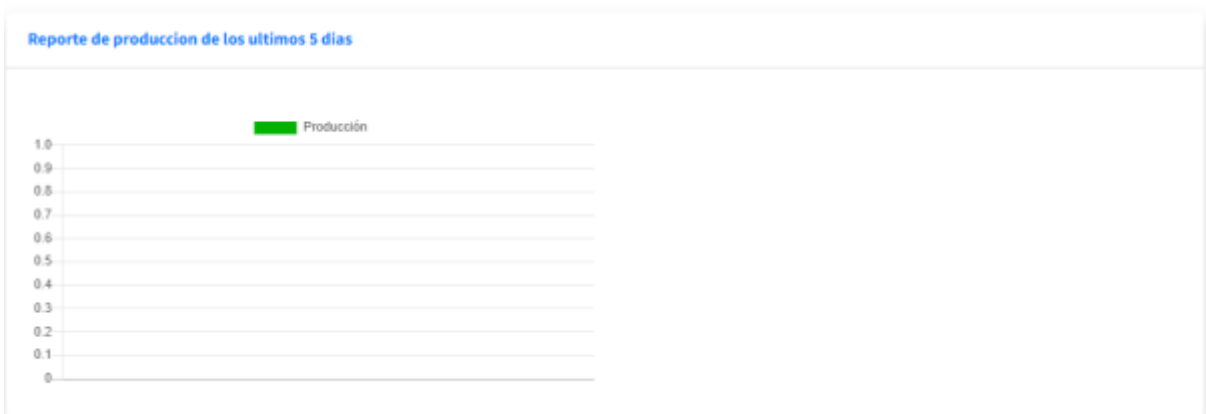
Producción de Leche

+ Agregar **Ver Datos Estadísticos**

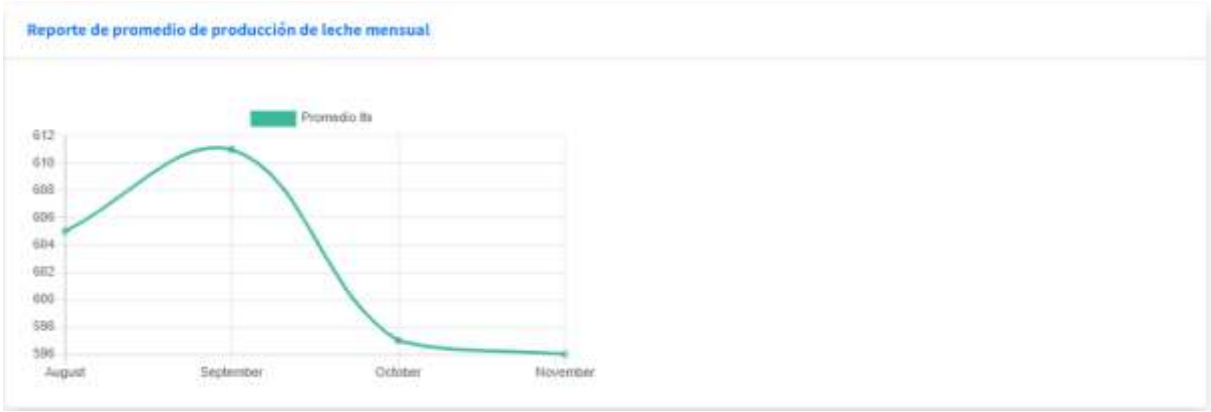
Show 10 entries Search:

Fecha	Medida	Litros	
2023/08/01	69.60	624	🗑️ 🗑️
2023/08/02	59.60	649	🗑️ 🗑️
2023/08/03	65.60	672	🗑️ 🗑️
2023/08/04	50.10	580	🗑️ 🗑️
2023/08/05	55.90	597	🗑️ 🗑️
2023/08/06	62.20	565	🗑️ 🗑️

Aquí se visualiza de manera gráfica un reporte de producción de los últimos 5 días.



En este otro apartado gráfico se observa un reporte promedio de producción de leche mensual.



## INFORMACIÓN GENERAL

En esta sección del manual, veremos aspectos cruciales relacionados con la gestión de información general de nuestra aplicación. Desde la asignación de parcelas hasta la creación y administración de grupos, así como la incorporación y gestión de recursos humanos, cada subtema abordará elementos esenciales para una operación eficiente dentro del sistema ProduAgro.

### Parcelas

Este apartado proporcionará instrucciones detalladas sobre cómo añadir y mantener la información de la ubicación de las parcelas, garantizando una gestión efectiva de tus activos.

Parcela	Ubicacion	
<input type="text" value="Search..."/>	<input type="text" value="Search..."/>	<input type="text" value="Search..."/>
Coyambe	Coy	<input checked="" type="checkbox"/> <input type="checkbox"/>
Inguaza	ing	<input checked="" type="checkbox"/> <input type="checkbox"/>
San Isidro	San	<input checked="" type="checkbox"/> <input type="checkbox"/>

Para añadir una nueva parcela damos clic en Agregar y llenamos los campos con la información específica.

Parcelas

**Parcela**

**Ubicacion**

## Grupos

En este apartado se crea, organiza y administra grupos a los cuales pertenecen los animales de la hacienda ProduAgro para optimizar la colaboración con las demás secciones.

Grupo	Cargos
Crianza	
Descarte	
Desteta	
Levante	
Neonato	
Preñada	

## Recursos humanos

En esta sección se integra y gestiona los recursos humanos que existen en la hacienda ProduAgro. Se añade nuevos miembros con su respectivo cargo que realiza en la hacienda para optimizar la gestión del equipo.

Nombre	Cargo	Cargos
Armando Nenjer	Jornalero	
Dolores Yamberla	Jornalero	
Fernando	Administrador	
Jose Benavides	Administrador	
Leopoldo Portilla	Jornalero	
Mauricio Tirira	Administrador	
Oscar Manosalvas	Jornalero	

## REPORTES

En esta sección del manual, tenemos la función de generación de informes de nuestro sistema informático. Los reportes son herramientas vitales para analizar y comprender la gestión y administración de toda la hacienda ProduAgro.

## Animales

En el reporte de animales se puede hacer una filtración de la información dependiendo de lo que se desea realizar la consulta:

**Animales**

Fecha de Inicio: dd/mm/aaaa

Fecha de Fin: dd/mm/aaaa

Parcela: Todos

Grupo: Todos

Estado: Todos

Show 10 entries

Search:

Número	Fecha de Nacimiento	Fotografía	Sexo	Número de la Madre	Grupo	Parcela	Estado
10-61	24/06/2010		Hembra	08-53	Rejo	San Isidro	Muerto
10-68	23/07/2010		Hembra	08-99	Rejo	San Isidro	Vivo


Puede generar reporte desde una fecha de inicio y final, consultar una respectiva parcela, grupo o estado, en este caso vamos a generar un reporte de ejemplo de todos los animales muertos. También tiene botones de imprimir y exportar a Excel.

## LISTA DE ANIMALES

Número	Fecha de Nacimiento	Fotografía	Sexo	Numero de la Madre	Grupo	Parcela	Estado
10-61	24/06/2010		Hembra	08-53	Rejo	San Isidro	Muerto
23-01	09/01/2023		Hembra	20-55	Vaconas	San Isidro	Muerto
23-06	18/01/2023		Hembra	17-38	Vaconas	San Isidro	Muerto
23-12	13/02/2023		Hembra	19-32	Vaconas	San Isidro	Muerto
23-17	09/03/2023		Hembra	17-15	Levante	San Isidro	Muerto
23-31	07/06/2023		Hembra	17-39	Destete	San Isidro	Muerto

## Inseminaciones

De igual manera todos los reportes tienen un filtro el cual puede ser escogido dependiendo de la consulta y posteriormente imprimir o exportar.




**Encargado**

**Tipo**

Número	Encargado	Tipo	Toro	Fecha	Notas
14-64	Leopoldo Portilla	Artificial	2000	2023-01-06	Inseminación y toro
20-70	Leopoldo Portilla	Artificial	[No aplica]	2023-01-08	
15-72	Mauricio Tirira	Artificial	[No aplica]	2023-01-03	
21-16	Leopoldo Portilla	Artificial	[No aplica]	2023-01-08	
17-32	Mauricio Tirira	Artificial	[No aplica]	2023-01-15	
18-45	Mauricio Tirira	Artificial	[No aplica]	2023-01-22	
18-65	Mauricio Tirira	Artificial	[No aplica]	2023-01-24	
17-72	Mauricio Tirira	Artificial	[No aplica]	2023-01-29	

## Chequesos



**Resultado**





Número	Resultado	Fecha de Chequeo	Descripción	Tratamiento	Estado Corporal
17-30	Positivo	2023-11-15	Preñada		3.00
17-32	Negativo	2023-11-15	PP-NORMAL UN-FH	Optimizador, GNDIUFNDEI	2.80
17-49	Negativo	2023-11-15	CLI	CIDR	2.80
17-49	Negativo	2023-11-15	CLI	CIDR	2.80
17-72	Negativo	2023-11-15	METRITIS	BIOCOBRE	2.80
17-78	Negativo	2023-11-15	CLI	PROSTAGLANDINA, OPTIMIZADOR	2.80
18-04	Positivo	2023-11-15	Preñada		3.00
18-29	Negativo	2023-11-15	CELO		2.80

## Partos


**♀ Partos** Estado Sexo

Todos ▼ Todos ▼

**Generar**

Número	Estado	Fecha de Parto	Sexo	Fotografía
20-55	Vivo	2023-01-09	Hembra	
18-53	Vivo	2023-07-30	Hembra	
17-24	Vivo	2023-07-26	Hembra	
18-04	Vivo	2023-07-28	Macho	

## Abortos

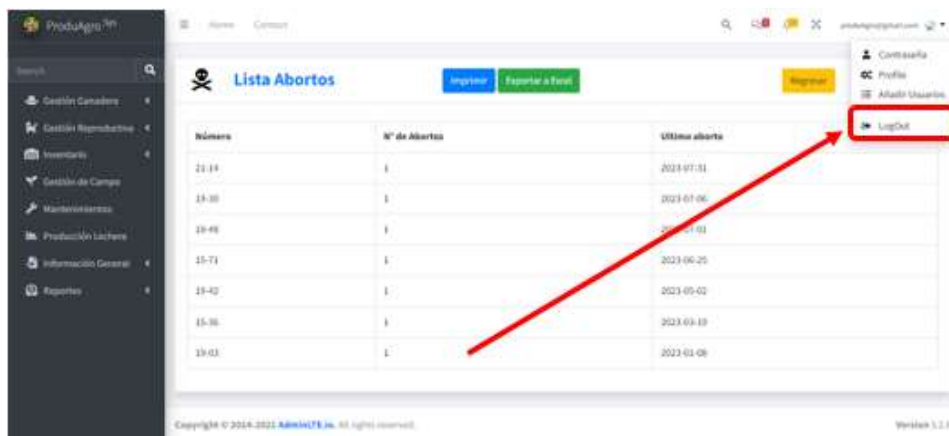
 **Lista Abortos** [Imprimir](#) [Exportar a Excel](#) [Regresar](#)

Número	N° de Abortos	Ultimo aborto
21-14	1	2023-07-31
19-30	1	2023-07-06
18-48	1	2023-07-01
15-71	1	2023-06-25
19-42	1	2023-05-02
15-36	1	2023-03-15
19-03	1	2023-01-08

## CIERRE DE SESIÓN

1. Para cerrar sesión, haga clic en el botón **"LogOut/Cerrar Sesión"** en la esquina superior derecha de la ventana.

2. Será redirigido a la pantalla de inicio de sesión.



## SOLUCION DE PROBLEMAS

Si encuentra algún problema con la aplicación, puede intentar lo siguiente:

1. Verifique que contenga conexión a internet estable.
2. Reinicie la aplicación y vuelva a intentarlo.
3. Borre los datos del cache del navegador que está usando.
3. Consulte el manual de usuario o la sección de ayuda en la aplicación.
4. Póngase en contacto con el soporte técnico para obtener asistencia adicional.

## DATOS SOPORTE TÉCNICO

- Teléfonos: 0986243268- 0968602784
- Nombre: Héctor Portilla- Fernando Mora
- Correo: hector.portilla@upec.edu.ec jefferson.mora@upec.edu.ec

## CONCLUSIÓN

El sistema informático ProduAgro proporciona una solución eficiente para la administración y gestión de información de la hacienda ProduAgro. Con su interfaz intuitiva y funcionalidades clave, facilita la toma de decisiones.

Esperamos que este manual de usuario le haya proporcionado la información necesaria para utilizar la aplicación de manera efectiva. Si tiene alguna pregunta adicional o necesita asistencia adicional, no dude en ponerse en contacto con soporte técnico.

**¡Gracias por utilizar ProduAgro Sys!**