



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE TECNOLOGÍAS DE LA INFORMACIÓN,
TELECOMUNICACIONES E INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES E INFORMÁTICOS**

TEMA:

**SISTEMA INFORMÁTICO APLICANDO PATRONES DE DISEÑO Y
PROGRAMACIÓN LIMPIA PARA LA PRODUCCIÓN PORCINA EN LA
EMPRESA PROINBA DEL CANTÓN CEVALLOS.**

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de
Ingeniero en Ingeniero en Sistemas Computacionales e Informáticos

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Ramírez Quinatoa Darío Javier

TUTOR: Ing. Edison Álvarez Mg.

Ambato - Ecuador

Abril, 2019

CERTIFICACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

"SISTEMA INFORMÁTICO APLICANDO PATRONES DE DISEÑO Y PROGRAMACIÓN LIMPIA PARA LA PRODUCCIÓN PORCINA EN LA EMPRESA PROINBA DEL CANTÓN CEVALLOS", del señor Darío Javier Ramírez Quinatoa, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad técnica de Ambato

Ambato, abril de 2019



Ing. Edison Álvarez Mg.

EL TUTOR

AUTORÍA DEL TRABAJO

El presente trabajo de investigación titulado: "SISTEMA INFORMÁTICO APLICANDO PATRONES DE DISEÑO Y PROGRAMACIÓN LIMPIA PARA LA PRODUCCIÓN PORCINA EN LA EMPRESA PROINBA DEL CANTÓN CEVALLOS". Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, abril de 2019



Darío Javier Ramírez Quinatoa

CC: 1804540522

APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Hernán Naranjo Mg e Ing. Félix Fernández PhD, revisó y aprobó el Informe Final del trabajo de graduación titulado "SISTEMA INFORMÁTICO APLICANDO PATRONES DE DISEÑO Y PROGRAMACIÓN LIMPIA PARA LA PRODUCCIÓN PORCINA EN LA EMPRESA PROINBA DEL CANTÓN CEVALLOS", presentado por el señor Darío Javier Ramírez Quinatoa de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.



Ing. Pilar Urrutia Mg.

PRESIDENTE DEL TRIBUNAL



Ing. Hernán Naranjo Mg.

DOCENTE CALIFICADOR



Ing. Félix Fernández PhD.

DOCENTE CALIFICADOR

DEDICATORIA

El presente trabajo lo quiero dedicar a Dios que me ha dado la fuerza y la voluntad de seguir adelante, así como también a mi madre María Quinatoa quien ha sido la persona que ha velado por mí, jamás dejo que me rinda, que ha estado en los peores momentos conmigo, que me ha dado toda su sabiduría durante toda la etapa de mis estudios, de tal manera que me han permitido llegar a cumplir un sueño y una meta.

A mi hermana Taña Ramírez quien ha sido mi constante apoyo y a toda mi familia que siempre ha estado ahí conmigo.

Darío Javier Ramírez Quinatoa

AGRADECIMIENTO

Quiero agradecer primero a Dios quien me ha permitido estar hoy aquí, por darme toda la entereza para no dejarme derrotar ante las adversidades, llevarme con bien en todo este camino y derramar siempre toda su bendición.

Agradecer también a mi familia que sin su preocupación y apoyo nada sería posible y las grandes virtudes que me fueron dadas.

A la persona que me guío en el presente trabajo a mi tutor de tesis el Ing. Edison Álvarez que siempre estuvo para ayudarme ante cualquier inquietud y brindarme todas las facilidades para terminar con gran éxito.

Y un agradecimiento especial a la empresa Proinba por haberme permitido realizar el proyecto de investigación.

Gracias infinitas.

Darío Javier Ramírez Quinatoa

ÍNDICE

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
APROBACIÓN COMISIÓN CALIFICADORA	iv
Dedicatoria	v
Agradecimiento	vi
Introducción	xvii
CAPÍTULO 1 El problema	1
1.1 Tema de Investigación	1
1.2 Planteamiento del problema	1
1.3 Delimitación	2
1.4 Justificación	2
1.5 Objetivos	3
1.5.1 General	3
1.5.2 Específicos	3
CAPÍTULO 2 Marco Teórico	4
2.1 Antecedentes Investigativos	4
2.2 Fundamentación teórica	6
2.2.1 Patrón de Diseño	6
2.2.1.1 Patrones de Construcción o Creacionales	7
2.2.1.2 Patrones de Estructuración	7
2.2.1.3 Patrones de Comportamiento	8
2.2.2 Programación Limpia	9
2.2.3 Metodología	9
2.2.4 Ciclo de Vida	9
2.2.5 Metodología Ágil	9

2.2.6	Extreme Programming (XP).....	10
2.2.6.1	Ciclo de vida de XP.....	10
2.2.6.2	Valores de la metodología XP.....	11
2.2.6.3	Pasos de la metodología XP	12
2.3	Propuesta de Solución	13
CAPÍTULO 3 Metodología		14
3.1	Modalidad de la investigación.....	14
3.2	Población y muestra.....	14
3.3	Recolección de información	14
3.4	Procesamiento de la información	15
3.5	Desarrollo del Proyecto	15
CAPÍTULO 4 Desarrollo de la propuesta		17
4.1	Programación Limpia	17
4.1.1	Reglas Generales de Programación Limpia.....	17
4.2	Patrones de Diseño	22
4.3	Metodología de Desarrollo Extreme Programming (XP).....	25
4.3.1	XP se estructura en tres prácticas.....	25
4.3.2	Metodología XP en el Desarrollo de Software.....	25
4.4	Desarrollo del Sistema del Control Porcícola basado en la Metodología Ágil XP	27
4.4.1	Fase de Exploración	27
4.4.1.1	Estudio de los procesos para la producción porcina.	27
4.4.1.2	Análisis de la comercialización de los porcinos. . .	30
4.4.1.3	Procesos del sistema para el control porcino . . .	31
4.4.2	Fase de Planeación de Entregas.....	45
4.4.3	Fase de Iteraciones	46
4.4.3.1	Iteraciones de las Historias de Usuario	47
4.4.3.2	Plan de Entregas	48
4.4.4	Fase de Producción	48
4.4.4.1	Base de Datos	48
4.4.4.2	Diagrama Casos de Uso	49
4.4.4.3	Diseño de las Interfaces	52
4.4.4.4	Patrones de diseño en el sistema web.....	60
4.4.4.5	Patrón Facade	61
4.4.4.6	Patrón Builder.....	64
4.4.4.7	Patrón Adapter.....	68

4.4.4.8	Programación Limpia en el Desarrollo del Sistema	70
4.4.5	Fase de Pruebas	75
4.4.5.1	Pruebas de Caja Negra	75
4.4.5.2	Pruebas de Funcionamiento del sistema.....	79
4.4.6	Implantación del sistema	93
4.4.6.1	Configuración de los equipos.....	93
4.4.6.2	Puesta en marcha del sistema.....	93
4.4.6.3	Capacitación	97
CAPÍTULO 5 Conclusiones y Recomendaciones		99
5.1	Conclusiones	99
5.2	Recomendaciones	99
Bibliografía		101
ANEXOS		105

ÍNDICE DE TABLAS

1	Clasificación Gof.	6
2	Reglas de Programación Limpia.	18
3	Reglas de Programación Limpia.	19
4	Reglas de Programación Limpia.	19
5	Reglas de Programación Limpia.	20
6	Reglas de Programación Limpia.	21
7	Reglas de Programación Limpia.	22
8	Patrones de Diseño.	23
9	Patrones de Diseño.	24
10	Patrones de Diseño.	24
11	Tabla Comparativa de Metodologías	25
12	Tabla Comparativa de Metodologías	26
13	Registro Usuarios.....	32
14	Criterios de Aceptación: Registro Usuarios	32
15	Registro Galpones.....	33
16	Criterios de Aceptación: Registro Galpones	33
17	Registro Corrales	34
18	Criterios de Aceptación: Registro Corrales.....	34
19	Registro Madres	34
20	Criterios de Aceptación: Registro Madres	35
21	Registro Machos.....	35
22	Criterios de Aceptación: Registro Machos.....	36
23	Registro Tipo Madres	36
24	Criterios de Aceptación: Registro Tipo Madres	37
25	Registro Inseminaciones	37
26	Criterios de Aceptación: Registro Inseminaciones.....	38
27	Registro Revisiones.....	38
28	Criterios de Aceptación: Registro Revisiones	39
29	Registro Nacimientos.....	39
30	Criterios de Aceptación: Registro Nacimientos	40

31	Registro Destetes	40
32	Criterios de Aceptación: Registro Destetes	41
33	Registro Detalle Destetes	41
34	Criterios de Aceptación: Registro Detalle Destetes	42
35	Registro Lotes	42
36	Criterios de Aceptación: Registro Lotes	42
37	Registro Detalle Lotes	43
38	Criterios de Aceptación: Registro Detalle Lotes	43
39	Registro Movimientos.....	43
40	Criterios de Aceptación: Registro Movimientos	44
41	Registro Mortalidad y Ventas.....	44
42	Criterios de Aceptación: Registro Mortalidad y Ventas	45
43	Estimación de las Historias de Usuarios.....	46
44	Iteraciones de las Historias de Usuarios	47
45	Plan de Entregas	48
46	Caso de Uso del Administrador	50
47	Caso de Uso del Empleado	51
48	Caja Negra: Inicio de Sesión.....	76
49	Caja Negra: Registro Usuarios.....	76
50	Caja Negra: Registro Galpones	76
51	Caja Negra: Registro Corrales	77
52	Caja Negra: Registro Madres	77
53	Caja Negra: Registro Machos	77
54	Caja Negra: Registro Tipo Madres	77
55	Caja Negra: Registro Inseminaciones	78
56	Caja Negra: Registro Nacimientos.....	78
57	Caja Negra: Registro Destetes.....	78
58	Caja Negra: Registro Movimientos.....	79
59	Caja Negra: Registro Mortalidad y Ventas	79
60	Casos de Prueba: Inicio de Sesión.....	80
61	Casos de Prueba: Registro de Usuario.....	81
62	Casos de Prueba: Registro de Galpones	82
63	Casos de Prueba: Registro de Corrales.....	83
64	Casos de Prueba: Registro de Madres	84
65	Casos de Prueba: Registro de Machos.....	85
66	Casos de Prueba: Registro de Tipo de Madres	86
67	Casos de Prueba: Registro Inseminaciones	88

68	Casos de Prueba: Registro Nacimientos	89
69	Casos de Prueba: Registro Destetes.....	90
70	Casos de Prueba: Registro Destetes.....	92
71	Casos de Prueba: Registro Mortalidad o Ventas.....	93
72	Registro Usuarios.....	105
73	Registro Galpones.....	105
74	Registro Corrales.....	106
75	Registro Madres.....	106
76	Registro Machos.....	106
77	Registro Tipo Madres.....	107
78	Registro Inseminaciones.....	107
79	Registro Revisiones.....	107
80	Registro Nacimientos.....	108
81	Registro Destetes.....	108
82	Registro Detalle Destetes.....	108
83	Registro Lotes.....	109
84	Registro Detalle Lotes.....	109
85	Registro Movimientos.....	109
86	Registro Mortalidad y Ventas.....	110

ÍNDICE DE FIGURAS

1	Trabajando con Extreme Programming.	11
2	Inseminación Artificial	28
3	Revisiones	28
4	Partos	29
5	Destetes	29
6	Recría	30
7	Engorde	30
8	Modelo Entidad Relación	49
9	Diagrama del Caso de Uso del Administrador	50
10	Diagrama del Caso de Uso del Empleado.....	51
11	Inicio de Sesión	52
12	Registrar Usuario	53
13	Registrar Galpones.....	53
14	Registrar Corrales	54
15	Registrar Madres.....	54
16	Registrar Machos	55
17	Registrar Tipo de Madres.....	55
18	Registrar Inseminaciones.....	56
19	Revisiones	56
20	Registrar Nacimientos.....	57
21	Registrar Destetes	57
22	Registro de Lote.....	58
23	Registro de Detalle Lote.....	58
24	Registrar Movimientos.....	59
25	Registro de Lote.....	59
26	Registro de Detalle Lote.....	60
27	Registro Mortalidad Venta.....	60
28	Aplicación de Patrones de Diseño	61
29	Sin Patrón de Diseño Facade.....	62
30	Con Patrón de Diseño Facade.....	62

31	Clase Conexión Base de Datos	63
32	Clase Conexión Base de Datos	63
33	Clase Conexión Base de Datos	63
34	Cadena de Conexión a la Base de Datos	64
35	Instanciación a la Base de Datos	64
36	Sin Patrón de Diseño Builder.....	65
37	Con Patrón de Diseño Builder	65
38	Clase BuilderMetodos.....	66
39	Clase Director	66
40	Clase DatosMetodos	67
41	Clase ConstructoMetodos.....	67
42	Instanciación del Builder.....	68
43	Sin Patrón de Diseño Adapter.....	68
44	Con Patrón de Diseño Adapter	69
45	Clase ConexionAdaptable	69
46	Interfaz del Adaptador.....	69
47	Adaptador.....	70
48	Instanciación al Adaptador	70
49	Reglas de Nombrado	71
50	Reglas de Nombrado	71
51	Reglas de Funciones.....	71
52	Reglas de Clases	72
53	Reglas de Comentarios.....	73
54	Reglas de Proceso de Error	73
55	Reglas de Proceso de Error	74
56	Reglas de Proceso de Error	74
57	Reglas de Proceso de Error	74
58	Reglas de Prueba de Unidad	75
59	Reglas de Prueba de Unidad	75
60	Prueba Funcional: Inicio de Sesión.....	80
61	Prueba Funcional: Registro de Usuario	81
62	Prueba Funcional: Registro de Galpones.....	82
63	Prueba Funcional: Registro de Corrales	83
64	Prueba Funcional: Registro de Madres.....	84
65	Prueba Funcional: Registro de Machos	85
66	Prueba Funcional: Registro Tipo de Madres	86
67	Prueba Funcional: Registro Inseminaciones	87

68	Prueba Funcional: Registro Inseminaciones	87
69	Prueba Funcional: Registro Nacimientos.....	88
70	Prueba Funcional: Registro Destetes	89
71	Prueba Funcional: Registro Lote.....	90
72	Prueba Funcional: Registro Detalle Lote.....	90
73	Prueba Funcional: Registro Movimientos.....	91
74	Prueba Funcional: Registro Lote.....	91
75	Prueba Funcional: Registro Detalle Lote.....	91
76	Prueba Funcional: Registro Mortalidad o Venta.....	92
77	Panel de Control	94
78	Características de Windows	94
79	Servidor Web IIS	95
80	Administrador IIS	95
81	Publicar el sistema.....	96
82	Carpeta creada al publicar.....	96
83	Agregar aplicación	97
84	Manual de Usuario: Inicio de Sesión.....	111
85	Manual de Usuario: Usuarios	112
86	Manual de Usuario: Galpones	113
87	Manual de Usuario: Corrales.....	114
88	Manual de Usuario: Madres	115
89	Manual de Usuario: Machos	116
90	Manual de Usuario: Tipo Madre.....	117
91	Manual de Usuario: Inseminaciones	118
92	Manual de Usuario: Revisiones.....	118
93	Manual de Usuario: Nacimientos.....	119
94	Manual de Usuario: Destetes.....	120
95	Manual de Usuario: Lote.....	120
96	Manual de Usuario: Movimientos.....	121
97	Manual de Usuario: Motalidad o Venta.....	122
98	Manual de Usuario: Reportes.....	123
99	Capacitación.....	124
100	Capacitación.....	124
101	Capacitación.....	125

Resumen Ejecutivo

Hoy en día la producción de cerdos ha ido en constante aumento en la provincia de Tungurahua así como en el Ecuador, gracias a que el tipo de clima y la zona geográfica en la que se encuentra es muy favorable para llevar a cabo esta actividad. Todas estas condiciones permiten que el producto final sea sano y con altos niveles de calidad.

Es importante mencionar que los cerdos tiene una gran variedad de productos que se pueden sacar a la población y que su demanda cada vez más va en aumento[1]. Por tal motivo el presente trabajo de investigación es el desarrollo de un sistema que permita realizar el control de la producción porcina en la granja Proinba ubicada en el Cantón Cevallos.

Para realizar el desarrollo del sistema se siguió una metodología ágil durante el proceso de desarrollo, en la parte de diseño se aplicó patrones de diseño y el código se realizó aplicando las reglas de programación limpia que necesitaba el sistema.

Abstract

Today the production of pigs has been constantly increasing in the province of Tungurahua as well as in Ecuador because the type of climate and the geographical area in which it is located is very favorable to carry out this activity. All these conditions allow the final product to be healthy and with high levels of quality.

It is important to mention that pigs have a great variety of products that can be taken out to the population and that their demand is increasingly increasing[1]. For this reason the present research work is the development of a system that allows control of pig production in the farm Proinba located in the Cevallos Canton, to carry out the development of the system followed an agile methodology throughout the development process In the design part, design patterns were applied and all the code was used the rules of clean programming needed by the system.

INTRODUCCIÓN

La actividad es asociada a la crianza, procesado y consumo del cerdo, constituye una realidad empresarial más o menos organizada, por lo que dicha actividad aún se encuentra en crecimiento.- Es de gran importancia conocer la manera en que se maneja la información, identificando el perfil de la producción tanto en los aspectos económicos y técnicos.- Por estas peculiaridades es necesario analizar los procesos de cómo se realiza la producción porcícola[2].

La información que se lleva de la granja porcícola se lo realiza de manera rudimentaria, aumentando tiempo en cada registro y al momento de buscar la información es difícil encontrarla puesto que los datos están guardados en diferentes lugares.

Para la crianza de los cerdos se realiza varios procesos, iniciando desde la etapa de inseminación de las cerdas, siguiendo a las revisiones, los partos, destetes, etapa de recría, etapa de engorde hasta finalizar con su comercialización.- Para lo cual se plantea implementar un sistema que permita controlar toda la producción de la granja porcícola y obtener información relevante que demuestre el estado en el que se encuentra la granja.

Para esta investigación se lo dividió en cinco capítulos que se muestran a continuación:

Capítulo I: se realiza el planteamiento del problema a cerca de la producción porcina en los últimos años, así como su limitación, la justificación y cuáles son los objetivos que se va a cumplir.

Capítulo II: se presentan los antecedentes investigativos de trabajos anteriormente relacionados a la producción porcina junto con la fundamentación teórica que dará los conceptos que se va a tratar.

Capítulo III: se presenta la modalidad de investigación que se va a llevar a cabo durante la realización del proyecto además de la manera que se va a recopilar la información y el procesamiento que se le dará, para al finalizar se darán actividades que se va a cumplir para lograr los objetivos planteados.

Capítulo IV: se presenta el desarrollo de la propuesta, donde se aplicará la metodología que fue seleccionada para llegar a cumplir con todos los objetivos.

Capítulo V: se tendrá las conclusiones y recomendaciones que se obtuvieron al finalizar la investigación.

CAPÍTULO 1

El problema

1.1. Tema de Investigación

Sistema informático aplicando patrones de diseño y programación limpia para la producción porcina en la empresa Proinba del Cantón Cevallos.

1.2. Planteamiento del problema

En los últimos años la manera de alimentarse de las personas ha ido variando con el tiempo, sin embargo la necesidad de consumir carne no ha cambiado en ningún aspecto.- Ya sea que se consuma pescado, carne de res, de cerdo o pollo, siendo la carne de cerdo uno de los más consumidos a nivel mundial por lo que tiene una alta demanda en el mercado, lo cual hace necesario que los procesos que se manejen y el control que se realice a la producción porcina se lo haga de una manera más sofisticada y aprovechando la tecnología actual.

A nivel mundial el sector porcino se encuentra cada vez en un mayor nivel de producción[3].- A pesar de haber tenido un aumento importante en los últimos años, debido a que algunas empresas porcícolas no cuentan con la información necesaria ni la más óptima, no permite tomar decisiones oportunas para detectar fallas, planificar la actividades, además de no tener un control de registros de producción diarios en las empresas[4].

Muchas veces los productores dedican un escaso tiempo a las actividades de gestión, por lo que se observa la ausencia de datos básicos confiables, la inexistencia de información automatizada en distintos programas no fiables o de difícil acceso al usuario, estas limitaciones imposibilita efectuar un análisis cualitativo y cuantitativo del sector porcícola con un alto grado de confiabilidad[4].

En el Ecuador la actividad porcina ha reducido considerablemente su producción, al nivel de llegar que las pequeñas empresas que realizan dicha actividad debieron cerrar sus operaciones[5].

Uno de los principales factores es el precio de la carne, puesto que en otros países como Colombia o Perú la carne del cerdo es mucho más barata.- Otro factor es

el precio de la materia prima que tiene un costo demasiado elevado, además de que el contrabando en el país está siempre latente[5].

En el país aún está a la baja dicho sector debido al poco nivel de formalidad que hay en el sector agropecuario, ya sea por una ineficiente administración o por el desconocimiento de la normativa vigente[6].

Con el fin de mejorar el rendimiento de la producción porcina, es importante orientar a los dueños de las granjas cual es el cumplimiento de la normativa que deben seguir para realizar dicha actividad, puesto que el sector porcino constituye uno de los mayores potenciales para nuestro país[6].

En la provincia de Tungurahua en el Cantón Cevallos la empresa Proinba lleva un control de los porcinos de manera rudimentaria, cada uno de los datos obtenidos se los almacena en Excel de manera desordenada lo cual al momento de solicitar algún informe la búsqueda de información es demasiado compleja, en algunas ocasiones no da el resultado esperado, puesto que los datos están en diferentes hojas lo que impide tener agilidad y rapidez al momento de saber el estado actual de la empresa porcina.

1.3. Delimitación

Delimitación de contenidos

- Área Académica: Software.
- Línea de Investigación: Desarrollo de software.
- Sublínea de Investigación: Aplicaciones web.

Delimitación espacial: El presente proyecto de investigación se realizará en la Empresa Proinba del Cantón Cevallos.

Delimitación Temporal: La presente investigación se desarrollará en el periodo Septiembre 2017 - Agosto 2018.

1.4. Justificación

La importancia del presente proyecto de investigación reside en implantar un sistema informático para la producción porcina de la empresa Proinba aplicando patrones de diseño lo que permite disminuir tiempos, aumentar la fiabilidad, disminuir el esfuerzo en el mantenimiento.- Al aplicar programación limpia en el desarrollo permite tener una estructuración o una organización correcta de los elementos del sistema, tener una legibilidad rápida y clara de lo que realiza

la codificación además de permitir tener una certeza en el funcionamiento del código.

Mediante la presente investigación se aporta de manera significativa a la transformación de la matriz productiva dentro de la empresa, aportando nuevas tecnologías con la elaboración del sistema permitiendo que la información sea manejada de mejor manera y con mayor facilidad[7].

El sistema ayuda de gran manera a optimizar tiempo y recursos, puesto que algunos de los registros que se realizan de forma manual se los realizará de manera automática, permitiendo agilidad en el trabajo y manteniendo una producción estable dentro de la empresa.

Los beneficiarios directos del proyecto son los propietarios de la empresa Proinba quienes mantendrán el control adecuado de la porcícola.

1.5. Objetivos

1.5.1. General

Implantar un sistema informático aplicando patrones de diseño y programación limpia para la producción porcina en la empresa Proinba.

1.5.2. Específicos

1. Analizar las características que debe cumplir el uso de patrones de diseño y de programación limpia al momento de desarrollar un sistema software.
2. Analizar los procesos de producción y comercialización de la granja porcícola.
3. Desarrollar el sistema informático para la producción porcícola aplicando patrones de diseño y programación limpia.
4. Implantar el sistema informático en la empresa.

CAPÍTULO 2

Marco Teórico

2.1. Antecedentes Investigativos

Para la presente investigación se han tomado proyectos relacionados con el control de la producción porcina y variables afines, mencionando los siguientes documentos:

De acuerdo al autor Dagner Arquímides Pillamango Mendoza en 2016, con el trabajo “ARQUITECTURA BASADA EN UNA CAPA DE CONTROL DE EXCEPCIONES PARA MEJORAR LA FIABILIDAD DE LA APLICACIÓN SOFTWARE DE PRÉSTAMOS BANCARIOS” estableciendo que:

“Se realizó un estudio sobre las formas de implementación de manejo de excepciones, conceptos de arquitectura, fiabilidad y de software, que permitieron mejorarlo.- Para lo cual se implementó una capa de excepciones que se pudo reutilizar en las diferentes capas de diseño en la arquitectura.- Los beneficios que se obtuvieron fueron: ahorro de tiempo, aumentando el porcentaje de tolerancia a fallos de 67 % a un 97 % y recursos económicos para los desarrolladores de software, además de tener un código limpio y que se ejecute de manera correcta”[8].

Del proyecto de Investigación realizado por los autores Carlos Máximo Castro Mendoza y Mariangel Isabella Murillo Sevillano en 2015, con el tema “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA AUTOMATIZADO PARA LA AUTOMATIZACIÓN Y CLIMATIZACIÓN DE UNA GRANJA PORCINA.” concluyen que:

“Mediante la utilización de un ciclo de dosificación de los alimentos se pudo garantizar una optimización de los recursos, dando la cantidad exacta de alimento para garantizar el físico de la cerda lactante, lo cual a través de una interfaz de comunicación remota, se puede verificar el total funcionamiento, garantizando el acceso adecuado para los procesos productivos diarios requeridos obteniendo las mejores condiciones para el desarrollo y crecimiento, reduciendo la mortalidad en los primeros años de vida”[9].

El autor Germán Andrés Cottura en 2014, mediante su proyecto de investigación con el tema “ADOPCIÓN DEL SISTEMA DE SEGUIMIENTO DE ACTIVI-

DADES PORCINAS EN LA PROVINCIA DE CÓRDOBA” establece que:

“Para conocer la información que se genera en la actividad porcina, es necesario contar con registros que permitan identificar el perfil de la producción en sus aspectos técnicos y económicos.- De tal manera que se puede obtener el mayor conocimiento de los usuarios sobre sus sistemas productivos porcinos.- Este se encuentra relacionado con la mayor implementación del programa SAP que mediante el manejo del programa informático, es fundamental para el exitoso funcionamiento del mismo”[10].

El autor David Palacio Correa en 2015, con el tema de proyecto de investigación “ELABORACIÓN DE UN SOFTWARE AGROPECUARIO PARA LA RECOPIACIÓN DE DATOS DE PRODUCCIÓN DE LA GRANJA SAN SILVESTRE” concluye que:

“Para que una granja tenga una alta productividad hay que manejar cada uno de los factores que son necesarios a controlar, los cuales deben ejecutarse de manera correcta llevando de manera clara todos los registros, logrando un control adecuado de cada una de las actividades que en ella se realice para poder detectar a tiempo posibles amenazas que puedan desfavorecer o que puedan generar problemas.- Por medio de la implementación del software ayudó a tener un mejor manejo integral de la producción, basados en principios y técnicas aplicables asegurando la protección de la higiene, salud humana y del medio ambiente con métodos económicamente factibles, seguros y confiables”[11].

El autor Cuauhtemoc Hohman Sandoval en 2014, con el proyecto “CREACIÓN DE FRAMEWORKS CON PATRONES DE DISEÑO PARA EL DESARROLLO DE APLICACIONES EMPRESARIALES” establece que:

“Para crear los frameworks se utilizaron varios patrones de diseño que permitieron ser utilizados a una gran variedad de distintas aplicaciones de TI y que admiten variar las implementaciones proporcionando soluciones más concretas a problemas de diseño.- Entre ellos se utilizó Singleton que permitió tener acceso a los datos haciendo referencia al mismo desde cualquier parte del sistema, también se utilizó Template Method que ayudo a expresar condiciones SQL simples y con el Composite combinarlas y crear condiciones más complejas.- Mientras que con el Abstract Factory permitió cambiar fácilmente entre las bases de datos y para la transaccionalidad se usó el Proxy de tal manera que libero al programador del código repetitivo de esa manera mejorando la legibilidad del código”[12].

Después de una búsqueda en las tesis realizadas en la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato no se encontraron investigaciones que tengan relación con el tema a estudio.

2.2. Fundamentación teórica

2.2.1. Patrón de Diseño

Los patrones de diseño fueron introducidos en 1995 con el libro llamado Gang of For (GoF) que en español significa “Banda de los Cuatro Autores” llamado Design Patterns – Elements of Reusable Object-Oriented Software el cuál fue escrito por los autores Erich Gamma, Richard Helm, Ralph Johnson y Jhon Vlissides con este libro se toma como referencia a cerca de los patrones de diseño.

El patrón de diseño es considerado como un diagrama de objetos que da una solución a un problema conocido y frecuente donde el diagrama de objetos se encuentra constituido por un conjunto de objetos descritos por clases y las distintas relaciones que enlazan a los objetos, generalmente los patrones responden a problemas de diseño de las aplicaciones en el marco de la Programación Orientada a Objetos (P.O.O), todo ellos se basa en soluciones conocidas y probadas cuyo diseño proviene de la experiencia de los distintos programadores, no se puede decir que existe algún aspecto teórico en los patrones sobre todo no existe una formalización de los patrones[13].

Para poder organizar el catalogo se toma en cuenta al Gof que se encuentra organizado por: Construcción, Estructuración y Comportamiento, tal como se puede observar en la Tabla 1.

Tabla 1: Clasificación Gof.

Fuente: Patrones de diseño para C#[13].

		PROPÓSITO		
		CREACIONAL	ESTRUCTURAL	COMPORTAMIENTO
ALCANCE	CLASE	Factory Method	Adapter	Interpreter Template Method
	OBJETO	Abstract Factory Builder Prototype Singleton	Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

2.2.1.1. Patrones de Construcción o Creacionales

El objetivo principal es la creación de objetos el cual consta de 5 patrones, estos son: Abstract Factory, Builder, Factory Method, Prototype y Singleton.

- Abstract Factory: Creación de objetos reagrupados sin tener la necesidad de conocer las clases concretas destinadas a la creación de objetos.
- Builder: Separa la construcción de objetos complejos de su implementación de manera que un cliente crea dichos objetos complejos con una implementación diferente.
- Factory Method: Presenta métodos abstractos para la creación de un objeto reportando a las subclases concretas la creación efectiva.
- Prototype: Crear objetos nuevos por duplicación de objetos existentes a quienes se les conoce como prototipos que disponen capacidad de clonación.
- Singleton: Asegura que de una clase existe una única instancia y proporciona un método único que la devuelve[13].

2.2.1.2. Patrones de Estructuración

Facilitan la jerarquía de clases y de sus relaciones el cual consta de 7 patrones, estos son: Adapter, Bridge, Composite, Decorator, Facade, Flyweigth y Proxy.

- Adapter: Convierte la interfaz de una clase ya existente en la interfaz esperada por los clientes también existentes que puedan trabajar de forma conjunta.
- Bridge: Separa los aspectos conceptuales de una jerarquía de clases de su implementación.
- Composite: Ofrece un marco de diseño de una composición de objetos con una profundidad de composición variable, lo cual se basa en el diseño de árbol.
- Decorator: Permite agregar dinámicamente funcionalidades suplementarias a un objeto.
- Facade: Reagrupa las interfaces de un conjunto de objetos en una interfaz unificada que resulta más fácil de utilizar.

- Flyweighth: Facilita la compartición de un conjunto importante de objetos con granularidad muy fina[13].
- Proxy: Permite proporciona un intermediario o un sustituto de manera que se pueda controlar el acceso[14].

2.2.1.3. Patrones de Comportamiento

Proporcionan soluciones para organizar las interacciones y para repartir el procesamiento entre los objetos el cual consta de 11 patrones, estos son: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method y Visitor.

- Chain of Responsibility: Su función principal es de evitar emparejar el objeto que envía una petición con el objeto que recibe la petición al permitir que más de un objeto pueda tomar la responsabilidad de la petición enviada.
- Command: Transforma una petición de un objeto de tal manera que se pueda catalogar clientes con diferentes peticiones manteniendo el registro de las peticiones y que soporte el deshacer las operaciones. Interpreter: Permite tener una representación de la gramática de algún lenguaje para usarla en un intérprete de lenguaje.
- Iterator: Permite de alguna manera acceder secuencialmente a los elementos de una colección sin poner a exposición su representación interna.
- Mediator: Permite crear un objeto que encapsula un conjunto de objetos lo que permite la organización de interacción de los objetos sin la necesidad de que entre ellos mismo se diferencien.
- Memento: Se mantiene el principio de encapsulación, captura y externaliza el estado interno de un conjunto de manera que el objeto puede regresar al estado anterior.
- Observer: Define la dependencia de uno a muchos entre objetos de tal manera que cuando un objeto cambie de estado todas las dependencias del objeto se notifiquen y actualicen automáticamente.
- State: Este puede cambiar su comportamiento al momento de cambiar su estado lo que hace parecer que el objeto cambia de clase.

- Strategy: Se crea un conjunto de algoritmos intercambiables que están encapsulados permitiendo que un algoritmo varíe independientemente de los clientes que los usan.
- Template Method: Las subclases que tienen el código se le da la opción de cambiar ciertos pasos o procedimientos sin la necesidad de cambiar la estructura del algoritmo.
- Visitor: Permite definir una operación nueva sin cambiar las clases de los elementos en los cuales se opera[15].

2.2.2. Programación Limpia

Cualquier código que se realice debe tener la facilidad de que no simplemente se ejecute sino también poder ser leído, interpretado y de fácil modificación sin importar en qué tipo de lenguaje sea codificado, cabe tomar el término que, código limpio es aquel que representa las intenciones o el propósito para que se decodifica y al desarrollo de todo el proceso de codificación se le conoce como programación limpia.

2.2.3. Metodología

Son métodos y técnicas que ayudan de manera abierta y homogénea cada una de las actividades del ciclo de vida de un proyecto de desarrollo de software, además que es un modo sistemático para implementar, mantener, administrar y gestionar el proyecto desde que surge el inicio del desarrollo hasta que se cumple con los objetivos permitiendo llegar con grandes posibilidades de éxito[16].

2.2.4. Ciclo de Vida

Son los pasos a seguir de las distintas fases intermedias que se requieren para poder validar el desarrollo de software que permite garantizar que el software cumpla con los requisitos que fueron establecidos por el cliente. Esto nace de una necesidad de resolver algún problema en particular y terminar cuando se da solución y su apropiado mantenimiento al software[17].

2.2.5. Metodología Ágil

Se puede considerar que un proceso es ágil cuando es incremental, es decir, se puede realizar entregas pequeñas y con ciclos rápidos, es cooperativo cuando el cliente y desarrolladores trabajan siempre juntos con una constante comunicación,

es sencillo mientras el método es fácil de aprender y modificar además de estar bien documentado y adaptable, que permita cambios de último momento[17].

2.2.6. Extreme Programming (XP)

Extreme Programming es una metodología ágil que promueve el trabajo en equipo potenciando así las relaciones interpersonales centrados siempre en el aprendizaje durante el desarrollo, dicha metodología se basa principalmente en la realimentación continua entre el cliente y las personas encargadas del desarrollo, permitiendo una comunicación fluida entre todos, simplicidad para dar soluciones y enfrentar los cambios que se den durante el desarrollo[18].

2.2.6.1. Ciclo de vida de XP

- Exploración: Aquí se plantean las historias de usuario que permitirán realizar la primera entrega del producto además de permitir que el equipo de desarrollo se adapte a las herramientas y tecnologías que se van a utilizar, dicha fase dura pocas semanas todo dependerá de la magnitud del proyecto y la familiarización de la tecnología con los desarrolladores[19].
- Planificación de la Entrega (Release): Se establecen las prioridades de cada una de las historias de usuario, se estima el esfuerzo que se va a realizar en cada una de ellas, se acuerda el contenido de las entregas y se determina un cronograma junto con el cliente, la entrega debe realizarse de manera rápida y unos pocos meses[19].
- Iteraciones: Está fase no debe tener más de tres semanas, en la primera iteración se puede establecer la arquitectura todo ello se realiza en base a las historias de usuario que el cliente vea conveniente, para tomarse en cuenta los elementos que conforman estas iteraciones son historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas, todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores[19].
- Producción: Está fase necesita de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea utilizado por parte del cliente, también se debe tomar en cuenta la inserción de nuevas características debido a los cambios que se vayan realizando en esta fase[19].

- **Mantenimiento:** El sistema debe estar funcionando al mismo tiempo que se desarrollan nuevas iteraciones para lo cual se necesita de tareas de soporte para el cliente, la velocidad de desarrollo puede bajar después la puesta en producción dicha fase puede requerir nuevo personal y cambios en la estructura[19].
- **Muerte del Proyecto:** Cuando ya no hay más historias de usuario por lo que hace necesario que se satisfagan otros aspectos como confiabilidad y rendimiento, se genera una documentación final y no se realizaran más cambios en la estructura. Otro motivo por el cual puede morir el proyecto puede ser porque el sistema no genera los beneficios esperados o cuando no hay medios para mantenerlo[19].

Con cada una de las fases que fueron definidas, se puede conocer cada uno de los procesos que se deben realizar durante el ciclo de vida XP tal como se puede observar en la Figura 1 aportando información relevante para ser ejecutado durante todo el desarrollo del proyecto.

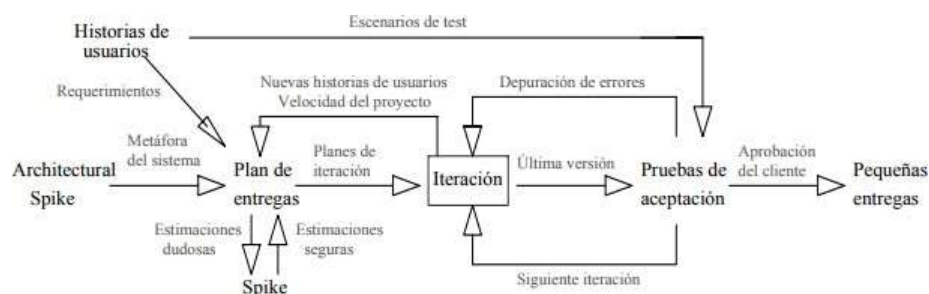


Figura 1: Trabajando con Extreme Programming.
Fuente: Introducción a la Metodología XP[20].

2.2.6.2. Valores de la metodología XP

- **Simplicidad:** Simplificar el diseño permitiendo agilizar el desarrollo y que el mantenimiento se lo realice mucho más fácil.
- **Comunicación:** Entre más simple sea el código ayudará a que la comunicación entre desarrolladores sea más simple incluyendo al cliente que también forma parte de ellos.
- **Retroalimentación (Feedback):** Durante todo el desarrollo del proyecto pueden surgir cambios de acuerdo a lo que el cliente requiera por lo que es conveniente realizar ciclos cortos que muestren resultados minimizando el riesgo de no cumplir algún requisito.

- Coraje o Valentía: En muchas ocasiones es necesario tener valentía para diseñar el código el de hoy y no dejarlo para mañana evitando así quedarse estancado en un solo punto o cuando el código diseñado no es lo más factible o correcto se lo desecha y empezar con otro nuevo[21].

2.2.6.3. Pasos de la metodología XP

- Desarrollo iterativo e incremental: Son mejoras pequeñas que se las realiza una tras otra.
- Pruebas unitarias continuas: Estas pruebas se las realiza frecuentemente repetidas y de manera automatizada en las que se incluye las pruebas de regresión es aconsejable escribir el código de las pruebas que se van a realizar antes de la codificación.
- Programación en parejas: Las tareas que se lleven a cabo durante el desarrollo es aconsejable realizarlo en parejas puesto que así mientras se escribe el código se va discutiendo y revisando las mejores alternativas de desarrollo.
- Frecuente integración del equipo de programación con el cliente o usuario: Es recomendable que una persona represente al cliente para que forme parte del equipo.
- Corrección de todos los errores antes de añadir nueva funcionalidad: Realizar entregas frecuentes.
- Refactorización del código: Reescribir algunas partes del código para mayor aumentar la legibilidad y el mantenimiento pero si cambiar el comportamiento.
- Propiedad del código compartido: Promueve que el personal de desarrollo pueda corregir y extender cualquier parte del proyecto, las frecuentes pruebas de regresión que se realicen garantizan las detecciones de posibles errores que se puedan encontrar.
- Simplicidad del código: Realizar algo simple y tener poco trabajo para realizar cualquier cambio sea lo más óptimo y aceptable[21].

2.3. Propuesta de Solución

En este proyecto se plantea el desarrollo de un sistema informático aplicando patrones de diseño y programación limpia que permita simplificar los procesos de la producción porcina que se realizan dentro de la empresa.

CAPÍTULO 3

Metodología

3.1. Modalidad de la investigación

El presente proyecto de investigación seguirá las siguientes modalidades:

- La modalidad de campo debido a que para la obtención de la información necesaria se acude directamente a la empresa.
- La modalidad bibliográfica o documental para poder obtener la información necesaria profundizando el tema a investigar mediante el uso de fuentes de artículos, libros, bibliotecas virtuales, repositorios y en temas de tesis similares aportando información valiosa para el proceso de investigación.
- La modalidad aplicada se realizará al aplicar los conocimientos que fueron obtenidos en todo el ciclo de estudio universitario.
- La modalidad de Desarrollo se utilizará para generar una aplicación web.

3.2. Población y muestra

La presente investigación por su característica no requiere población y muestra.

3.3. Recolección de información

Para que la presente investigación sea ejecutada, se realizara un trabajo en colaboración con las personas de la empresa para poder conocer de manera más detallada los requerimientos de los procesos que se deberán considerar para el desarrollo del proyecto, de esa manera se obtendrá la información más óptima y correcta.

De igual manera para poder recabar más información se utilizará la técnica de investigación documental de tipo informativa lo que permitirá obtener información valiosa y relevante de diferentes fuentes confiables además de analizar cada una de ellas que servirán de la mejor manera para el desarrollo del proyecto.

3.4. Procesamiento de la información

El procesamiento y análisis de los datos se los realizara de la siguiente manera:

- Recopilación de la información por medio de la investigación documental.
- Analizar la información obtenida.
- Interpretación de los resultados.

3.5. Desarrollo del Proyecto

1. Análisis de las distintas características de la programación limpia que se deben tomar en cuenta al momento de desarrollar software.
2. Análisis de los patrones de diseño más utilizados al momento de desarrollar software.
3. Desarrollo de la metodología XP

3.1. Fase de Exploración

- Estudio de los procesos para la producción porcina.
- Análisis de la comercialización de los porcinos.
- Análisis de los requerimientos del sistema.
- Realizar las historias de usuario.

3.2. Fase de Planeación de Entrega

- Estimar el tiempo de duración y priorización de cada una de las historias de usuario.

3.3. Fase de Iteraciones

- Iteraciones de las historias de usuario.
- Planeación de entregas.

3.4. Fase de Producción

- Diseño del aplicativo web para la producción porcina aplicando patrones de diseño.
- Desarrollo del aplicativo web aplicando programación limpia.

3.5. Fase de Pruebas y Mantenimiento

- Pruebas de aceptación del sistema informático.

4. Implantación del sistema informático en la empresa Proinba

- Configuración los equipos y el sistema.
- Puesta en marcha del sistema.
- Capacitación.

CAPÍTULO 4

Desarrollo de la propuesta

4.1. Programación Limpia

No siempre es fácil crear código limpio para poder tenerlo se debe practicarlo constantemente y fallar, solo de esa manera se puede conseguir un código aceptable en ello se incluye ver la forma en que otros practican o los desarrollan incluyendo las fallas que cometen y observar la forma en que se recuperan, la organización y la decisión que se tome en ese instante pueden ser fatales en el desarrollo del código[22].

“Me gusta que el código sea elegante y eficaz. La lógica debe ser directa para evitar errores ocultos, las dependencias deben ser mínimas para facilitar el mantenimiento, el procesamiento de errores completo y sujeto a una estrategia articulada, y el rendimiento debe ser óptimo para que los usuarios no tiendan a estropear el código con optimizaciones sin sentido. El código limpio hace bien una cosa.” Bjarne Stroustrup[22].

Para realizar un buen código se podría tomar la regla del Boy Scout

Con el hecho de escribir código de manera correcta no basta, siempre se debe actualizar o limpiar con el tiempo, debido a que con el tiempo deja de ser el más óptimo o se corrompe.

Por lo cual los norteamericanos practican una regla y que se puede aplicar al código: “Dejar el campamento más limpio de lo que se ha encontrado”[22].

Lo que nos indica que cada vez que se lea un código se lo debería entregar mucho más limpio de lo que se ha recibido de tal manera que se evitara que sea corrompido.

4.1.1. Reglas Generales de Programación Limpia

Mediante la información obtenida se puede tener de manera clara y sencilla los beneficios que ofrece la programación limpia al momento de desarrollar software, tal como se puede observar en las tablas 2, 3, 4, 5, 6 y 7

Tabla 2: Reglas de Programación Limpia.

Fuente: Darío Ramírez

NOMBRADO	
REGLA	OBJETIVO
Nombres con sentido	Los nombres deben revelar las intenciones para el cual van a ser utilizados, ya sea para una variable, función, método o clase, si un nombre requiere algún tipo de comentario significa que no revela el cometido.
Evitar la desinformación	Al momento de desarrollar código se deben evitar dejar pistas que sean falsas lo cual dificulta el significado del código, no se deben colocar palabras que se alejen de lo que se pretende hacer en el desarrollo.
Realizar distinciones con sentido	Podrían utilizar los prefijos mientras cada uno de ellos tenga un sentido distinto, permitiendo reconocer el para qué fueron creados.
Usar nombres que se puedan pronunciar	Es poder explicar de manera clara y sencilla algún nombre evitando así que se cambie el significado de la palabra al momento de su pronunciación.
Usar nombres que se puedan buscar	Se deben colocar nombres con palabras explícitas de tal manera que se pueda realizar una búsqueda sencilla y encontrarlo rápidamente.
No exceder con el atractivo	Evitar que las palabras sean mal interpretadas o se las tomen a burla, siempre se debe optar por claridad antes que el entretenimiento.

Tabla 3: Reglas de Programación Limpia.

Fuente: Darío Ramírez

FUNCIONES	
REGLA	OBJETIVO
Tamaño reducido	Toda función que se realice debe ser lo más pequeña posible que se pueda de esa manera se podrá leer mejor y con mayor claridad el propósito de la misma.
Bloques y sangrado	Todos los bloques deben contener una línea de longitud que pueda señalar la invocación de una función, de esta manera no solo se reduce el tamaño sino que también añade valor documental.
Realizar distinciones con sentido	Podrían utilizar los prefijos mientras cada uno de ellos tenga un sentido distinto, permitiendo reconocer el para qué fueron creados.
Verbos y palabras clave	Mejora la explicación de su cometido de la misma manera como el orden y el cometido de los argumentos los cuales deben tener un verbo y un sustantivo.
Mejor excepciones que devolver códigos de error	Permite que el código de procesamiento del error se pueda simplificar y permita separar el código de ruta.

Tabla 4: Reglas de Programación Limpia.

Fuente: Darío Ramírez

CLASES	
REGLA	OBJETIVO
Organización de clases	Es muy importante organizar todo por clases para que el código no se mezcle, realice de manera independiente cada función y que pueda ser leído fácilmente.
Encapsulación	Permite que las variables y funciones puedan ser privadas y solo se utilicen en esa clase.
Las clases deben ser de tamaño reducido	Que cada clase sea lo más reducido posible para evitar la confusión y que sea de fácil manejo para cualquier actualización.
Principio de responsabilidad única	Una clase solo debe cambiar por uno y solo un motivo manteniendo la responsabilidad y la directriz para el tamaño de la clase.
Organizar los cambios	Si se organiza las clases reducirá el riesgo de que el código no funcione por los cambios que se realice.

Tabla 5: Reglas de Programación Limpia.

Fuente: Darío Ramírez

COMENTARIOS	
REGLA	OBJETIVO
El realizar un comentario no compensa el error	Es muy superior un código expresivo y claro con mínimos comentarios que dar la explicación del error que comete cierta línea de código.
Explicarse en el código	Comentar a todo lo que se haga no es lo ideal de hecho se sabe mucho más al crear código mismo que en escribir el comentario.
Código de calidad	Escribirlos correctamente son de gran beneficio y necesarios.
Comentarios legales	Por motivos legales en algunas corporaciones al momento de crear el código obligan a crear determinados comentarios.
Comentarios informativos	Proporcionar una información básica con un comentario es de gran ayuda para conocer la funcionalidad de una función, método, etc.
Explicar la intención	Realizar un comentario es más que información útil sobre la implementación y proporcionar la intención de una decisión.
Advertir las consecuencias	En ocasiones comentar ciertas secciones de código advirtiendo de las consecuencias que se produce al ejecutarlos.
Asignaciones y menciones	Permiten recordar quien ha añadido qué y cuándo fue realizado lo ideal es escribirlo en el código fuente.

Tabla 6: Reglas de Programación Limpia.

Fuente: Darío Ramírez

REGLAS DE PROCESO DE ERROR	
REGLA	OBJETIVO
Usar excepciones en lugar de códigos devueltos	Es mejor crear una excepción que permita detectar el error, por lo que el código de invocación es más limpio y en su lógica no se va a oscurecer por el control de errores.
Crear la instrucción try-catch-finally	Permite que al ejecutar el código en la parte del try de una instrucción try-catch-finally se indica que la ejecución se puede cancelar en cualquier momento y a continuación retornar en el catch.
Ofrecer contexto junto a las excepciones	Redactar los mensajes de error que sean informativos junto a las excepciones, se debería colocar la operación fallida y el tipo de fallo.
Evitar pasar y devolver null	Se evita generar problemas al momento de invocar al método o a la función que realice el llamado y al evitar pasar null el código se ejecutará sin producir errores en tiempo de ejecución.

Tabla 7: Reglas de Programación Limpia.

Fuente: Darío Ramírez

REGLAS DE PRUEBAS DE UNIDAD	
REGLA	OBJETIVO
Las tres leyes DGP	<p>Primera Ley: No debe crear código de producción hasta que haya creado una prueba de unidad que falle.</p> <p>Segunda Ley: No debe crear más de prueba de unidad que baste como fallida y no compilar se considera un fallo.</p> <p>Tercera Ley: No debe crear más código de producción que el necesario para superar la prueba de fallo actual.</p>
Realizar pruebas limpias	Para realizar este tipo de pruebas no siempre es necesario que las variables contengan nombres adecuados o que las funciones sean breves y descriptivas no que el código este bien diseñado, basta con que funcione y contenga el código de producción pero el tener pruebas incorrectas es igual o peor que no tener prueba alguna.
Una afirmación por regla	Que cada una de las pruebas que se realice debe llegar a una misma conclusión que se entienda de forma rápida y sencilla.
Un solo concepto por prueba	Evita tener funciones extensas en las que se prueben cosas diferentes una tras otra lo más aconsejable es dividir las pruebas diferentes que permitirá conocer su funcionalidad de mejor manera para que se utiliza cada una de ellas.
F.I.R.S.T.	<p>Rapidez: La distintas reglas deben ser rápidas y ejecutarse rápidamente</p> <p>Independencia: Las pruebas se deben ejecutar de manera independiente y en cualquier orden, no deben depender de otras pruebas.</p> <p>Repetición: En cualquier entorno las pruebas se deben repetir sin ningún problema.</p> <p>Validación Automática: Cada una de las pruebas que se realice debe tener un resultado booleano, es decir, o fallan o aciertan.</p> <p>Puntualidad: La pruebas que se vaya a realizar deben crearse en el momento exacto, antes de que el código entre a producción, siempre se debe tomar en cuenta que se debe diseñar código que se pueda probar.</p>

4.2. Patrones de Diseño

En mucho de los sistemas orientados a objetos que se han diseñado se han podido encontrar objetos de comunicación recurrente y patrones de clases cuya finalidad es de resolver problemas específicos de diseño lo cual al momento de diseñar lo hace más reusable y flexible ante cualquier cambio que se pueda realizar a un futuro de esa manera siempre obtener un software de calidad.

Elementos principales de un patrón de diseño

Nombre

Nos indica de manera clara el problema del diseño.

El Problema

Indica en qué momento se debe aplicar el dicho patrón.

La Solución

Permite describir los elementos que componen sus relaciones, la colaboración, el diseño y las responsabilidades que estos implican.

Mediante la investigación realizada se determinó los patrones de diseño más utilizados al momento de desarrollar software tal como se puede observar en las tablas 8,9 y 10

Tabla 8: Patrones de Diseño.

Fuente: Darío Ramírez

PATRONES DE CREACIÓN	
Mediante este tipo de patrones ayuda que el sistema pueda ser independiente de cómo se crean, representan sus objetos o de que se componen.	
Factory Method	
PROPÓSITO	MOTIVACIÓN
Es aquella que ayuda a decidir a qué clase instanciar es decir, permite que la clase difiera la instanciación a las subclases.	Mantener las relaciones entre los objetos mediante la utilización de clases abstractas
Abstract Factory	
PROPÓSITO	MOTIVACIÓN
Sin la necesidad de especificar las clases concretas se puede proporcionar una interfaz para poder crear clases de objetos dependientes o relacionales.	Definir la interfaz que permita soportar distintas ventanas.
Builder	
PROPÓSITO	MOTIVACIÓN
Construir un objeto complejo donde solo se pueda especificar su contenido y tipo, ocultando detalles de la construcción de dicho objeto.	Un ejemplo claro puede ser al convertir un tipo de texto en distintas representaciones diferentes las cuales a un futuro sean necesario definir nuevos formatos.
Singleton	
PROPÓSITO	MOTIVACIÓN
Indicar que una clase tiene solo una finalidad especifica además de otorgar un acceso global.	Necesariamente algunas de las clases deben cumplir una sola finalidad.

Tabla 9: Patrones de Diseño.

Fuente: Darío Ramírez

PATRONES ESTRUCTURALES	
Aquí se trata de evitar que algunos cambios que se realicen a alguna aplicación no se alteren de ninguna manera las relaciones entre los objetos, las cuales el uso se encuentra relacionadas con las interfaces que puedan soportar los objetos.	
Adapter	
PROPÓSITO	MOTIVACIÓN
Convertir la interfaz de una clase en otra que esperan los clientes.	Aunque una interfaz no sea exactamente a lo que se requiere de una aplicación se puede utilizar su biblioteca.
Bridge	
PROPÓSITO	MOTIVACIÓN
Mediante la implementación de una abstracción permite que cada uno evolucione de manera independiente.	Por medio de la herencia, la abstracción que se realice permitirá tener varias implementaciones.
Proxy	
PROPÓSITO	MOTIVACIÓN
Mediante la implementación de una abstracción permite que cada uno evolucione de manera independiente.	Es la sustitución de un objeto con otro objeto para controlar el acceso hacia este.
Composite	
PROPÓSITO	MOTIVACIÓN
Crear objetos más complejos mediante una composición recursiva de otros objetos similares.	Algunas aplicaciones gráficas poseen componentes que se puedan agrupar para formar unos más complejos.
Decorator	
PROPÓSITO	MOTIVACIÓN
Permite asignar nuevas responsabilidades de manera dinámica.	No es necesario repartir nuevas responsabilidades a toda la clase sino solo a los objetos individuales necesarios.
Facade	
PROPÓSITO	MOTIVACIÓN
Simplificar los accesos a un conjunto de objetos de tal manera que todos los clientes puedan comunicarse con dicho conjunto.	Minimizar las comunicaciones y dependencias entre los subsistemas.

Tabla 10: Patrones de Diseño.

Fuente: Darío Ramírez

PATRONES DE COMPORTAMIENTO	
Mediante este tipo de patrones ayuda que el sistema pueda ser independiente de cómo se crean, representan sus objetos o de que se componen.	
Template Method	
PROPÓSITO	MOTIVACIÓN
Construir la base de un algoritmo sin tomar en cuenta los detalles que tienen las subclases.	El tipo de algoritmo que se realice no es necesario mostrar la manera de cómo se creó o los detalles de las subclases que pueda contener.
Strategy	
PROPÓSITO	MOTIVACIÓN
Tener un conjunto de algoritmos, que sea encapsulados de tal manera que se pueda intercambiar uno por otro.	Un procesador de textos, en la cual una clase de composición es la que se encargara de gestionar las líneas del texto en el cuál se le delega un algoritmo de compositor.

4.3. Metodología de Desarrollo Extreme Programming (XP)

4.3.1. XP se estructura en tres prácticas

Metodología de Programación

Diseño sencillo, testing, refactorización y codificación con estándares.

Metodología de Equipo

Propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente.

Metodología de Procesos

Cliente, entregas frecuentes y planificación[16].

4.3.2. Metodología XP en el Desarrollo de Software

La elección de la metodología se la realiza en base a las siguientes tablas 11 y 12, tomando en cuenta el USO, CAPACIDAD DE AGILIDAD, APLICABILIDAD, PROCESOS Y PRODUCTOS, así como valorándolos de acuerdo al siguiente rango: 1= (si cumple) y 0= (no cumple).

Tabla 11: Tabla Comparativa de Metodologías

Fuente: Darío Ramírez

COMPARACIÓN DE METODOLOGÍAS				
ASPECTO		XP	SCRUM	KANBAN
USO	Respeto de las fechas de entrega.	1	1	0
	Cumplimiento de los requisitos.	1	1	1
	Satisfacción del usuario final.	1	1	1
	Aumento de la productividad.	1	1	1
CAPACIDAD DE AGILIDAD	Iteraciones cortas	1	1	0
	Colaboración	1	1	1
	Integración de los cambios.	1	1	1
	El plan de trabajo puede cambiar.	0	1	1
	Los recursos humanos pueden cambiar.	1	1	1
	Cambiar los indicadores.	1	1	1

Tabla 12: Tabla Comparativa de Metodologías

Fuente: Darío Ramírez

APLICABILIDAD	Tamaño del proyecto. (Pequeño, Mediano o Grande)	1	1	1
	Complejidad del proyecto.	1	1	1
	Los riesgos del proyecto.	1	1	0
	El tamaño del equipo. (Grande, Mediano o Pequeño)	1	1	1
	Interacción con el cliente	1	1	1
	Interacción con el usuario final.	1	1	0
	Interacción con los miembros del equipo.	1	1	0
PROCESOS Y PRODUCTOS	Gestión de proyectos.	0	1	0
	Descripción de procesos.	1	1	0
	Definición de requisitos.	1	1	1
	Modelado	1	1	1
	Código	1	1	1
	Pruebas unitarias.	1	1	1
	Pruebas de integración.	1	1	0
	Pruebas de aceptación.	1	1	0
	Pruebas del sistema.	1	1	0
TOTAL		24	26	16

De acuerdo al valor obtenido del rango de criterios Scrum tiene la valoración más alta, sin embargo se utilizará Extreme Programming (XP) puesto que su valor no está muy distante de Scrum, además de que es el más adecuado para aplicarlo al presente trabajo de investigación.

Los roles que maneja XP son los siguientes:

- Cliente: Proinba
- Programador: Darío Ramírez
- Encargado de pruebas (Tester): Ing. Ricardo Pazmiño
- Encargado de seguimiento (Tracker): Ing. Ricardo Pazmiño
- Entrenador (Coach): Ing. Edison Álvarez
- Gestor (Big Boss): Sra. Aida Catuta

Las fases de la metodología XP son las siguientes:

- Exploración
- Planeación de entregas
- Iteraciones
- Producción
- Pruebas

4.4. Desarrollo del Sistema del Control Porcícola basado en la Metodología Ágil XP

Para poder desarrollar el proyecto de investigación es necesario seguir todas las fases de la metodología, de esa manera permitirá tener un sistema que controle todo el proceso de la producción porcícola.- Así obteniendo beneficios para la granja porcícola.

4.4.1. Fase de Exploración

En esta fase el cliente es la persona encargada de definir los requisitos que el sistema necesita, esto se lo puede realizar mediante historias de usuario, además de que los programadores son las personas que se encargan de estimar los tiempos de desarrollo lo que permite tener una visión general del sistema.

4.4.1.1. Estudio de los procesos para la producción porcina.

Para poder obtener una carne de alta calidad se debe cumplir cada uno de los procesos minuciosamente y con estrictos controles en el crecimiento de los cerdos, para lo cual se debe seguir los siguientes procedimientos.

- Inseminación Artificial
- Revisiones
- Partos
- Destetes
- Recría
- Engorde

Inseminación Artificial

Principalmente la inseminación artificial es la colecta, almacenamiento y el procesamiento a bajas temperaturas de semen para luego introducirlo en el aparato reproductor de la cerda con la finalidad de su fecundación.

La cerda a la cual se va a inseminar debe encontrarse en celo lo cual sucede cada 21 días, en el primer día la cerda se encuentra muy receptiva al macho, para que la cerda se encuentre en el punto máximo de estimulación es necesario que el verraco (cerdo macho) este cerca, de esta manera la inseminación que se lo realice, sea de manera exitosa como se puede observar en la Figura 2.



Figura 2: Inseminación Artificial
Elaborado por: Pregón Agropecuario[23]

Revisiones

Para que una cerda se encuentre en estado de fecundación debe pasar tres revisiones, la primera revisión se la realiza a las 3 semanas de su inseminación, la segunda se la realiza a las 6 semanas y la última revisión se la hace a las 9 semanas en la cual es la confirmación de su fecundación.- Si no pasa la revisión en cualquiera de sus etapas se la vuelve a inseminar y se realiza el mismo proceso, así como se puede observar en la Figura 3.



Figura 3: Revisiones
Elaborado por: La Porcicultura[24]

Partos

Una vez terminada la etapa de preñez la cerda es bajada a maternidad para su parto, al momento de nacer los lechones generalmente debe estar una persona responsable del cuidado de los lechones, lo cual al momento de su nacimiento se debe tener mucho cuidado para evitar que la mortalidad sea elevada o ninguna, tal como se observa en la Figura 4.



Figura 4: Partos
Elaborado por: Todo Cerdos[25]

Destete

Este se lo realiza pasado las tres semanas del nacimiento de los lechones, durante esta etapa se alimenta solo de la madre, de igual manera siempre se debe estar pendiente de cada uno de ellos y tenerles un cuidado especial, para luego pasar a la etapa de recría, tal como se observa en la Figura 5.



Figura 5: Destetes
Elaborado por: Patricia Roldán[26]

Recría

Se encarga del crecimiento de los lechones destetados, en la cual se puede dividir por lotes de 40 a 50 lechones dependiendo el espacio que se tenga, como se puede observar en la Figura 6.



Figura 6: Recría
Elaborado por: Antonio Palomo[27]

Engorde

Esta es la etapa final en la cual se realiza el engorde de los cerdos, en esta instancia llegan con 24 semanas de edad y con el peso deseado aunque dependiendo de la alimentación se puede tomar una semana más para su venta, se lo puede observar en la Figura 7.



Figura 7: Engorde
Elaborado por: El Productor[28]

En cada una de las etapas tanto a la madre como a los lechones se les aplica sus respectivas vacunas, además de que la alimentación que se les da es distinta de acuerdo a la etapa en la que se encuentren.

4.4.1.2. Análisis de la comercialización de los porcinos.

Para la venta de los cerdos se los puede realizar desde la etapa de Recría, puesto que por motivos de salud del lechoncito, la venta no se lo puede realizar antes, para de esa manera al final obtener un producto de alta calidad.

Para la etapa de Recría se lo vende de acuerdo a la edad en semanas que tenga el lechón, además de sacar un costo de la crianza del lote, con ello venderlo de tal manera que se pueda obtener una ganancia significativa.

En la etapa de Engorde la venta se la realiza de acuerdo a la cantidad de kilogramos que el cerdo ha ganado durante toda la etapa de crecimiento, la cual

dura unas 24 semanas, luego el valor total se lo obtiene al multiplicarlo por el valor actual al mercado.

4.4.1.3. Procesos del sistema para el control porcino

Personal Administrativo

Registrar Usuarios

Personal Empleado

Registro de Galpones

Registro de Corrales

Registro de las Madres

Registro de los Machos o Pajuelas

Registro Tipo de la Madre

Registro de las Inseminaciones

Registro de las Revisiones

Registro de los Nacimientos

Registro de los Destetes

Registro Detalle Destetes

Registro Lotes

Registro Detalle Lotes

Registro de Movimientos

Registro de Mortalidad y Ventas

Historias de Usuario

Las historias de usuario son descripciones cortas y que se encuentran esquematizadas que permite de fácil manera que los requerimientos del sistema sean convertidos en tareas.

Personal Administrativo

Tabla 13: Registro Usuarios

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador
Nombre de la historia: Registro de usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los Usuarios y debe tener datos como: Usuario y Contraseña	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 14: Criterios de Aceptación: Registro Usuarios

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador
Cuando:	Se realiza el registro de los datos perteneciente a los Usuarios y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador
Cuando:	Se realiza el cambio de los datos perteneciente a los Usuarios y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Personal Empleado

Tabla 15: Registro Galpones

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador o Empleado
Nombre de la historia: Registro de Galpones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registro debe pertenecer a los galpones y debe tener datos como: Número y Tipo.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 16: Criterios de Aceptación: Registro Galpones

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a los Galpones y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a los Galpones y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 17: Registro Corrales

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador o Empleado
Nombre de la historia: Registro de Corrales	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los corrales y debe tener datos como: Número y Tipo.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 18: Criterios de Aceptación: Registro Corrales

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a los Corrales y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a los Corrales y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 19: Registro Madres

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador o Empleado
Nombre de la historia: Registro de las Madres	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a las Madres y debe contar con los siguientes datos: Número, Raza, Número de Partos, Galpón y Corral.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 20: Criterios de Aceptación: Registro Madres

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a las Madres y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a las Madres y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 21: Registro Machos

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 5	Usuario: Administrador o Empleado
Nombre de la historia: Registro del Macho o Pajuela	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los Machos o Pajuelas y debe tener datos como: Número, Raza, Galpón y Corral.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 22: Criterios de Aceptación: Registro Machos

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a los Machos y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a los Machos y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 23: Registro Tipo Madres

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador o Empleado
Nombre de la historia: Registro del Tipo de Madre	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer al Tipo de Madre y debe tener datos como: Tipo y Descripción.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 24: Criterios de Aceptación: Registro Tipo Madres

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a los Tipos de Madres y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a los Tipos de Madres y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 25: Registro Inseminaciones

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 7	Usuario: Administrador o Empleado
Nombre de la historia: Registro de las Inseminaciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registro debe pertenecer a las Inseminaciones y debe tener datos como: Número, Día, Semana, Número Madre, Tipo Madre, Macho o Pajuela.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 26: Criterios de Aceptación: Registro Inseminaciones

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a las Inseminaciones y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a las Inseminaciones y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 27: Registro Revisiones

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 8	Usuario: Administrador o Empleado
Nombre de la historia: Registro de las Revisiones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a las Revisiones y debe contar con los siguientes datos: Número Cerda, Número Inseminación, 1ra Revisión, 2da Revisión, Confirmación.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 28: Criterios de Aceptación: Registro Revisiones

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realizó el registro de los datos perteneciente a las Revisiones y se presionó el botón guardar al realizar el registro de las Inseminaciones
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a las Revisiones y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

Tabla 29: Registro Nacimientos

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 9	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Nacimientos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registro debe pertenecer a las Nacimientos y debe contar con los siguientes datos: Número Nacimiento, Semana Inseminación, Semana Real Nacimiento, Día Real Nacimiento, Lechones Vivos Nacimiento, Lechones Muertos Nacimiento, Lechones Momias Nacimiento, Adoptados Nacimiento, Donados Nacimiento.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 30: Criterios de Aceptación: Registro Nacimientos

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a los Nacimientos y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a las Nacimientos y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 31: Registro Destetes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Destetes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registro debe pertenecer a las Destetes y debe contar con los siguientes datos: Número Lote, Semana, Día, Galpón, Corral.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 32: Criterios de Aceptación: Registro Destetes

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a los Destetes y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a los Destetes y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 33: Registro Detalle Destetes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 11	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Detalles Destete	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer al Detalles Destete y debe contar con los siguientes datos: Número Madre, Número Partos, Cant. Lechones Machos, Cant. Lechones Hembras, Peso Total	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 34: Criterios de Aceptación: Registro Detalle Destetes

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realizó el registro de los datos perteneciente al Detalle Destetes al haber presionado el botón guardar al realizar el registro de los Destetes.
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a los Detalle Destetes y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

Tabla 35: Registro Lotes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 12	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Lotes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los Lotes, debe contar con los siguientes datos: Código, Número. Corral Recría, Corral Engorde, Vendido.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 36: Criterios de Aceptación: Registro Lotes

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realizó el registro de los datos perteneciente a los Lotes al haber presionado el botón guardar al realizar el registro de los Destetes.
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a los Lotes y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

Tabla 37: Registro Detalle Lotes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 13	Usuario: Administrador o Empleado
Nombre de la historia: Registro del Detalle Lotes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registro debe pertenecer a Detalle Lotes, debe contar con los siguientes datos: Código Lote, Número Lote, Acción.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 38: Criterios de Aceptación: Registro Detalle Lotes

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realizó el registro de los datos perteneciente al Detalle Lotes al haber presionado el botón guardar al realizar el registro de los Destetes.
Entonces:	Los datos se almacenan en la base de datos

Tabla 39: Registro Movimientos

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 14	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Movimientos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Programador responsable: Darío Ramírez	
Descripción: El registro debe pertenecer a Movimientos, debe contar con los siguientes datos: Galpón Origen, Corral Origen, #Lote, #Cerdos, Semana, Día, Galpón Destino, Corral Destino.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 40: Criterios de Aceptación: Registro Movimientos

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a la Movimientos y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a Movimientos y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

Tabla 41: Registro Mortalidad y Ventas

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 15	Usuario: Administrador o Empleado
Nombre de la historia: Registro de la Mortalidad y Ventas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a Mortalidades y Ventas, debe contar con los siguientes datos: Mortalidad, Venta, Galpón, Corral, Lote, #Cerdos, Semana, Día, Número Nota, Observación.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 42: Criterios de Aceptación: Registro Mortalidad y Ventas

Fuente: Darío Ramírez

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el registro de los datos perteneciente a la Mortalidad y Ventas y se presiona el botón guardar
Entonces:	Los datos se almacenan en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Se realiza el cambio de los datos perteneciente a la Mortalidad y Ventas y se presiona en actualizar
Entonces:	Los datos se actualizarán en la base de datos

CRITERIOS DE ACEPTACIÓN	
Dado:	Usuario de tipo administrador o empleado
Cuando:	Al presionar en eliminar, saldrá un mensaje de confirmación y al presionar aceptar
Entonces:	Los datos se eliminaran de la base de datos

La aceptación de las Historias de Usuario se puede observar en el Anexo A.

4.4.2. Fase de Planeación de Entregas

Aquí se da la prioridad a cada una de las historias de usuarios y poder realizar una estimación de los tiempos que van a durar cada una de las historias, como se puede observar en la tabla 43.

Tabla 43: Estimación de las Historias de Usuarios

Fuente: Darío Ramírez

Nº	HITORIA DE USUARIO	DÍAS ESTIMADOS	HORAS ESTIMADAS	SEMANAS ESTIMADAS	ROLES
1	Registro de Usuarios	3	8	0.6	Programmer /Tracker
2	Registro Galpones	1	8	0.2	Programmer /Tracker
3	Registro Corrales	1	8	0.2	Programmer /Tracker
4	Registro Madres	3	8	0.6	Programmer /Tracker
5	Registro Machos	3	8	0.6	Programmer /Tracker
6	Registro Tipo Madre	1	8	0.2	Programmer /Tracker
7	Registro Inseminaciones	3	8	0.6	Programmer /Tracker
8	Registro Revisiones	3	8	0.6	Programmer /Tracker/Big Boss
9	Registro Nacimientos	3	8	0.6	Programmer /Tracker
10	Registro Destetes	4	8	0.8	Programmer /Tracker
11	Registro Detalle Destetes	3	8	0.6	Programmer /Tracker
12	Registro Lote	2	8	0.4	Programmer /Tracker
13	Registro Detalle Lote	2	8	0.4	Programmer /Tracker/Big Boss
14	Registro Movimientos	5	8	1	Programmer /Tracker
15	Registro Mortalidad y Ventas	5	8	1	Programmer /Tracker/Big Boss
TIEMPO TOTAL ESTIMADO		42	120	8.4	

Los roles que intervienen son:

- Programmer: Darío Ramírez
- Tracker: Ing. Ricardo Pazmiño
- Big Boss: Sra. Aida Catuta

4.4.3. Fase de Iteraciones

Una iteración no puede pasar de las tres semanas de duración, al finalizar cada iteración se debe realizar un entregable y que sea funcional del sistema que se está desarrollando.

4.4.3.1. Iteraciones de las Historias de Usuario

Tabla 44: Iteraciones de las Historias de Usuarios

Fuente: Darío Ramírez

ITERACIÓN	N°	HISTORIAS DE USUARIO	PRIORIDAD ENTREGA	DEPENDENCIA(N° HITORIA DE USUARIO)	RIESGO(ALTA-MEDIA-BAJA)	DÍAS ESTIMADOS	HORAS ESTIMADAS	SEMANAS ESTIMADAS
PRIMERA	1	Registro Usuarios	1	Ninguna	Alta	3	8	0.6
	2	Registro Galpones	1	Ninguna	Media	1	8	0.2
	3	Registro Corrales	1	2	Media	1	8	0.2
	4	Registro Madres	2	Ninguna	Alta	3	8	0.6
	5	Registro Machos	2	Ninguna	Alta	3	8	0.6
	6	Registro Tipo Madre	1	Ninguna	Media	1	8	0.2
	7	Registro Inseminaciones	3	4,5,6	Alta	3	8	0.6
	8	Registro Revisiones	3	7	Alta	3	8	0.6
	TOTAL ITERACIÓN						18	64
SEGUNDA	9	Registro Nacimientos	1	7,8	Alta	3	8	0.6
	10	Registro Destetes	2	9	Alta	4	8	0.8
	11	Registro Detalle Destetes	3	10	Media	3	8	0.6
	12	Registro Lote	3	10	Media	2	8	0.4
	13	Registro Detalle Lote	3	10	Media	2	8	0.4
TOTAL ITERACIÓN						14	40	2.8
TERCERA	14	Movimientos	1	7,8,9,10,12	Alta	5	8	1
	15	Mortalidad y Ventas	2	7,8,9,10,12	Alta	5	8	1
	TOTAL ITERACIÓN						10	16

4.4.3.2. Plan de Entregas

Tabla 45: Plan de Entregas

Fuente: Darío Ramírez

Procesos	N°	HISTORIA DE USUARIO	DÍAS ESTIMADOS	HORAS ESTIMADAS	SEMANAS ESTIMADAS	N° ITERACIÓN	N° ENTREGA
Administración	1	Registro de Usuarios	3	8	0.6	1	1
Administración o Empleados	2	Registro Galpones	1	8	0.2	1	1
	3	Registro Corrales	1	8	0.2	1	1
	4	Registro Madres	2	8	0.4	1	2
	5	Registro Machos	2	8	0.4	1	2
	6	Registro Tipo Madre	1	8	0.2	1	1
	7	Registro Inseminaciones	2	8	0.4	1	3
	8	Registro Revisiones	2	8	0.2	1	3
	9	Registro Nacimientos	2	8	0.4	2	1
	10	Registro Destetes	3	8	0.6	2	2
	11	Registro Detalle Destetes	2	8	0.4	2	3
	12	Registro Lote	2	8	0.4	2	3
	13	Registro Detalle Lote	2	8	0.4	2	3
	14	Registro Movimientos	2	8	0.4	3	1
	15	Registro Mortalidad y Ventas	2	8	0.6	3	2

4.4.4. Fase de Producción

En esta fase es donde se desarrolló el sistema web para el control de la granja porcícola, con cada uno de los requerimientos que fueron brindados por parte de la empresa.

4.4.4.1. Base de Datos

Para el diseño de la base de datos se utilizó el modelo entidad relación, mediante el cual nos permite observar de manera gráfica la percepción a un problema del mundo real, mediante las relaciones que hay entre cada una de sus entidades,

permite comprender funcionalidad del sistema así como las interacciones que hay entre cada entidad.

En la figura 8 se puede observar el diseño de la base de datos.

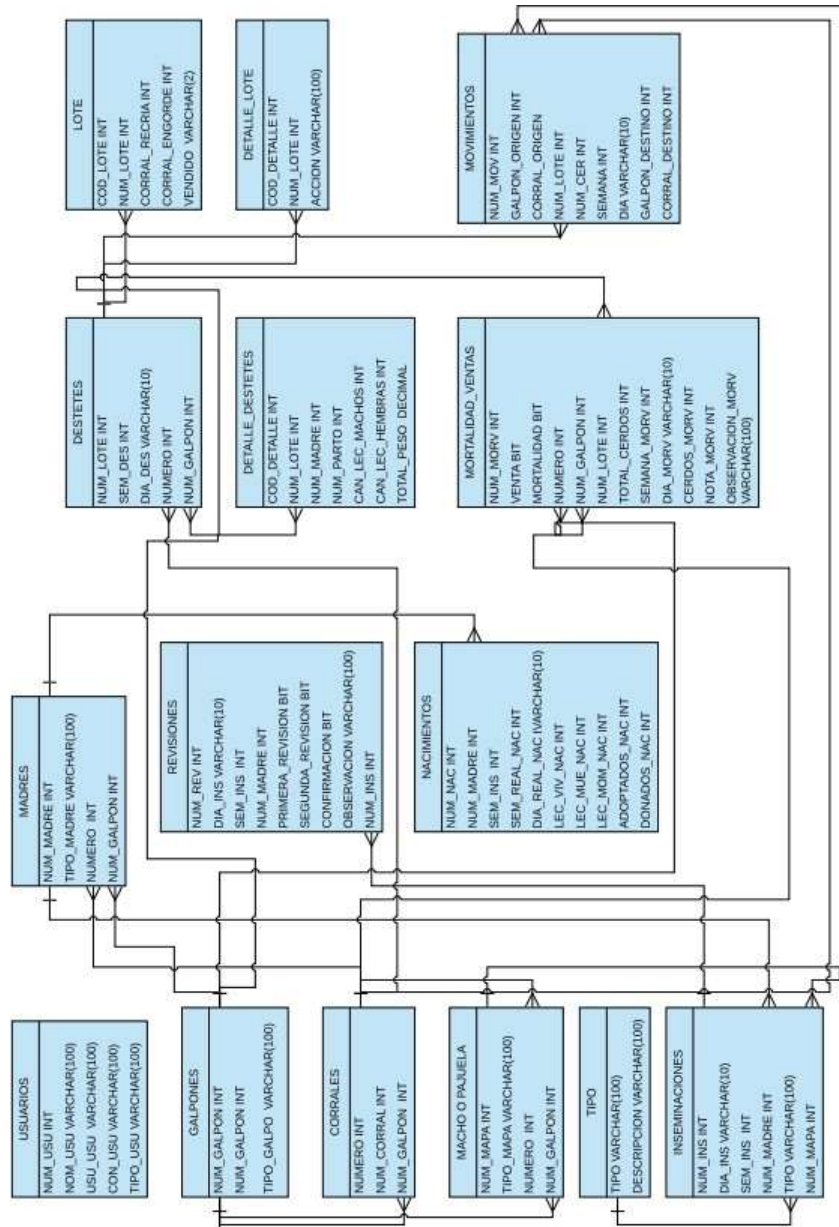


Figura 8: Modelo Entidad Relación
Elaborado por: Darío Ramírez

4.4.4.2. Diagrama Casos de Uso

Por medio del diagrama se puede conocer cuáles son los actores que van a ser encargados del sistema así como cada uno de los procesos que van manejar cada uno de ellos, tal como se puede observar en la figura 9 y 10.

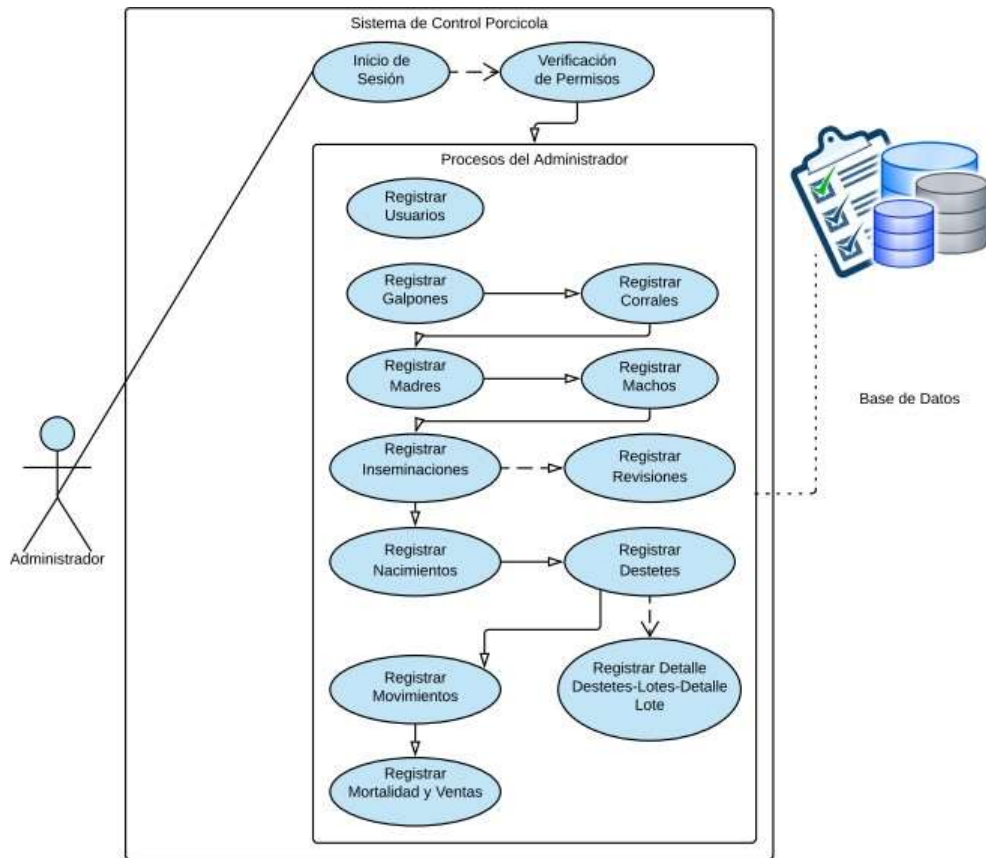


Figura 9: Diagrama del Caso de Uso del Administrador
Elaborado por: Darío Ramírez

Como se muestra en la tabla 46 se puede observar la especificación para el caso de uso del administrador.

Tabla 46: Caso de Uso del Administrador

Fuente: Darío Ramírez

Nombre:	Procesos de registros del administrador
Actores:	Administrador
Precondiciones	El usuario debe estar guardado en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1.- El usuario debe ingresar por medio de un usuario y contraseña. 2.- El sistema verificará los permisos de ese usuario. 3.- El usuario puede realizar el registro de un nuevo usuario y darle el permiso de un administrador o empleado, también podrá registrar todos los procesos que intervienen en el control porcícola.
Flujo Alternativo	El sistema presenta mensajes de información.
Poscondiciones	El sistema le permite ingresar al usuario y tener un completo control.

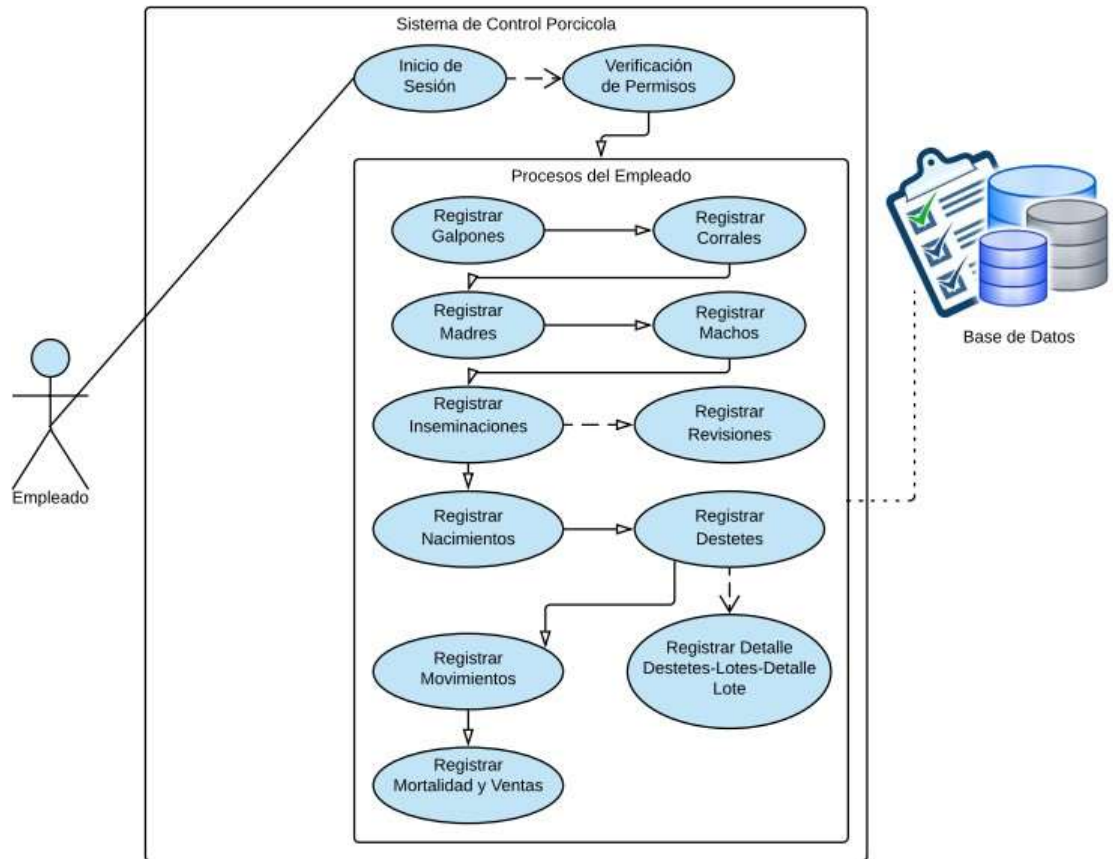


Figura 10: Diagrama del Caso de Uso del Empleado
Elaborado por: Darío Ramírez

Como se muestra en la tabla 47 se puede observar la especificación para el caso de uso del empleado.

Tabla 47: Caso de Uso del Empleado

Fuente: Darío Ramírez

Nombre:	Procesos de registros del empleado
Actores:	Empleado
Precondiciones	El usuario debe estar guardado en la base de datos. Solo debe tener acceso a los procesos del control porcícola.
Flujo Normal	1.- El usuario debe ingresar por medio de un usuario y contraseña. 2.- El sistema verificará los permisos de ese usuario. 3.- El usuario puede realizar el registro de todos los procesos que intervienen en el control porcícola.
Flujo Alternativo	El sistema presenta mensajes de información.
Poscondiciones	El sistema le permite ingresar al usuario.

4.4.4.3. Diseño de las Interfaces

Inicio de Sesión

Por medio de un usuario y contraseña permitirá que el usuario ingrese al sistema de acuerdo a los permisos que se le hayan otorgado, para que de esa manera maneje solo los procesos para el cual fueron asignados cada uno.

The image is a screenshot of a web application's login page. At the top, there is a blue header with the text 'Inicio de Sesión'. Below the header, there are two input fields: 'Usuario' with a user icon and 'Contraseña' with a lock icon. A blue button labeled 'Ingresar' is positioned below the password field.

Figura 11: Inicio de Sesión
Elaborado por: Darío Ramírez

Al iniciar la sesión de acuerdo al usuario que ingrese tendrá acceso a todo el sistema o solo para los cuales el usuario fue dado el permiso, si es como administrador tendrá la capacidad de ingresar un nuevo usuario dándole los permisos de administrador o simplemente como un empleado además de controlar todos los procesos del control porcino el, si es como empleado solo tendrá acceso a los procesos.

Registrar Usuarios

Al ingresar con el perfil de administrador, se podrá realizar el registro de un nuevo usuario, otorgándole los permisos de administrador o empleado al nuevo usuario, para que pueda acceder al sistema.

Registro Usuario

Nombre

Usuario

Contraseña

Tipo

Figura 12: Registrar Usuario
Elaborado por: Darío Ramírez

Ya sea que se ingrese al sistema con el perfil de administrador o con el perfil de empleado, podrá realizar todos los registros de la producción porcina de la empresa.

Registrar Galpones

Aquí se realizará el registro de un nuevo galpón, colocando el tipo de galpón correspondiente, mientras que el número de galpón es automático.

Galpones

Número Galpón

Tipo Galpón

Figura 13: Registrar Galpones
Elaborado por: Darío Ramírez

Registrar Corrales

Para realizar el registro de un corral, solo se deberá escoger el galpón que previamente fue registrado y el número correspondiente de corral será de manera automática.

Corrales

Galpón

Corral #

Guardar Cancelar

Figura 14: Registrar Corrales
Elaborado por: Darío Ramírez

Registrar Madres

Para registrar una nueva madre todos los campos deben ser llenados, además de seleccionar el galpón y el corral de destino.

Madres

Número Madre

Raza Madre

Número de Partos

Galpón

Corral

Guardar Cancelar

Figura 15: Registrar Madres
Elaborado por: Darío Ramírez

Registrar Machos

Para registrar un nuevo macho se debe llenar todos los campos, además de seleccionar el galpón y el corral de destino.

Figura 16: Registrar Machos
Elaborado por: Darío Ramírez

Registrar Tipo de Madres

El tipo de madre es como la cerda se encuentra en ese momento, es decir, si ya ha sido destetada (se le quito ya a los lechones), quedada (la inseminación no fue exitosa) o es primeriza (es la primera vez que se insemino).

Figura 17: Registrar Tipo de Madres
Elaborado por: Darío Ramírez

Registrar Inseminaciones

Se registra las inseminaciones que se ha ido realizando en la semana a cada una de las cerdas.

Inseminaciones

Día Inseminación:

Semana Inseminación:

Número Madre:

Tipo Madre:

Tipo Macho o Pajuela:

Figura 18: Registrar Inseminaciones
Elaborado por: Darío Ramírez

Al momento en el que se registró una nueva inseminación, de manera automática los datos de las revisiones también fueron guardados en la base de datos.- Aquí el usuario deberá ir chequeando cada revisión que se realice a las cerdas, si fueron o no revisadas, también podrá colocar si la cerda fue quedada, es decir, no paso la revisión, para que sea nuevamente inseminada la cerda.- Como se puede observar en la figura 19.

	#MADRE	SEMANA INS.	DIA INS.	1ra REVISION	2da REVISION	CONFIRMACION	OBSERVACION
Totales	210	34	11-01-2019	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 19: Revisiones
Elaborado por: Darío Ramírez

Registrar Nacimientos

Se registrará todos los nacimientos que haya en la semana.

Figura 20: Registrar Nacimientos
Elaborado por: Darío Ramírez

Registrar Destetes

Se registrará todos los destetes que haya en la semana.

Figura 21: Registrar Destetes
Elaborado por: Darío Ramírez

Al registrar un nuevo destete, de manera automática los datos del lote fueron guardados, tomando en cuenta que después de realizar el destete se van a recría y el lote no se encuentra vendido. Como se puede observar en la figura.- Como se puede observar en la figura 22.

Lote					
<input type="text" value="#Lote"/> <input type="button" value="Buscar"/>					
		#LOTE	#CORRAL RECRÍA	#CORRAL ENGORDE	VENDIDO
Editar	Eliminar	219	2	0	N

Figura 22: Registro de Lote
Elaborado por: Darío Ramírez

Aquí solo se registrará la acción que se realizó, que puede ser ingreso o un movimiento.- Al momento de realizar el destete la acción que se guardará será de Ingreso. Como se puede observar en la figura 23.

Detalle	
#LOTE	ACCIÓN
219	INGRESO

Figura 23: Registro de Detalle Lote
Elaborado por: Darío Ramírez

Registrar Movimientos

Después de la etapa de recría, se realizará el movimiento de los cerdos, es decir, se los pasará a la etapa de engorde para lo cual se registra todos los cambios de los cerdos que haya en la semana.- Como se puede observar en la figura 24.

Movimientos

DESDE

Galpón: Seleccione un Galpón ▼ Corral: Seleccione un Corral ▼

Lote: Lote # Cerdos: # Cerdos

Semana: Semana Dia: Dia

HACIA

Galpón: Seleccione un Galpón ▼ Corral: Seleccione un Corral ▼

Guardar Cancelar

Figura 24: Registrar Movimientos
Elaborado por: Darío Ramírez

Una vez que los datos de los movimientos fueron guardados, de manera automática se llena el campo de engorde que se encuentra en la misma tabla del Lote.- Como se puede observar en la figura 25.

Lote

#Lote: [] Buscar

		#LOTE	#CORRAL RECRÍA	#CORRAL ENGORDE	VENDIDO
Editar	Eliminar	219	2	1	N

Figura 25: Registro de Lote
Elaborado por: Darío Ramírez

Al realizar el registro del movimiento, también se guardará la acción que fue realizada, que en este caso es un Movimiento, que se registra en la misma tabla de Detalle Lote.- Como se puede observar en la figura.

Detalle	
#LOTE	ACCIÓN
219	INGRESO
219	MOVIMIENTO

Figura 26: Registro de Detalle Lote
Elaborado por: Darío Ramírez

Registrar Mortalidad y Ventas

Aquí es donde se registrará una venta o una mortalidad que haya en la semana.

Mortalidad y Ventas ^

VENTA MORTALIDAD

Galpón	<input type="text" value="Seleccione un Galpón"/>	Lote	<input type="text" value="Lote"/>
Corral	<input type="text" value="Seleccione un Corral"/>	# Cerdos	<input type="text" value="# Cerdos"/>
Semana	<input type="text" value="Semana"/>	Día	<input type="text" value="Día"/>
# Cerdos	<input type="text" value="X Cerdos"/>	Número Nota	<input type="text" value="Número Nota"/>
Observación	<input type="text" value="Observación"/>		

Figura 27: Registro Mortalidad Venta
Elaborado por: Darío Ramírez

4.4.4.4. Patrones de diseño en el sistema web

En el punto 4.2 se realizó el análisis de los patrones de diseño que más se utilizan en el desarrollo de software.- Los patrones de diseño que se eligió son: Facade, Builder y Adapter, de acuerdo a la necesidad que el sistema requería.

Al desarrollar software ya no es suficiente solo digitar el código y que al final el sistema se ejecute de manera correcta.- Se necesita de otras herramientas que ayuden a mejorar el código para que sea más eficiente, tal como se puede utilizar los patrones de diseño, que permiten mejorar el diseño del sistema.

Pero también se puede aplicar reglas de programación limpia al momento de realizar el código, ya que ayuda de gran manera a que el desarrollo sea más claro y ordenado.

Tanto el uso de patrones de diseño y las reglas de programación limpia permiten tener un complemento entre ambos al momento de desarrollar el software, para de esa manera tener como resultado un sistema ágil y efectivo, pero también con código limpio.

Por medio de la figura 28 nos ayudará a comprender el interacción que tiene el patrón de diseño con la programación limpia al momento de desarrollar el sistema.

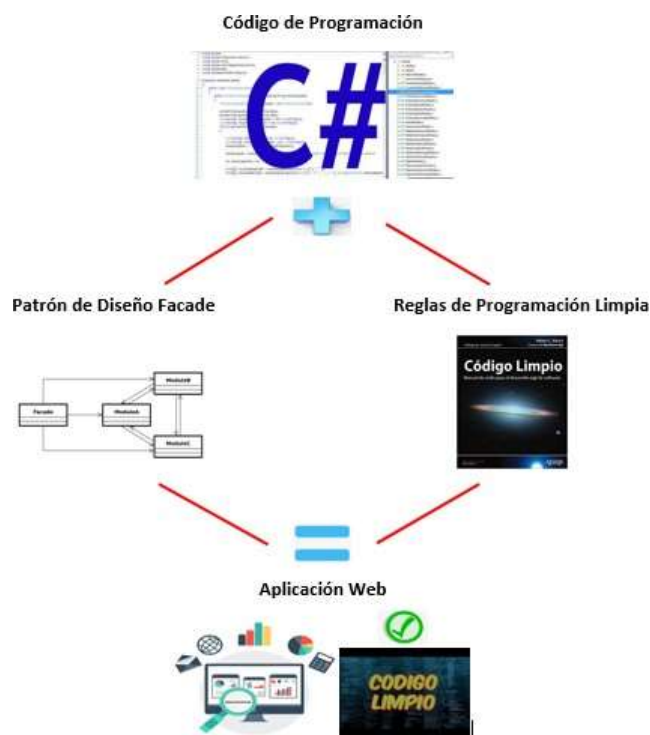


Figura 28: Aplicación de Patrones de Diseño
Elaborado por: Darío Ramírez

4.4.4.5. Patrón Facade

El patrón Facade se eligió porque permite simplificar el acceso a un conjunto de clases, de tal manera que una sola clase podrá ser utilizada por todos los clientes, para tener comunicación con ese conjunto de clases.

Sin utilizar el patrón de diseño facade, se realizar una cadena de conexión individual por cada Web Form nuevo que se desea agregar, lo cual hace que sea repetitivo y menos eficiente, porque al cambiar la conexión se las deberá realizar a cada una de ellas presentado molestias y lentitud al realizar el mantenimiento del sistema.- Como se puede observar en la figura 29.

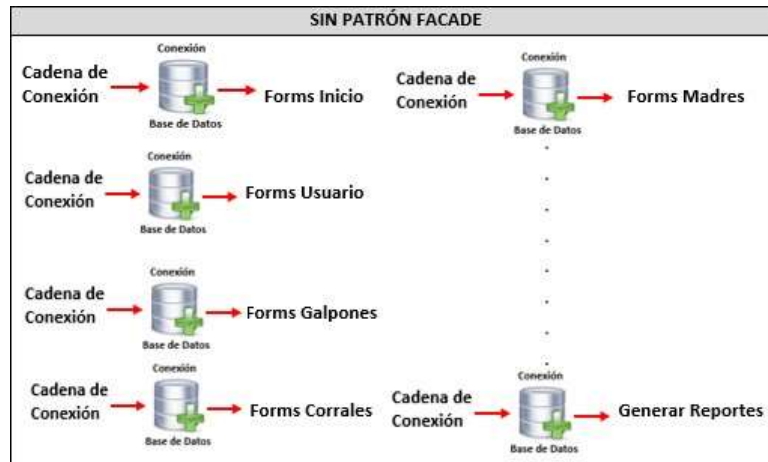


Figura 29: Sin Patrón de Diseño Facade
Elaborado por: Darío Ramírez

Con el uso del patrón de diseño facade solo se debes realizar una cadena de conexión evitando que sea repetitivo y más eficiente, porque al cambiar la conexión solo deberá realizarlo en nuestra cadena de conexión principal sin interferir en el resto del sistema permitiendo tener agilidad y rapidez al realizar el mantenimiento del sistema.- Como se puede observar en la figura 30.

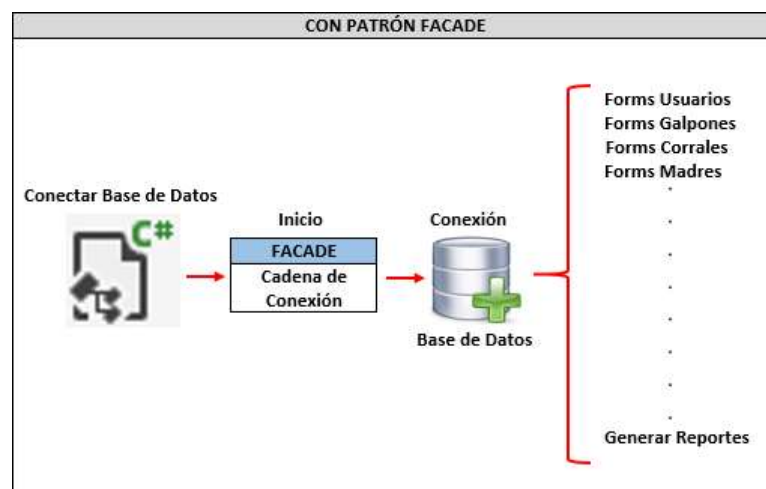


Figura 30: Con Patrón de Diseño Facade
Elaborado por: Darío Ramírez

Para poder crear el Patrón Facade se creó una Clase llamada ConectarBaseDatos dentro de un BussinesObjetc, en cuál se creó datos privados que solo podrán ser accedidos desde la misma clase, también se creó métodos públicos estáticos que permitirán crear nuestra cadena de conexión. La figura 31, 32 y 33 representa nuestra clase.

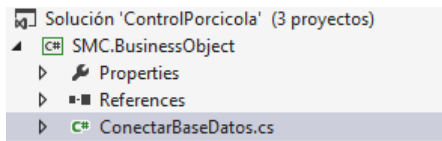


Figura 31: Clase Conexión Base de Datos
Elaborado por: Darío Ramírez

```

namespace SMC.BusinessObject
{
    SMC.PresentationLayer(5)
    {
        public class ConectarBaseDatos
        {
            #region Datos

            private static string _cadenaConexion;

            //Datos adicionales
            private static string _servidor;
            private static string _baseDatos;
            private static string _usuario;
            private static string _contraseña;
            private static string _parametro;

            #endregion
        }
    }
}

```

Figura 32: Clase Conexión Base de Datos
Elaborado por: Darío Ramírez

```

#region Propiedades
/// <summary>
/// Permite manipular toda la información que contenga la propiedad
/// "ConnectionString" del objeto "Connection".
/// </summary>
65 referencias
public static string CadenaConexion
{
    get{
        return _cadenaConexion;}
    set{
        _cadenaConexion = value;}
}
0 referencias
public static string Parametro...
/// <summary> ...
0 referencias
public static string Servidor...
0 referencias
public static string BaseDatos...
/// <summary> ...
0 referencias
public static string Usuario...

```

Figura 33: Clase Conexión Base de Datos
Elaborado por: Darío Ramírez

En la figura 34 es donde se creará nuestra cadena de conexión que nos servirá para todo el sistema.

```

public partial class Inicio : System.Web.UI.Page
{
    0 referencias
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection conexion = new SqlConnection();

        try
        {
            //Guardar la cadena de conexion en la propiedad estatica para que mantenga para toda la aplicacion.
            ConectarBaseDatos.CadenaConexion = "Data Source=NICTRON;Initial Catalog=PROINBA;User ID=sa;Password=sa";

            conexion.ConnectionString = ConectarBaseDatos.CadenaConexion;

            //Abrir la conexion.
            conexion.Open();
        }
    }
}

```

Figura 34: Cadena de Conexión a la Base de Datos
Elaborado por: Darío Ramírez

Se debe crear una instanciación para conectarse a la base desde cualquier Web Form, como se observa en la figura 35.

```

public DataTable GetDataPadres()
{
    SqlDataAdapter dta = new SqlDataAdapter();
    //Instanciar un "Connection".
    SqlConnection conexion = new SqlConnection();
    conexion.ConnectionString = ConectarBaseDatos.CadenaConexion;

    DataSet ds = new DataSet();
    SqlCommand cmd = new SqlCommand("SELECT M.NUM_MAPA, M.TIPO_MAPA, G.TIPO_GALPO, C.NUM_CORRAL FROM Macho_Pajuela M inne

    dta.SelectCommand = cmd;
    dta.SelectCommand.Connection = conexion;
    dta.Fill(ds);
    return ds.Tables[0];
}

```

Figura 35: Instanciación a la Base de Datos
Elaborado por: Darío Ramírez

4.4.4.6. Patrón Builder

El patrón Builder se eligió porque permite construir objetos complejos que solo podrán especificar su contenido y tipo, de esa manera ocultando detalles de la construcción.

Sin usar el Patrón Builder para construir los reportes, se lo puede realizar en una sola clase, de tal manera que el cliente conoce todo el proceso de como se crean esos reportes, cómo se observa en la figura 36.

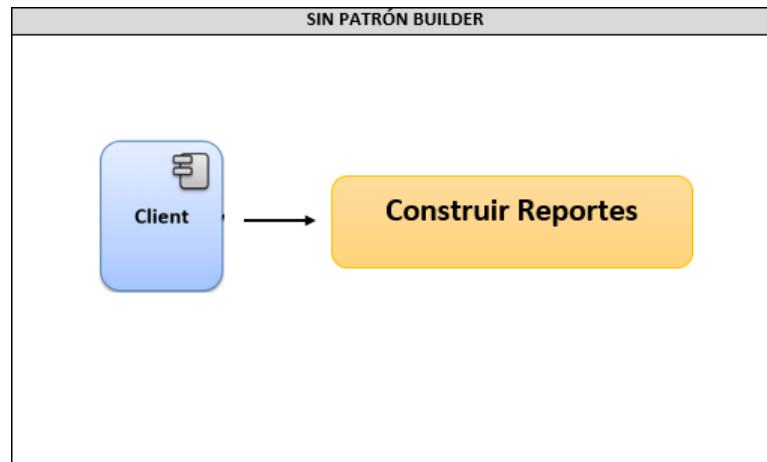


Figura 36: Sin Patrón de Diseño Builder
Elaborado por: Darío Ramírez

Con el uso del Patrón Builder, se divide las partes que van a ser creadas de manera individual y que al final se puede ensamblar todo creando el objeto completo.- Además de que los procesos de creación se guardan en un solo punto, permitiendo crear representaciones diferentes, así ocultando los detalles del proceso de construcción, como se observa en la figura 37.

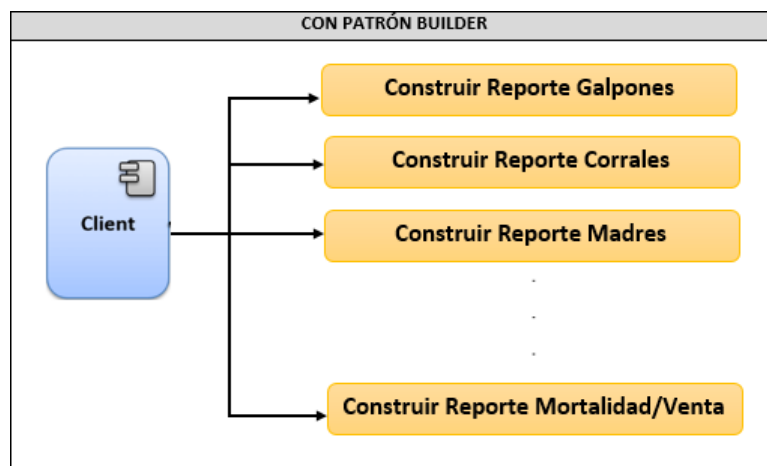


Figura 37: Con Patrón de Diseño Builder
Elaborado por: Darío Ramírez

Para crear el patrón Builder se crea una clase abstracta BuilderMetodos, que tendrá los datos de los métodos, además de tener los métodos que serán construidos por la clase que los va a heredar.

```

public abstract class BuilderMetodos
{
    protected DatosMetodos datosMetodos;

    0 referencias
    public DatosMetodos getDatosMetodos()
    {
        return this.datosMetodos;
    }

    1 referencia
    public void crearNuevosMetodos()
    {
        this.datosMetodos = new DatosMetodos();
    }

    //Métodos que seran contruidos por la clase que hereden
    2 referencias
    public abstract void contruirCorrales();
    2 referencias
    public abstract void contruirGalpones();
}

```

Figura 38: Clase BuilderMetodos
Elaborado por: Darío Ramírez

A continuación, se crea la clase Director, que es donde se crea el método que va a tener de parámetro de la clase Builder.

```

1 referencia
public class Director
{
    //Se crea el constructor de la clase
    0 referencias
    public Director()
    {
    }

    //Se crea el metodo que va a tener de parametro de la clase Builder
    0 referencias
    public void construir(BuilderMetodos builder)
    {
        builder.crearNuevosMetodos();

        builder.contruirCorrales();
        builder.contruirGalpones();
    }
}

```

Figura 39: Clase Director
Elaborado por: Darío Ramírez

Luego, se crea la clase DatosMetodos, que es donde se realiza los datos y las propiedades de cada método.

```

public class DatosMetodos
{
    //Se inicializa las propiedades
    private DataTable corrales;
    private DataTable galpones;

    // Se crea el constructor de la clase
    1 referencia
    public DatosMetodos()
    {
    }

    //Son las propiedades de cada metodo
    0 referencias
    public DataTable getCorrales()
    {
        return this.corrales;
    }

    0 referencias
    public void setCorrales(DataTable corrales)
    {

```

Figura 40: Clase DatosMetodos
Elaborado por: Darío Ramírez

En la clase ConstructorMetodos es donde se realiza la herencia de los métodos, en el cual también se crean los métodos que se encuentran en el BuilderMetodos.

```

//Se realiza la heredación de los metodos
2 referencias
public override void contruirCorrales(){
}

2 referencias
public override void contruirGalpones()
{
}

//Se crean los metodos
0 referencias
public DataTable GetDataCorrales()
{
    SqlDataAdapter dta = new SqlDataAdapter();
    //Instanciar un "Connection".
    SqlConnection conexion = new SqlConnection();
    conexion.ConnectionString = ConectarBaseDatos.CadenaConexion;

```

Figura 41: Clase ConstructoMetodos
Elaborado por: Darío Ramírez

Para finalizar se crean los objetos del Builder junto al ConstructorMetodos, se instancian los métodos y se los llama.

```

protected void btnCorralesBuilder_Click(object sender, EventArgs e)
{
    //Crear el objeto Director
    Director metodos = new Director();

    //Crear los objetos ConcreteBuilder
    BuilderMetodos corrales = new ConstructorMetodos();
    BuilderMetodos galpones = new ConstructorMetodos();

    //Construir el corral y el galpón
    metodos.construir(corrales);
    DatosMetodos getCorrales = corrales.getDatosMetodos();
    metodos.construir(galpones);
    DatosMetodos getGalpones = galpones.getDatosMetodos();

    DatosMetodos obtenerCorrales = new DatosMetodos();
    DatosMetodos obtenerGalpones = new DatosMetodos();

    ReportViewer1.LocalReport.Refresh();
}

```

Figura 42: Instanciación del Builder
Elaborado por: Darío Ramírez

4.4.4.7. Patrón Adapter

El patrón Adapter se eligió porque permite que un conjunto de clases con interfaces distintas puedan cooperar entre sí y es la que los clientes esperan utilizar.

Sin usar el Patrón Adapter, el cliente puede ver todo el código y hacer cualquier tipo de modificación en la clase original, ya que no hay otra interfaz con cual interactuar, como se observa en la figura 43.

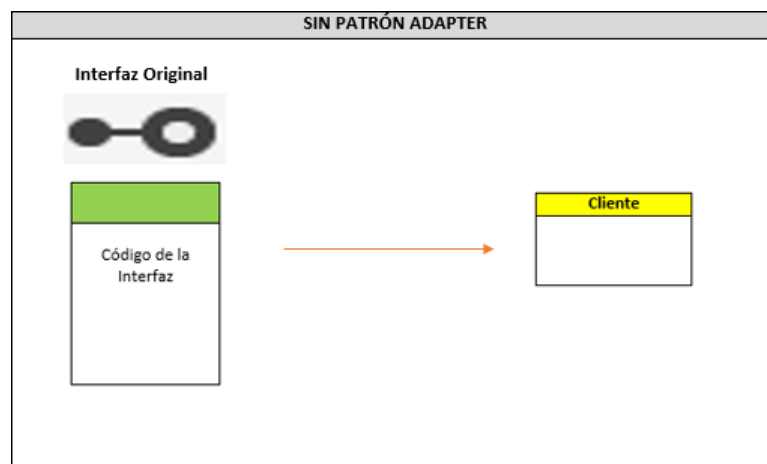


Figura 43: Sin Patrón de Diseño Adapter
Elaborado por: Darío Ramírez

Con el Patrón Adapter permite convertir la interfaz de una clase, en una interfaz que el cliente espera, lo que evita tener que conocer la clase que se utiliza para implementar esa interfaz. cómo se observa en la figura 44.

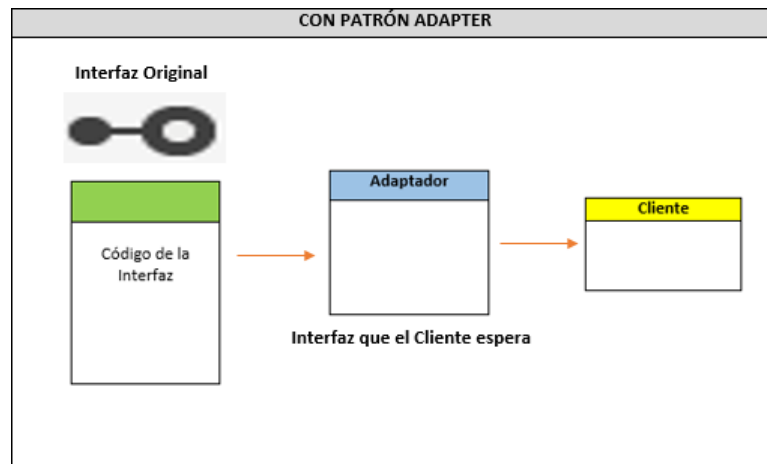


Figura 44: Con Patrón de Diseño Adapter
Elaborado por: Darío Ramírez

Para crear el Patrón Adapter se crea la clase ConexionAdaptable que es la que contendrá todos los métodos necesarios.

```
public class ConexionAdaptable
{
    private string conexion;
    1 referencia
    public ConexionAdaptable()
    {
    }

    1 referencia
    public string obtenerConexion()
    {
        return this.conexion;
    }
}
```

Figura 45: Clase ConexionAdaptable
Elaborado por: Darío Ramírez

Se crea la interfaz, donde se define el método que implementará el objeto de tipo Adaptador.

```
1 referencia
public interface IAdaptador
{
    1 referencia
    public abstract void conexion();
}
```

Figura 46: Interfaz del Adaptador
Elaborado por: Darío Ramírez

Luego se crea la clase Adaptador que hereda de la interfaz, y la que el cliente utilizará.

```

public class Adaptador : IAdaptador
{
    ConexionAdaptable conexionadaptable = new ConexionAdaptable();

    1 referencia
    public Adaptador()
    {
    }

    1 referencia
    public override void conexion()
    {
    }

    1 referencia
    public void getConexion()
    {
        SqlConnection conexion = new SqlConnection();
        conexion.ConnectionString = ConectarBaseDatos.CadenaConexion;
        conexionadaptable.obtenerConexion();
    }
}

```

Figura 47: Adaptador
Elaborado por: Darío Ramírez

Al finalizar en el cliente se utiliza el Adaptador realizando la instanciación y llamando al método de esa clase.

```

public void CargarddlGalpones()
{
    Adaptador conexion = new Adaptador();
    conexion.getConexion();
    {
        DataTable dt = new DataTable();
        string query = "SELECT NUM_GALPON, TIPO_GALPO FROM Galpones";
        SqlCommand cmd = new SqlCommand(query, conexion);
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(dt);

        ddlGalpones.DataSource = dt;
        ddlGalpones.DataValueField = "NUM_GALPON";
        ddlGalpones.DataTextField = "TIPO_GALPO";
        ddlGalpones.DataBind();
    }
}

```

Figura 48: Instanciación al Adaptador
Elaborado por: Darío Ramírez

4.4.4.8. Programación Limpia en el Desarrollo del Sistema

Para tener un código limpio se aplicó varias reglas de programación limpia durante todo el desarrollo del sistema, que se detallan a continuación:

Nombres con sentido, Nombres que se pueden pronunciar y No exceder con el atractivo

El método representa para lo que fue creado que es cargar los galpones, también se observa el nombre de la clase que es Conectar Base de Datos, incluyendo la

variable que representa la conexión.

```
public void CargarddlGalpones()
{
    //Instanciar un "Connection".
    SqlConnection conexion = new SqlConnection();
    conexion.ConnectionString = ConectarBaseDatos.CadenaConexion;
    {
        DataTable dt = new DataTable();
        string query = "SELECT NUM_GALPON, TIPO_GALPO FROM Galpones";
        SqlCommand cmd = new SqlCommand(query, conexion);
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(dt);

        ddlGalpones.DataSource = dt;
        ddlGalpones.DataValueField = "NUM_GALPON";
        ddlGalpones.DataTextField = "TIPO_GALPO";
        ddlGalpones.DataBind();
    }
}
```

Figura 49: Reglas de Nombrado
Elaborado por: Darío Ramírez

Realizar distinciones con sentido

Se utiliza prefijos que representan para que fueron creados, como el txtBuscarSemana que es un texto y btnBuscar que es un botón.

```
<div align="left">
<asp:TextBox ID="txtBuscarSemana" runat="server" placeholder="Semana" onkeypress
<asp:TextBox ID="txtBuscarNota" runat="server" placeholder="Nota" onkeypress="re
<asp:TextBox ID="txtBuscarObservacion" runat="server" placeholder="Observación"
<asp:Button ID="btnBuscar" runat="server" Text="Buscar" class="btn btn-app btn-g
...
```

Figura 50: Reglas de Nombrado
Elaborado por: Darío Ramírez

Funciones de tamaño reducido, Distinciones con sentido, Verbos y palabras clave

La función tiene un tamaño reducido, el nombre validarLetras representa para lo que fue creado y la palabra clave patrón nos indica los argumentos que vamos a utilizar en la función.

```
<script>
function validarLetras(e) {
    tecla = (document.all) ? e.keyCode : e.which;
    if (tecla == 8) return true;
    patron = /^[0-9A-Za-z\s]+$/g;
    te = String.fromCharCode(tecla);
    return patron.test(te);
}
</script>
```

Figura 51: Reglas de Funciones
Elaborado por: Darío Ramírez

Clases organizadas, Encapsulación, Principios de responsabilidad única y Organiza los cambios

La clase se encuentra dentro de la capa de negocios lo que permite tener una organización adecuada de la clase, las variables son privadas, que solo pueden ser utilizadas dentro de esa clase, la clase solo será para crear los datos y las propiedades manteniendo un principio de responsabilidad única, de esta manera organizada se reducirá los riesgos que el código no funcione.

```
namespace SMC.BusinessObject
{
    55 referencias
    public class ConectarBaseDatos
    {
        #region Datos

        private static string _cadenaConexion;

        //Datos adicionales
        private static string _servidor;
        private static string _baseDatos;
        private static string _usuario;
        private static string _contraseña;
        private static string _parametro;

        #endregion
    }
}
```

Figura 52: Reglas de Clases
Elaborado por: Darío Ramírez

Comentarios explicando el código, Comentarios informativos y Explicar las intenciones

Por medio de los comentarios damos explicación de cada parte para que fue creado la línea de código lo que permite dar información relevante, además de explicar la intención del porqué fue creado.

```

protected void Page_Load(object sender, EventArgs e)
{
    //Utilizar el SqlConnection para realizar la conexion a la BD.
    SqlConnection conexion = new SqlConnection();

    try
    {
        //Guardar la cadena de conexion en la propiedad estatica para que mantenga para toda la aplicacion
        ConectarBaseDatos.CadenaConexion = "Data Source=NICTRON;Initial Catalog=PROINBA;User ID=sa;Pas:

        conexion.ConnectionString = ConectarBaseDatos.CadenaConexion;

        //Abrir la conexion.
        conexion.Open();

    }
    catch (SqlException excepcion)
    {
        ////Tratar y administrar las excepciones de BD.
        Excepciones.Gestionar(excepcion);
    }
}

```

Figura 53: Reglas de Comentarios
Elaborado por: Darío Ramírez

Procesos de error usando excepciones, Crear instrucciones try-catch-finally, Contexto junto a las excepciones y Evitar pasar y devolver null

Mediante la clase de Excepciones se puede controlar los errores que se puedan dar en el sistema, como se observa en la figura 54.

```

        ////Tratar y administrar las excepciones de BD.
        Excepciones.Gestionar(excepcion);

        lblMensaje.Text = "Error de C#:" +
            "<br>" +
            Excepciones.MensajePersonalizado;
    }
}

```

Figura 54: Reglas de Proceso de Error
Elaborado por: Darío Ramírez

Se crea instrucciones try para ejecutar la instrucción, con el finally se puede cancelar la instrucción regresando al catch, además de evitar devolver valores nulos, como se observa en las figuras 55, 56 y 57.

```

try
{
    //Guardar la cadena de conexion en la propiedad estatica para que mantenga para toda :
    ConectarBaseDatos.CadenaConexion = "Data Source=NICTRON;Initial Catalog=PROINBA;User :

    conexion.ConnectionString = ConectarBaseDatos.CadenaConexion;

    //Abrir la conexion.
    conexion.Open();
}

```

Figura 55: Reglas de Proceso de Error
Elaborado por: Darío Ramírez

```

catch (SqlException excepcion)
{
    ////Tratar y administrar las excepciones de BD.
    Excepciones.Gestionar(excepcion);

    ////Personalizar los mensajes de error.
    //MessageBox.Show(Excepciones.MensajePersonalizado,
    //                "Error de SQL Server",
    //                MessageBoxButtons.OK,
    //                MessageBoxIcon.Error);

    lblMensaje.Text = "Error de SQL Server:" +
        "<br>" +
        Excepciones.MensajePersonalizado;
}

```

Figura 56: Reglas de Proceso de Error
Elaborado por: Darío Ramírez

```

finally
{
    //Cerrar la conexion.
    conexion.Close();
    //Liberar recursos.
    conexion.Dispose();
}

```

Figura 57: Reglas de Proceso de Error
Elaborado por: Darío Ramírez

Pruebas de unidad limpias, Un solo concepto por prueba

Como se puede observar en la figura 58, se realiza la prueba por unidad manteniéndolo limpio y descriptivo.

Registro Usuario
^

Nombre

Usuario

Contraseña

Tipo

		NOMBRE	USUARIO	CONTRASEÑA	TIPO
Editar	Eliminar	DARIO RAMIREZ	ADMIN	123DAR	ADMINISTRADOR
Editar	Eliminar	JOSE	JOSE	JOSE	EMPLEADO

Figura 58: Reglas de Prueba de Unidad
Elaborado por: Darío Ramírez

En la figura 59, representa un solo concepto de las pruebas.- Cada prueba es independiente y de manera rápida.

CRITERIO	CONTENIDO
ID	P02
Nombre	Registro de Usuario
Descripción	El registro de un nuevo usuario solo le será permitido al administrador así como su actualización y eliminación.
Precondiciones	El usuario debe estar registrado como administrador en la base de datos.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de usuario se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Figura 59: Reglas de Prueba de Unidad
Elaborado por: Darío Ramírez

4.4.5. Fase de Pruebas

4.4.5.1. Pruebas de Caja Negra

Aquí solo se toma en cuenta las entradas y salidas que pueda haber del sistema, sin la necesidad de preocuparnos del código fuente, para poderlos obtener hay que basarse en los requerimientos del sistema.

Tabla 48: Caja Negra: Inicio de Sesión

Fuente: Autor

N° 1	PRUEBAS CAJA NEGRA	
Datos de Entrada Inicio de Sesión-Usuario y Contraseña	Resultado Acceso al sistema	
Tipo de flujo de datos Se los realizará por medio de un login de acceso.		
La estructura de datos que viaja con el flujo Usuario y Contraseña		
Comentarios: Al momento que se ingresa su usuario y contraseña el sistema verifica el tipo de permiso que tiene ese usuario y permite el ingreso al sistema.		

Tabla 49: Caja Negra: Registro Usuarios

Fuente: Autor

N° 2	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Usuario-Nombre, Usuario, Contraseña y Tipo	Resultado Nuevo usuario registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Nombre-Usuario-Contraseña y Tipo		
Comentarios: Solo el administrador podrá ingresar un nuevo usuario, es de suma importancia que ponga que tipo es puesto que con ello se le dará el acceso al sistema con el permiso correspondiente.		

Tabla 50: Caja Negra: Registro Galpones

Fuente: Autor

N° 3	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Galpones-Tipo Galpón.	Resultado Nuevo galpón registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Número y Tipo Galpón.		
Comentarios: El usuario solo debe colocar el tipo de galpón puesto que el número de galpón se lo ingresa automáticamente.		

Tabla 51: Caja Negra: Registro Corrales

Fuente: Autor

N° 4	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Corrales-Tipo Galpón.	Resultado Nuevo corral registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Tipo Galpón y Número Corral.		
Comentarios: El usuario solo debe colocar el tipo de galpón puesto que el número de corral se lo ingresa automáticamente.		

Tabla 52: Caja Negra: Registro Madres

Fuente: Autor

N° 5	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Madres-Número, Raza, #Partos, Galpón y Corral.	Resultado Nueva madre registrada en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Número-Raza-#Partos-Galpón y Corral.		
Comentarios: Para que se registre la madre es obligado llenar todos los campos.		

Tabla 53: Caja Negra: Registro Machos

Fuente: Autor

N° 6	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Machos-Número, Raza, Galpón y Corral.	Resultado Nuevo macho registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Número-Raza -Galpón y Corral.		
Comentarios: Para que se registre la madre es obligado llenar todos los campos.		

Tabla 54: Caja Negra: Registro Tipo Madres

Fuente: Autor

N° 7	PRUEBAS CAJA NEGRA	
Datos de Entrada Registro Tipo Madre – Tipo y Descripción.	Resultado Nuevo tipo registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Tipo y Descripción.		
Comentarios: En el tipo es donde se especificará a la madre su estado en el que se encuentra en ese momento, puede ser Destetada(DTT), Quedada(QDD),...etc.		

Tabla 55: Caja Negra: Registro Inseminaciones

Fuente: Autor

N° 8	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Inseminaciones-Día, Semana, #Madre, Tipo, #Macho.	Resultado Nueva inseminación y revisión registrada en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Día-Semana-#Madre-Tipo y #Macho.		
Comentarios: Tanto la Inseminación como las Revisiones se hacen al mismo con los mismos datos de ingreso excepto el Tipo y el #Macho.		

Tabla 56: Caja Negra: Registro Nacimientos

Fuente: Autor

N° 9	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Nacimientos-#Madre, Sem. Ins, Sem. Real, Día Real, Vivos, Muertos, Momias, Adoptados, Donados	Resultado Nuevo nacimiento registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo #Madre-Sem. Ins-Sem. Real-Día Real-Vivos-Muertos-Momias-Adoptados y Donados.		
Comentarios: Para que se registre la madre es obligado llenar todos los campos.		

Tabla 57: Caja Negra: Registro Destetes

Fuente: Autor

N° 10	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Destetes-#Lote, Semana, Día, Galpón, #Corral, #Madre, #Parto, Machos, Hembras y Peso	Resultado Nuevo destete-detalle destete-Lote y Detalle Lote registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo #Lote-Semana-Día-Galpón-#Corral-#Madre-#Parto-Machos-Hembras y Peso		
Comentarios: Al momento de registrar un nuevo destete se ingresarán los otros registros automáticamente y llevar un mejor control del sistema.		

Tabla 58: Caja Negra: Registro Movimientos

Fuente: Autor

N° 11	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Movimientos-Galpón Origen, #Corral Origen, #Lote, Cerdos Lote, Día, Semana, Galpón Destino y #Corral Destino.	Resultado Nuevo movimiento registrado en la base de datos	
Tipo de flujo de datos Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Galpón Origen-#Corral Origen-#Lote-Cerdos Lote-Día-Semana-Galpón Destino y #Corral Destino.		
Comentarios: Al momento de realizar un registro de movimiento la tabla lote se actualizará al igual que el detalle lote.		

Tabla 59: Caja Negra: Registro Mortalidad y Ventas

Fuente: Autor

N° 12	PRUEBAS CAJA NEGRA	
Datos de Entrada Registrar Mortalidad o Venta-Venta, Mortalidad, Galpón, #Corral, #Lote, Cerdos Lote, Día, Semana, Cantidad Cerdos, Nota y Observación.	Resultado Nueva mortalidad o venta registrado en la base de datos	
Tipo de flujo de datos de ingreso Se los realizará por medio de un formulario.		
La estructura de datos que viaja con el flujo Venta-Mortalidad-Galpón-#Corral-#Lote-Cerdos Lote-Día-Semana-Cantidad Cerdos-Nota y Observación.		
Comentarios: Al ingresar solo puede ser o bien una mortalidad o bien una venta pero jamás van a ser los dos, por lo cual al seleccionar uno el otro se bloquea.		

4.4.5.2. Pruebas de Funcionamiento del sistema

Estas pruebas se basan en la revisión, ejecución y retroalimentación de las funcionalidades del sistema desarrollado, de esta manera se podrá probar y validar que el sistema realiza todo lo se especificó en los requerimientos. Para lo cual se utilizó los Casos de Prueba de Funcionalidad.

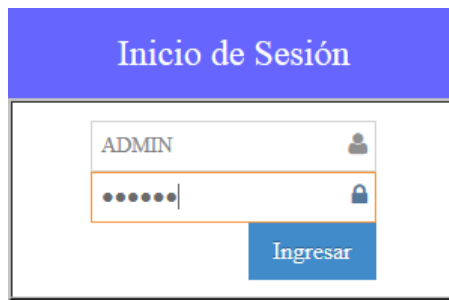


Figura 60: Prueba Funcional: Inicio de Sesión
Elaborado por: Autor

Tabla 60: Casos de Prueba: Inicio de Sesión

Fuente: Autor

CRITERIO	CONTENIDO
ID	P01
Nombre	Inicio de sesión
Descripción	El sistema permitirá el acceso al sistema mediante el tipo de usuario para el que ha sido asignado.
Precondiciones	El usuario debe estar registrado en la base de datos.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. El usuario ingresa el usuario y su clave 2. Verifica que tipo de usuario es. 3. No puede quedar campos en blanco.
Resultado esperado	El usuario puede acceder al sistema
Estado	Ejecutado.
Resultado obtenido	El inicio de sesión se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Registro Usuario
^

Nombre

Usuario

Contraseña

Tipo

		NOMBRE	USUARIO	CONTRASEÑA	TIPO
Editar	Eliminar	DARIO RAMIREZ	ADMIN	123DAR	ADMINISTRADOR
Editar	Eliminar	JOSE	JOSE	JOSE	EMPLEADO

Figura 61: Prueba Funcional: Registro de Usuario
Elaborado por: Autor

Tabla 61: Casos de Prueba: Registro de Usuario

Fuente: Autor

CRITERIO	CONTENIDO
ID	P02
Nombre	Registro de Usuario
Descripción	El registro de un nuevo usuario solo le será permitido al administrador así como su actualización y eliminación.
Precondiciones	El usuario debe estar registrado como administrador en la base de datos.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de usuario se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Galpones

Número Galpón

Tipo Galpón

Tipo

		#GALPÓN	TIPO
Editar	Eliminar	1	ENGORDE
Editar	Eliminar	2	GESTACION
Editar	Eliminar	3	RECRIA
Editar	Eliminar	4	MATERNIDAD
Editar	Eliminar	5	MAT
Editar	Eliminar	6	ROM

Figura 62: Prueba Funcional: Registro de Galpones
Elaborado por: Autor

Tabla 62: Casos de Prueba: Registro de Galpones

Fuente: Autor

CRITERIO	CONTENIDO
ID	P03
Nombre	Registro de Galpones
Descripción	El registro de un galpón lo podrá realizar tanto el administrador como el empleado.
Precondiciones	El usuario debe iniciar sesión.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de galpones se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Corrales

Galpón: ENGORDE

Corral #: 3

Guardar Cancelar

Galpón: Buscar

	#CORRAL	GALPÓN
Eliminar	1	ENGORDE
Eliminar	2	ENGORDE
Eliminar	1	GESTACION
Eliminar	2	GESTACION
Eliminar	1	RECRIA
Eliminar	2	RECRIA

Figura 63: Prueba Funcional: Registro de Corrales
Elaborado por: Autor

Tabla 63: Casos de Prueba: Registro de Corrales

Fuente: Autor

CRITERIO	CONTENIDO
ID	P04
Nombre	Registro de Corrales
Descripción	El registro de un corral lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Para que hay un corral debe estar registrado en la base un galpón.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de corrales se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Madres
^

Número Madre	<input type="text" value="310"/>
Raza Madre	<input type="text" value="TOPIC"/>
Número de Partos	<input type="text" value="3"/>
Galpón	<input style="border-bottom: 1px solid #ccc;" type="text" value="GESTACION"/> ▼
Corral	<input style="border-bottom: 1px solid #ccc;" type="text" value="1"/> ▼

<input type="text" value="# Madre"/>	<input type="button" value="Buscar"/>
--------------------------------------	---------------------------------------

		#MADRE	RAZA	#PARTOS	#CORRAL	GALPÓN
Editar	Eliminar	210	LANDRAX	5	1	ENGORDE
Editar	Eliminar	512	HYPOR	4	1	GESTACION

Figura 64: Prueba Funcional: Registro de Madres
Elaborado por: Autor

Tabla 64: Casos de Prueba: Registro de Madres

Fuente: Autor

CRITERIO	CONTENIDO
ID	P05
Nombre	Registro de Madres
Descripción	El registro de una madre lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Para registrar una madre en la base debe haber un galpón y un corral.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de madres se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Machos o Pajuelas
^

Número Macho o Pajuela	<input type="text" value="102"/>
Raza Macho o Pajuela	<input type="text" value="LANDRAX"/>
Galpón	<input style="border-bottom: 1px solid #ccc;" type="text" value="GESTACION"/> ▼
Corral	<input style="border-bottom: 1px solid #ccc;" type="text" value="2"/> ▼

<input type="text" value="# Padre"/>	<input type="button" value="Buscar"/>
--------------------------------------	---------------------------------------

		#MACHO	RAZA	#CORRAL	GALPÓN
Editar	Eliminar	100	HYPOR	2	GESTACION
Editar	Eliminar	101	LAN	1	GESTACION

Figura 65: Prueba Funcional: Registro de Machos
Elaborado por: Autor

Tabla 65: Casos de Prueba: Registro de Machos

Fuente: Autor

CRITERIO	CONTENIDO
ID	P06
Nombre	Registro de Machos
Descripción	El registro de un macho lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Para registrar una macho en la base debe haber un galpón y un corral.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de machos se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Tipo Madre
^

Tipo

Descripción

Descripción

		TIPO	DESCRIPCIÓN
Editar	Eliminar	DTT	DESTETADA
Editar	Eliminar	PRZ	PRIMERIZA
Editar	Eliminar	QDD	QUEDADA

Figura 66: Prueba Funcional: Registro Tipo de Madres
Elaborado por: Autor

Tabla 66: Casos de Prueba: Registro de Tipo de Madres

Fuente: Autor

CRITERIO	CONTENIDO
ID	P07
Nombre	Registro de Tipo de Madres
Descripción	El registro de un tipo de madre lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Ninguna.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de tipo de madres se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Inseminaciones

Día Inseminación: 08/02/2019

Semana Inseminación: 12

Número Madre: 512

Tipo Madre: PRZ

Tipo Macho o Pajuela: HYPOR

Guardar **Cancelar**

Semana: **Buscar**

		DÍA	SEMANA	#MADRE	TIPO	MACHO
Editar	Eliminar	11-01-2019	34	210	DESTETADA	HYPOR

Figura 67: Prueba Funcional: Registro Inseminaciones
Elaborado por: Autor

Revisiones

	#MADRE	SEMANA ISS	DIA INS.	1da REVISION	2da REVISION	CONFIRMACION	OBSERVACION
Editar	210	34	11-01-2019	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 68: Prueba Funcional: Registro Inseminaciones
Elaborado por: Autor

Tabla 67: Casos de Prueba: Registro Inseminaciones

Fuente: Autor

CRITERIO	CONTENIDO
ID	P08
Nombre	Registro de Inseminaciones
Descripción	El registro de una inseminación lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Debe estar registrado la madre, el macho y el tipo de la madre.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos y el registro de una revisión se ingresará al mismo tiempo.
Estado	Ejecutado.
Resultado obtenido	El registro de inseminaciones y Revisiones se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Nacimientos ^

Número Madre	<input type="text" value="210"/>		Semana Inseminación	<input type="text" value="34"/>
Semana Real Nacimiento	<input type="text" value="34"/>		Día Real Nacimiento	<input type="text" value="13/02/2019"/>
Lechones Vivos	<input type="text" value="5"/>		Lechones Muertos	<input type="text" value="1"/>
Lechones Momias	<input type="text" value="1"/>		Lechones Adoptados	<input type="text" value="0"/>
Lechones Donados	<input type="text" value="0"/>			

Semana

		#MADRE	SEMANA INS.	SEMANA REAL	DIA REAL	VIVOS	MUERTOS	MOMIAS	ADOPTADOS	DONA
Editar	Eliminar	512	12	13	31-01-2019	4	6	5	6	4

Figura 69: Prueba Funcional: Registro Nacimientos
Elaborado por: Autor

Tabla 68: Casos de Prueba: Registro Nacimientos

Fuente: Autor

CRITERIO	CONTENIDO
ID	P09
Nombre	Registro de Nacimientos
Descripción	El registro de un nacimiento lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Debe estar registrado la inseminación.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos.
Estado	Ejecutado.
Resultado obtenido	El registro de Nacimientos se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Destetes ^

Número Lote	<input type="text" value="319"/>	Semana Destete	<input type="text" value="34"/>	Día Destete	<input type="text" value="06/02/2019"/>
Destino Destete	<input style="border: none; background-color: #f0f0f0; width: 100%;" type="text" value="RECRIA"/>	Número Madre	<input style="border: none; background-color: #f0f0f0; width: 100%;" type="text" value="512"/>		
	<input style="border: none; background-color: #f0f0f0; width: 100%;" type="text" value="1"/>	Número Parto	<input style="border: none; background-color: #f0f0f0; width: 100%;" type="text" value="4"/>		
Lechones Machos	<input style="border: none; background-color: #f0f0f0; width: 100%;" type="text" value="3"/>	Lechones Hembras	<input style="border: none; background-color: #f0f0f0; width: 100%;" type="text" value="3"/>	Peso Total	<input style="border: none; background-color: #f0f0f0; width: 100%;" type="text" value="12,5"/>

Semana

		#LOTE	SEMANA	DÍA	GALPÓN	CORRAL	#MADRE	#PARTOS	LECHONES MACHOS	LECHONES HEMBRAS	PESO TOTAL
Editar	Eliminar	219	23	01-02-2019	RECRIA	1	210	4	2	4	34,560

< >

Figura 70: Prueba Funcional: Registro Destetes

Elaborado por: Autor

Lote					
<input type="text" value="#Lote"/>		<input type="button" value="Buscar"/>			
		#LOTE	#CORRAL RECRÍA	#CORRAL ENGORDE	VENDIDO
Editar	Eliminar	219	2	0	N

Figura 71: Prueba Funcional: Registro Lote
Elaborado por: Autor

Detalle	
#LOTE	ACCIÓN
219	INGRESO

Figura 72: Prueba Funcional: Registro Detalle Lote
Elaborado por: Autor

Tabla 69: Casos de Prueba: Registro Destetes

Fuente: Autor

CRITERIO	CONTENIDO
ID	P10
Nombre	Registro de Destetes
Descripción	El registro de un destete lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Deben estar registrado los nacimientos.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos, se guardará también el lote y el detalle del lote al mismo tiempo.
Estado	Ejecutado.
Resultado obtenido	El registro de Destetes se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Movimientos

DESDE

Galpón: RECRIA Corral: 2

Lote: 219 # Cerdos: 6

Semana: 45 Día: 08/05/2019

HACIA

Galpón: ENGORDE Corral: 1

Guardar **Cancelar**

Semana: **Buscar**

		GALPÓN ORIGEN	CORRAL ORIGEN	#LOTE	#CERDOS	SEMANA	DIA	GALPÓN DESTINO	CORRAL DESTINO
Editar	Eliminar	RECRIA	2	219	6	34	07-03-2019	ENGORDE	1

Figura 73: Prueba Funcional: Registro Movimientos
Elaborado por: Autor

Lote

#Lote: **Buscar**

		#LOTE	#CORRAL RECRIA	#CORRAL ENGORDE	VENDIDO
Editar	Eliminar	219	2	1	N

Figura 74: Prueba Funcional: Registro Lote
Elaborado por: Autor

Detalle

#LOTE	ACCIÓN
219	INGRESO
219	MOVIMIENTO

Figura 75: Prueba Funcional: Registro Detalle Lote
Elaborado por: Autor

Tabla 70: Casos de Prueba: Registro Destetes

Fuente: Autor

CRITERIO	CONTENIDO
ID	P11
Nombre	Registro de Movimientos
Descripción	El registro de un movimiento lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Deben estar registrado el destete.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco.
Resultado esperado	Los datos se guardarán en la base de datos, se actualizará también el lote y el detalle del lote al mismo tiempo que se registra el movimiento.
Estado	Ejecutado.
Resultado obtenido	El registro de Movimientos se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

Figura 76: Prueba Funcional: Registro Mortalidad o Venta
Elaborado por: Autor

Tabla 71: Casos de Prueba: Registro Mortalidad o Ventas

Fuente: Autor

CRITERIO	CONTENIDO
ID	P12
Nombre	Registro de Mortalidad o Ventas
Descripción	El registro de una mortalidad o ventas lo podrá realizar tanto el administrador como el empleado.
Precondiciones	Deben estar registrado el destete o movimiento.
Pasos y condiciones de ejecución	<ol style="list-style-type: none"> 1. Se debe llenar todo el formulario y guardarlo. 2. El usuario puede ser Administrador o Empleado. 3. No puede quedar campos en blanco. 4. Si es mortalidad no puede ser venta al mismo tiempo y viceversa.
Resultado esperado	Los datos se guardarán en la base de datos, pero puede ser o solo venta solo mortalidad pero no los dos al mismo tiempo.
Estado	Ejecutado.
Resultado obtenido	El registro de Mortalidad o Venta se realizó con éxito.
Errores asociados	Ninguno.
Responsable diseño	Darío Ramírez
Responsable ejecución	Darío Ramírez
Comentarios	

4.4.6. Implantación del sistema

4.4.6.1. Configuración de los equipos

Se instalará un gestor de base de datos que almacenará toda la información que la empresa vaya a manejar, para lo cual se instaló el SQL Server, puesto que ayuda a manejar los procedimientos almacenados, tener seguridad, además de incluir un entorno gráfico potente que permite el uso de comandos DDL Y DML gráficamente.

4.4.6.2. Puesta en marcha del sistema

Para poner en marcha el sistema se realizó algunas configuraciones que se detallan a continuación:

Configuración del servidor

Para alojar el sistema se va a utilizar el servidor web IIS (Internet Information Services).

Un servidor web, son también conocidos como servidores http cuya finalidad es la de distribuir un contenido web o la transferencia de documentos en redes locales o desde el internet, de acuerdo a las solicitudes que los clientes lo realicen[29].

Un servidor web IIS, es un conjunto de servicios para servidores usando Microsoft Windows, que permiten desde la creación, configuración y administración de sitios web para que los usuarios envíen y reciban información[30][31].

Habilitar el IIS

Para poder habilitar el servidor web IIS se debe seguir los siguientes pasos: Seleccionar Panel de Control > Programas y características, se escoge Activar o desactivar las características de Windows.



Figura 77: Panel de Control
Elaborado por: Autor

A continuación hay que buscar Internet Information Services y es recomendable seleccionar todo pero de manera individual, luego en Aceptar.



Figura 78: Características de Windows
Elaborado por: Autor

Para probar que se encuentre correctamente activado el IIS, solo se coloca localhost en el Mozilla o el Google Chrome.



Figura 79: Servidor Web IIS
Elaborado por: Autor

Al terminar de configurar el servidor IIS se crea una dirección para nuestro servidor que contendrá todos los archivos de configuración que permitirán a nuestras aplicaciones ser publicadas. La dirección es C:\inetpub\wwwroot. Para empezar a publicar se debe abrir el administrador de IIS, se puede ingresar desde el buscador del Windows.

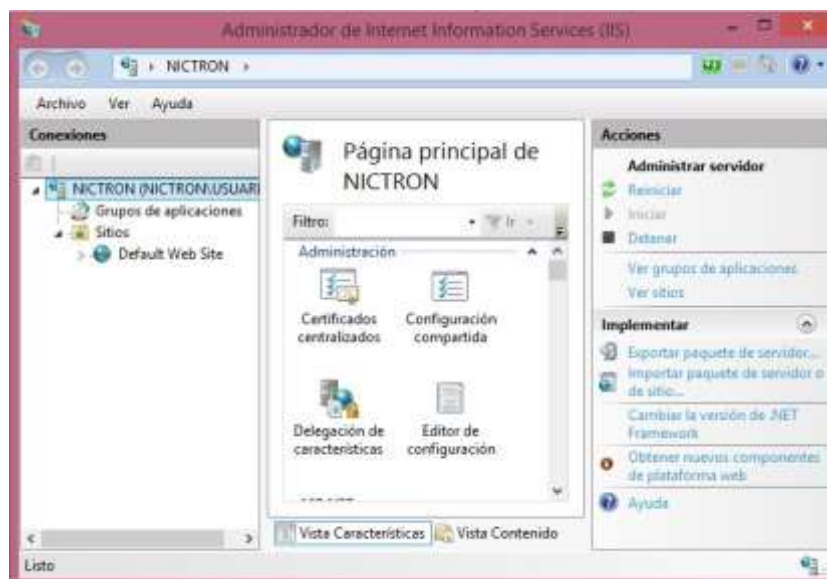


Figura 80: Administrador IIS
Elaborado por: Autor

Para publicar el sistema se da click derecho y publicar sitio web y se da la

ubicación de destino.

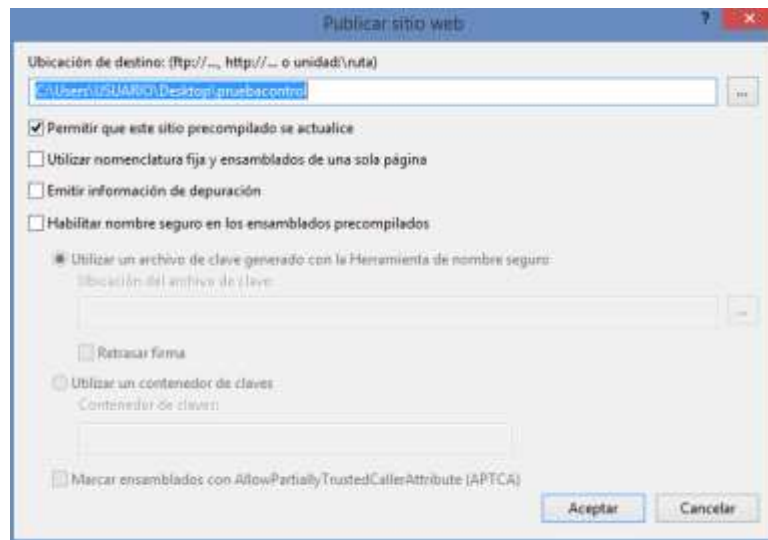


Figura 81: Publicar el sistema
Elaborado por: Autor

A continuación la carpeta creada se copia dentro C:\inetpub\wwwroot.

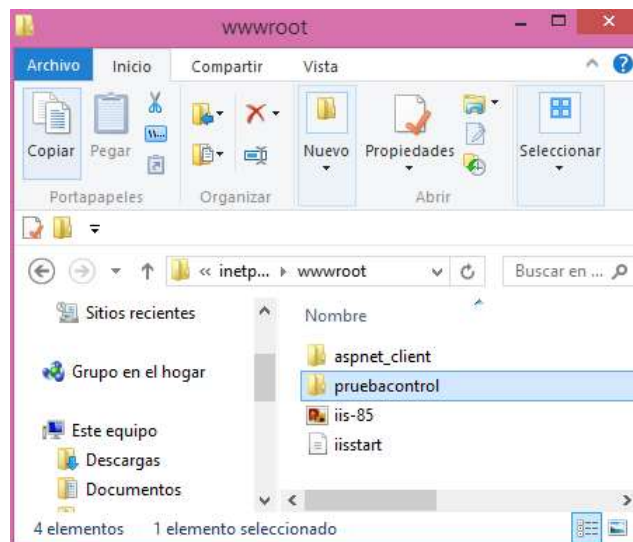


Figura 82: Carpeta creada al publicar
Elaborado por: Autor

Luego en el administrador del IIS la carpeta que se creó se da click derecho y se selecciona agregar aplicación.

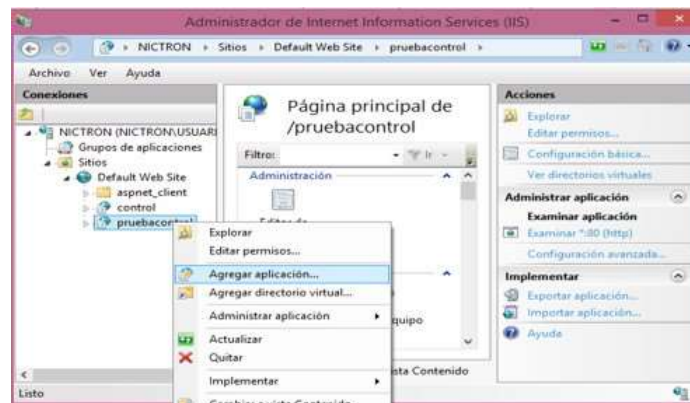


Figura 83: Agregar aplicación
Elaborado por: Autor

Para finalizar se selecciona administrar aplicación en explorar y el sistema se encuentra publicado.

4.4.6.3. Capacitación

Para la capacitación se dio una charla del manejo del sistema al personal de la empresa.

La capacitación se dio el día sábado 12 de enero de 2019 y el lunes 14 de enero de 2019 con los siguientes puntos.

El 12 de enero de 2019

- Se dio a conocer los tipos de usuario que pueden ingresar al sistema, como son el administrador y empleado y que procesos puede realizar cada usuario.
- Se explicó cada una de las interfaces que conforman el sistema, la manera en que la información será llenada y guardada en la base de datos.
- Cuál es la manera en que los datos serán visualizados después de realizar el registro y como buscarlos.
- Se indicó como se actualizarán los datos así como su eliminación.

El 14 de enero de 2019

Se indicó como generar los reportes:

- En que parte se debe ubicar para generar y de que módulos se realizará el reporte
- La manera en que se debe generar el reporte una vez seleccionado el módulo.

- Como guardar el reporte y cómo realizar su respectiva impresión.

Ante cualquier tipo de inquietud que tenga el personal de la empresa, se procedió a entregar el manual de usuario respectivo, tal como se muestra en el Anexo B. En el Anexo C se muestra la capacitación.

CAPÍTULO 5

Conclusiones y Recomendaciones

5.1. Conclusiones

- Con la implementación del sistema de control de producción porcina disminuyeron los tiempos de respuesta a la hora de obtener información detallada durante el ciclo de vida que cumple un porcino en toda su crianza, permitiendo tener un control diario de los procesos o fases que se realiza en la granja porcícola.
- Al aplicar las reglas de nombrado, funciones, clases, comentarios, procesos de error y pruebas de unidad al momento de desarrollar el software, el código es más ordenado, expresivo y legible, además ayuda a mejorar que tanto las clases como los métodos sean más claros y simples, permitiendo obtener el sistema con código limpio.
- Al momento de realizar el diseño en el desarrollo del sistema, la utilización del patrón de diseño Facade evito que la conexión sea repetitiva para todos los procesos.- Con el patrón Builder se creó un objeto complejo ocultando los detalles de la construcción de ese objeto.- El patrón Adapter permitió convertir la interfaz de una clase a otra.- Con lo cual se logró que el sistema sea maleable.
- A través de la información que se presenta al final de los procesos de la producción porcícola, se obtiene un control frecuente de la granja, iniciando con la etapa de inseminación de las cerdas madres y el nacimiento de los lechones hasta su comercialización.
- El sistema permite reducir tiempo de 10s a 5s, además de disminuir el esfuerzo ya que algunos de sus procesos se los realiza de manera automática al registrar los datos, asegurando información clara y confiable.

5.2. Recomendaciones

- Para trabajos a futuro es recomendable agregar un proceso de control de alimentación que tiene que tener cada porcino puesto que con ello se

controlaría el gasto que se tiene en cada uno de ellos y el tipo de alimentación que deben consumir ya que en cada etapa de crianza del porcino es diferente.

- Para la obtención de reportes e información ya sea por parte del personal administrativo o por parte de los empleados se podría implementar un aplicativo móvil que les permita estar siempre pendientes de los porcinos aumentando la eficiencia en el trabajo.
- Para un posible trabajo a futuro se recomienda realizar un módulo de auditoria que ayudará a mantener un control de todos los cambios que se realice al sistema al cual solo el administrador será el encargado de ir revisando puesto que con ello mantendrá un control constante de la granja porcícola.
- Para trabajos a futuro se puede realizar el desarrollo de cualquier proceso, utilizando las reglas de programación limpia, ya que permiten mejorar el código que se realice, además de al momento de realizar el diseño se trabaje con patrones de diseño, puesto que permiten la reutilización y la modificación del diseño en cualquier momento.

Bibliografía

- [1] O. A. M. Msc., “Plan de negocios para un emprendimiento en producción de carne porcina,” 2010.
- [2] X. M. y. E. M. Josep Gasa, Sergi López Vergé, “Iniciación a la producción y manejo del ganado porcino,” 2015.
- [3] J. L. Chris Wright, Chris Harris, “El mercado mundial de las carnes,” 2016. [Online] Available: <http://www.elsitioporcino.com/articulos/2678/el-mercado-mundial-de-las-carnes/>.
- [4] J. S. N. E. A. T. J. P. J. C. J. Productivas Cottura, G Brunori, “Desarrollo y administracion de un sistema informatico para monitoreo permanente de gestiones en actividades productivas porcinas en argentina,” pp. 1–12, 2016.
- [5] R. Líderes, “La producción porcina en el país está a la baja,” p. 1, 2017.
- [6] E. M. B. Garzón, “Mejora de procesos y optimización de la producción porcícola en la granja de la universidad de las américas,” 2017.
- [7] J. A. M. L. y Maria Belen Villao Loor, “Software de gestion de procesos de produccion en la unidad de docencia, investigacion y vinculacion hato porcino de la espam mfl,” 2014.
- [8] D. A. P. Mendoza, “Arquitectura basada en una capa de control de excepciones para mejorar la fiabilidad de la aplicación software de préstamos bancarios,” 2016.
- [9] C. M. C. M. y Mariangel Isabella Murillo Sevillano, “Diseno e implementacion de un prototipo de sistema automatizado para la alimentacion y climatizacion de una granja porcina,” 2015.
- [10] C. M. C. M. y Mariangel Isabella Murillo Sevillano, “Adopcion del sistema de seguimiento de actividades porcinas en la provincia de cordoba,” 2014.
- [11] D. P. Correa, “Elaboracion de un software agropecuario para la recopilacion de datos de produccion en la granja san silvestre,” 2015.

- [12] C. H. Sandoval, “Creacion de frameworks con patrones de diseno para el desarrollo de aplicaciones empresariales,” 2014.
- [13] L. Debrauwer, *Patrones de diseno de C*. 2012.
- [14] D. Garcia, “Patrones estructurales (vii): Patron proxy,” 2014. [Online] Available: <https://danielggarcia.wordpress.com/2014/04/07/patrones-estructurales-vii-patron-proxy/>.
- [15] J. C. B. Osorio, “Desarrollo de aplicaciones con uso de patrones de comportamiento,” 2013.
- [16] E. G. M. y Julian Pacienza, “Metodologias de desarrollo de software,” 2015.
- [17] A. A. M. Loayza, “Ciclo de vida y metodologias de desarrollo de softyware,” 2015.
- [18] J. Bautista, “Programacion extrema (xp),” *Tendencias en Computacion*, p. 7, 2013.
- [19] C. Sanchez, “Ciclo de vida de un proyecto xp,” 2010. [Online] Available: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
- [20] G. F. Escribano, “Introduccion a extreme programming,” p. 4, 2002.
- [21] A. Fallis, “Metodologia actual metodologia xp,” *Journal of Chemical Information and Modeling*, pp. 5–7, 2014.
- [22] R. C. Martin, *Código Limpio Manual de estilo para el desarrollo ágil de software*. 2012.
- [23] P. Agropecuario, “Inseminación artificial en cerdas,” 2009. [Online] Available: <http://www.pregonagropecuario.com/cat.php?txt=11339>.
- [24] K. G. Martínez, “Técnicas de palpación de la cerda,” 2018. [Online] Available: <https://laporcicultura.com/reproduccion-porcina/tecnicas-palpacion-la-cerda/>.
- [25] T. Cerdos, “El parto,” 2015. [Online] Available: <http://www.todocerdos.com.ar/notas.asp?nid=754&sid=3>.
- [26] P. Roldan-Santiago, “Destete de los lechones,” 2014. [Online] Available: https://www.researchgate.net/figure/FIGURA-4-A-menudo-el-destete-de-los-lechones-es-realizado-entre-la-tercer-fig2_266673337.

- [27] A. P. Yagüe, “Diarrea post-destete en lechones: prácticas preventivas,” 2017. [Online] Available: https://www.3tres3.com/articulos/diarrea-post-destete-en-lechones-practicas-preventivas_38334/.
- [28] E. Productor, “Cerdos de engorde,” 2014. [Online] Available: <http://elproductor.com/articulos-tecnicos/articulos-tecnicos-salud-animal/a-los-cerdos-de-engorde-les-gusta-tener-su-casa-bien-limpia/>.
- [29] D. Guide, “Servidor web, definición, historia y programas,” 2016. [Online] Available: <https://www.ionos.es/digitalguide/servidores/know-how/servidor-web-definicion-historia-y-programas/>.
- [30] Willmer, “Iis-internet-information-server,” 2015. [Online] Available: <https://wiwiloz.wordpress.com/iis-internet-information-server/>.
- [31] Fahozo, “Definición de servidor iis,” 2016. [Online] Available: <https://es.scribd.com/doc/56767327/Definicion-de-Servidor-IIS>.

Anexos y Apéndices

Anexo A

Anexo A

Aceptación de las Historias de Usuario

Personal Administrativo

Tabla 72: Registro Usuarios

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador
Nombre de la historia: Registro de usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los Usuarios y debe tener datos como: Usuario y Contraseña	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Personal Empleado

Tabla 73: Registro Galpones

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador o Empleado
Nombre de la historia: Registro de Galpones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los galpones y debe tener datos como: Número y Tipo.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 74: Registro Corrales

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador o Empleado
Nombre de la historia: Registro de Corrales	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los corrales y debe tener datos como: Número y Tipo.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 75: Registro Madres

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador o Empleado
Nombre de la historia: Registro de las Madres	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a las Madres y debe contar con los siguientes datos: Número, Raza, Número de Partos, Galpón y Corral.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 76: Registro Machos

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 5	Usuario: Administrador o Empleado
Nombre de la historia: Registro del Macho o Pajuela	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los Machos o Pajuelas y debe tener datos como: Número, Raza, Galpón y Corral.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 77: Registro Tipo Madres

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador o Empleado
Nombre de la historia: Registro del Tipo de Madre	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer al Tipo de Madre y debe tener datos como: Tipo y Descripción.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 78: Registro Inseminaciones

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 7	Usuario: Administrador o Empleado
Nombre de la historia: Registro de las Inseminaciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a las Inseminaciones y debe tener datos como: Número, Día, Semana, Número Madre, Tipo Madre, Macho o Pajuela.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 79: Registro Revisiones

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 8	Usuario: Administrador o Empleado
Nombre de la historia: Registro de las Revisiones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a las Revisiones y debe contar con los siguientes datos: Número Cerda, Número Inseminación, 1ra Revisión, 2da Revisión, Confirmación.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 80: Registro Nacimientos

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 9	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Nacimientos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a las Nacimientos y debe contar con los siguientes datos: Número Nacimiento, Semana Inseminación, Semana Real Nacimiento, Día Real Nacimiento, Lechones Vivos Nacimiento, Lechones Muertos Nacimiento, Lechones Momias Nacimiento, Adoptados Nacimiento, Donados Nacimiento.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 81: Registro Destetes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Destetes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a las Destetes y debe contar con los siguientes datos: Número Lote, Semana, Día, Galpón, Corral.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 82: Registro Detalle Destetes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 11	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Detalles Destete	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer al Detalles Destete y debe contar con los siguientes datos: Número Madre, Número Partos, Cant. Lechones Machos, Cant. Lechones Hembras, Peso Total	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 83: Registro Lotes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 12	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Lotes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a los Lotes, debe contar con los siguientes datos: Código, Número. Corral Recría, Corral Engorde, Vendido.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 84: Registro Detalle Lotes

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 13	Usuario: Administrador o Empleado
Nombre de la historia: Registro del Detalle Lotes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 2	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a Detalle Lotes, debe contar con los siguientes datos: Código Lote, Número Lote, Acción.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 85: Registro Movimientos

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 14	Usuario: Administrador o Empleado
Nombre de la historia: Registro de los Movimientos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a Movimientos, debe contar con los siguientes datos: Galpón Origen, Corral Origen, #Lote, #Cerdos, Semana, Día, Galpón Destino, Corral Destino.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	

Tabla 86: Registro Mortalidad y Ventas

Fuente: Darío Ramírez

HISTORIA DE USUARIO	
Número: 15	Usuario: Administrador o Empleado
Nombre de la historia: Registro de la Mortalidad y Ventas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Programador responsable: Darío Ramírez	
Descripción: El registró debe pertenecer a Mortalidades y Ventas, debe contar con los siguientes datos: Mortalidad, Venta, Galpón, Corral, Lote, #Cerdos, Semana, Día, Número Nota, Observación.	
Observaciones: Los datos deben ser almacenados en un gestor de base de datos.	



Ing. Ricardo Pazmiño

Anexo B

Anexo B

Manual de Usuario

Iniciar la Sesión



Figura 84: Manual de Usuario: Inicio de Sesión
Elaborado por: Autor

1. Cuadro de texto donde se inserta el usuario.
2. Cuadro de texto donde se inserta la contraseña del usuario.
3. Después de haber ingresado usuario y contraseña se da click en el botón para ingresar al sistema.

Usuarios

Solo el usuario que tenga permiso de administrador será quien ingrese un usuario nuevo al sistema.



Figura 85: Manual de Usuario: Usuarios
Elaborado por: Autor

1. En los cuadros de texto se insertará el nombre, usuario, contraseña y el tipo de usuario.
2. Después de haber ingresado todos los campos se da click en el botón para guardar un nuevo usuario.
3. El botón cancelar es opcional, si se desea cancelar el registro de un nuevo usuario.
4. Al seleccionar editar permitirá actualizar los datos.
5. Al seleccionar en eliminar permitirá borrar un dato.

Galpones

Registro Usuario

Nombre
Usuario
Contraseña
Tipo

Guardar Cancelar

		NOMBRE	USUARIO	CONTRASEÑA	TIPO
Editar	Eliminar	DARIO RAMIREZ	ADMIN	123DAR	ADMINISTRADOR
Editar	Eliminar	JOSE	JOSE	JOSE	EMPLEADO
Editar	Eliminar	DARIO	DARIO	DARIO	ADMIN

Figura 86: Manual de Usuario: Galpones
Elaborado por: Autor

1. En los cuadro de texto se insertará tipo de galpón, en el cuadro de texto de número de galpón no se insertará ningún dato porque es automático.
2. Después de haber ingresado todos los campos se da click en el botón para guardar un nuevo galpón.
3. El botón cancelar es opcional, si se desea cancelar el registro de un galpón.
4. En el cuadro de texto de ingresará por el tipo de galpón que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Corrales



Figura 87: Manual de Usuario: Corrales
Elaborado por: Autor

1. Una vez insertado el galpón se seleccionará el tipo de galpón, en el cuadro de texto de número de corral no se insertará ningún dato porque es automático.
2. Después de haber ingresado todos los campos se da click en el botón para guardar un nuevo corral.
3. El botón cancelar es opcional, si se desea cancelar el registro de un corral.
4. En el cuadro de texto de ingresará por el galpón que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar en eliminar permitirá borrar un dato.

Madres

The screenshot shows a web application window titled 'Madres'. It contains a form with the following fields: 'Número Madre' (text input), 'Raza Madre' (text input), 'Número de Partos' (text input), 'Galpón' (dropdown menu), and 'Corral' (dropdown menu). Below the form are 'Guardar' and 'Cancelar' buttons. A blue arrow labeled '1' points to the form fields. Below the form is a search section with a text input labeled 'Madre', a 'Buscar' button, and two buttons labeled 'Editar' and 'Eliminar'. A blue arrow labeled '4' points to the search input, and a blue arrow labeled '5' points to the 'Buscar' button. Below the search section is a table with the following data:

		#MADRE	RAZA	#PARTOS	#CORRAL	GALPÓN
Editar	Eliminar	210	LANDRAX	5	1	ENGORDE
Editar	Eliminar	312	HYPOR	4	1	GESTACION

Blue arrows labeled '2' and '3' point to the 'Guardar' and 'Cancelar' buttons respectively. A blue arrow labeled '6' points to the 'Editar' button in the table, and a blue arrow labeled '7' points to the 'Eliminar' button in the table.

Figura 88: Manual de Usuario: Madres
Elaborado por: Autor

1. En los cuadros de texto se insertará número de la madre, la raza y el número de partos, se seleccionará el galpón y el corral.
2. Después de haber ingresado todos los campos se da click en el botón para guardar una nueva madre.
3. El botón cancelar es opcional, si se desea cancelar el registro de una madre.
4. En el cuadro de texto de ingresará el número de la madre que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Machos

Form fields: Número Macho o Pajuela, Raza Macho o Pajuela, Galpón, Corral. Buttons: Guardar, Cancelar.

		#MACHO	RAZA	#CORRAL	GALPÓN
Editar	Eliminar	100	HYPOR.	2	GESTACION
Editar	Eliminar	101	LAN	1	GESTACION

Figura 89: Manual de Usuario: Machos
Elaborado por: Autor

1. En los cuadros de texto se insertará número del padre y la raza, se seleccionará el galpón y el corral.
2. Después de haber ingresado todos los campos se da click en el botón para guardar un nuevo padre.
3. El botón cancelar es opcional, si se desea cancelar el registro de un padre.
4. En el cuadro de texto de ingresará el número del padre que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Tipo Madre



Figura 90: Manual de Usuario: Tipo Madre
Elaborado por: Autor

1. En los cuadros de texto se insertará el tipo y la descripción.
2. Después de haber ingresado todos los campos se da click en el botón para guardar el tipo de madre.
3. El botón cancelar es opcional, si se desea cancelar el registro del tipo de madre.
4. En el cuadro de texto de ingresará descripción del tipo de madre que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Inseminaciones

1

2

3

4

5

6

7

	DIA	SEMANA	#MADRE	TIPO	MACHO
Editar	11-01-2019	34	210	DESTETADA	HYPOR

Figura 91: Manual de Usuario: Inseminaciones
Elaborado por: Autor

1. En los cuadros de texto se insertará el día y la semana, se seleccionará el número de la madre, el padre y el tipo de la madre.
2. Después de haber ingresado todos los campos se da click en el botón para guardar las inseminaciones.
3. El botón cancelar es opcional, si se desea cancelar el registro de la inseminación.
4. En el cuadro de texto de ingresará la semana que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Revisiones

1

2

3

	#MADRE	SEMANA INS.	DIA INS.	1ra REVISION	2da REVISION	CONFIRMACION	OBSERVACION
Editar	210	34	11-01-2019				

Figura 92: Manual de Usuario: Revisiones
Elaborado por: Autor

Al registrar una nueva inseminación, también se ingresará una nueva revisión.

1. En el cuadro de texto de ingresará la semana que se va a buscar.
2. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
3. Al seleccionar editar permitirá actualizar los datos.

Nacimientos

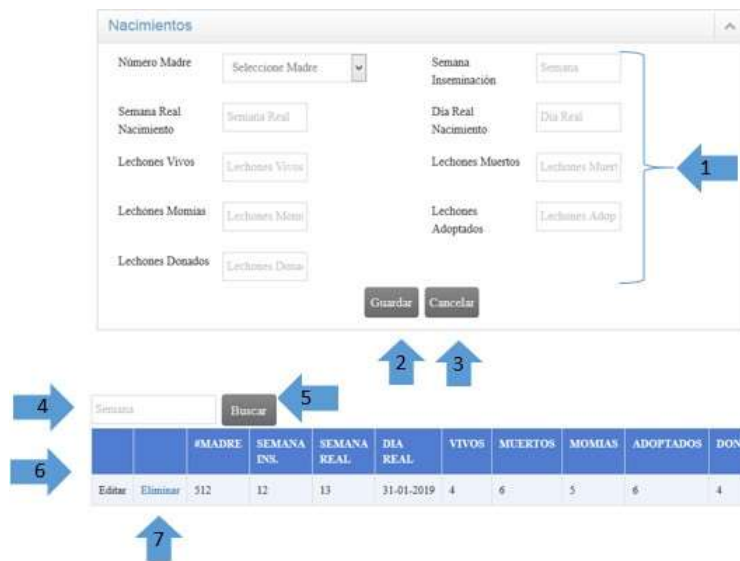


Figura 93: Manual de Usuario: Nacimientos
Elaborado por: Autor

1. Se selecciona el número de la madre y en los cuadros de texto se insertará la semana de inseminación, semana nacimiento, día, lechones vivos, muertos, momias, adoptados y donados.
2. Después de haber ingresado todos los campos se da clic en el botón para guardar los nacimientos.
3. El botón cancelar es opcional, si se desea cancelar el registro de los nacimientos.
4. En el cuadro de texto de ingresará la semana que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Destetes



Figura 94: Manual de Usuario: Destetes
Elaborado por: Autor

1. En los cuadros de texto se insertará el número de lote, semana, día, número de parto, lechones machos, hembras y el peso total, se seleccionará del galpón, el corral y el número de madre.
2. Después de haber ingresado todos los campos se da click en el botón para guardar los Destetes.
3. El botón cancelar es opcional, si se desea cancelar el registro de los Destetes.
4. En el cuadro de texto de ingresará la semana que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Lote



Figura 95: Manual de Usuario: Lote
Elaborado por: Autor

Al registrar un nuevo destete se ingresará también el detalle del destete, el lote y el detalle de lote.

1. En el cuadro de texto de ingresará el lote que se va a buscar.
2. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
3. Al seleccionar editar permitirá actualizar los datos.
4. El detalle del lote solo servirá para observar la acción que se realiza.

Movimientos

Figura 96: Manual de Usuario: Movimientos
Elaborado por: Autor

1. En los cuadros de texto se insertará la semana y el día, se seleccionará el galpón de origen y el número de corral origen, con lo cual de manera automática el lote y el número de cerdos se llenarán los cuadros de texto y también se debe seleccionar el galpón de destino y el número de corral destino.
2. Después de haber ingresado todos los campos se da clic en el botón para guardar los movimientos.
3. El botón cancelar es opcional, si se desea cancelar el registro un movimiento.
4. En el cuadro de texto de ingresará la semana que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Mortalidad o Venta

Mortalidad y Ventas

VENTA MORTALIDAD

Galpón: Seleccione un Galpón

Corral: Seleccione un Corral

Semana: Semana

Cerdos: X Cerdos

Observación: Observación

Lote: Lote

Cerdos: # Cerdos

Día: Día

Número Nota: Número Nota

Guardar Cancelar

4 Semana Nota Observación Buscar 5

VENTA	MORTALIDAD	GALPÓN	#CORRAL	#LOTE	CANTIDAD LOTE	SEMANA	DÍA	C
Editar	Eliminar	RECRIA	2	219	6	13	30-01-2019	3-

6 7

Figura 97: Manual de Usuario: Motalidad o Venta
Elaborado por: Autor

1. Al seleccionar el galpón y el número de corral, de manera automática el lote y el número de cerdos de ese lote se llena los cuadros de texto, se debe seleccionar o una venta o una mortalidad pero no los dos al mismo tiempo, también se debe llenar la semana, el día, cantidad de cerdos, la nota y la observación.
2. Después de haber ingresado todos los campos se da click en el botón para guardar la mortalidad o la venta.
3. El botón cancelar es opcional, si se desea cancelar el registro una mortalidad o venta.
4. En el cuadro de texto de ingresará la semana que se va a buscar.
5. Al dar clic en el botón buscar, se realizará la búsqueda deseada.
6. Al seleccionar editar permitirá actualizar los datos.
7. Al seleccionar en eliminar permitirá borrar un dato.

Reportes



Figura 98: Manual de Usuario: Reportes
Elaborado por: Autor

1. Se va al menú y seleccionar Reportes.
2. Luego se escoge el módulo del cual se quiere obtener el reporte, se da click en el botón y se generará el reporte.

Anexo C

Anexo C



Figura 99: Capacitación
Elaborado por: Autor



Figura 100: Capacitación
Elaborado por: Autor



Figura 101: Capacitación
Elaborado por: Autor