



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA EN COMPUTACIÓN**

“Trabajo de grado previo a la obtención del Título de Ingeniero en Computación”

**TRABAJO DE GRADUACIÓN**

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA EL  
CONTROL Y REGISTRO COMPUTACIONAL DE GANADO VACUNO  
UTILIZANDO LA TECNOLOGÍA RFID”**

**Autor:** VILLEGAS OÑATE JUAN MIGUEL

**Director:** Ing. Aníbal LLanga

Riobamba – Ecuador  
AÑO  
2015

Los miembros del Tribunal de Graduación del proyecto de investigación de título:  
**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA EL CONTROL Y REGISTRO COMPUTACIONAL DE GANADO VACUNO UTILIZANDO LA TECNOLOGÍA RFID”**

Presentado por: Juan Miguel Villegas Oñate y dirigido por: Ing. Anibal Llanga

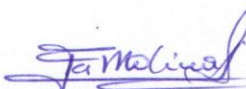
Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

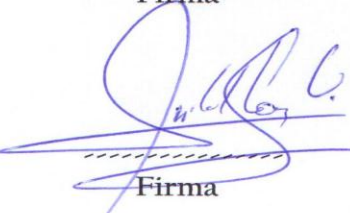
-----  
Ing. Danny Velazco  
Presidente del Tribunal

  
-----  
Firma

-----  
Ing. Fernando Molina  
Miembro del Tribunal

  
-----  
Firma

-----  
Ing. Anibal Llanga  
Director de Tesis

  
-----  
Firma

## AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Juan Miguel Villegas Oñate y Ing. Anibla LLanga Director del Proyecto; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”

JUAN MIGUEL VILLEGAS OÑATE  
060292669-3



## GRADECIMIENTO

Doy gracias a dios por haber sido mi guía, compañía y fortaleza. Además por haber puesto en mi camino a personas de gran espíritu y solidaridad para compartir sus conocimientos para la culminación del presente proyecto.

A mi esposa y a mi hijo, por haberme apoyado siempre han sido la fortaleza necesaria para haber llevado a cabo el presente proyecto

A mi tutor Ing. Anibal Llanga por su valioso tiempo para la revisión y corrección de este proyecto.

A la universidad nacional de Chimborazo que través de su espacio físico y personal docente, me ha permitido alcanzar un respaldo académico de tan alto reconocimiento.

A todas aquellas personas de una forma u otra me apoyaron para cumplir con este objetivo personal.

Juan Miguel Villegas Oñate

## DEDICATORIA

Este trabajo de tesis está dedicado a mi esposa e hijo pues son mi razón de vivir.

A mis Padres ya que con su motivación y apoyo constante y sus sabios consejos han logrado que cumpla mis metas y objetivos propuestos.

Juan Miguel.

# ÍNDICE GENERAL

## ÍNDICE DE CUADROS

<b>TABLA 1</b> EVOLUCIÓN DE LA TECNOLOGÍA RFID .....	6
<b>TABLA 2</b> COMPARATIVA ENTRE TAG PASIVOS Y TAG ACTIVOS.....	10
<b>TABLA 3</b> CARACTERÍSTICAS DEL MICROCONTROLADOR ATMEGA 164P .....	17
<b>TABLA 4</b> OPERACIONALIZACIÓN DE LA VARIABLE INDEPENDIENTE .....	35
<b>TABLA 5</b> OPERACIONALIZACIÓN DE LA VARIABLE DEPENDIENTE.....	36
<b>TABLA 6</b> ACCESO A INFORMACIÓN COMPLETA CON EL REGISTRO MANUAL.....	37
<b>TABLA 7</b> ACCESO A LA INFORMACIÓN DEL REGISTRO ACTUAL .....	38
<b>TABLA 8</b> PORCENTAJE DE REGISTRO OPORTUNO DE LA INFORMACIÓN MANUAL.....	39
<b>TABLA 9</b> PORCENTAJE DE GARANTÍA DE INFORMACIÓN CON REGISTRO MANUAL.....	40
<b>TABLA 10</b> ACCESO A LA INFORMACIÓN DEL REGISTRO MANUAL .....	41
<b>TABLA 11</b> FRECUENCIA OBSERVADA.....	42
<b>TABLA 12</b> FRECUENCIAS ESPERADAS.....	42
<b>TABLA 13</b> PRUEBA DEL CHI CUADRADO.....	43
<b>TABLA 14</b> ACCESO A INFORMACIÓN MANUAL VS SISTEMA .....	44
<b>TABLA 15</b> RECURSOS DE HARDWARE PARA EL DESARROLLO DEL SISTEMA.....	67
<b>TABLA 16</b> RECURSOS DE SOFTWARE PARA EL DESARROLLO DEL SISTEMA .....	67
<b>TABLA 17</b> DICCIONARIO DE DATOS ANIMAL.....	72
<b>TABLA 18.</b> DICCIONARIO DE DATOS PROPIETARIO .....	73
<b>TABLA 19</b> DICCIONARIO DE DATOS CHEQUEOS VETERINARIO .....	73
<b>TABLA 20</b> DICCIONARIO DE VACUNAS .....	74
<b>TABLA 21</b> DICCIONARIO DE DATOS PARTOS .....	74
<b>TABLA 22</b> ESPECIFICACIONES TÉCNICAS MÍNIMAS PARA EL FUNCIONAMIENTO DE COREGVAC..	88

## ÍNDICE DE FIGURAS

<b>FIGURA 1</b> FUNCIONAMIENTO DE LA TECNOLOGÍA RFID .....	6
<b>FIGURA 2</b> COMPONENTES DE UN SISTEMA RFID .....	7
<b>FIGURA 3</b> CIRCUITO INTELIGENTE DE UNA ETIQUETA .....	9
<b>FIGURA 4</b> DIFERENTES TIPOS DE TAG'S .....	11
<b>FIGURA 5</b> DIFERENTES TIPOS DE LECTORES RFID .....	11
<b>FIGURA 6</b> DIAGRAMA DE BLOQUES DE UN LECTOR RFID .....	13
<b>FIGURA 7</b> DISPOSITIVO RFIDUDINO .....	14
<b>FIGURA 8</b> CONECTORES DE LA PLACA RFIDUDINO .....	14
<b>FIGURA 9</b> DISEÑO DEL SOFTWARE RFID.....	15
<b>FIGURA 10</b> BLOQUES FUNCIONALES DE UN MICROCONTROLADOR .....	16
<b>FIGURA 11</b> CONFIGURACIÓN DE PINES DEL ATMEGA 164P .....	17
<b>FIGURA 12</b> CONECTORES MODULO ARDUINO 1.....	18
<b>FIGURA 13</b> ALGUNOS TIPOS DE SHIELDS .....	19
<b>FIGURA 14</b> TIPOS DE ANTENAS DE MÓDULOS XBEE.....	22
<b>FIGURA 15</b> CONEXIONES REQUERIDAS PARA EL XBEE.....	23
<b>FIGURA 16</b> CONFIGURACIÓN PUNTO A PUNTO XBEE.....	24
<b>FIGURA 17</b> CONFIGURACIÓN PUNTO A MULTIPUNTO .....	25
<b>FIGURA 18</b> INTERFAZ GRÁFICA DEL SOFTWARE XCTU .....	27
<b>FIGURA 19:</b> VISUAL 2008 .....	29
<b>FIGURA 20</b> ARETES PARA GANADO.....	31
<b>FIGURA 21</b> ARETES PARA GANADO CON CÓDIGO DE BARRAS .....	31
<b>FIGURA 22</b> ARETE RFID .....	32
<b>FIGURA 23</b> ACCESO A INFORMACIÓN COMPLETA CON EL REGISTRO MANUAL.....	37
<b>FIGURA 24</b> ACCESO A INFORMACIÓN DE UN REGISTRO ANIMAL.....	38
<b>FIGURA 25</b> REGISTRO OPORTUNO DE LA INFORMACIÓN DE UN MANUAL.....	39
<b>FIGURA 26</b> GARANTÍA DE INFORMACIÓN CON REGISTRO MANUAL.....	40
<b>FIGURA 27</b> ACCESO A LA INFORMACIÓN DEL REGISTRO MANUAL .....	41
<b>FIGURA 28</b> PRUEBA DEL CHI - CUADRADO.....	44
<b>FIGURA 29</b> DIAGRAMA DE MÓDULOS DEL SISTEMA .....	52
<b>FIGURA 30</b> TARJETA DE DESARROLLO RFIDUINO.....	53
<b>FIGURA 31</b> MÓDULO RFID UDINO .....	53
<b>FIGURA 32:</b> VENTANA DE PROGRAMACIÓN PARA ARDUINO .....	55
<b>FIGURA 33</b> DETERMINACIÓN DEL PUERTO SERIAL PARA PRUEBAS ARDUINO .....	55
<b>FIGURA 34</b> COMPROBACIÓN DEL ID DE LOS TAGS EN ARDUINO .....	56
<b>FIGURA 35</b> DIAGRAMA DE PINES VISTA SUPERIOR.....	56
<b>FIGURA 36</b> CONEXIÓN DE UN XBEE CON SU PUERTO COM.....	57

<b>FIGURA 37</b> CONFIGURACIÓN DIRECCIÓN DEL XBEE .....	58
<b>FIGURA 38</b> CONFIGURACIÓN SERIAL TRANSMISOR.....	58
<b>FIGURA 39</b> DIRECCION UNICA MAC DEL XBEE RECEPTOR .....	59
<b>FIGURA 40</b> CONEXIÓN DE UN XBEE RECEPTOR CON SU PUERTO COM.....	59
<b>FIGURA 41</b> CONFIGURACIÓN DIRECCIÓN DEL XBEE RECEPTOR.....	60
<b>FIGURA 42</b> CONFIGURACIÓN SERIAL RECEPTOR.....	60
<b>FIGURA 43</b> DIAGRAMA DE FLUJO PARA LA PROGRAMACIÓN DEL PIC .....	61
<b>FIGURA 44</b> SIMULACIÓN Y PRUEBAS DE RECEPCIÓN.....	62
<b>FIGURA 45</b> CARGANDO EL PROGRAMA DEL PIC EN EL SIMULADOR .....	63
<b>FIGURA 46</b> PRUEBAS DE ENVÍO Y RECEPCIÓN DE DATOS .....	63
<b>FIGURA 47</b> DIAGRAMA ESQUEMÁTICO DEL MÓDULO DE RECEPCIÓN.....	63
<b>FIGURA 48</b> CIRCUITO DE RECEPCIÓN EN EL PROTOBOARD .....	64
<b>FIGURA 49</b> NUEVA CONEXIÓN DE HYPERTERMINAL.....	64
<b>FIGURA 50</b> PRUEBAS DE INTERCONEXIÓN ENTRE LOS DOS DISPOSITIVOS TRANSMISORES .....	65
<b>FIGURA 51</b> PLACA DE CIRCUITO IMPRESO DEL RECEPTOR .....	65
<b>FIGURA 52</b> CASO DE USO ACCESO AL SISTEMA .....	68
<b>FIGURA 53</b> CASO DE USO VISUALIZAR INFORMACIÓN DE LA RES .....	68
<b>FIGURA 54</b> CASO DE USO AGREGAR REGISTROS .....	69
<b>FIGURA 55</b> CASO DE USO MODIFICAR REGISTROS EN EL SISTEMA .....	69
<b>FIGURA 56</b> CASO DE USO ELIMINAR REGISTRO .....	70
<b>FIGURA 57</b> DISEÑO SISTEMA COREGVAC.....	70
<b>FIGURA 58</b> MODELO ENTIDAD RELACIÓN.....	71
<b>FIGURA 59</b> TABLAS EN LA BASE DE DATOS .....	72
<b>FIGURA 60</b> INGRESO AL SISTEMA COREGVAC.....	75
<b>FIGURA 61</b> INTERFACE GENERAL DE DATOS EN EL SISTEMA COREGVAC .....	76
<b>FIGURA 62</b> VENTANA PARA AGREGAR UN REGISTRO.....	76
<b>FIGURA 63</b> VENTANA PARA ELIMINAR UN REGISTRO.....	77
<b>FIGURA 64</b> VENTANA PARA ELIMINAR UN REGISTRO.....	77
<b>FIGURA 65</b> CONFIGURACIÓN DEL PUERTO VIRTUAL EN PROTEUS.....	78
<b>FIGURA 66</b> BÚSQUEDA DE PUERTOS CON EL SOFTWARE.....	78
<b>FIGURA 67</b> SELECCIÓN DEL PUERTO DE COMUNICACIÓN COM .....	79
<b>FIGURA 68</b> ENVÍO DE DATOS DESDE PROTEUS HACIA EL SOFTWARE COREGVAC .....	79
<b>FIGURA 69</b> SISTEMA COREGVAC.....	80
<b>FIGURA 70</b> EJECUTAR PROGRAMA DE INSTALACIÓN .....	88
<b>FIGURA 71</b> ASISTENTE DE INSTALACIÓN.....	89
<b>FIGURA 72</b> SELECCIÓN CARPETA DE INSTALACIÓN.....	89
<b>FIGURA 73</b> ARCHIVO DE BASE DE DATOS .....	90
<b>FIGURA 74</b> INGRESO AL SISTEMA.....	90
<b>FIGURA 75</b> VENTANA PRINCIPAL DE INFORMACIÓN .....	91



<b>FIGURA 76</b> PANEL DE CONEXIÓN CON LE RECEPTOR .....	91
<b>FIGURA 77</b> CINTA DE OPCIONES .....	92
<b>FIGURA 78</b> VENTANA PARA NUEVO REGISTRO ANIMAL.....	92
<b>FIGURA 79</b> VENTANA PARA NUEVO REGISTRO PROPIETARIO .....	93
<b>FIGURA 80</b> VENTANA PARA NUEVO REGISTRO VACUNAS.....	93
<b>FIGURA 81</b> VENTANA PARA NUEVO REGISTRO PARTOS .....	94
<b>FIGURA 82</b> VENTANA PARA NUEVO REGISTRO CHEQUEOS VETERINARIO .....	94
<b>FIGURA 83</b> VENTANA PARA ELIMINAR REGISTRO.....	95
<b>FIGURA 84</b> VENTANA PARA MODIFICAR REGISTROS.....	95
<b>FIGURA 85</b> MODO DE BÚSQUEDA SIN CONEXIÓN.....	96
<b>FIGURA 86</b> FINALIZAR APLICACIÓN.....	96
<b>FIGURA 87</b> CIRCUITO RECEPTOR EN EL PROTOBOARD.....	97
<b>FIGURA 88</b> MONTAJE Y SOLDADURA DE ELEMENTOS EN EL CIRCUITO IMPRESO DEL RECEPTOR	97
<b>FIGURA 89</b> REVISIÓN SUELDA DE ELEMENTOS EN EL CIRCUITO IMPRESO DEL RECEPTOR .....	98
<b>FIGURA 90</b> REVISIÓN FINAL ELEMENTOS EN EL CIRCUITO IMPRESO DEL RECEPTOR.....	98
<b>FIGURA 91</b> MONTAJE MÓDULO XBEE RECEPTOR EN EL MÓDULO .....	99
<b>FIGURA 92</b> MONTAJE FINAL DEL MÓDULO DE RECEPCIÓN .....	99
<b>FIGURA 93</b> MONTAJE MÓDULO DE LECTURA Y TRANSICIÓN .....	100
<b>FIGURA 94 95</b> MONTAJE ALIMENTACIÓN MÓDULO DE LECTURA Y TRANSICIÓN .....	100
<b>FIGURA 96</b> MONTAJE, CABLEADO MÓDULO DE LECTURA Y TRANSICIÓN.....	101
<b>FIGURA 97</b> ASEGURAMIENTO DE LA ANTENA MÓDULO DE LECTURA Y TRANSICIÓN.....	101
<b>FIGURA 98</b> MÓDULOS DE LECTURA/TRANSMISIÓN Y RECEPCIÓN LISTOS .....	102
<b>FIGURA 99</b> PRUEBAS MÓDULO DE RECEPCIÓN .....	102
<b>FIGURA 100</b> MÓDULOS EN OPERATIVIDAD .....	103
<b>FIGURA 101</b> PRUEBAS INICIALES DE LECTURA Y TRANSMISIÓN .....	103
<b>FIGURA 102</b> PRUEBAS SISTEMA COMPLETO.....	104
<b>FIGURA 103</b> PRESENTACIÓN DE TODO EL SISTEMA.....	104
<b>FIGURA 104</b> ANIMALES DEL HATO GANADERO .....	105
<b>FIGURA 105</b> ZONA DE PASTOREO .....	105
<b>FIGURA 106</b> DISTRIBUCIÓN DE ANIMALES EN LA ZONA DE PASTOREO .....	106
<b>FIGURA 107</b> UBICACIÓN DE LA INFRAESTRUCTURA .....	106
<b>FIGURA 108</b> MÓDULO DE LECTURA Y TRANSICIÓN EN LA ZONA DE PASTOREO .....	107
<b>FIGURA 109</b> PRUEBA1 DE LECTURA .....	107
<b>FIGURA 110</b> PRUEBA2 DE LECTURA.....	108
<b>FIGURA 111</b> BOSQUEJO DE INTERFAZ PRINCIPAL DEL SISTEMA.....	109
<b>FIGURA 112</b> BOSQUEJO DE INTERFAZ AGREGAR MODIFICAR .....	109
<b>FIGURA 113</b> BOSQUEJO DE INTERFAZ ELIMINAR REGISTRO.....	109

## CONTENIDOS

RESUMEN.....	xiii
SUMARY.....	xiii
INTRODUCCIÓN.....	1
CAPÍTULO I.....	2
1.    MARCO REFERENCIAL.....	2
1.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA.....	2
1.2. FORMULACIÓN DEL PROBLEMA.....	2
1.3. OBJETIVOS.....	3
1.3.1.    General.....	3
1.3.2.    Específicos.....	3
1.4. JUSTIFICACIÓN.....	3
CAPÍTULO II.....	5
2.    FUNDAMENTACIÓN TEORICA.....	5
2.1. Tecnología RFID Breve Historia.....	5
2.2. Definición RFID.....	6
2.2.1.    Componentes de un Sistema RFID.....	7
2.2.2    Tipos de Sistemas RFID.....	7
2.3 Transpondedor o Tag.....	9
2.3.1    Clasificación de las etiquetas RFID.....	9
2.4 Lectores.....	11
2.4.1    Operación Básica de un lector RFID.....	12
2.4.2    Componentes de un lector RFID.....	13
2.4.3    Tipos y selección de dispositivos RFID.....	14
2.4.4    Lector RFIDudino.....	14
2.5 Software RFID.....	15
2.6 Microcontroladores.....	16
2.6.1    Microcontrolador ATMEGA 164PA.....	16
2.7 Arduino.....	18
2.7.1    Lenguaje de Programación Arduino.....	19
2.7.2    Expandir Arduino con Shields.....	19
2.8 Transceptores.....	20

2.8.1	Módulo XBee .....	20
2.8.2	Tipos de Antenas Xbee.....	22
2.8.3	Circuito Básico para el XBee .....	23
2.8.4	Modos de operación .....	23
2.9	Software X-CTU .....	27
2.10	Sistema Gestor de Bases de Datos .....	28
2.10.1	Aplicaciones de los sistemas de las bases de datos .....	28
2.10.2	Gestor de base de datos Access .....	28
2.11	Acceso a Bases de Datos desde Programas de Aplicación.....	29
2.11.1	Microsoft Visual Basic 2008.....	29
2.12	Modos de identificación animal.....	30
2.12.1	Manejo y Rastreabilidad.....	30
2.12.2	Aretes visuales.....	30
2.12.3	Aretes de códigos de barras.....	31
2.12.4	Aretes RFID .....	31
CAPTULO III .....		33
3.	METODOLOGÍA .....	33
3.1.	Diseño de Investigación .....	33
3.2.	Tipos de investigación.....	33
3.3.	Método de Investigación .....	33
3.3.1	Método inductivo.-.....	33
3.4.	Población y Muestra.....	34
3.4.1	Población.....	34
3.4.2	Muestra.....	34
3.5.	Procedimientos .....	34
3.5.1	Técnica de Investigación.....	34
3.6.	Operacionalización de variables.....	35
3.6.1.	Variable Independiente.....	35
3.6.2.	Variable Dependiente .....	36
3.7.	Procesamiento y Análisis .....	37
3.7.1.	Comprobación de la Hipótesis.- .....	42
2.12.4.1	Establecimiento de la Hipótesis. ....	43
2.12.4.2	Establecer el nivel de Significancia.....	43

2.12.4.3	Elección de la prueba estadística.....	43
2.12.5	Decisión de la hipótesis .....	44
2.12.6	Comparativa del Acceso a la información y operatividad entre el sistema Manual y el sistema COREGVAC.....	44
CAPITULO IV .....		45
4.	RESULTADOS Y DISCUSIÓN .....	45
4.1.	Resultados .....	45
4.2.	Discusión .....	46
CAPITULO V .....		47
5.	CONCLUSIONES Y RECOMENDACIONES.....	47
5.1.	Conclusiones .....	47
5.2.	Recomendaciones .....	48
CAPITULO VI.....		49
6.	PROPUESTA.....	49
6.1.	Título de la Propuesta.....	49
6.2.	Introducción .....	49
6.3.	Objetivos .....	50
6.3.1.	Objetivo General .....	50
6.3.2.	Objetivos Específicos .....	50
6.4.	Requisitos de diseño e implementación del prototipo y selección de herramientas .....	50
6.5.	Diseño e implementación del hardware del prototipo.....	51
6.5.1.	Diagrama de bloques del hardware del prototipo.....	51
6.6.	Implementación del Hardware de Lectura y Transmisión .....	52
6.6.1.	Módulo de Lectura con RFIDudino y Arduino .....	52
6.6.2.	Programación y Pruebas del Arduino 1 y RFIDudino para conformar el modulo Lector.....	53
6.6.3.	Configuración XBee para Transmisión y Recepción .....	56
6.7.	Implementación del hardware de Recepción e Interface al Computador ..	60
6.7.1.	Programación Pic en el módulo de recepción .....	61
6.7.2.	Simulación en Proteus .....	62
6.8.	Montaje y Pruebas en el Protoboard .....	63
6.8.1.	Montaje del Circuito en el protoboard .....	64
6.9.	Diseño de la Placa del Circuito .....	65

6.10.	Desarrollo del Sistema .....	66
6.10.1.	Desarrollo de la Aplicación Ingeniería de sistema COREGVAC	66
6.10.2.1	Antecedentes y Problematización .....	66
6.10.2.2	Definición de Requerimientos .....	67
6.10.2.3	Análisis del requerimiento.....	67
6.10.2.4	Factibilidad técnica.....	67
6.10.2.5	Casos del uso del sistema .....	68
6.10.2.	Base de datos .....	70
6.10.1.2	Modelo entidad relación .....	71
6.10.1.3	Diccionario de datos .....	72
6.10.3	Diseño del sistema.....	75
6.11	Pruebas de Comunicación virtual con el software COREGVAC y Proteus.....	78
6.12	Resultado Integración y pruebas del Hardware y el software Resultado	80
6.13	CONCLUSIONES Y RECOMENDACIONES .....	81
6.13.2	Conclusiones .....	81
6.13.3	Recomendaciones .....	82
CAPITULO VII .....		83
7 BIBLIOGRAFIA .....		83
ANEXOS .....		85

## RESUMEN

La tesis que se presenta tuvo como objetivo principal diseñar un prototipo (software y hardware), para automatizar el proceso de control y registro de ganado vacuno en un hato específico, además mejorar el tratamiento de la información permitiendo llevar un registro estricto de las características propias de cada animal. Datos que a su vez permitan hacer un seguimiento y un control oportuno, optimizando el tiempo en búsquedas y generando resultados eficientes. Además de lo antes mencionado, garantizando la permanencia de los datos de forma segura y ordenada en un sistema que permita gestionar dicha información de modo continuo a través de una interfaz sencilla para el usuario y de rápidos resultados.

El prototipo terminado tiene como base en su hardware la tecnología de identificación por radio frecuencia RFID. Ésta está integrada con módulos arduinos, obteniendo un dispositivo electrónico el cual se encarga de obtener un código único existente en el arete de identificación del animal. La adquisición de datos de la etiqueta al computador se la hace mediante una transmisión de radiofrecuencia para lo cual se integran y programan los módulos de radiofrecuencia de la series Xbee, teniendo así dos dispositivos uno de transmisión y uno de recepción. Este módulo esta creado en base a un pic atmega el cual estandariza el código y lo envía al pc mediante la etapa de interface serial a USB.

El prototipo en su software trabaja con una base de datos en la cual se tiene todas las características del animal como son su peso, su raza, fotografía del mismo, etc. Todo el manejo de la información en la base de datos es interrelacionada con un software que tiene una interfaz amigable con el usuario además que es muy sencillo de manejar, es decir muestra de forma concreta la información. Un código RFID nos ayuda a visualizar en tiempo real cuantos animales están registrados y cuantos animales hay en la base de datos. De esta forma se obtiene la información de forma efectiva, lo que conlleva a mejorar los accesos a la misma supliendo los procesos manuales de registro y control para tener un seguimiento más eficiente.



UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA  
CENTRO DE IDIOMAS



Lic. Ruth Molina

11 de Mayo del 2015

### SUMMARY

The thesis that is present had like a main objective to design a prototype (software and hardware) to automate the process of control and record of dairy cattle in a specific herd, further to improve the treatment of the information, allowing to take a strict record of the proper characteristics of every animal. Information that permit to do a tracing and opportune control, optimizing the time in searches and generating efficient results. Additionally, it will guarantee permanence of the information in a safe and organized way in a system to allow manage this information in a continuous mode through a simple interface to the user and fast results.

The finished prototype has like a base in its hardware the technology of identification by radio frequency RFID. It is integrated by modules arduinos, obtaining an electronic device its function is to obtain an existing unique code in the earring of the animal identification. The acquisition of information of the tag to the computer is made by means of a radio frequency transmission for which they integrate and program the radio frequency modules of the series Xbee, so having two devices, of transmission and one of reception. This module is created based on a pic atmega which standardizes the code and send it to the PC by means of stage of serial interface to the USB

The prototype at its software works with a database which contains all the characteristics of the animal such us: its weight, race, picture, etc. The complete management of the information in the database is interrelated with a software that has an attractive interface with the user farther It is very simply to handle. It show the information in a specific way. A RFID code helps to visualize in real time how many animals are there registered and how many animals are there into the database. Thus, the information is obtained more effectively, which faster the accesses to get it replacing the traditional processes of record and control for getting more efficient tracing.

CENTRO DE IDIOMAS



COORDINACION

## **INTRODUCCIÓN.**

El control y registro de animales vacunos se ha convertido en una herramienta importante para monitorear, erradicar enfermedades evitar fraudes y sobre todo inyectar un valor agregado a los animales ya que los productores utilizan esta herramienta como un elemento de trazabilidad y los consumidores tienen una seguridad alimenticia. Es así que se han ido implementado varios tipos de sistemas en donde se puede hacer un seguimiento o trazabilidad del animal obteniendo procedimientos muy sofisticados altos en presupuestos etc.

La tecnología RFID en la actualidad presenta una alternativa robusta de desarrollo para sistemas en donde se necesite tener un seguimiento de un producto dentro de un proceso o también llamado trazabilidad saber de dónde vino el producto y hacia donde llegó lo que le agrega seguridad y calidad hacia el consumidor. Aplicando los mismos principios de identificación y utilizando alternativas tecnológicas como la RFID conlleva a desarrollar sistemas de identificación animal más económicos modulares y sobre todo fáciles de integrar y manejar dentro de una hato.

Para el diseño y la implementación del prototipo de control y registro de ganado vacuno o COREGVAC se realizaron dos fases la fase de construcción de Hardware y la fase de construcción del software, utilizando módulos de transmisión y recepción RF para su conectividad, lo que conlleva a tener un sistema diferente a los convencionales ya que es independiente a otros medios tradicionales de transmisión y recepción (internet wirelles, etc) además su costo es muy bajo comparado con algunos sistemas de registro, ya que no es necesario comprar módulos de lectura adicionales si no que ya se integró en un solo dispositivo lectura/transmisión de datos hacia un receptor convirtiéndose en una herramienta eficiente de bajo costo y obteniendo un resultado eficaz al momento de implementarlo.



# CAPÍTULO I

## 1. MARCO REFERENCIAL

### 1.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA.

A nivel regional no se cuenta con un sistema de control y registro computacional de animales vacunos confiable que permita conocer acontecimientos que se realizan en los ganados como el de vacunación, peso, etc.; de manera fácil, automática y eficaz, con un tratamiento de la información en tiempo real y sencillo de manejar.

Por el desconocimiento de tecnologías que ayuden a recolectar esta información importante en un hato de producción es que ganaderos de nuestra región llevan sus registros de forma inconsistente (cuadernos, hojas , etc.). de forma tal que implica pérdidas de información seguimiento y control bajo, además como la crianza de esta especie involucra movilización a lugares de difícil acceso y a distancias moderadas de su infraestructura (hacienda, fincas etc.) no se tiene muchas veces toda la información en tiempo real de cada animal es decir el conjunto de estas prácticas conlleva a tener baja productividad, pérdidas de los animales propagación de enfermedades etc.

Debido a las reformas agrarias en el Ecuador y al crecimiento en la población de animales vacunos, se hace imprescindible un sistema de control y registro computacional que permita transmitir dicha información de forma confiable y a distancias y sobre todo tener un sistema en el cual se obtenga mediante el código único asignado a cada animal toda la información relacionada a este.

### 1.2. FORMULACIÓN DEL PROBLEMA.

¿Cómo incide el diseño e implementación de un prototipo en el control y registro computacional de ganado vacuno utilizando la tecnología RFID?

### **1.3. OBJETIVOS.**

#### **1.3.1. General**

Diseñar e Implementar un prototipo para el control y registro computacional de ganado vacuno utilizando la tecnología RFID

#### **1.3.2. Específicos**

- Investigar el funcionamiento de la Tecnología RFID para aplicarla en la identificación en hatos ganaderos.
- Sustentar teóricamente la investigación analizando los dispositivos electrónicos con su tecnología que incidan en la eficiencia y eficacia del prototipo.
- Implementar un prototipo para el registro y control basado en la tecnología RFID y en la transmisión inalámbrica.

### **1.4. JUSTIFICACIÓN**

Realizar una herramienta que ofrezca la mayor efectividad en el proceso de crianza (trazabilidad) en el ganado vacuno, elevar la competitividad del agro, fortaleciéndolo técnicamente, para alcanzar con altos estándares de productividad, bajos costos unitarios y calidad, que garanticen la presencia estable y creciente de la producción en los mercados internos, además erradicar las enfermedades como la aftosa, son entre otros los objetivos del ministerio de agricultura en nuestro país; los mismos que no pudieran ser llevados a cabo sin el debido proceso previo de registro y control de la especie en cuestión.

La falta de un Sistema de control y registro computacional de animales que permita tener un seguimiento claro del nivel de cumplimiento de los requerimientos que conlleva la crianza y derivados (carne, leche, etc.) para que autoridades de control, faenadores o compradores puedan tomar decisiones tiene

las siguientes causas desconocimiento de tecnologías modernas, crecimiento en la población de animales vacunos, falta de definición de los procesos y seguimiento. Si no se implementa un sistema para el control y registro computacional de animales podrían surgir los siguientes efectos: La mala imagen del lugar en donde se hace la crianza por no existir la inversión suficiente en tecnología, además puede ocasionar pérdidas y generar desconfianza a nivel de autoridades control y baja o nula competitividad en lo más importante que son los consumidores.

## CAPÍTULO II

### 2. FUNDAMENTACIÓN TEORICA

#### 2.1. Tecnología RFID Breve Historia

Durante la segunda guerra mundial aparece la base de la tecnología de Identificación por radio frecuencia o RFID<sup>1</sup> con la finalidad de identificar aeroplanos y es usada por la Fuerza Aérea Británica, para diferenciar sus propios aeroplanos de los aeroplanos alemanes. Luego en la década de los 70 Estados Unidos implementa sistemas para el de seguimiento de materiales nucleares y de ganado vacuno, al transcurrir el tiempo se mercantilizó los sistemas de identificación a base de radio frecuencia.

A principio de los 90 la empresa IBM patenta un sistema RFID UHF (Ultra-High Frequency), logrando integrar todo el circuito en un solo chip y bajando el costo, siendo la base para el desarrollo acelerado y es así que el Instituto de Massachussets (MIT) investiga la posibilidad de colocar etiquetas RFID de bajo costo en productos para hacer un seguimiento en la cadena de suministros.

El avance más importante se da entre los años 1999 y 2003 cuando Auto-Id Center obtiene la ayuda de numerosas empresas que lideraban el comercio y la industria además de le departamento de Defensa de los Estados Unidos y muchos vendedores de sistemas RFID, es así que se desarrollan los primeros estándares para la captura de datos entre ellos el Código Electrónico del producto EPC<sup>2</sup>, el cual está formado por una serie de números que identifican de una forma única a objetos (cajas, artículos, etc.) a lo largo de la cadena de suministros. En el 2003 se creó EPC Global el mismo que tenía por objetivo de velar por el desarrollo de la red EPC y sus estándares (Epc global Gen1 y EPC global Gen 2). (INTECO, Instituto Nacional de Tecnologías de la Comunicación Española, 2010)

---

<sup>1</sup> Radio Frequency Identification o Identificación por Radio Frecuencia

<sup>2</sup> Electronic Product Code o Código Electrónico de Producto

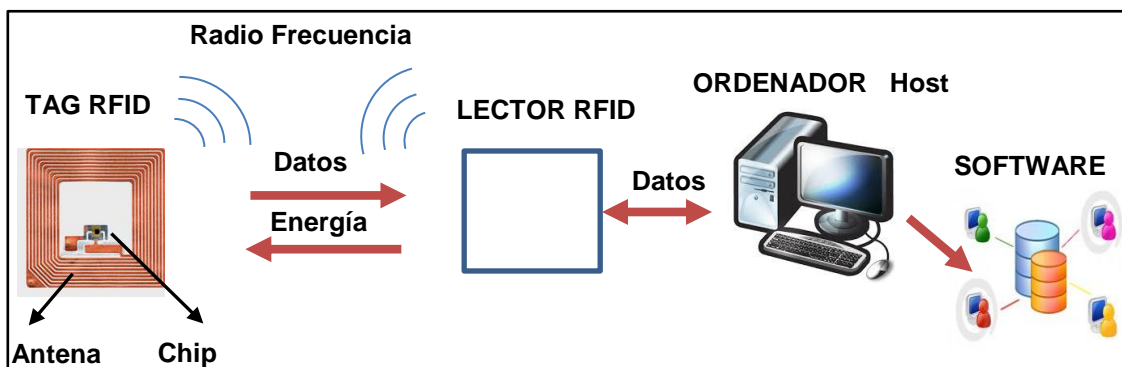
1948	Se inventa para identificar aviones amigos en la segunda guerra mundial RFID
1950-1960	Se realizan las primeras investigaciones y experimentos en laboratorios, sobre la tecnología RFID.
1960-1970	En esta década se realizan Pruebas de Campo en el desarrollo de la Teoría de la identificación por radiofrecuencia
1970-1980	Empieza un desarrollo acelerado ya que se logra integrar en un solo chip y aparecen las primeras implementaciones adaptadas con RFID.
1980-1990	La identificación por radio frecuencia tiene mayor importancia en aplicaciones Comerciales
1990-2000	Estándares RFID aparecen esto conjuntamente con varas entidades industriales y de defensa.
2000-2014	RFID se combina con servicios móviles, subcutáneos para animales y humanos, y llega a formar parte del diario vivir. supliendo varios procesos en cuanto a la identificación seguimiento trazabilidad en animales o cosas. las aplicaciones mejoran cada vez más

**Tabla 1** Evolución de la tecnología RFID

Fuente: El Autor

## 2.2. Definición RFID

Se trata de una tecnología de transmisión de datos automática que emplea ondas electromagnéticas que permiten la captura y grabación de datos entre el lector y la etiqueta, tarjetas, o tags RFID siendo el propósito fundamental el transmitir la identidad de un objeto similar a un número de serie único mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas Auto ID (Automatic Identification o Identificación Automática). (LIBERA, 2010)

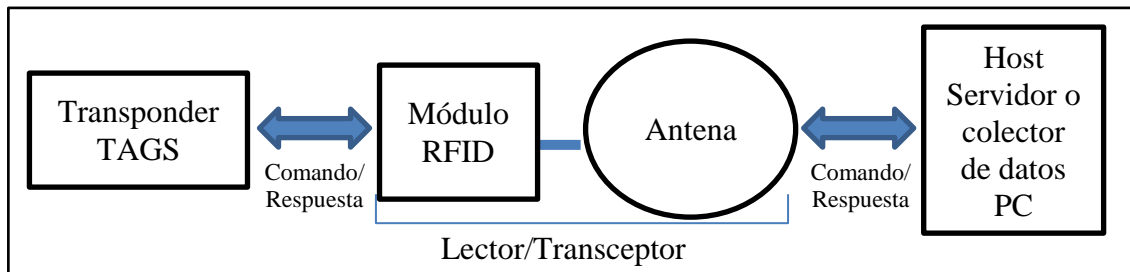


**Figura 1** Funcionamiento de la tecnología RFID

Fuente: El Autor

### 2.2.1. Componentes de un Sistema RFID

Los sistemas de RFID están constituidos por varios componentes principales y estos son: tags, lectores, antenas y un host (computadora central).



*Figura 2 Componentes de un sistema RFID*

*Fuente: El Autor*

### 2.2.2 Tipos de Sistemas RFID

La clasificación de un sistema RFID se da en función de varias cuantificaciones como: el rango de frecuencias en el que opera, el modo de comunicaciones entre el lector y las etiquetas, el principio de funcionamiento.

#### 2.2.2.1 Según los Rangos de Frecuencia

La frecuencia a la que trabaja un sistema RFID influye directamente en las características del sistema, como por ejemplo rango y velocidad de lectura, diseño de las antenas y la manera en la que la etiqueta y el lector interactúan entre sí, en consecuencia dependiendo de la frecuencia de trabajo, el sistema puede que sea o no apropiado para una determinada aplicación. **(Telectrónica, 2006)**. De esta manera se tiene los siguientes tipos de sistemas:

- Sistemas RFID de baja frecuencia (Low- Frequency): Su frecuencia de funcionamiento esta entre 125 -134,2 Khz.
- Sistemas RFID de alta frecuencia (High- Frequency): Su frecuencia de funcionamiento es 13.56Mhz.
- Sistemas RFID de Ultra alta frecuencia (Ultra High- Frequency): Su frecuencia de funcionamiento esta entre 860 – 960Mhz

- Sistemas RFID de microondas: Su frecuencia de funcionamiento es mayor a los 2.4 Ghz, son muy poco utilizados por no haber regulaciones globales para su uso.

### 2.2.2.2 Según el Tipo de Comunicación entre Lector y Etiqueta

Para establecer la comunicación entre lector y etiqueta esta se puede dar de dos modos Full dúplex o Half dúplex

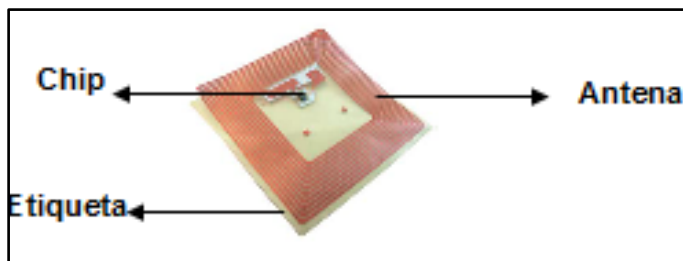
- **Full-duplex (FDX):** En este tipo de comunicación la señal de retomo empieza tan pronto como empieza la señal de interrogación y el acumulador se ha cargado. La señal de retomo se recibe repetidamente y sin interrupción mientras se mantenga la señal de interrogación. Consecuentemente, un transponder FDX no necesita almacenar energía para ser capaz de devolver su código entero. (**Telectrónica, 2006**)
- **Half-duplex (HDX):** En este tipo de comunicación la señal de retomo empieza sólo después de terminar a señal de interrogación y si el acumulador se ha cargado totalmente. La señal de retomo sólo se envía una vez, dado que el transponder vacía su acumulador después de enviar su código, el modo de comunicación HDX en un sistema de baja frecuencia, en el cual se utiliza una frecuencia de transmisión variable: 124,2 Khz. para codificar un "1" y 134,2 Khz. para codificar un "0". (**Telectrónica, 2006**)

### 2.2.2.3 Según el Acoplamiento entre el Lector y la Etiqueta

El acoplamiento es el mecanismo que determina el modo en que el circuito de la etiqueta y el circuito del lector se comunican mutuamente permitiendo la transmisión y recepción de datos y energía. El tipo de acoplamiento afecta directamente al rango de lectura del sistema se acoplan de dos formas. El acoplamiento Inductivo se usa para comunicaciones de baja y alta frecuencia (LF,HF) y acoplamiento capacitivo para frecuencias UHF y microondas.

## 2.3 Transpondedor o Tag

Un Tag RFID está compuesto por un microchip y una antena flexible instalada sobre un sustrato PET flexible. Este sustrato o “inlay” es luego instalado en una etiqueta con adhesivo de base a pesar de que los chips son pequeños, las antenas no lo son ellas necesitan ser lo suficientemente grandes como para captar la señal emitida por el lector. La antena permite que una etiqueta pueda leerse a una distancia de 3 metros o más, incluso a través de distintos materiales, El tamaño de la antena tiende a determinar el tamaño de una etiqueta RFID, en la siguiente figura se muestra los componentes de una etiqueta o Tag



*Figura 3 Circuito Inteligente de una Etiqueta*  
*Fuente: El Autor*

### 2.3.1 Clasificación de las etiquetas RFID

Las Etiquetas RFID se clasifican en:

#### ➤ Según la configuración de memoria que poseen

- **Solo lectura:** El código de identificación único y es personalizado durante su fabricación.
- **De lectura y escritura:** El código en el tag puede ser modificado a voluntad del usuario. (Con un lector RFID de escritura/escritura).
- **De una sola escritura:** Se puede escribir el código una sola vez y esta se vuelve solo de lectura.



➤ **Según la forma de energizar la etiqueta**

- **Pasivas :** Las etiquetas no poseen alimentación eléctrica. Se activan solo cuando el lector induce una corriente eléctrica mínima suficiente para generar un campo magnético y la etiqueta se energiza para responder al lector esta respuesta es conocida como “Backscatter”.
- **Activas:** Poseen una fuente de alimentación autónoma de energía que hace funcionar a sus circuitos integrados y el transmisor para propagar su señal al lector. Estos tipos de etiquetas son capaces de transmitir a largas distancias . Pueden incorporar una pequeña memoria que les permite almacenar mayor cantidad de información su limitación principal es la batería porque su tiempo de vida se limita a esta.

A continuación se hace una comparativa entre los tipos de tags

<b>Tag Pasivo</b>	<b>Tag Activo</b>
Funciona sin batería, Relativamente económico	Funciona con batería, Relativamente costoso
Ciclo de vida ilimitado	Ciclo de vida limitado por la batería
Poco peso - Alcance limitado (3 - 5m)	Mayor peso - Mayor alcance (100 m)
Sensible al ruido	Mayor inmunidad ante presencia de ruido
Dependencia de la señal del dispositivo lector	Transmisor propio
Requiere dispositivos lectores potentes	Relaja el requisito de potencia de los lectores
Velocidad de transmisión baja	Velocidad de transmisión alta
Lectura simultánea baja , Alta sensibilidad de orientación	Lectura simultánea alta, Menor sensibilidad de orientación

**Tabla 2** Comparativa entre Tag Pasivos y Tag Activos

**Fuente:** El Autor

➤ **Capacidad de Almacenamiento de Datos**

- 1 bit (anti-hurto, como los de supermercados)
- Capacidades de memoria de varios Kbits



**Figura 4** Diferentes tipos de Tag's

**Fuente:** <http://www.coresonant.com/html/rfid-tags-for-solar-module-india.html>

## 2.4 Lectores

Los dispositivos de lectura son los encargados de leer o escribir la información que tiene el Tag, estos suministran energía a las etiquetas pasivas para poder comunicarse con ellas podría decirse que los lectores son la parte fundamental de una sistema RFID ya que estos sirven de interfaz entre la etiqueta, el sistema de almacenamiento y el procesamiento de la información usualmente soportan múltiples protocolos. (Telectrónica, 2006)



**Figura 5** Diferentes tipos de Lectores RFID

**Autor:** <http://www.rfidpoint.com/wp-content/uploads/lector.jpg>

### 2.4.1 Operación Básica de un lector RFID

Para la mayoría de las aplicaciones, los lectores operarán automáticamente o como equipos dirigidos. Los lectores utilizan la banda de 902-928 MHz (USA) dividida en aproximadamente 60 canales. La metodología de modulación es denominada Frequency Hopping Spread Spectrum y ha sido establecida por la FCC para minimizar la interferencia con otros dispositivos de RF. Algoritmos anti-colisión son utilizados para leer y clasificar ingresos múltiples y simultáneos de distintos tags. **(INTECO, Instituto Nacional de Tecnologías de la Comunicación Española, 2010)**

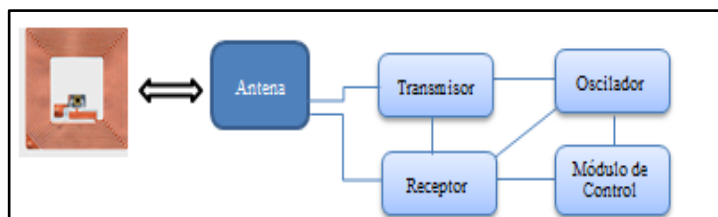
- **Lectura Fija:** Un lector puede setearse para operar en forma constante realizando lecturas fijas y acumulando listas de tags en su memoria. Las listas de tags representan la población actual de etiquetas en su rango de lectura . A medida que los tags responden a las emisiones del lector son incluidos en la lista. Si no responden, son eliminados de la lista acumulada en la memoria. Un lapso de tiempo determinado es fijado para establecer cuando un tag debe ser removido de la lista. Un sistema central puede recibir una lista de etiquetas desde el lector cuando desee actualizar sus registros. La información disponible que recibe el host incluye la ubicación del lector, el tiempo de lectura, el tamaño de la lista de tags, y la identificación de cada etiqueta en la lista. **(Telectrónica, 2006)**
- **Modo Directo / Interactivo:** Los lectores que operan bajo esta modalidad responderán a los comandos del host. El host puede indicar al lector reunir una lista de etiquetas dentro del rango de lectura o buscar una etiqueta específica. En ambos casos el lector comienza por recoger una lista. Una vez completado el comando instruido por el host, el lector espera hasta recibir el siguiente. **(Telectrónica, 2006)**

## 2.4.2 Componentes de un lector RFID

Un lector RFID consta de los siguientes componentes:

**Antena:** Es el elemento más sensible de todo el sistema RFID ya que este capta las señales de baja potencia que irradian las etiquetas estas pueden variar su ubicación dependiendo de la aplicación que se le va a dar y así las antenas pueden ser:

- **De puerta:** Son utilizadas en control de accesos o en sistemas antirrobo generalmente están asociadas a una alarma.
  - **De sobremesa:** Se utilizan para lectura y escritura de etiquetas se incorporan a los puntos de identificación y control de terminales de servicios.
- 
- **Módulo de Radiofrecuencia:** Encargado de proveer una señal portadora al modulador del transmisor y una señal de referencia al demodulador en el receptor.
  - **Transmisor:** Se encarga de crear una señal de radio frecuencia modulada que contiene comandos e información para las etiquetas RFID.
  - **Receptor:** Se encarga de demodular la señal que recibe el lector.
  - **Unidad de Control:** Se encarga de procesar los datos transmitidos y recibidos tiene su memoria para almacenar los datos que reciben de sus etiquetas, controla todos los componentes del lector y es el encargado de interactuar con el usuario.



*Figura 6 Diagrama de bloques de un lector RFID*

*Fuente: El Autor*

### 2.4.3 Tipos y selección de dispositivos RFID

#### 2.4.4 Lector RFIDudino

Es un componente (Shield ), ofrece la capacidad de expandirse de forma inalámbrica tiene un socket para XBee; añade más capacidades a un arduino haciendo a los sistemas más robustos y confiables sobre todo modulares consta de: RFIDuino Shield, RFIDuino Antena, 20cm Cable de conexión (Arduino, 2013)

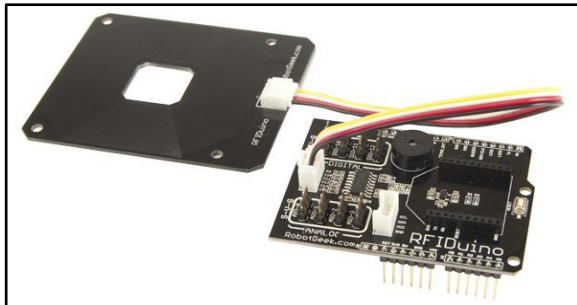


Figura 7 Dispositivo RFIDudino

Fuente: <http://www.trossenrobotics.com/rfiduino>

##### 2.4.4.1 Características

- XBee Socket para la comunicación inalámbrica
- Antena RFID está separado de la placa principal
- Fácil acceso 3 pines de señal-Vcc-Ground , pines analógicos y digitales
- Incluidos en la placa un Zumbador y LEDs

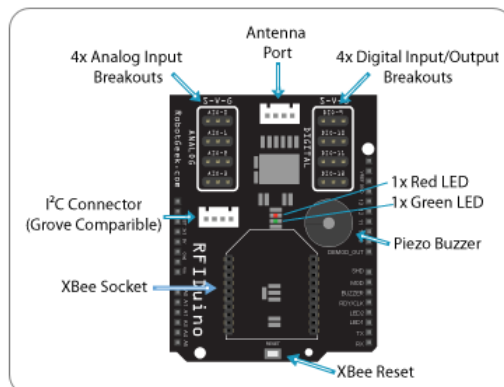


Figura 8 Conectores de la placa RFIDudino

Fuente: <http://www.trossenrobotics.com/rfiduino>

#### 2.4.4.2 Salidas del Shield

- 1 x Buzzer - Digital Pin 5, 1 x LED Rojo - Digital Pin 3, 1 x LED verde - Digital Pin 2

#### 2.4.4.3 Conectores Externos

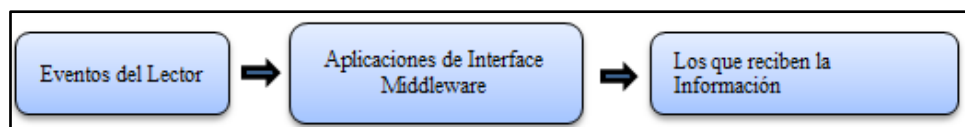
- 1 x XBee socket - Digital Pines 0/1 RX/TX
- 4 x 3-pin RobotGeek Digital I/O Ports - Digital Pins 9, 10, 11, 12
- 4 x 3-pin RobotGeek Analog Output Ports - Analógica Pins 0,1,2,3
- 1 x 4-pin Puerto I2C - Analógica Pines 4/5 , 1 x 4-pin Puerto para la antena

#### 2.4.4.4 Rango de operación

- El rango promedio de lectura del lector es 70-100mm. Esto puede variar dependiendo de la cantidad de energía que SE dar el sistema y el tipo de etiqueta que está utilizando.
- El lector utilizan el estándar EM4102.
- Etiquetas son de sólo lectura. Cada etiqueta viene con un número de identificación único predeterminada de 40 bits (5 bytes).
- El RFIDuino sólo está diseñado para trabajar con los módulos XBee 1mW

### 2.5 Software RFID

El Software RFID debe ser capaz de administrar la interacción entre el lector, las etiquetas y la aplicación para el usuario. Para realizar estas actividades el software RFID ha sido dividido en 3 capas las cuales se ilustran en la figura N° 2.5, y se detallan a continuación:

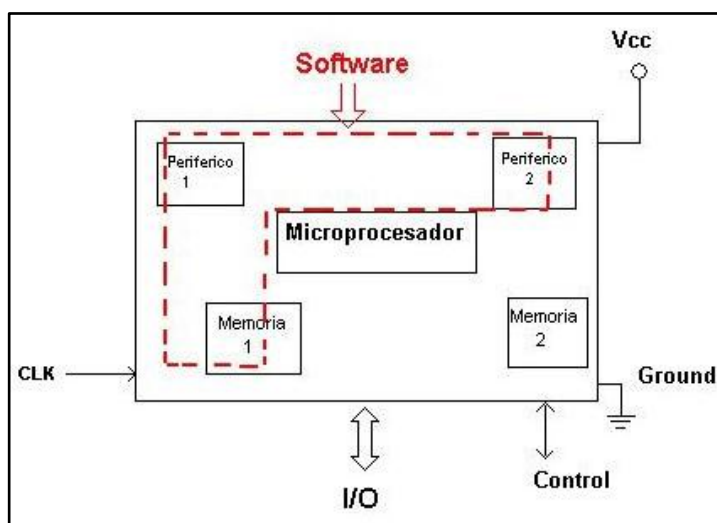


*Figura 9* Diseño del software RFID

*Fuente:* El Autor

## 2.6 Microcontroladores

Un microcontrolador (abreviado  $\mu\text{C}$ , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres unidades funcionales principales: unidad central de procesamiento (CPU), memoria y periféricos de entrada y salida (I/O). (Academia.edu, 2014)



*Figura 10* Bloques Funcionales de un microcontrolador

*Fuente:* <http://medialab.usal.es/facartec/files/2012/10/B%C3%A1sicos-Arduino.pdf>

Un microcontrolador para que pueda realizar algún proceso es necesario crear y luego grabar en la memoria EEPROM algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para Microcontroladores; debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que lo hace trabajar cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

### 2.6.1 Microcontrolador ATMEGA 164PA

Es un microcontrolador CMOS de ocho bits de bajo consumo basado en arquitectura RISC mejorada. Sus instrucciones se ejecutan en un ciclo de

máquina, el ATmega 164 P consigue transferencia de información alrededor de 1MIPS por MHz admitido por el sistema, permitiendo al diseñador del sistema optimizar el consumo de energía versus la velocidad de procesamiento (Atmel, 2014)

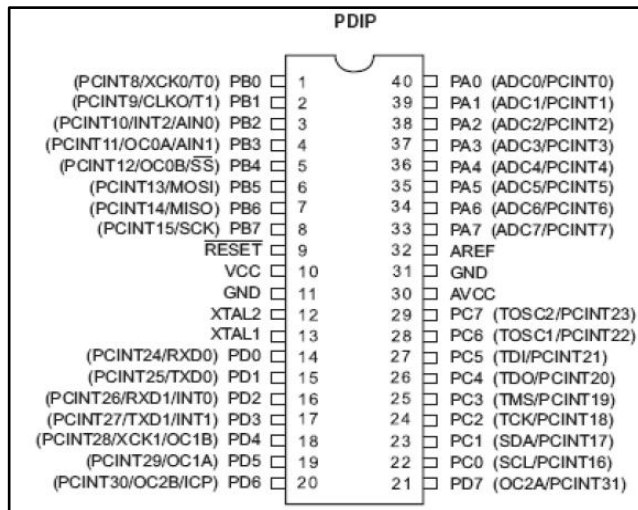


Figura 11 Configuración de pines del ATmega 164P  
Fuente: ATmega164P guía.pdf

### 2.6.1.1 Características el ATmega provee las siguientes características

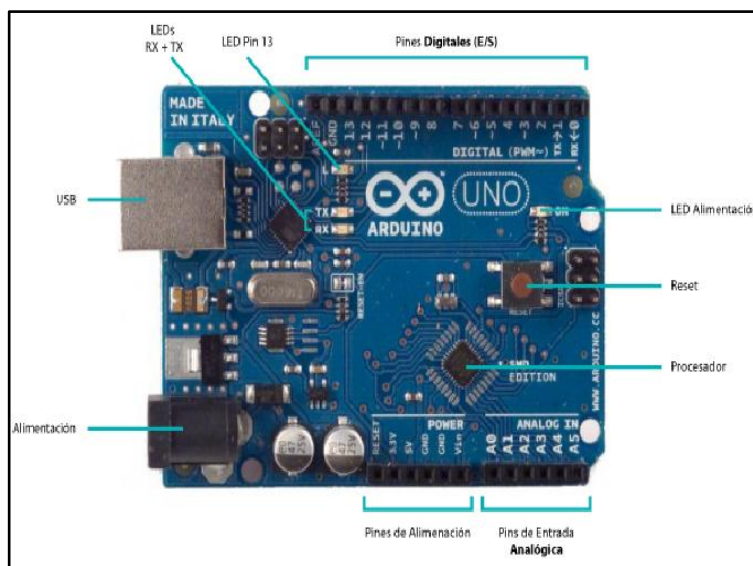
<b>Sistema de Flash Programable</b>	<b>16K Bytes</b>
<b>Capacidad de lectura y escritura en la EEPROM</b>	512Kb
<b>Capacidad de lectura y escritura en la SRAM</b>	1Kb
<b>Pines de Propósito general E/S</b>	32
<b>Registros de propósito general</b>	32
<b>Timers</b>	3
<b>Canales ADC</b>	8 –de 10 bits con opción de entrada diferencial

Tabla 3 Características del microcontrolador ATmega 164p  
Fuente: El Autor



## 2.7 Arduino

Es una plataforma de desarrollo de computación física (physical computing) de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software (programas) para su placa. se dise también que es una herramienta capaz de controlar el mundo físico través de un ordenador Arduino se utiliza para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos estos proyectos pueden ser autónomos o comunicarse con un programa (software) que se ejecute en un ordenador. La placa podemos crearla o adquirirla ya lista para usar, y el software de desarrollo es abierto y se lo puede descargar gratis. El lenguaje de programación de Arduino es una implementación de Wiring<sup>3</sup>, una plataforma de computación física parecida, que a su vez se basa en Processing, un entorno de programación multimedia. (Arduino, 2013)



**Figura 12** Conectores Modulo Arduino 1

**Fuente:** [medialab.usal.es/facartec/files/2012/10/Básicos-Arduino.pdf](http://medialab.usal.es/facartec/files/2012/10/Básicos-Arduino.pdf)

<sup>3</sup> **Wiring.**- Es un marco de programación de código abierto para microcontroladores para controlar a los dispositivos conectados a una amplia gama de placas

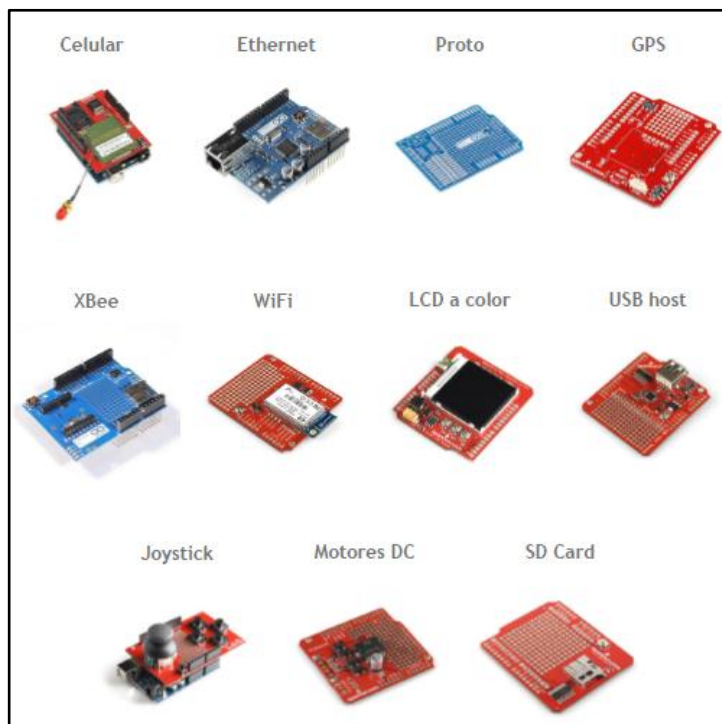
### 2.7.1 Lenguaje de Programación Arduino

Arduino se programa en el lenguaje de alto nivel C/C++ y generalmente tiene los siguientes componentes para elaborar el algoritmo:

- Estructuras
- Variables
- Operadores matemáticos, lógicos y booleanos
- Estructuras de control (Condicionales y ciclos)
- Funciones

### 2.7.2 Expandir Arduino con Shields

Un Shield es una placa que permite expandir funcionalidades a un Arduino, con lo cual podemos conectar motores, o a la red celular, a una red WiFi, a una red Ethernet o tener un MP3 en el Arduino, entre muchos más.



**Figura 13** Algunos tipos de Shields

**Fuente:** <https://learn.sparkfun.com/tutorials/what-is-an-arduino>

## 2.8 Transceptores

Un transceptor ( Transceiver) no es más que un equipo que usa circuitos comunes con el fin de proveer las capacidades de transmisión y recepción.

### 2.8.1 Módulo XBee

Un módulo Xbee son dispositivos integrados que brinda un medio inalámbrico para la interconexión y comunicación entre dispositivos trabajan en la banda de 2,4Ghz, estos módulos utilizan el protocolo de red llamado IEEE 802.15.4 para crear redes punto a multipunto o para redes punto a punto, fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia, y una sincronización de comunicación predecible estos módulos son muy fáciles de usar y son de propiedad de Digi basado en el protocolo Zigbee<sup>4</sup> .

Existen varias series de estos tipos de módulos entre esas tenemos el XBee series1, XBee series pro, XBeeZnet 2.5, variando únicamente en características de alcance y qué tipo de firmware a utilizar, además que no son compatibles por ejemplo no puedo conectar un serie 1 con un serie Znet. (Xbee.cl Ingenieria MCI Ltda, 2014)

#### 2.8.1.1 XBee 802.15.4

La Serie 1 viene con firmware 802.15.4 que permite redes punto a punto o de topología en estrella. Este firmware ofrece ADC (conversión analógica-digital), entradas y salidas digitales, y 128-bit AES

- **Capa física.-** Tipo de modulación y frecuencia (2.4 Ghz, 16-QAM, 915 Mhz y 868 MhzDSSS-BPSK) y el formato del paquete.

---

<sup>4</sup> Zigbee: - Es una alianza y un estándar de redes MESH de eficiencia energética y de costos. XBee utiliza el estándar Zigbee y lo agrega y envuelve en un pequeño empaque elegante

- **Control de acceso al medio.-** timing, asociación con un nodo coordinador, direccionamiento, prevención de colisiones, retransmisión, encriptación, ruteo de dos saltos a través de un nodo coordinador, etc...

### 2.8.1.2 El protocolo ZigBee,

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE\_802.15.4, el protocolo está preparado para poder controlar en la misma red la cantidad de hasta 65535 equipos. (MCI Electronics, 2010)

Entre las necesidades que satisface el módulo se encuentran:

- Bajo costo
- Ultra-bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.
- Instalación barata y simple.
- Redes flexibles y extensibles.

#### ➤ **Capa de red**

Permite múltiples saltos utilizando el ruteo por redes MESH<sup>5</sup>. Asociación a una red (no sólo a un nodo coordinador como el caso de 802.15.4) Más tipos de encriptación y autenticación. (MCI Electronics, 2010)

#### ➤ **Capa de Aplicación**

Permite a los nodos exponer valores individuales, como sensores, switches o actuadores. Perfiles para varios tipos de dispositivos, etc. (MCI Electronics, 2010)

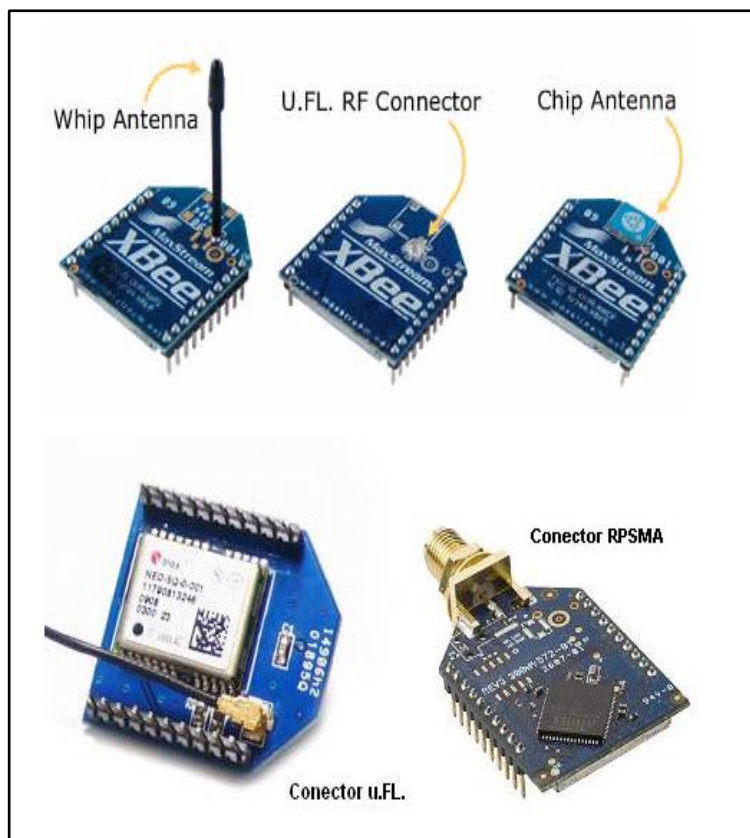
---

<sup>5</sup> **Redes MESH.-** Permiten acceder a un punto remoto, utilizando módulos intermedios para llegar como routers.

## 2.8.2 Tipos de Antenas Xbee

Existen varios tipos de antenas para estos módulos estos son:

- **Chip Antena.-** Básicamente es un pequeño chip que actúa como antena
  - **Wire Antena .-** (Wip Antenna).- Es un pequeño cable que sobresale del dispositivo.
  - **u.FL Antena.-** Un conector pequeño para conectar nuestra propia antena si tiene un equipo una caja y deseamos que la antena este afuera de esta.
  - **RPSMA Antena.-** Es un conector más grande para conectar nuestra propia antena, igual como el anterior es perfecto sin nuestro dispositivo se encuentren una caja y deseamos que el antena baya afuera de esta.
- (Xbee.cl Ingenieria MCI Ltda, 2014)

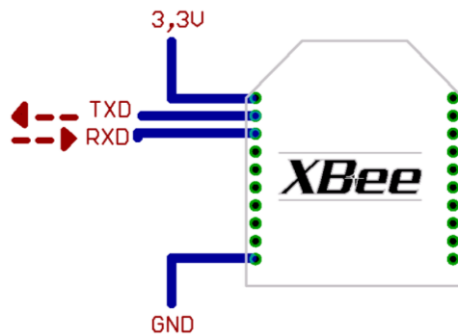


**Figura 14** Tipos de antenas de módulos XBee

**Fuente:** [http://www.olimex.cl/tutorials.php?page=tut\\_xbee](http://www.olimex.cl/tutorials.php?page=tut_xbee)

### 2.8.3 Circuito Básico para el XBee

A continuación se muestran las condiciones mínimas que necesita el módulo XBee para poder ser utilizado, luego de esto, se debe configurar según el modo de operación que se desea para la aplicación requerida por nosotros



*Figura 15 Conexiones Requeridas para el XBee*  
*Fuente: Xbee-Guia Usuario.pdf*

El módulo requiere una alimentación desde 2.8 a 3.4V, la conexión a tierra y las líneas de transmisión de datos UART (TXD y RXD), para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún conversor adecuado para los niveles de voltaje. (MCI Electronics, 2010)

### 2.8.4 Modos de operación

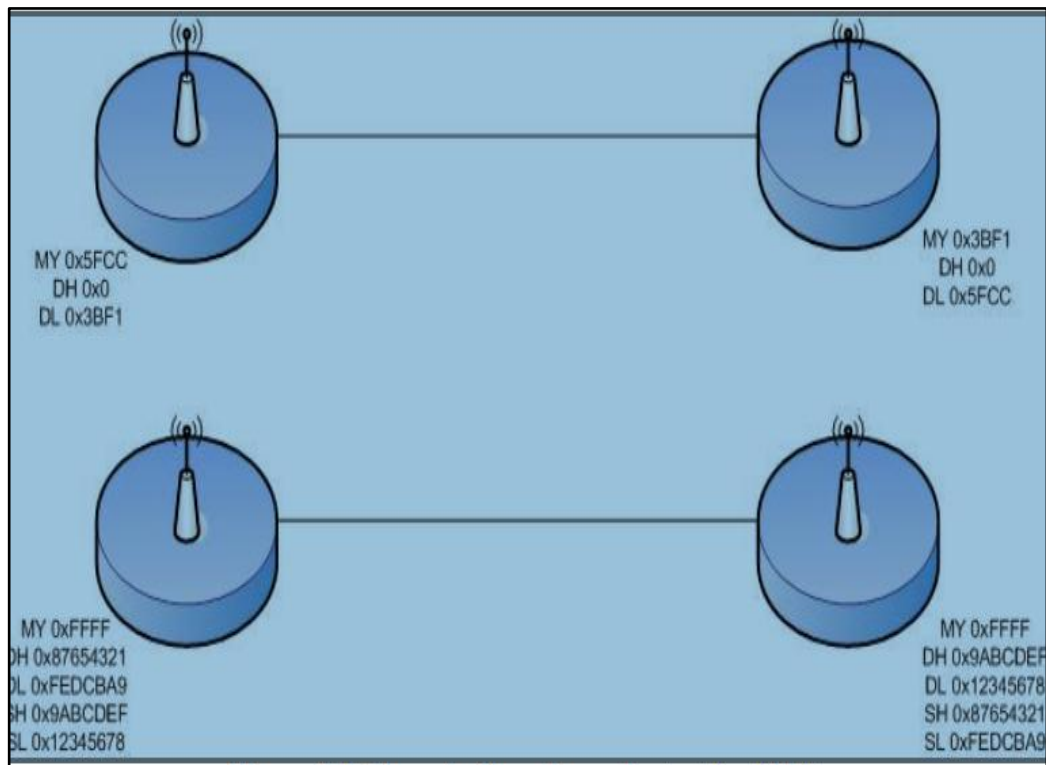
Los módulos XBee pueden operar en los siguientes modos:

#### 2.8.4.1 Modo Transparente

Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control también se usa para reemplazar alguna conexión Serial por un cable ya que esta configuración más sencilla posible y no requiere una mayor configuración. Básicamente todo lo que pasa por puerto

UART<sup>6</sup>( DIN,pin3), es enviado el módulo deseado, y lo que llega al módulo, es enviado de vuelta por el mismo puerto UART(DOUT, pin2). Es así que existen cuatro tipos de conexión transparente la diferencia principal radica en el número de nodos o puntos de acceso, y la forma en que estos interactúan si y estas conexiones son:

- **Punto a punto.-** Es la conexión ideal para reemplazar comunicación Serial Por un cable sólo se debe configurar la dirección. Para ello se utiliza los comandos MY y el DL. Para que el modo punto a punto funcione los módulos deben pertenecer a la misma PAN<sup>7</sup> ID<sup>8</sup> y al mismo canal. (MCI Electronics, 2010)



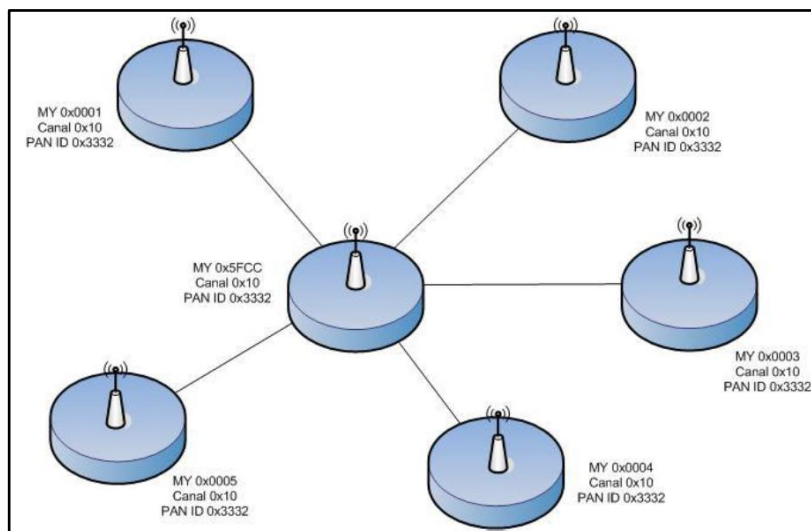
**Figura 16** Configuración Punto a Punto XBee  
 Autor: Xbee-Guía Usuario.pdf

<sup>6</sup> **UART** son las siglas de "Universal Asynchronous Receiver-Transmitter" (en español, Transmisor-Receptor Asíncrono Universal). Éste controla los puertos y dispositivos serie.

<sup>7</sup> PAN= Personal Área Network –Red de área personal Personal Area Network. Se entiende por PAN una red de comunicaciones que incluye un Coordinador de red y uno o más routers o dispositivos finales (end points).

<sup>8</sup> ID=Dirección pan del modulo

- Punto a Multipunto.-** Esta configuración permite transmitir información a uno o varios módulos conectados a la misma red de manera más controlada ya que se necesitan las direcciones de los otros módulos por lo que existe mayor seguridad. La elección del canal debe ser cuidadosa ya que otras tecnologías como wi-fi o Bluetooth utiliza el mismo espectro de frecuencias, por lo que se podría producir interferencia. En esta configuración es necesario pertenecer tanto al mismo canal como a la misma red es así que la información es más controlada además para enviar información se debe ingresar la dirección del módulo de destino por lo que es necesario el conocimiento completo de la red. En los módulos XBee pro el reconocimiento de la red se realiza automáticamente. (MCI Electronics, 2010)



**Figura 17** Configuración punto a Multipunto  
 Autor: Xbee-Guia Usuario.pdf

- Broadcast.-** Esta configuración permite el envío de información desde un nodo a varios nodos en una misma red, es decir la información recibida es la misma para todos los cualquier módulo que reciba un paquete con una dirección de destino por Broadcast ser aceptado.
- Cable virtual I/O.-** Esta configuración permite crear un canal de comunicación de manera transparente entre los pines de un módulo y otro, cada pin de entrada tiene su propio pin de salida ya definido entre nodos, lo



que permite una forma totalmente simple de envía información controlada por medio de manera sencilla y rápida, sin necesidad de complicadas configuraciones.

#### **2.8.4.2 Modo de operación API**

Este modo es más complejo, pero permite el uso de frames con cabeceras que aseguran la entrega de los datos, al estilo TCP, extiende el nivel en el cual aplicación del cliente puede interactuar con las capacidades de red del módulo cuando se está en este modo toda la información que entra y sale es empaquetada en frames, que definen operaciones eventos dentro del modo. **(MCI Electronics, 2010)**

#### **2.8.4.3 Modo recibir / transmitir**

La información transmitida puede ser directa o indirecta en modo directo la información se envía inmediatamente a la dirección de destino en el modo indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita. **(MCI Electronics, 2010)**

#### **2.8.4.4 Modo de bajo consumo**

El modo de bajo consumo hace que el módulo rf tenga bajo consumo cuando no se encuentra en uso, para poder entrar en modo de bajo consumo se debe cumplir las siguientes condiciones:

- Sleep\_ RQhay no transmicion ni rec (pin9) está en alto y el módulo está en pin sleep mode (SM=1,2 o 5)
- El módulo están reposo es decir no hay transmisión y recepción de datos, la cantidad de tiempo está definida por ST(time before Sleep) **(MCI Electronics, 2010)**

### 2.8.4.5 Modo de comando

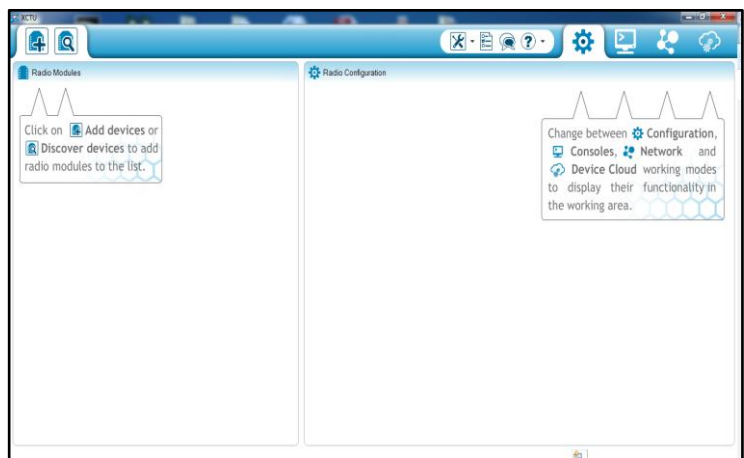
El modo comando permite ingresar comandos AT al módulo XBee, para configurar o ajustar o modificar parámetros, como son la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para esto es necesario utilizar el Hyperterminal de Windows, el X-CTU o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria.

### 2.8.4.6 Idle

Esta configuración es cuando el módulo no se está en ninguno de los otros modos, es decir si no está ni transmitiendo ni recibiendo, ni ahorrando energía ni el modo de comandos, entonces se dice que se encuentre un estado al que llama IDLE.

## 2.9 Software X-CTU

El software X-CTU es una aplicación basada en Windows que provee una interfaz gráfica para la configuración e interacción con módulos de radiofrecuencia del fabricante, en la figura se muestra la interfaz de entrada o ventana principal en donde se puede establecer los parámetros de configuración de los dispositivos de RF. (Digi, 2013)



**Figura 18** Interfaz Gráfica del Software XCTU  
**Fuente:** El Autor

## **2.10 Sistema Gestor de Bases de Datos**

Un sistema gestor de base de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente y se diseñan para gestionar grandes cantidades de información.

### **2.10.1 Aplicaciones de los sistemas de las bases de datos**

Las bases de datos son ampliamente usadas y forman una parte esencial de casi todas las empresas en la actualidad.

- Banca
- Universidades
- Telecomunicaciones
- Finanzas
- Ventas
- Producción
- Recursos humanos

### **2.10.2 Gestor de base de datos Access**

Microsoft Access es el SGBD (sistema gestor de base de datos) de Microsoft Office. Con este programa podemos almacenar datos de manera sistemática y acceder a ellos de forma rápida y estructurada. Access también puede considerarse un sistema de desarrollos rápido de aplicaciones RAD (Rapid Application Development) y por tanto su objetivo final será crear una aplicación que gestione los datos estructurados de manera sencilla para el usuario. ( **Microsoft. Tareas básicas en Access, 2014**)

Access es una herramienta de diseño e implementación de aplicaciones de base de datos que se puede usar para realizar un seguimiento de la información importante. Puede conservar los datos en el equipo o publicarlos en la Web, de forma que otras personas puedan usar la base de datos con un explorador web. ( **Microsoft. Tareas básicas en Access, 2014**)

## **2.11 Acceso a Bases de Datos desde Programas de Aplicación**

Para interrelacionar con una base de datos se necesita un programa de aplicación, estos se escriben usualmente en lenguaje anfitrión, tal como C, C++,C#, Visual , o Java, etc.

### **2.11.1 Microsoft Visual Basic 2008**

Es un lenguaje de programación orientada a objetos, para programar en visual Basic generalmente es utilizar un entorno de desarrollo integrado Microsoft visual studio, todos los programas basados en visual basic requieren un frameworks para ejecutarse.

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C#y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y hace más sencilla la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. ( **Microsoft. Visual Studio, 2007**)



*Figura 19: Visual 2008*

*Fuente:*[http://blogs.msdn.com/blogfiles/jasonlan/WindowsLiveWriter/VisualStudio2008Beta2now/available\\_2D47/image.png](http://blogs.msdn.com/blogfiles/jasonlan/WindowsLiveWriter/VisualStudio2008Beta2now/available_2D47/image.png)

## **2.12 Modos de identificación animal**

Los modos de identificación animal del ganado es esencial ya que refuerza y moderniza todo el manejo exitoso de su información siendo al principio el método más utilizado el del herrado (marca de hierro candente), o marcas específicas basados en el color de la piel o en los patrones de color, tatuajes. Existen dos razones fundamentales para la identificación animal que es la prueba de propiedad y el manejo/rastreabilidad.

### **2.12.1 Manejo y Rastreabilidad.**

Esta se ha convertido en una necesidad que ha ido evolucionando, tanto así que en muchos casos el confirmar la propiedad no es el tema central al contrario los animales se registran para ver su progreso en términos de ganancia de peso, fertilidad, que tan propenso son a enfermedades, por lo tanto facilitar la selección y el manejo; además identificar animales se hace necesario para un proceso de diagnóstico de enfermedades de manera que estos sean desechados cuando se detectan un problema a lo largo de la cadena de producción.

### **2.12.2 Aretes visuales**

Este método se ha usado por décadas y muchos productores han usado aretes escritos a mano como herramienta de manejo, esto se ha cuestionado ya que eran propenso a borrarse caerse o simplemente deteriorarse haciéndose ilegibles.

En la producción de aretes se ha evolucionado tanto que existen aretes a prueba de adulteración que son dobles y son impresos con tecnología láser, tienen una alta tasa de retención y permanecen legibles por muchos años, son fáciles de colocar por medio de un aplicador además pueden imprimirse con códigos alfanuméricos para identificar de manera única a cada animal siendo legibles a una distancia de 2 metros.



*Figura 20 Aretes para ganado*  
*Fuente: <http://www.agromundo.com.mx/aretos.html>*

### **2.12.3 Aretes de códigos de barras**

Para eliminar el error humano en la lectura aparecen los códigos de barras en los aretes dándole un avance adicional y dando una prestación un tanto más automática en el proceso de lectura de este código ya que se lo realiza por medio de un lector de código de barras y utilizarlo en sistemas de registro. La mayor desventaja radica en la presencia de suciedad en el código de barras del arete. Otra desventaja un sistema de cómputo ligado a un lector.



*Figura 21 Aretes para ganado con código de barras*  
*Fuente: <http://www.agromundo.com.mx/aretos.html>*

### **2.12.4 Aretes RFID**

El desarrollo más reciente dentro de la identificación con aretes es el uso de dispositivos de identificación por radiofrecuencia RFID, avanzaron más que con los códigos de barras; haciéndolos más robustos y no dependen de una lectura visual si no que al contrario dependen de radiofrecuencia, un lector se encarga de

energizar al chip para que este emita un código único que tendrá el ganado haciéndolos mucho más seguros y fáciles de manejar.



**Figura 22** Arete RFID

**Fuente:** [http://www.rfidecuador.ec/es/components/com\\_jshopping/files/img\\_products/](http://www.rfidecuador.ec/es/components/com_jshopping/files/img_products/)

## CAPTULO III

### 3. METODOLOGÍA

#### 3.1. Diseño de Investigación

El tipo de estudio empleado es de diseño experimental, se trata de determinar los correctivos de la parte práctica de cada proceso, para un funcionamiento eficiente con la implementación de la tecnología RFID y la transmisión y recepción por RF.

#### 3.2. Tipos de investigación

Los tipos de investigación aplicados de acuerdo al propósito en el presente trabajo o las finalidades perseguidas son: exploratoria, descriptiva y explicativa.

- **Exploratorio.-** Cuerpo de la investigación con el fenómeno para conocerlo o detallarlo tipo pilotaje
- **Descriptivo.-** Caracterizar cada variable, manifestaciones o comportamiento de las variables
- **Asociación de variables.-** Medir el impacto de la variable independiente sobre la variable dependiente.

#### 3.3. Método de Investigación

##### 3.3.1 Método inductivo.-

Se aplicó el método inductivo ya que la investigación requirió el análisis de los registros o la información de cada animal del hato ganadero y la forma como se la trataba. Y se utilizó los siguientes pasos que implica este método:

- Determinación de problema
- Planteamiento de hipótesis
- Inferencia inductiva procesos de investigación realizados



- Comprobación de la hipótesis
- Generalización de los resultados

### **3.4.Población y Muestra**

#### **3.4.1 Población**

Para establecer el conjunto de animales vacunos para este estudio se estableció que sería en el hato del ganadero conformado por 40 animales.

#### **3.4.2 Muestra**

En vista de que se trata de una población pequeña no se calcula una muestra por lo tanto se trabajará con el universo en su totalidad.

### **3.5. Procedimientos**

Ya que la información del hato ganadero se lo llevaba en apuntes u hojas de registro el acceso a dicha información de cada animal a consultar es tediosa y tomaba mucho tiempo en ubicarla, muchas de las veces ya no existía.

#### **3.5.1 Técnica de Investigación**

Las técnicas e instrumentos que se utilizan para la recolección de la información y datos; fundamentalmente son las siguientes:

- **Entrevistas:** Dirigida a al dueño del Hato ganadero y al encargado del cuidado de los animales.
- **Encuestas:** Para conocer las consideraciones del dueño del Hato ganadero sobre la organización y funcionalidad del sistema.
- **Observación:** Para verificar la realización de la actividad conjunta referida a la transmisión y recepción de datos vía RF.

### 3.6.Operacionalización de variables

#### 3.6.1. Variable Independiente

Variable	Definición	Categorías	Indicador	Técnicas
<b>SISTEMA DE CONTROL Y REGISTRO COMPUTACIONAL DE ANIMALES VACUNOS</b>	Herramienta capaz de registrar almacenar controlar y transmitir información general de animales vacunos	Sistema Prototipo	<ul style="list-style-type: none"> <li>➤ Alcance</li> <li>➤ Costos</li> <li>➤ Eficiencia</li> <li>➤ Integridad de los datos</li> <li>➤ Portabilidad</li> </ul>	Observación
		Registro y control de ganado vacuno	<ul style="list-style-type: none"> <li>➤ Operatividad</li> <li>➤ Tiempo, acceso a la información,</li> <li>➤ Integridad de datos</li> <li>➤ Usabilidad</li> </ul>	Pruebas de campo.

**Tabla 4** Operacionalización de la variable Independiente

*Fuente: El Autor*

### 3.6.2. Variable Dependiente

Variable	Definición	Categorías	Indicador	Técnicas
TOMA OPORTUNA DE DECISIONES A NIVEL DE PRODUCCIÓN	La toma oportuna permite generar competitividad, inyecta valor agregado y confinaza en un hato ganadero de producción.	Tiempo	➤ Respuesta optima en el acceso a la información	Pruebas de campo.
		Resultados	<ul style="list-style-type: none"> <li>➤ Registro oportuno</li> <li>➤ Seguimiento</li> <li>➤ Capacidad de obtener información en tiempo real.</li> </ul>	Observación directa monitoreo Registro y evaluación

*Tabla 5 Operacionalización de la variable Dependiente*

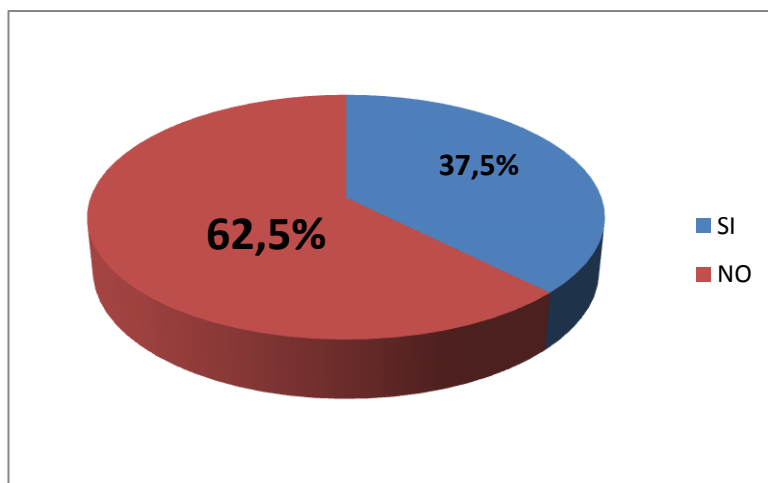
*Fuente: El Autor*

### 3.7. Procesamiento y Análisis

**Pregunta N° 1.- ¿Del Hato ganadero, de cuántos animales existe la información completa; como las características propias, controles veterinarios, etc.?**

Variable	Frecuencia	Porcentaje %
SI	15	37,5%
NO	25	62.5%
<b>TOTAL</b>	<b>40</b>	<b>100%</b>

*Tabla 6 Acceso a Información completa con el registro manual  
Fuente: Encuesta Sistema COREGVAC*



*Figura 23 Acceso a Información completa con el registro manual  
Fuente: El Autor Tabla6*

#### **Análisis:**

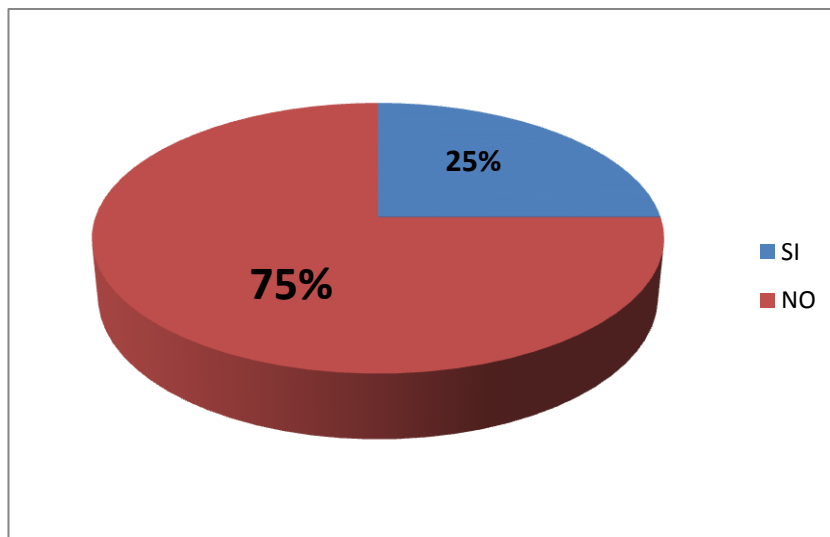
El dueño del hato indica que de los 40 animales que equivalen al 100% , solo tiene la certeza de saber los datos generales y que existe una información completa desde el momento que ingresaron a su hato ganadero solo 15 animales que corresponden al 37.5% .

**Pregunta N° 2.- ¿Del Total de animales del Hato de cuántos animales tiene acceso a la información en menor tiempo y con resultados que visualicen todas y cada uno de sus eventos?**

Variable	Frecuencia	Porcentaje %
SI	10	25%
NO	30	75%
<b>TOTAL</b>	<b>40</b>	<b>100%</b>

*Tabla 7 Acceso a la Información del registro actual*

*Fuente: Encuesta Sistema COREGVAC*



*Figura 24 Acceso a Información de un registro animal*

*Fuente: El Autor Tabla 7*

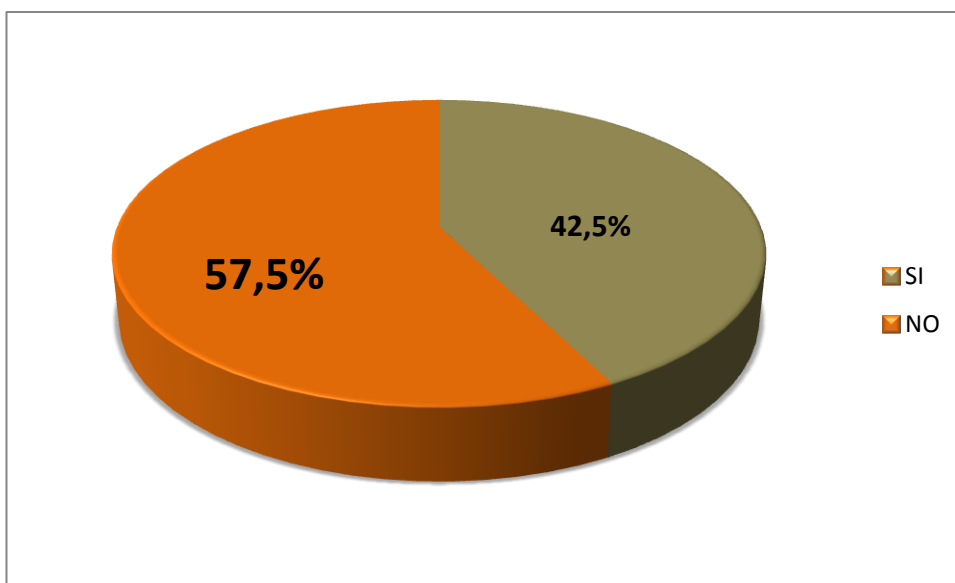
### **Análisis:**

El dueño del hato indica que del 100% de los animales, solo de 10 tiene la certeza de recordar, y acceder a la información y visualizarla ya que la mayoría dicha información a veces se les pasa por alto como por ejemplo la última fecha de vacunación y que tipo de vacuna tuvo el animal además en muchos casos tiene que recurrir apuntes u hojas de registro en diferentes lugares.

**Pregunta N° 3.- ¿Del Total de animales del Hato de cuántos animales tiene o realiza un registro oportuno?**

Variable	Frecuencia	Porcentaje %
SI	17	42.5%
NO	23	57.5%
<b>TOTAL</b>	<b>40</b>	<b>100%</b>

*Tabla 8 Porcentaje de Registro oportuno de la información manual  
Fuente: Encuesta Sistema COREGVAC*



*Figura 25 Registro oportuno de la información de un manual  
Fuente: El Autor Tabla 8*

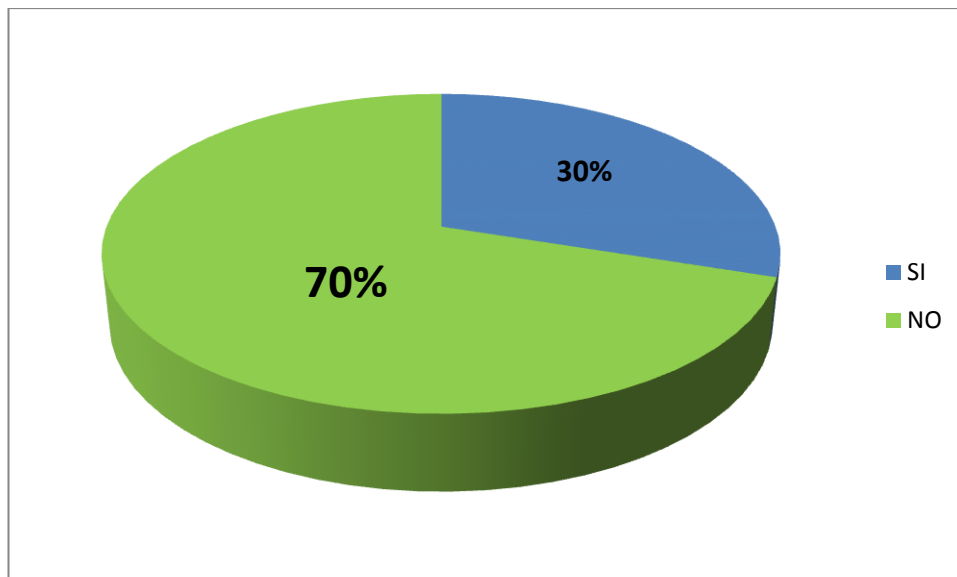
**Análisis:**

El dueño del hato indica que de los 40 animales, solo de 17 hace el registro oportuno por el tiempo que esto conlleva es decir un evento lo apunta y luego lo lleva a otro registro para actualizarlo por ejemplo el último parto y que el sexo de la cría, ya que esto lleva de una tratamiento al animal; solo es de forma visual esto hasta que se los aretea y se los trata como nuevos animales en el hato.

**Pregunta N° 4.- ¿Del Total de animales del Hato de cuántos animales tiene un registro de datos íntegro y completo que puede garantizar la información de este para hacer un seguimiento de la res?**

Variable	Frecuencia	Porcentaje %
SI	12	30%
NO	28	70%
<b>TOTAL</b>	<b>40</b>	<b>100%</b>

*Tabla 9 Porcentaje de Garantía de información con registro manual  
Fuente: Encuesta Sistema COREGVAC*



*Figura 26 Garantía de información con registro manual  
Fuente: El Autor Tabla 9*

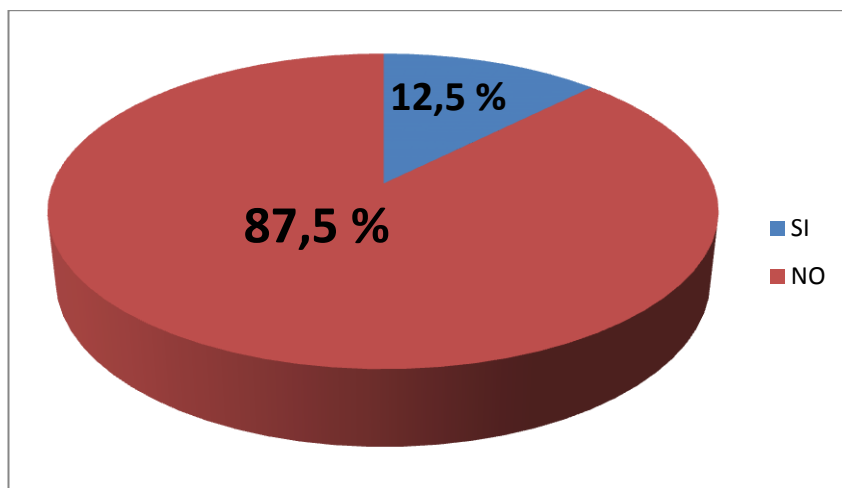
### **Análisis**

El dueño del hato indica que del total de animales existe un registro de datos íntegro y completo de 12 de los 40 animales, que puede garantizar la información de este.

**Pregunta N° 5.- ¿Del Total de animales del Hato relacionando con toda la información existente de cuántos animales cree usted que la información es completa y confiable para en caso de tomar una decisión tenga la certeza de hacerlo?**

Variable	Frecuencia	Porcentaje %
SI	5	12,5%
NO	35	87,5%
<b>TOTAL</b>	<b>40</b>	<b>100%</b>

*Tabla 10 Acceso a la Información del registro manual  
Fuente: Encuesta Sistema COREGVAC*



*Figura 27 Acceso a la Información del registro manual  
Fuente: El Autor*

### **Análisis**

El dueño del hato indica que de los animales de hato 35 que equivale al 87,5% no tiene la información completa y confiable para en caso de tomar una decisión tenga la certeza de hacerlo solo tiene de 5 que equivale al 12,5% .



### 3.7.1. Comprobación de la Hipótesis.-

Acceso a la Información de datos de los animales en el Hato de producción.- Para acceder a la información o a los registros de forma confiable segura y con un acceso inmediato a la información dentro del hato de 40 animales que tiene el productor se tiene la siguiente tabla de contingencia observada.

Acceso	Indicador					
	Registro Oportuno	Respuesta Optima en el Acceso a la Información	Seguimiento	Capacidad de obtener información en tiempo real	Confiabilidad en la información	Total
Manual	15	10	17	12	5	59
por el Sistema COREGVAC	40	40	40	40	40	200
<b>TOTAL</b>	55	50	57	52	45	259

*Tabla 11 Frecuencia Observada*

*Fuente: El Autor*

Acceso	Indicador					
	Registro oportuno	Respuesta Optima en el Acceso a la Información	Seguimiento	Capacidad de obtener información en tiempo real	Confiabilidad en la información	Total
Manual	12,53	11,39	12,98	11,85	10,25	59
Sistema COREGVAC	42,47	38,61	44,02	40,15	34,75	200
<b>TOTAL</b>	55	50	57	52	45	259

*Tabla 12 Frecuencias esperadas*

*Fuente: El Autor*

### 2.12.4.1 Establecimiento de la Hipótesis.

**H0=** La implementación del sistema Prototipo de control y registro computacional de ganado vacuno incide en la toma oportuna de decisiones a nivel de producción.

**H1=** La implementación del sistema Prototipo de control y registro computacional de ganado vacuno no incide en la toma oportuna de decisiones a nivel de producción.

### 2.12.4.2 Establecer el nivel de Significancia

Nivel de significancia  $\alpha = 0,05$  con 4Gl. ( $X^2_{\text{tabla}} = 9.49$ ) **Gl=** grados de libertad

**Dónde:**  $Gl=(p-1)(q-1)$  **p=** número de filas **q=** número de columnas

$Gl=(2-1)(5-1) = 1*4= 4$

### 2.12.4.3 Elección de la prueba estadística

Para el análisis estadístico se ha escogido la prueba de Chi cuadrado.

$$X^2 = \sum \frac{(fo - fe)^2}{fe}$$

#### Resultados obtenidos

<b>fo</b>	<b>fe</b>	<b><math>X^2 = \frac{(fo-fe)^2}{fe}</math></b>
15	12,53	0,4874
40	42,47	0,1438
10	11,39	0,1696
40	38,61	0,0500
17	12,98	1,2417
40	44,02	0,3663
12	11,85	0,0020
40	40,15	0,0005
5	10,25	2,6897
40	34,75	0,7934
	<b>X<sup>2</sup></b>	<b>5,9447</b>

**Dónde:**  
**X<sup>2</sup>** = Chi cuadrado  
**fo**=Frecuencia Observada  
**fe**=Frecuencia Esperada

**Tabla 13** Prueba del chi cuadrado

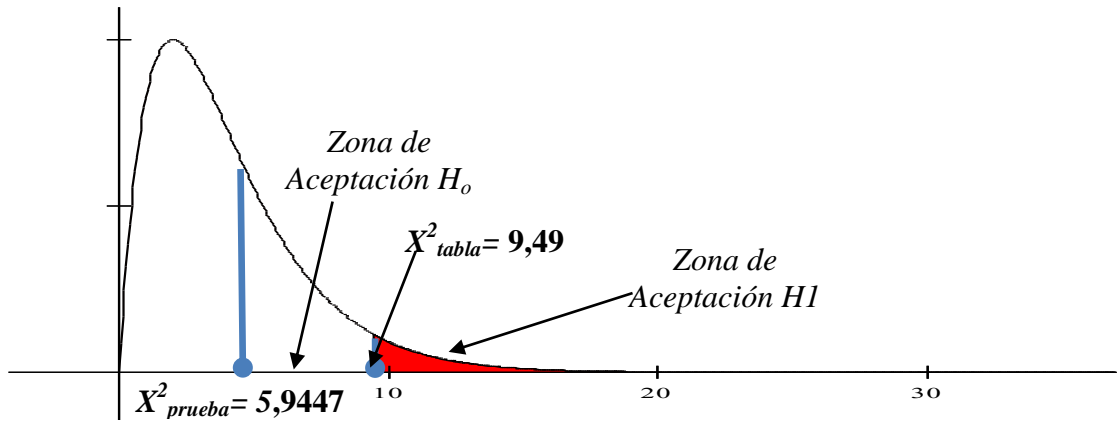
**Fuente:** El Autor

### 2.12.5 Decisión de la hipótesis

**Regla de decisión:**

Si  $X^2_{prueba} \leq X^2_{tabla} = 9,49$  no se rechaza  $H_0$

Si  $X^2_{prueba} > X^2_{tabla} = 9,49$  se rechaza  $H_0$



**Figura 28** Prueba del Chi - cuadrado  
Fuente: El Autor

Entonces:  $X^2_{prueba} = 5,9447 < X^2_{tabla} = 9,49$ . El  $X^2$  calculado 5,9447 es menor que  $X^2$  de la tabla 9,49 por lo tanto la hipótesis ( $H_0$ ) del investigador La implementación del sistema Prototipo de control y registro computacional de ganado vacuno incide en la toma oportuna de decisiones a nivel de producción. Es aceptada.

### 2.12.6 Comparativa del Acceso a la información y operatividad entre el sistema Manual y el sistema COREGVAC.

Acceso	Indicador				
	Registro oportuno	Respuesta Óptima en el Acceso a la Información	Seguimiento	Capacidad de obtener información en tiempo real	Confiabledad en la información
Manual	37,5%	25%	42,5%	30%	12,5%
por el Sistema COREGVAC	100%	100%	100%	100%	100%

**Tabla 14** Acceso a información Manual Vs Sistema  
Fuente: El Autor

## CAPITULO IV

### 4. RESULTADOS Y DISCUSIÓN

#### 4.1. Resultados

El sistema de Control y Registro de Ganado Vacuno es una herramienta completa por los siguientes aspectos:

- Fácil de Instalar y Configurar.
- La tecnología de transmisión es modular es decir se lo puede adaptar para aumentar el alcance de transmisión tan solo con cambiar los transceptores.
- Puede ser integrada a otros hatos de Producción
- Es una herramienta autónoma

El sistema COREGVAC tiene como principal característica la aplicación de la Norma ISO 9216, y además cumple con las características de Requisitos del sistema, Aplicaciones disponibles, Soporte, Interoperabilidad, Comercio, Administración.

La norma 9216 es utilizada a nivel mundial para el mejoramiento de Software gracias a que cumple con los siguientes parámetros:

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

El Sistema de Control y Registro de Ganado Vacuno dispone de la flexibilidad en la creación de contenidos para realizar cambios, el cual permite realizar una

aplicación rápida que se adapta a los requerimientos del dueño del hato ganadero, para gestionar, ingreso de más reses, seguimiento y controles de sanidad, apertura de nuevos registros de animales, consultas, etc. Además permite tener un acceso inmediato y dinámico a toda la información de los animales del hato registrados en la aplicación mediante un código único de identificación.

En relación al hardware es un prototipo modular es decir capaz de con solo variar módulos como los de transmisión se puede obtener mayor alcance de transmisión siempre con una línea de vista entre dispositivos TX-RX, además es portable y depende de sí mismo es decir no necesita otros canales o tecnologías de transmisión (internet . wirrelles, bluetooth, etc) más que solo la de RF (Radio frecuencia) sin interferencia con otros dispositivos y con seguridad en su transmisión. por esto tiene las siguientes características

- Conectividad, Portabilidad
- Independencia de wirelles o internet
- Autonomía, Modularidad

#### **4.2.Discusión**

En otras investigaciones realizadas con este tipo de tecnologías e inducidas al control y registro de animales vemos que existen varios prototipos en los cuales utilizan los medios comunes de transmisión como internet wirelles, dependiendo de estos además utilizan lectores industriales de mayor costo como el de bastón, para las lecturas tipo caravana, con este prototipo primero el lector es integrado con tecnología abierta y de bajo costo; para realizar la transición y recepción TX del RX se utiliza radiofrecuencia esto los hace independientes a otros tipos de transmisión sin necesidad de otros medios como módems o routers es decir es una herramienta autónoma. El prototipo funciona porque cumple con las necesidades del dueño del hato en el control y registro de los animales, además se tiene confiabilidad y seguridad de que su información está presente y almacenada de forma ordenada.

## CAPITULO V

### 5. CONCLUSIONES Y RECOMENDACIONES

#### 5.1.Conclusiones

- Al investigar el funcionamiento de la tecnología RFID, se concluye que es una herramienta tecnológica que tiene excelentes prestaciones para la recolección de datos, lo cual permite elaborar prototipos o herramientas mismas que ayuden automatizar el delicado trabajo que es el del tratamiento de información de forma autónoma y eficiente.
- Al existir diversos tipos de lectores y etiquetas a bajo costo se puede integrar dispositivos que ayuden a realizar trabajos de trazabilidad en cadenas de producción a bajo costo y con resultados eficientes.
- La creación del prototipo ha logrado cumplir con los objetivos propuestos en el plan de titulación, es decir se diseñó e implemento un sistema de control y registro computacional de ganado vacuno utilizando como base la tecnología RFID además dispositivos electrónicos de bajo costo y eficientes en su desempeño.
- Aplicando esta tecnología en hatos ganaderos se obtiene una herramienta de capacidades enormes esta herramienta integrada con su hardware y su software ofrece ayuda muy importante en el control y registro de ganado vacuno en un hato específico de producción, ofreciendo valor agregado y seguridad sanitaria al utilizarlo además se puede implementar en otros hatos de producción.

## 5.2.Recomendaciones

- Es recomendable utilizar el sistema de identificación animal por medio de aretes RFID, ya que garantiza una automatización en la recolección de datos con una mínima inversión y con resultados óptimos generando integridad en la información a diferencia de otros métodos como los de marcas o lecturas manuales de identificación animal que conlleva a inconsistencia en los registros
- Al ser una tecnología que tiene muchas prestaciones en la identificación y trabaja en base a radiofrecuencia es recomendable utilizar elementos de bajo ruido y para futuras investigaciones buscar los mejores dispositivos para integrarlos y así tener un sistema libre de errores en lecturas y recolección de datos.
- Es recomendable siempre interactuar con las personas involucradas y hacerle saber este tipo de beneficios tecnológicos para ofrecerles la solución a su problema de identificación, estableciendo alcances y costos.

## CAPITULO VI

### 6. PROPUESTA

#### 6.1. Título de la Propuesta

Diseño de un sistema computacional para el control y registro de animales vacunos en base a la tecnología RFID, y transmisión remota mediante radio frecuencia.

#### 6.2. Introducción

El control y registro de animales vacunos se ha convertido en una herramienta importante para monitorear, erradicar enfermedades evitar fraudes y sobre todo inyectar un valor agregado a los animales ya que los productores utilizan esta herramienta como un elemento de trazabilidad y los consumidores tienen una seguridad alimenticia.

Para el diseño y la implementación del prototipo de control y registro de ganado vacuno se realizaron dos fases la de construcción de Hardware y la de construcción del software. Para la construcción del Hardware se utilizó dispositivos Arduino, transceptores XBee S1, microcontrolador ATmega164P, shields de regulación de tensión de alimentación, shields para inteface de RS232 a usb para establecer una comunicación entre el hardware y el computador.

Para que este hardware funcione se desarrollaron aplicaciones de software utilizando diferentes lenguajes de programación; para el microcontrolador se utilizó Bascom, pero antes de esto se hizo simulaciones en proteus, para los demás dispositivos arduino C++, para configurar los transeptores XBee s1 se utilizó el software que el fabricante recomienda como es el XCTU; además una parte esencial de este sistema es la creación de la base de datos con un motor en Microsoft Access.



Utilizando un lenguaje de aplicación que en este caso es Visual Basic 2008 para realizar todas las gestiones en la misma y mostrar las interfaces amigables con el usuario. Para el desarrollo de la placa del circuito impreso se utilizó el software Eagle.

### **6.3.Objetivos**

#### **6.3.1. Objetivo General**

Desarrollar un prototipo para el control y registro de ganado vacuno, utilizando los conocimientos aprendidos de microcontroladores PIC, comunicación serial inalámbrico, tecnologías de tarjetas RFID, el uso y gestión de bases de datos.

#### **6.3.2. Objetivos Específicos**

- Integrar la tecnología RFID con dispositivos de transmisión y recepción de información
- Implementar un circuito de transmisión y recepción de código de un tag RFID
- Crear una base de datos con su respectivo software de gestión en el cual se almacenara toda la información general de un animal vacuno a tratar.

### **6.4.Requisitos de diseño e implementación del prototipo y selección de herramientas**

Para implementar el prototipo fue necesario establecer cuáles serían los requisitos que debería cumplir tanto para el hardware como para el software, y estos son:

- Un elemento principal de lectura de RFID para leer los tags RFID
- Un módulo de transmisión de datos el que envía el código único del TagRFID
- Un módulo de recepción de datos que recibe el código único del TagRFID
- Un módulo de comunicación serial para conectar el prototipo a un computador personal

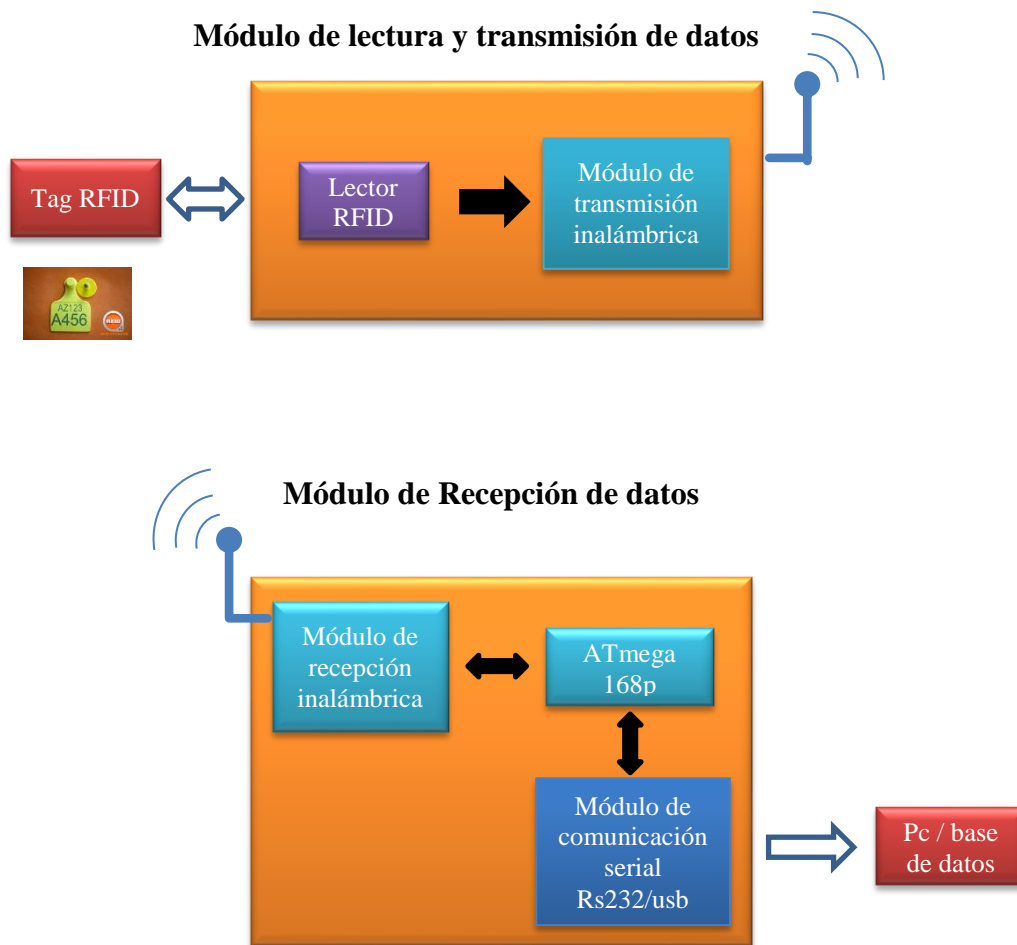
- Un algoritmo que permita al microcontrolador estandarizar el código de recepción del Tag RFID
- Un algoritmo que permita al dispositivo de Arduino obtener el código mediante el lector RFID
- Una base de datos en la cual se almacena la toda la información de cada animal con su código único de identificación y las características propias.
- Un algoritmo para establecer una comunicación serial entre el receptor el pc y la base de datos.
- Un algoritmo para realizar todas las gestiones con la base de datos además interactuar con el hardware del prototipo.
- Un Software de aplicación para interrelacionar con la base de datos y que sea amigable con el usuario.

## **6.5.Diseño e implementación del hardware del prototipo**

En esta sección se explica las pruebas de funcionamiento de los dispositivos utilizados en el proyecto, como son el lector RFIDudino, los shield de interface de serial a mini usb, los XbeeS1, así como también el desarrollo del proyecto a nivel de hardware. Las pruebas realizadas la construcción misma del hardware partiendo de un diagrama propuesta de bloques en donde se puede evidenciar los diferentes módulos que conforman este elemento electrónico de mucha importancia para el funcionamiento y la integración misma de todo el sistema.

### **6.5.1. Diagrama de bloques del hardware del prototipo**

De acuerdo a los requisitos de la sección anterior se determinó que hardware del prototipo debía tener los módulos tanto de transmisión como de recepción de forma inalámbrica y cada uno de estos con módulos internos que realizan operaciones específicas como el de la lectura del código el de estandarización del código obtenido, la interface de rs232 al computador, etc. elementos que se muestra a continuación en la figura.

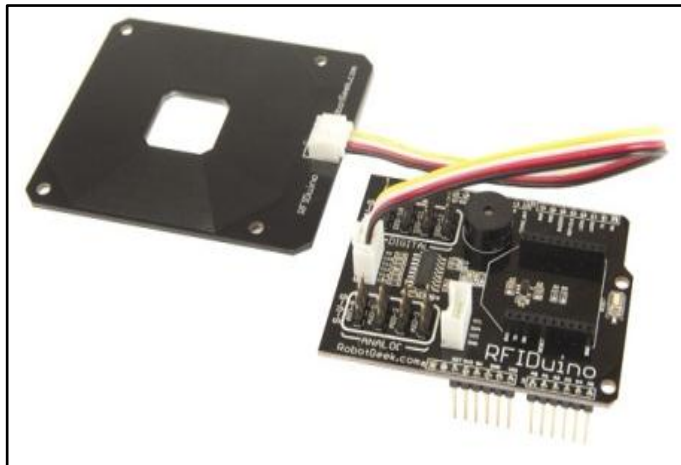


*Figura 29 Diagrama de Módulos del Sistema*  
*Fuente: El. Autor*

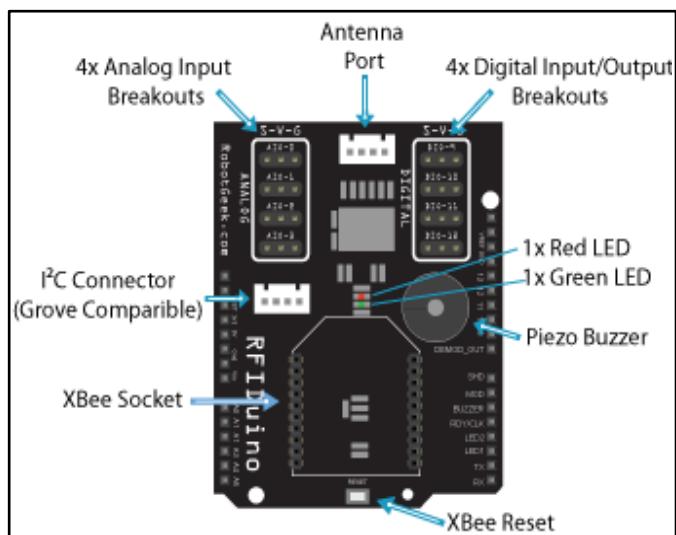
## 6.6. Implementación del Hardware de Lectura y Transmisión

### 6.6.1. Módulo de Lectura con RFIDudino y Arduino

La interconexión entre los elemento de lectura, el RFIDudino y Arduino al elemento de transmisión XBee es fundamental para crear un módulo de transmisión el cual se encarga de obtener el ID (Código de Identificación) de las etiquetas RFID y luego transmitirlo de forma inalámbrica hacia el receptor todo el esquema de conexiones y el módulo RFIDudino, claro con su respectiva programación esto se muestra en la siguiente figura.



**Figura 30** Tarjeta de desarrollo RFIDuino  
**Fuente:** <http://www.trossenrobotics.com/rfiduino>



**Figura 31** Módulo RFID uduino  
**Fuente:** [www.trossenrobotics.com/shared/images/productGuides/rfiduino/RFIDuino\\_connectors.png](http://www.trossenrobotics.com/shared/images/productGuides/rfiduino/RFIDuino_connectors.png)

### 6.6.2. Programación y Pruebas del Arduino 1 y RFIDuino para conformar el modulo Lector

Para comprobar el funcionamiento de este dispositivo con el lector Rfiduino , cargamos el programa y realizamos las pruebas con el mismo software de programación del dispositivo Arduino1 para esto se siguen los siguientes pasos:

- Compilamos el código de programación de nuestro programa

```
#include <RFIDuino.h> // librería ausar del RFIDuino
//Configuración de los pines de las salidas
#define R_LED 3
```

```

#define G_LED 2
#define BUZZER 5

byte tagData[5];
byte tagDataBuffer[5];
int readCount = 0;
boolean verifyRead = false;
boolean tagCheck = false;
RFIDuino myrfid;
void setup()
{
  Serial.begin(9600);
  pinMode(R_LED,OUTPUT);
  pinMode(G_LED,OUTPUT);
  pinMode(BUZZER,OUTPUT);
  digitalWrite(R_LED, LOW);
  digitalWrite(G_LED, LOW);
  digitalWrite(BUZZER, LOW);
}

void loop()
{
  tagCheck = myrfid.decodeTag(tagData);
  if (tagCheck==true)
  {
    readCount++;
    if(readCount==1)
    {
      myrfid.transferToBuffer(tagData, tagDataBuffer);
    }

    else if(readCount==2)
    {
      verifyRead = myrfid.compareTagData(tagData, tagDataBuffer);
      if (verifyRead == true)
      {
        for(int n=0;n<5;n++)
        {
          Serial.print(tagData[n],DEC);
          {
          }
        }
        Serial.print("\n\r");
        digitalWrite(G_LED,HIGH);
        digitalWrite(BUZZER, HIGH);
        delay(1000);
        digitalWrite(BUZZER, LOW);
        digitalWrite(G_LED,LOW);
      }
    }
  }
}

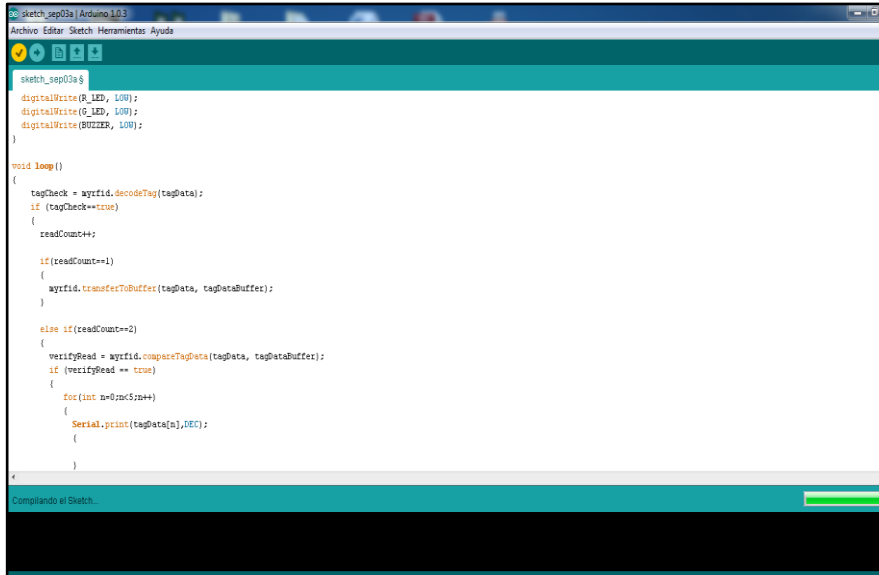
```

```

    }
    readCount=0;
  }
}
}
}

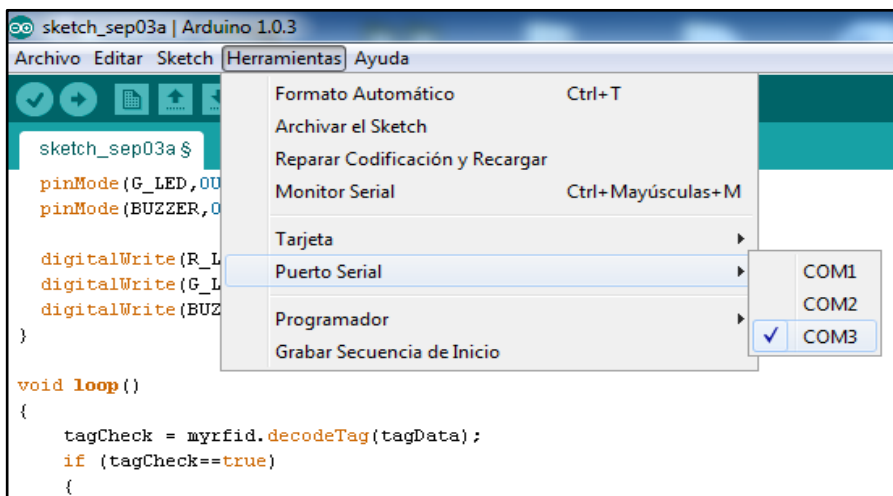
```

- Cargamos y compilamos el código de programación de nuestro programa de la tarjeta arduino



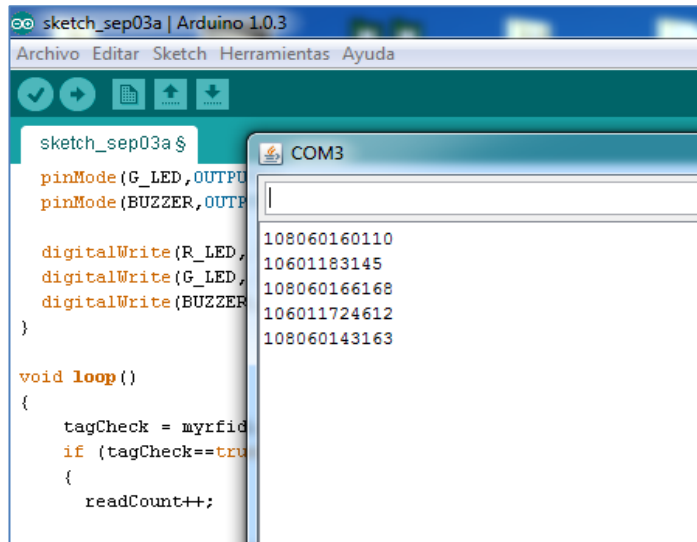
**Figura 32:** Ventana de Programación para Arduino  
**Fuente:** El Autor

- Escogemos el puerto de comunicación COM en este caso con el puerto a usar es el COM3.



**Figura 33** Determinación del Puerto serial para pruebas Arduino  
**Fuente:** El Autor

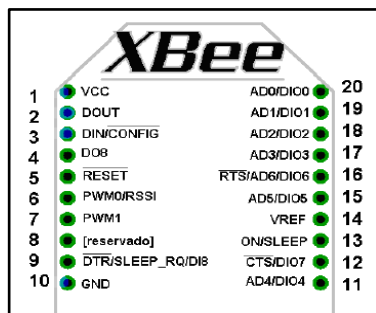
- Hacemos la comprobación de lectura del ID del TAG RFID al pasar por la antena los tags, el lector muestra el resultado que es el código único que este tiene y se visualiza en pantalla. Si observamos aquí el código obtenido no está estandarizado



*Figura 34 Comprobación del Id de los Tags en Arduino*  
*Fuente: El Autor*

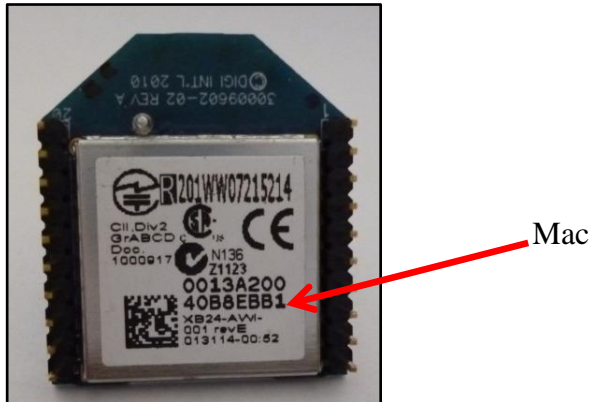
### 6.6.3. Configuración XBee para Transmisión y Recepción

Los dispositivos XBee se comprobaron y se configuraron tanto para transmitir como para recibir el código único de cada Tag y está configurado en modo transparente o modo de puente serial (como que fuera un cable conectado) para esto se utiliza el software que el fabricante recomienda el cual es el XCTU, y los pasos siguientes son:



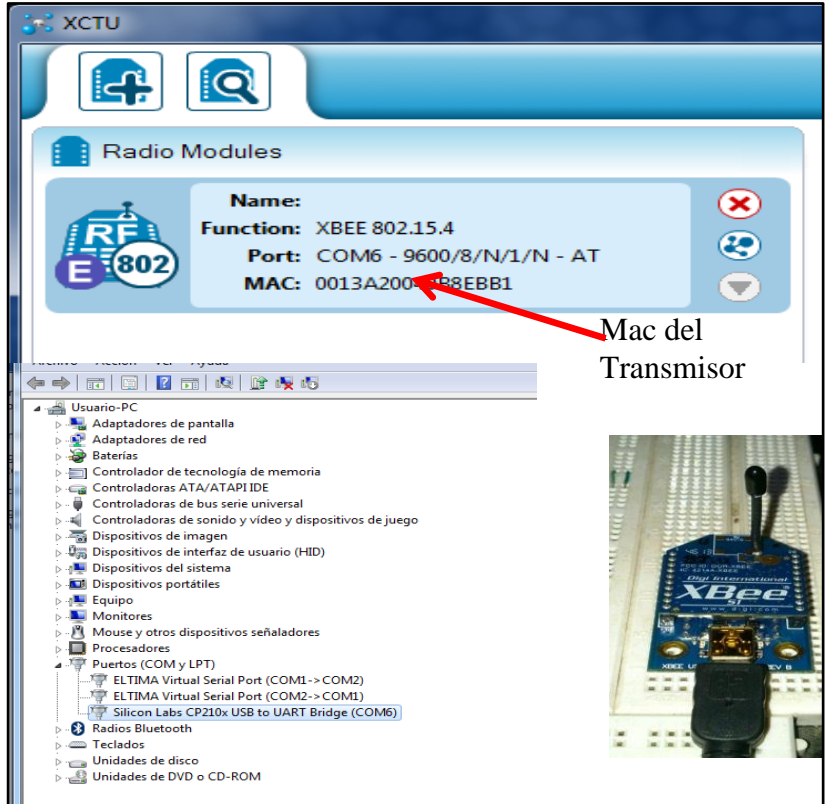
*Figura 35 Diagrama de pines vista superior*  
*Fuente: Xbee-Guía Usuario.pdf*

- **XBee Transmisor.-** Ubicamos la dirección mac del dispositivo esta se encuentra en la parte posterior del XBee es una dirección única.



**Figura** Dirección Única Mac del Xbee Transmisor  
**Fuente:** El Autor

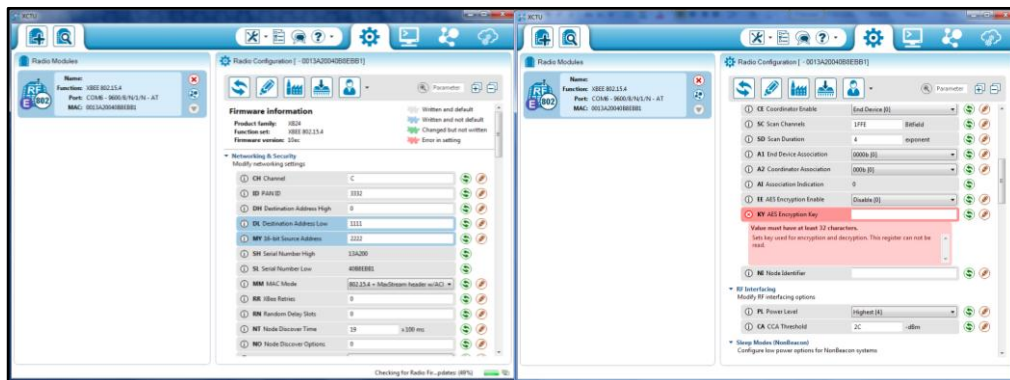
- Para conectar el dispositivo con el pc se utiliza el puerto COM6, el Xbee es acoplado a un Shield explorer de interface Mini USB a UART ademas polariza y regula.



**Figura 36** Conexión de un Xbee con su puerto com  
**Fuente:** El Autor

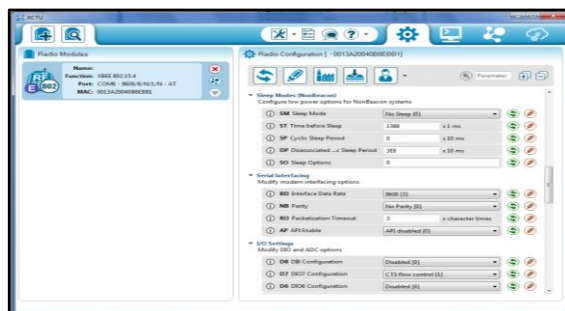


- Aquí nos fijamos en la opción de En destination address Low y destination address
- Configuramos el Destination Address Low DL con 1111 y MY ( 16-bits Source Address) con 2222. Los parametros de configuracion se vizualizan en la siguinete figura.
- El PANID es 3332 este vine por defecto y los dos transeptores deben tener el mismmo.



**Figura 37** Configuración dirección del Xbee  
Fuente: El Autor

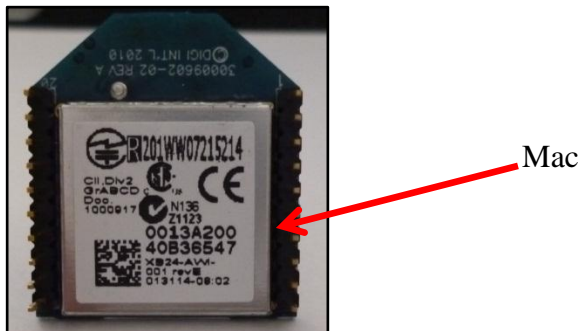
- Las opciones están por defecto pero debemos configurar los siguientes parámetros :
  - BR.- (Baud Rate) = 9600 bps
  - NB.- (parity)= No Parity
- Los demás parámetros están por defecto.



**Figura 38** Configuración serial Transmisor  
Fuente: El Autor

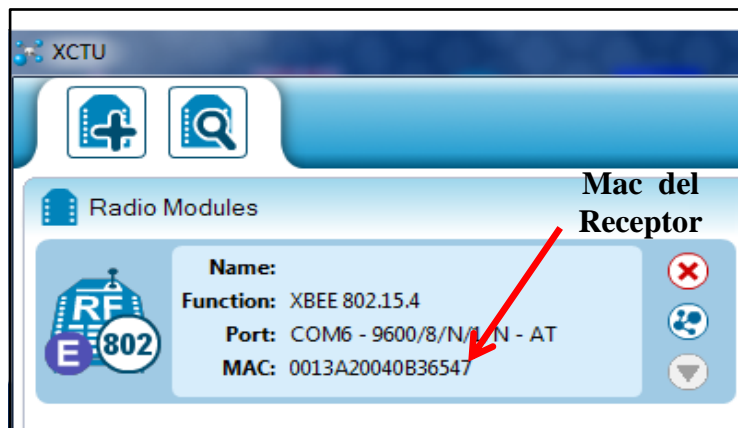
- **XBee Receptor**

Al igual que el transmisor; en el receptor tambien ubicamos la dirección mac del dispositivo esta se encuentra en la parte posteri del XBee es una dirección única.



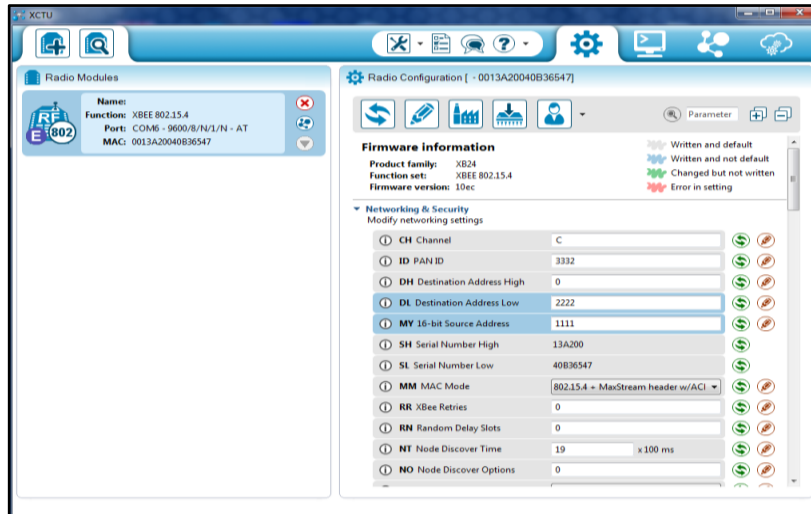
*Figura 39* Dirección Unica Mac del Xbee Receptor  
*Fuente:* El Autor

- Para conectar el dispositivo con el pc se utiliza el puerto COM6, el XBee es acoplado a un Shield explorer de interface Mini USB a UART ademas lo polariza y regula.



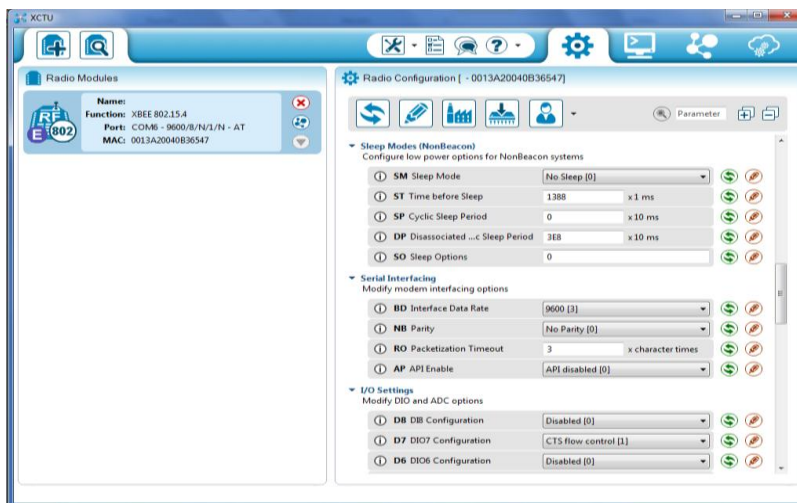
*Figura 40* Conexión de un Xbee receptor con su puerto com  
*Fuente:* El Autor

- Seguimos todos los pasos anteriores que era para el transmisor en este caso cambiamos el DL por 2222 y el MY por 1111
- El PANID es 3332 este viene por defecto y los dos transeportes deben tener el mismo .



**Figura 41** Configuración dirección del Xbee Receptor  
**Fuente:** El Autor

- Se mantiene los parámetros de BD a 9600 Bps, NB = no parity (0), entre los más importantes.



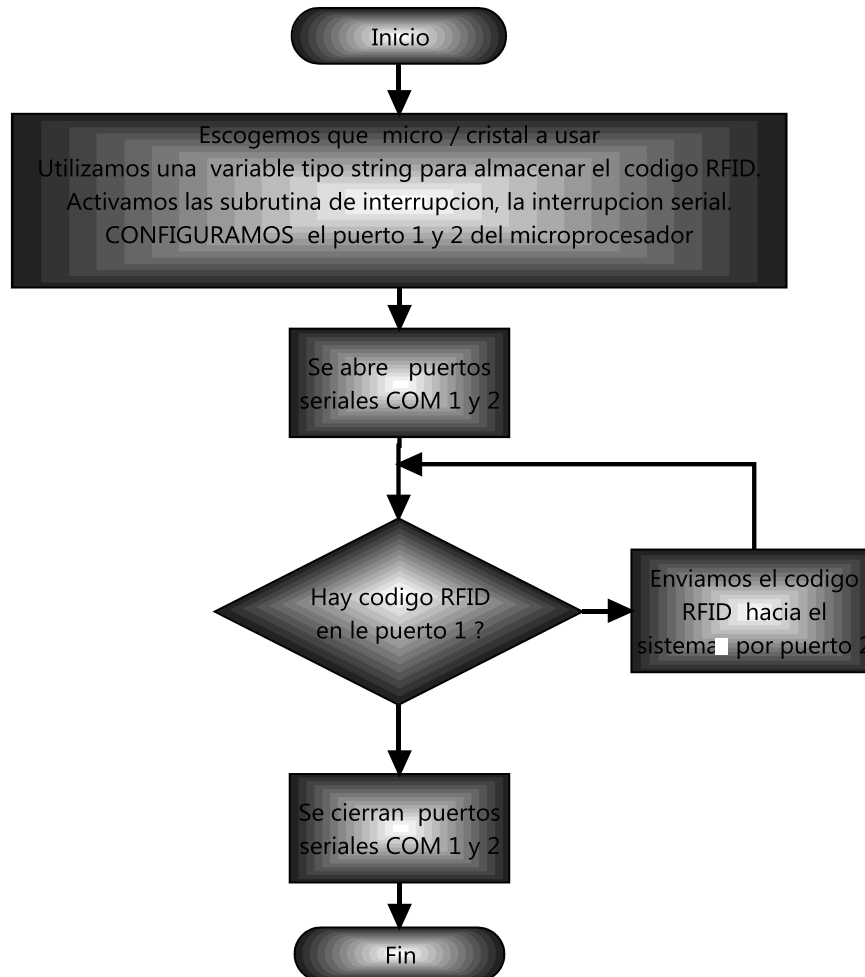
**Figura 42** Configuración serial Receptor  
**Fuente:** El Autor

## 6.7. Implementación del hardware de Recepción e Interface al Computador

En esta sección se muestra como se realizó la programación del pic el diseño de circuito impreso el montaje y pruebas de la etapa de recepción e interface a hacia el computador.

### 6.7.1. Programación Pic en el módulo de recepción

- Diagrama de Flujo



*Figura 43 Diagrama de flujo para la programación del pic*

*Fuente: El Autor*

- Código fuente

```
$regfile = "M164pdef.dat"    'DECLARO MICRO A USAR
$crystal = 8000000          'DECLARO CRISTAL A USAR
$hwstack = 32              'HARDWARE STACK COM. SERIAL
$swstack = 10              'SW STACK COM. SERIAL
$framesize = 40            'FRAME SPACE COM. SERIAL
Dim S As String * 11       'STRING PARA ALMACENAR CÓDIGON RFID
On Urxc Subserial          'ACTIVAR LA SUBROUTINA DE INTERRUPCION
Enable Interrupts          'ACTIVAR TODAS LAS INTERRUPCIONES DEL
MICRO
Enable Urxc                'ACTIVAR LA INTERRUPCION SERIAL DEL MICRO
```

```

Config Com1 = 9600 , Synchron = 0 , Parity = None , Stopbits = 1 , Databits =
8 , Clockpol = 0      'CONFIGURACIÓN PUERTO SERIAL 1
Config Com2 = 9600 , Synchron = 0 , Parity = None , Stopbits = 1 , Databits =
8 , Clockpol = 0      'CONFIGURACIÓN PUERTO SERIAL 2

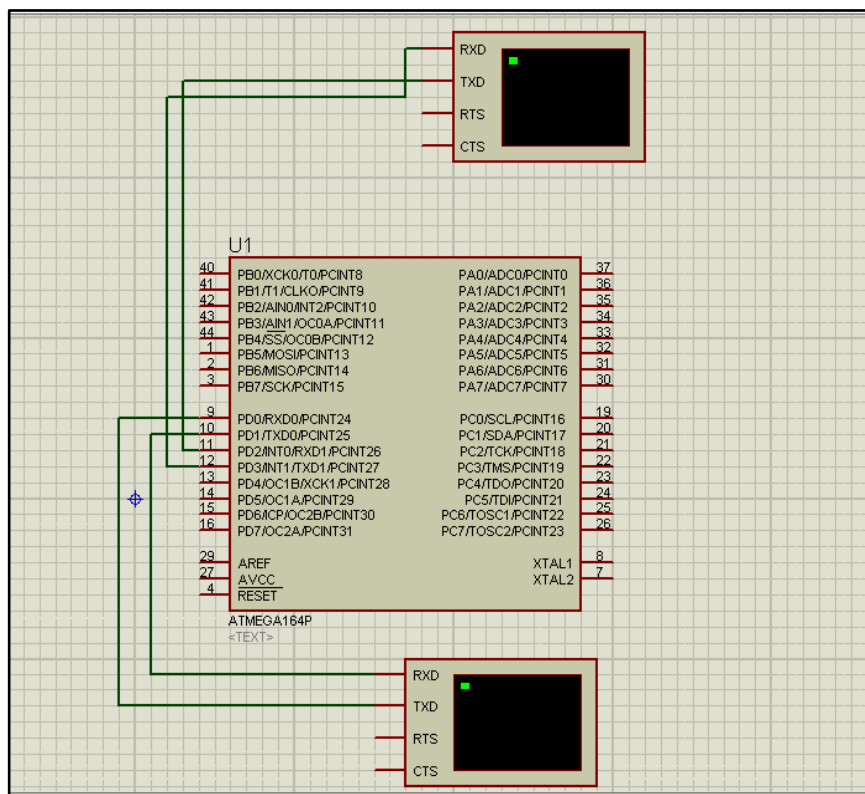
```

```

'ABRIR UARTS O PUERTOS SERIALES 1 Y 2
Open "Com1:" For Binary As #1
Open "Com2:" For Binary As #2
'LAZO INFINITO DO LOOP
Do
Loop
Subserial:      'SUBRRUTINA DE INTERRUPCIÓN SERIAL
Input #1 , S    'SE RECIBE CÓDIGO RFID POR PUERTO 1
Print #2 , S    'SE ENVÍA CÓDIGO RFID POR PUERTO 2
Return          'RETORNO DE INTERRUPCIÓN
'CERRAR UARTS O PUERTOS SERIALES 1 Y 2
Close #1
Close #2
End

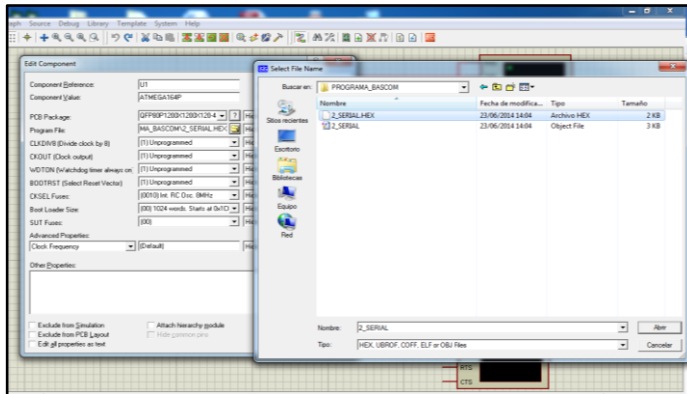
```

### 6.7.2. Simulación en Proteus

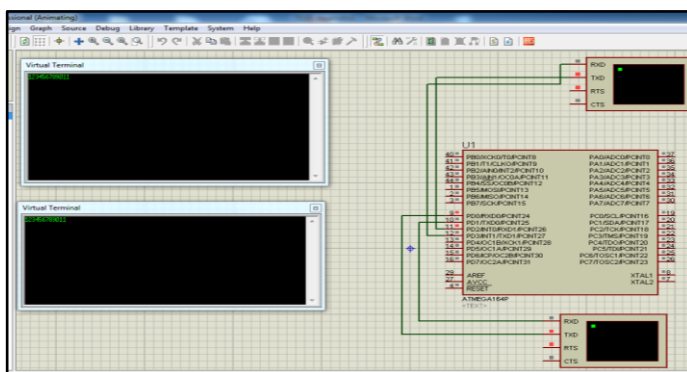


**Figura 44** Simulación y pruebas de recepción

*Fuente: El Autor*



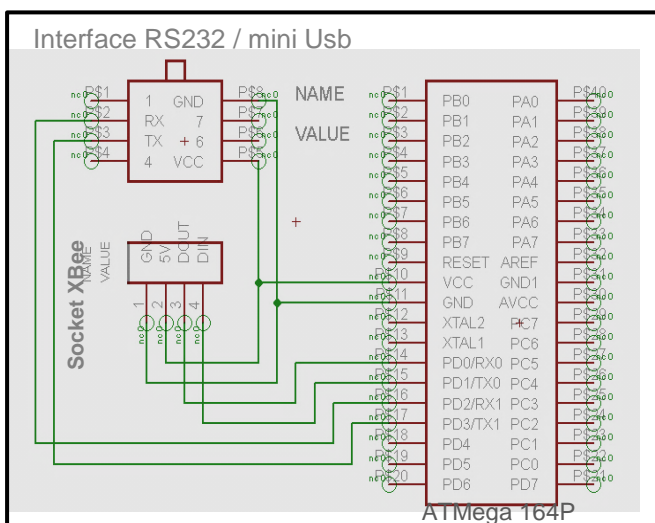
**Figura 45** Cargando el programa del pic en el simulador  
**Fuente:** El Autor



**Figura 46** Pruebas de envío y recepción de datos  
**Fuente:** El Autor

## 6.8. Montaje y Pruebas en el Protoboard

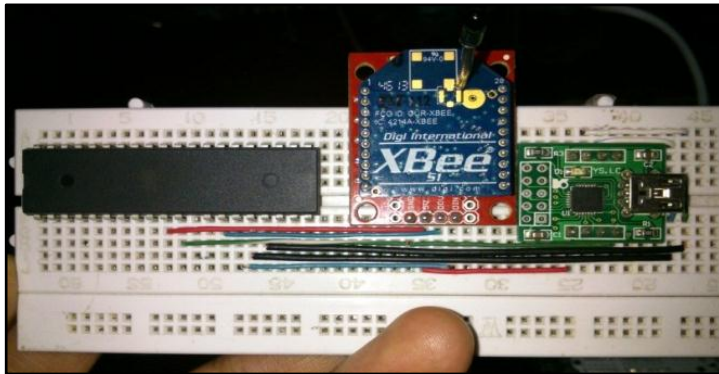
En base al siguiente diagrama esquemático se elaboró el circuito en el protoboard realizando las respectivas pruebas de recepción.



**Figura 47** Diagrama Esquemático del Módulo de Recepción  
**Fuente:** El Autor

### 6.8.1. Montaje del Circuito en el protoboard

Para realizar las pruebas iniciales es necesario hacer el montaje de los elementos y los módulos en un protoboard para comprobar en la parte física ya el funcionamiento de lo que será el receptor del código.



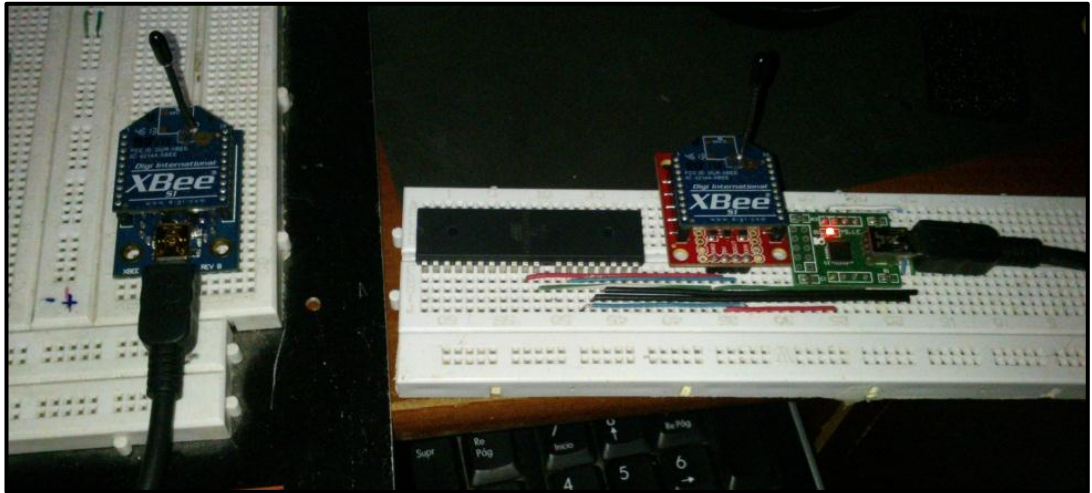
*Figura 48* Circuito de Recepción en el protoboard  
*Fuente: El Autor*

- Para comprobar el funcionamiento creamos una conexión con el Hyperterminal de Windows
- Creamos una nueva conexión llamada Lector RFID y escogemos el Icono que creamos conveniente



*Figura 49* Nueva conexión de Hyperterminal  
*Fuente: El Autor*

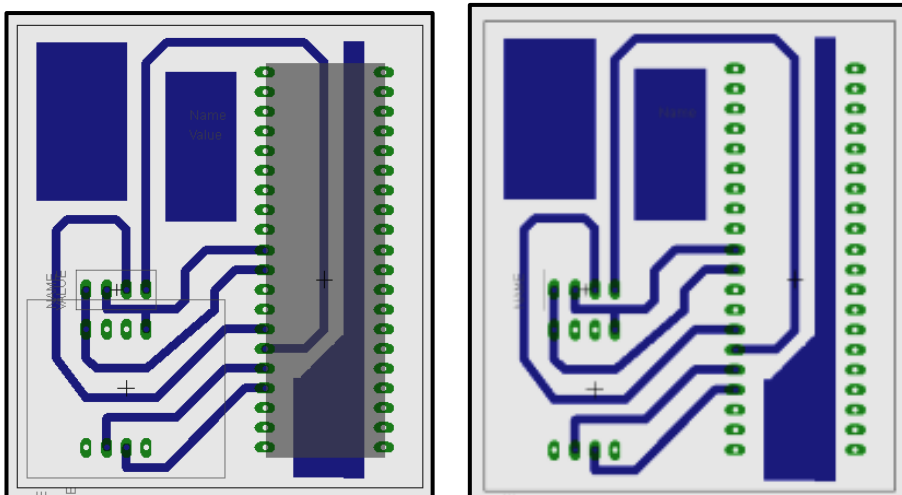
- Determinamos El puerto de conexión
- Configuramos las propiedades de la conexión
- Establecemos conexión y hacemos las pruebas



*Figura 50 Pruebas de interconexión entre los dos dispositivos transmisores*  
*Fuente: El Autor*

## 6.9. Diseño de la Placa del Circuito

Después de haber realizado las pruebas en el protoboard, y estar seguros de la recepción el circuito ya probado lo llevamos a la plaqueta o circuito impreso para hacerlo el montaje final del circuito. En la figura se muestra el diseño de las pistas en la placa de cobre se lo realizó en el Software electrónico Eagle.



*Figura 51 Placa de circuito impreso del Receptor*  
*Fuente: El Autor*



## **6.10. Desarrollo del Sistema**

En esta sección se explica la implementación del sistema tanto de la base de datos como de su respectivo lenguaje de aplicación con su respectiva ingeniería de software.

### **6.10.1. Desarrollo de la Aplicación Ingeniería de sistema COREGVAC**

Para interactuar con la base de datos es necesario realizar un sistema capaz de cumplir con los requisitos necesarios para llevar un control y registro de información del hato ganadero, para esto se utiliza Visual 2008 ya que es una herramienta potente de programación orientada a objetos.

#### **6.10.2.1 Antecedentes y Problematización**

Los Propietarios del hato ganadero ubicado en el cantón penipe y específicamente en la parroquia matus, realizan una actividad agrícola, ganadera teniendo una enorme responsabilidad de obtener productos de buena calidad; en la producción ganadera inyectar valor agregado y seguridad sanitaria a su base de producción que son las reses los productos que se obtienen de estos; para ello es necesario mejorar la forma del control y el registro que estos procesos lo ameritan; la toma de información se la hace de forma manual es decir la información se lo lleva en hojas o cuadernos siendo el acceso a estos datos una forma inconsistente y en muchos casos tediosa de encontrar, en casos extremos no existe dicha información, el dueño del hato ganadero no cuenta con un sistema que realice este trabajo de control y registro teniendo un problema considerable porque toda la información se la debe tener en tiempo real y de forma inmediata, ordenada para establecer parámetros de mejoras en los procesos de crianza vacunación prevención etc..

### 6.10.2.2 Definición de Requerimientos

La especificación de requerimientos para el desarrollo de este sistema se ha realizado gracias a la colaboración del dueño y el ayudante del hato ganadero.

### 6.10.2.3 Análisis del requerimiento

- No cuentan con un sistema computacional para llevar a cabo la recolección y manejo de datos
- Los datos recolectados que son las características propias de cada animal necesitan tener una base de datos accesible a consultas.
- Un sistema que facilite la tarea de recolectar datos de cada animal y hacerle un tratamiento ordenado, y sobre todo darle permanencia y seguridad de los mismos
- Facilitar en control y el registro de los animales.

### 6.10.2.4 Factibilidad técnica

- **Recursos hardware**

<b>PC</b>	Computador Portátil
<b>Procesador/ Velocidad</b>	Core I5/ 2.3Ghz
<b>Memoria</b>	4 Gb
<b>Disco Duro</b>	600Gb
<b>Módulo para recibir datos</b>	Hardware de recepción creado en el prototipo

*Tabla 15 Recursos de Hardware para el desarrollo del sistema*

*Fuente: El Autor*

- **Recursos Software**

<b>Sistema Operativo</b>	Microsoft Windows 7 SPI
<b>Lenguaje de programación</b>	Microsoft Visual Estudio 2008. Ver. prueba
<b>Base de datos</b>	Microsoft Access

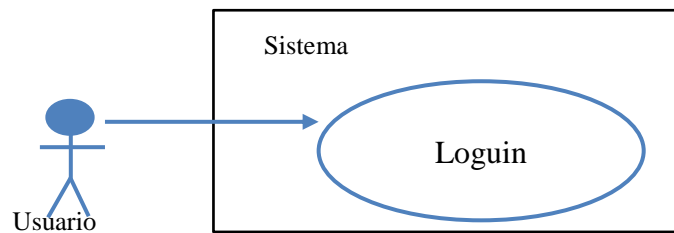
*Tabla 16 Recursos de Software para el desarrollo del sistema*

*Fuente: El Autor*

### 6.10.2.5 Casos del uso del sistema

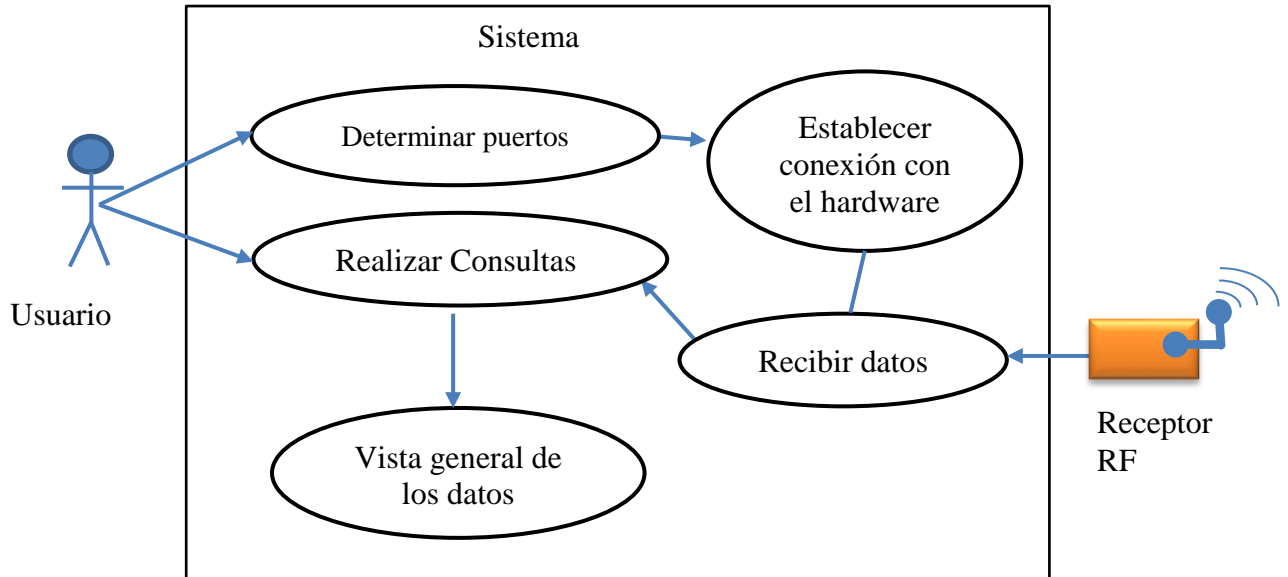
Los casos de uso se realizan para comprender los procesos del o los usuarios que van a interrelacionarse con el sistema, son una breve descripción. Estos se detallan a continuación.

- **Casos de uso ingresar al sistema**



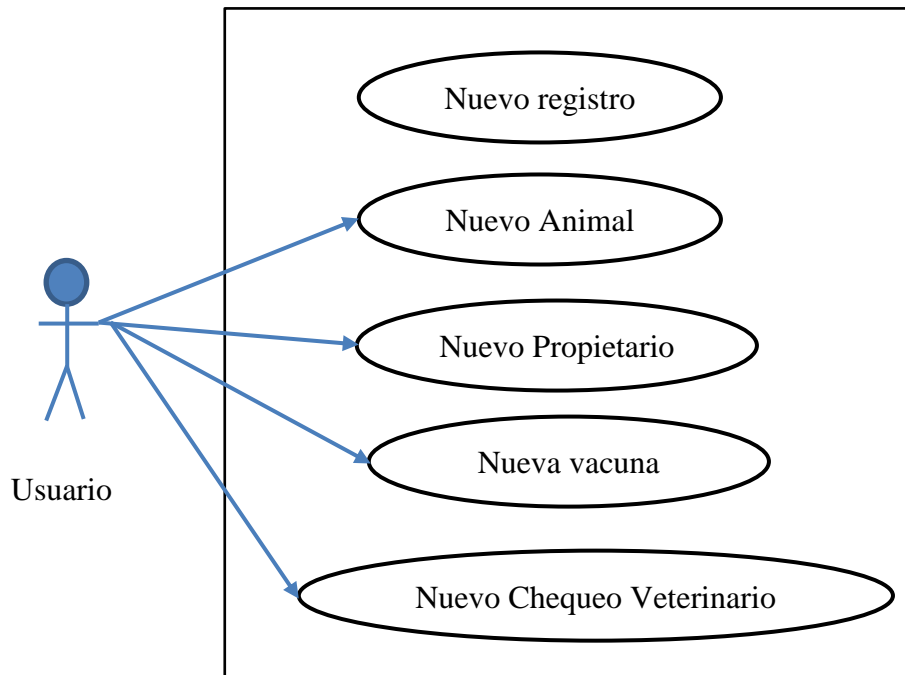
*Figura 52 Caso de uso Acceso al sistema*  
*Fuente: El Autor*

- **Caso de uso mostrar información de la Res**



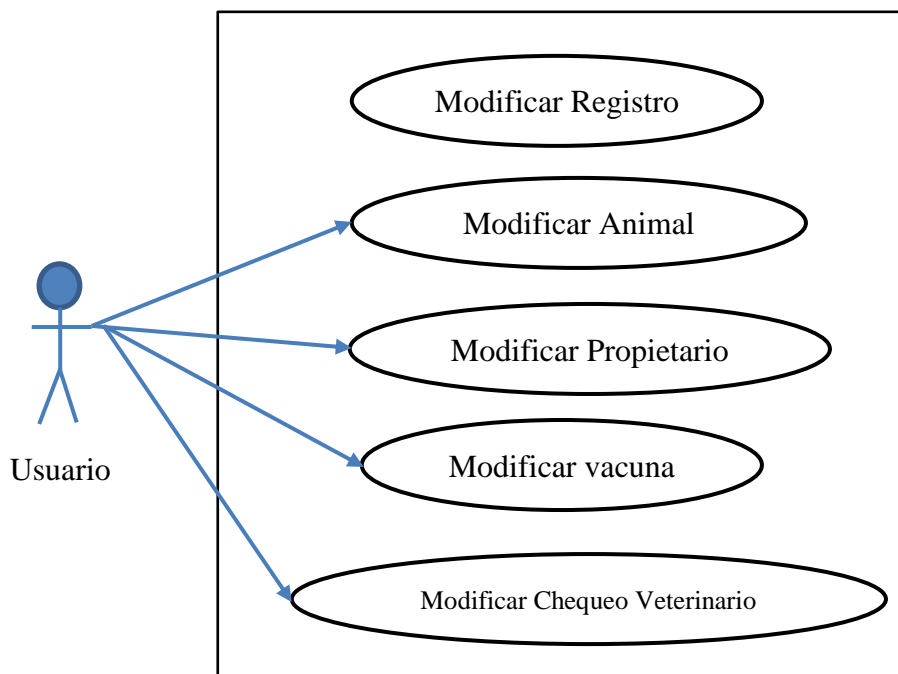
*Figura 53 Caso de uso Visualizar información de la Res*  
*Fuente: El Autor*

- **Caso de uso Agregar Registro**



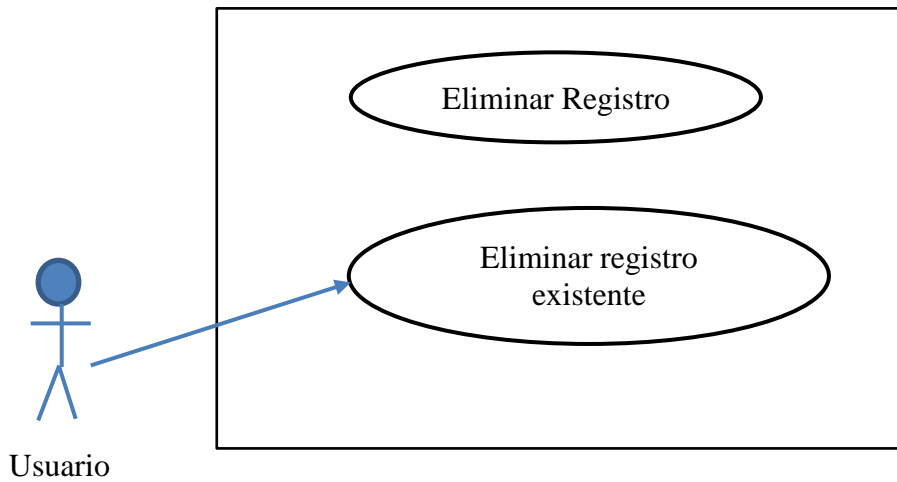
*Figura 54* Caso de uso Agregar registros  
*Fuente: El Autor*

- **Caso de Uso Modificar Registro**



*Figura 55* Caso de uso Modificar registros en el sistema  
*Fuente: El Autor*

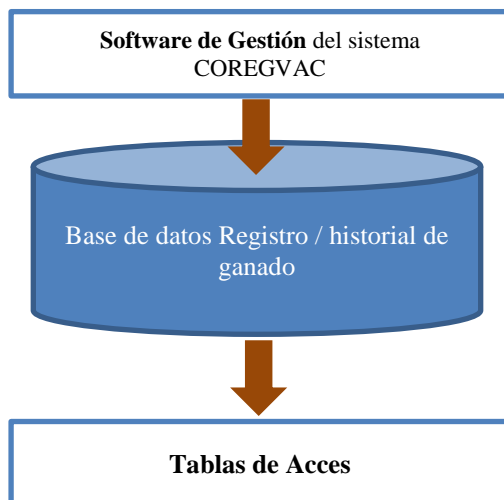
- **Caso de uso Eliminar registro**



**Figura 56** Caso de uso Eliminar Registro  
**Fuente:** El Autor

### 6.10.2. Base de datos

La base de datos presenta de manera sencilla el manejo y la organización de los datos haciendo que los procesos se lleven a cabo de forma segura, es decir el sistema minimiza los procesos y el tiempo en el que se trata la información y los resultados que esto implica, es una de las partes fundamentales del sistema ya que en esta se deposita toda la información a recolectar esta tiene el siguiente diseño:



**Figura 57** Diseño Sistema COREGVAC  
**Fuente:** El Autor

### 6.10.1.2 Modelo entidad relación

Este modelo permite representar los datos de entra y salida en el sistema, las relaciones entre objetos y los atributos que definen las propiedades de estos.

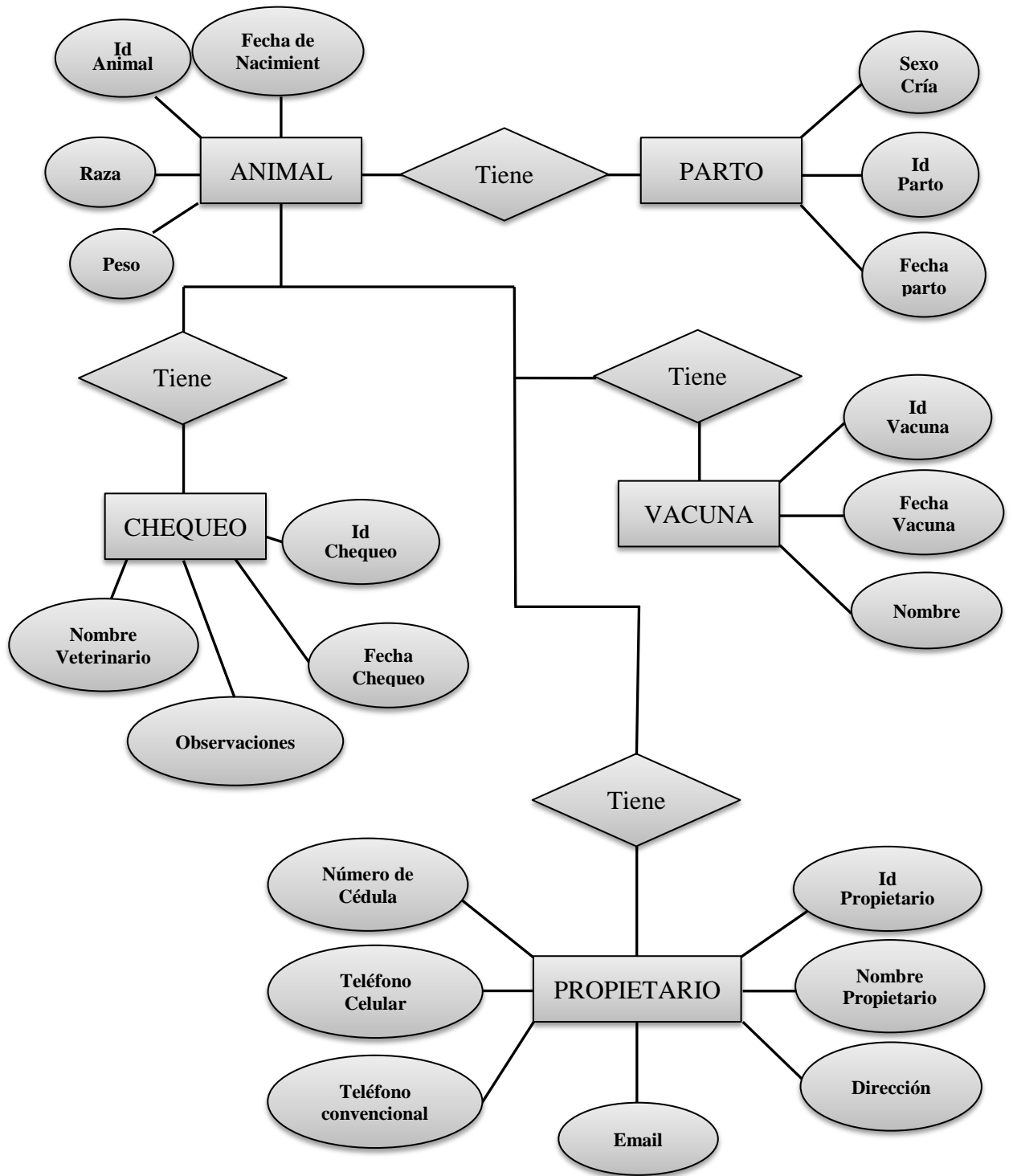
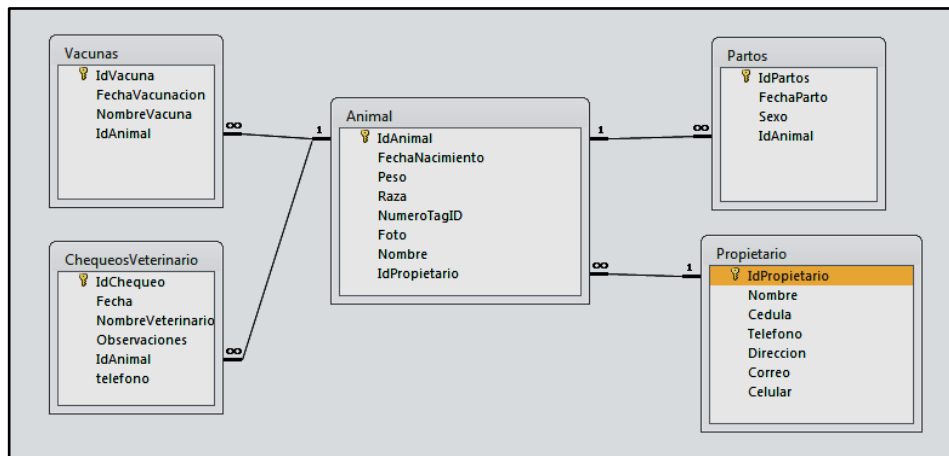


Figura 58 Modelo Entidad Relación  
Fuente: El Autor

- Diseño en Access de las tablas de la base de datos



**Figura 59** Tablas en la base de Datos

*Fuente: El Autor*

### 6.10.1.3 Diccionario de datos

El diccionario de datos es el que abarca las características lógicas del sistema, nombre, descripción, alias, contenido y organización. Este diccionario identifica los procesos donde se emplean los datos.

#### Nombre de la Tabla: ANIMAL

**Descripción:** Para registrar un animal

Nombre del dato	Usos	Tipo	PK
Id Animal	Llave primaria de la Tabla	Int	Si
FechaNacimiento	Fecha en que nace la cría / animal	FechaHora	no
Peso	Es el peso que tiene el animal	Texto	no
Raza	Es la raza que tiene el animal	Texto	no
NumeroTagID	Es el numero identificador único del animal en el tagRFID	Texto	no
Foto	Es la imagen del animal	Objeto Ole	no
Nombre	Es el nombre que se le da al animal	Texto	no
IdPropietario	Llave foránea de la tabla animal	Numérico	no

**Tabla 17** Diccionario de Datos Animal

*Fuente: El Autor*

**Nombre de la Tabla:** PROPIETARIO**Descripción:** Para registrar los datos personales del propietario

Nombre del dato	Usos	Tipo	PK
Id Propietario	Llave primaria de la Tabla	Int	Si
Nombre	Nombre del propietario	Texto	no
Cedula	Es la cedula del propietario	Texto	no
Teléfono	Es el número telefónico Convencional	Numero	no
Celular	Es el número telefónico Celular	Texto	no
Dirección	Es la dirección de domicilio	Texto	no
E mail	Es el nombre que se le da al animal	Texto	no

*Tabla 18. Diccionario de Datos Propietario**Fuente: El Autor***Nombre de la Tabla:** CHEQUEOS VETERINARIO**Descripción:** Para registrar y hacer un seguimiento del proceso de saneamiento y prevención del animal vacuno.

Nombre del dato	Usos	Tipo	PK
IdChequeo	Llave primaria de la Tabla	Int	Si
Fecha	Fecha en que se realiza un chequeo	Fecha/Hora	no
Nombre Veterinario	Nombre del Encargado del animal	Texto	no
Observaciones	Observaciones anotadas por el veterinario del animal	Texto	no
IdAnimal	Llave foránea de la tabla	Int	no
teléfono	Es el teléfono del veterinario	Texto	no

*Tabla 19 Diccionario de Datos Chequeos Veterinario**Fuente: El Autor*



**Nombre de la Tabla:** VACUNAS

**Descripción:** Para registrar y hacer un seguimiento del proceso de vacunación y dosis suministrada a cada animal vacuno.

Nombre del dato	Usos	Tipo	PK
IdVacunas	Llave primaria de la Tabla	Int	Si
Fecha Vacunación	Fecha en que a un animal se le dosifica una vacuna	Fecha/Hora	no
Nombre Vacuna	Es el nombre del medicamento	Texto	no
IdAnimal	Llave foránea de la tabla	Int	no

*Tabla 20 Diccionario de Vacunas*

*Fuente: El Autor*

**Nombre de la Tabla:** PARTOS

**Descripción:** Cantidad de partos que tiene un animal.

Nombre del dato	Usos	Tipo	PK
IdPartos	Llave primaria de l a Tabla	Int	Si
FechaParto	Fecha en que un animal nace	Fecha/Hora	no
Sexo	Si el animal nace macho o hembra	Texto	no
IdAnimal	Llave foránea de la tabla	Int	no

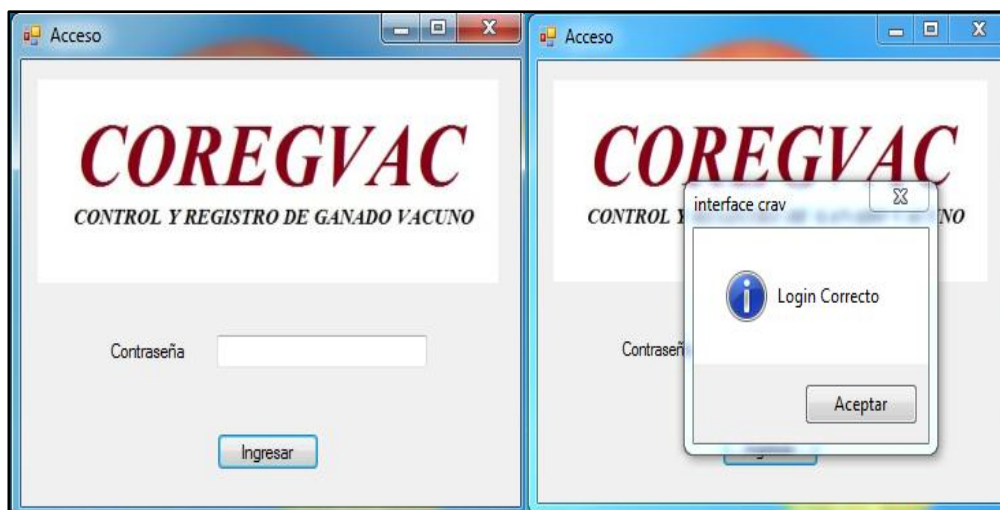
*Tabla 21 Diccionario de Datos Partos*

*Fuente: El Autor*

### 6.10.3 Diseño del sistema

Mediante un lenguaje de programación orientado a objetos se presenta el diseño del sistema COREGVAC, además previamente interactuando con el usuario se le presenta primero unos bosquejos (ir anexo Bosquejos de interfaces de usuario) de forma general de cómo será el sistema consiguiendo con algunas modificaciones quede de la siguiente manera.

- **Logín de Usuario.**- El usuario ingresa la contraseña e ingresa al sistema



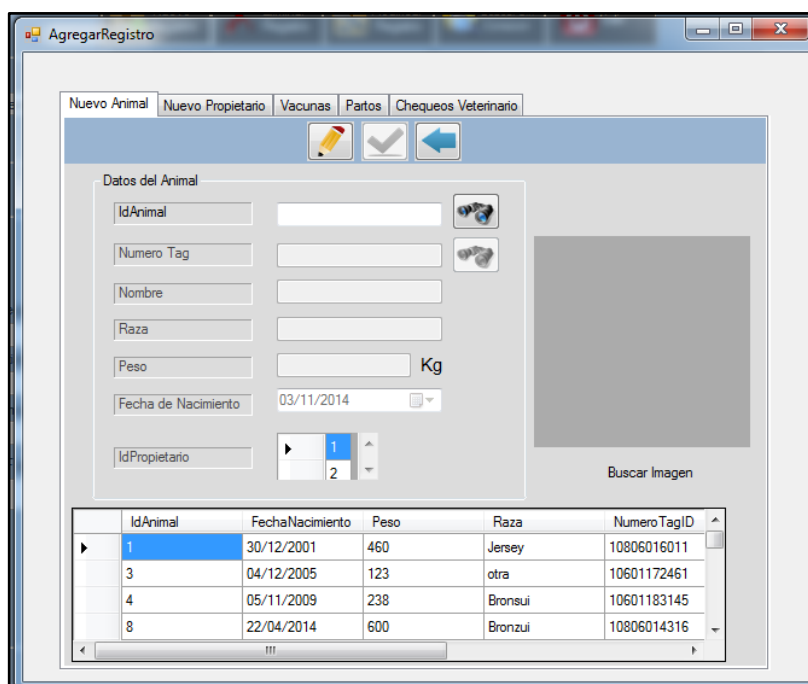
*Figura 60 Ingreso al sistema COREGVAC*  
*Fuente: El Autor*

- **Datos Reses.**- Interface principal del software donde se muestra la información de la res además está la opción en donde se gestiona la conexión con su respectivo puerto de comunicación COM .



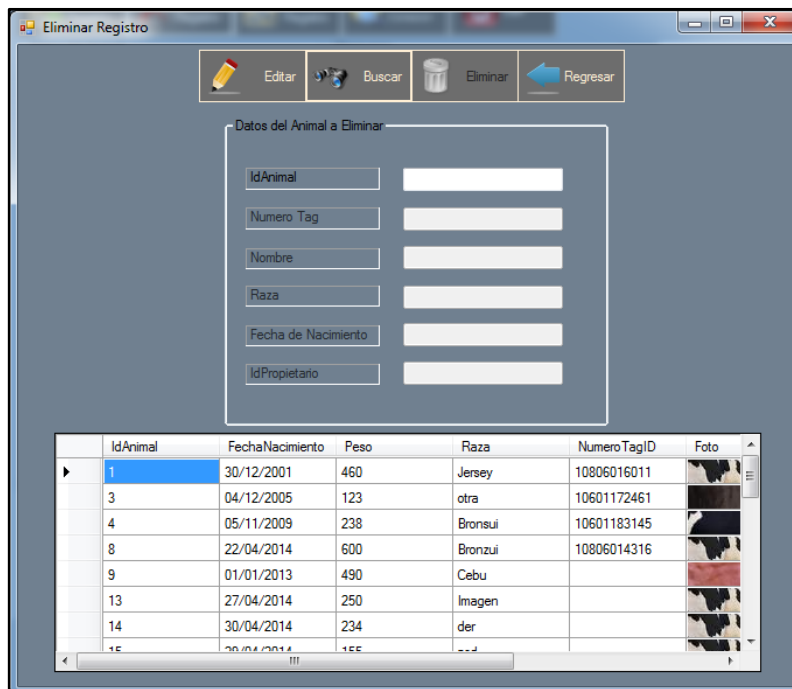
**Figura 61** Interface general de datos en el sistema COREGVAC  
**Fuente:** El Autor

- **Agregar Registro.-** Permite agregar nuevos registros



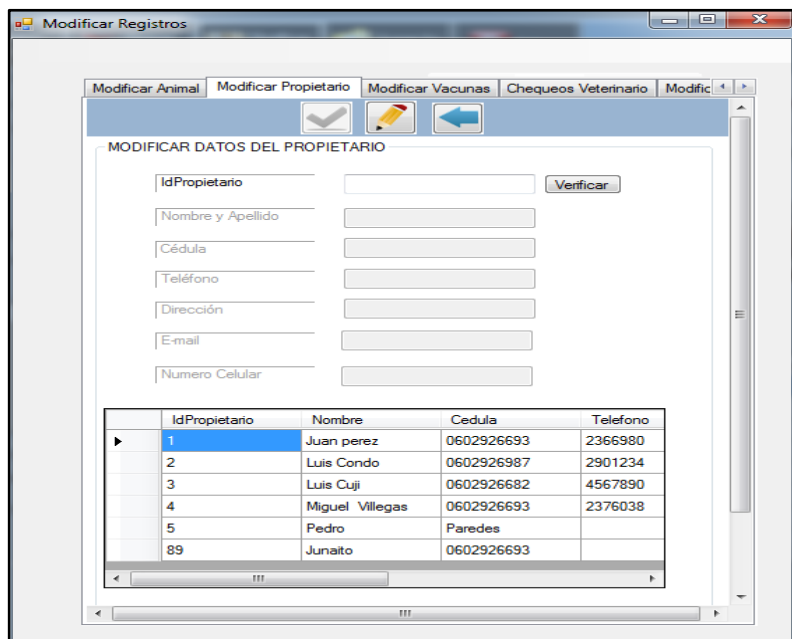
**Figura 62** Ventana para Agregar un registro  
**Fuente:** El Autor

- **Eliminar Registro.-** Permite eliminar registros de los animales existentes



*Figura 63 Ventana para Eliminar un Registro*  
Fuente: El Autor

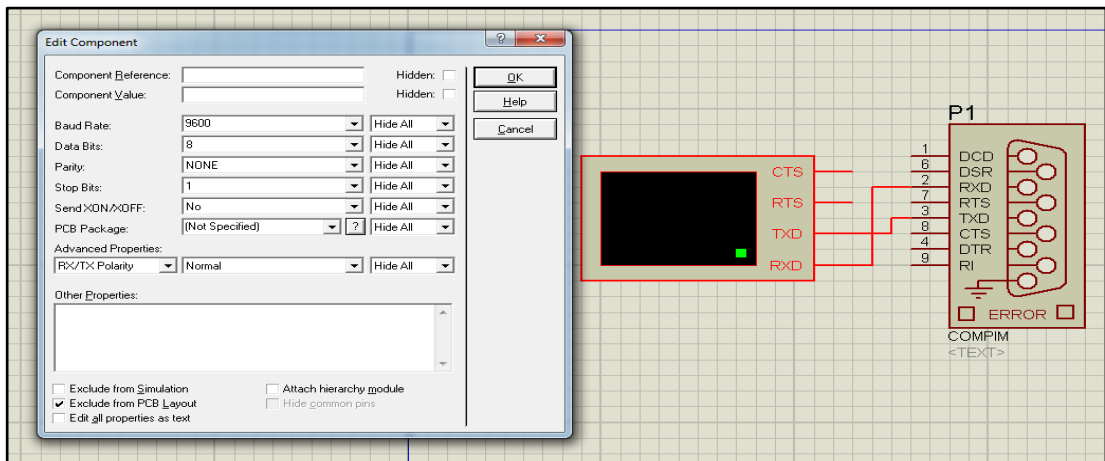
- **Modificar Registros.-** Permite Modificar los registros existentes



*Figura 64 Ventana para Eliminar un Registro*  
Fuente: El Autor

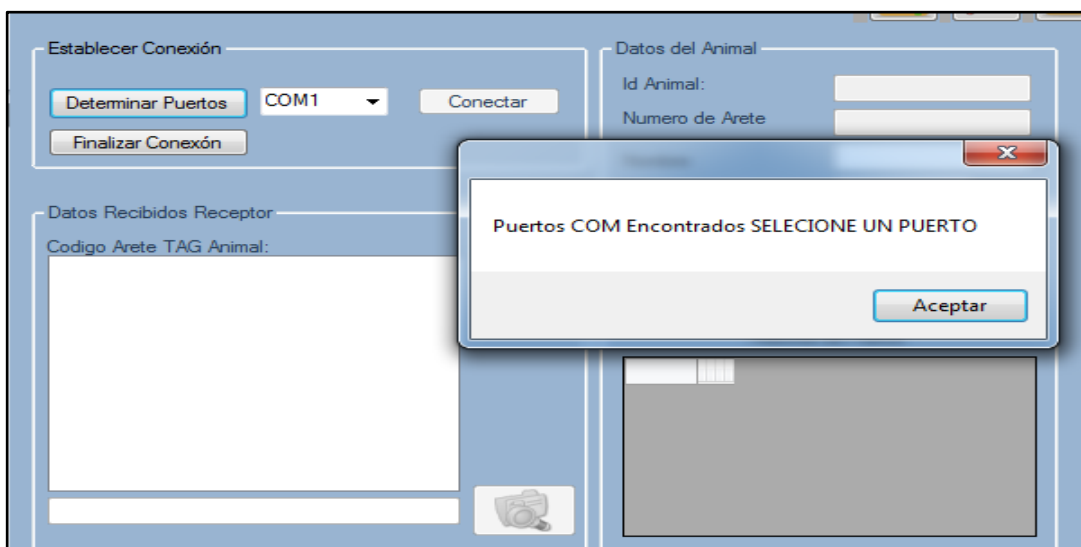
## 6.11 Pruebas de Comunicación virtual con el software COREGVAC y Proteus

Para estas pruebas se utilizó el simulador proteus estableciendo un COM o Puerto de comunicación virtual, configurado con las propiedades como se muestra en la figura.



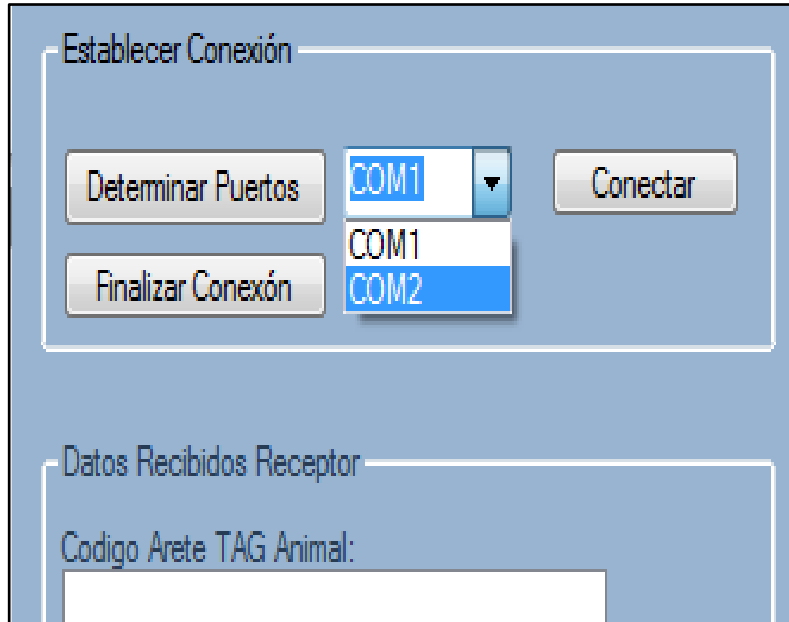
**Figura 65** Configuración del puerto virtual en proteus  
**Fuente:** El Autor

- Determinamos los puertos existentes en nuestro pc esto lo realizamos ya con nuestro software de aplicación.



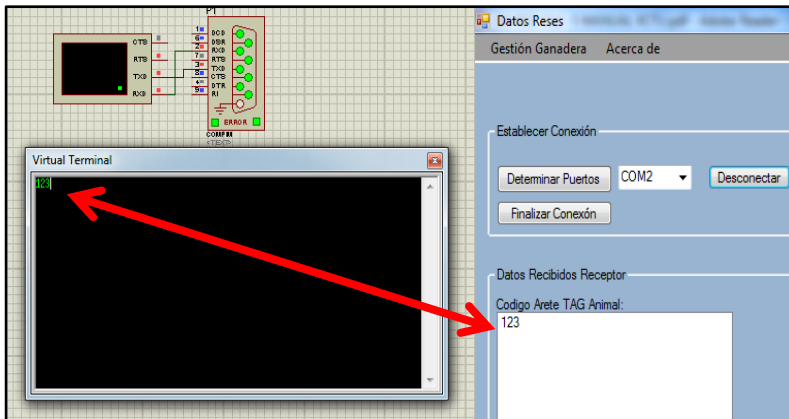
**Figura 66** Búsqueda de puertos con el software  
**Fuente:** El Autor

- Seleccionamos un puerto existente en este caso es el COM2 que es un puerto virtual y está asociado al pc, el COM1 es el que está configurado en el Proteus. Hay que tener mucho cuidado en esto.



**Figura 67** Selección del Puerto de Comunicación con  
**Fuente:** El Autor

- Enviamos un datos desde el proteus y comprobamos la transmisión y se visualiza en pantalla en la base de datos



**Figura 68** Envío de datos desde proteus hacia el software COREGVAC  
**Fuente:** El Autor

## 6.12 Resultado Integración y pruebas del Hardware y el software Resultado

En las Pruebas finales el dispositivo transmite con una línea de vista de forma óptima, en la figura se puede apreciar todo el sistema funcionando. Además en las pruebas de identificación individual se obtuvo un 100% de resultados efectivos, el único parámetro a tomar en cuenta es la cobertura del radio enlace es de aproximadamente 100 metros en línea de vista esto durante las pruebas.

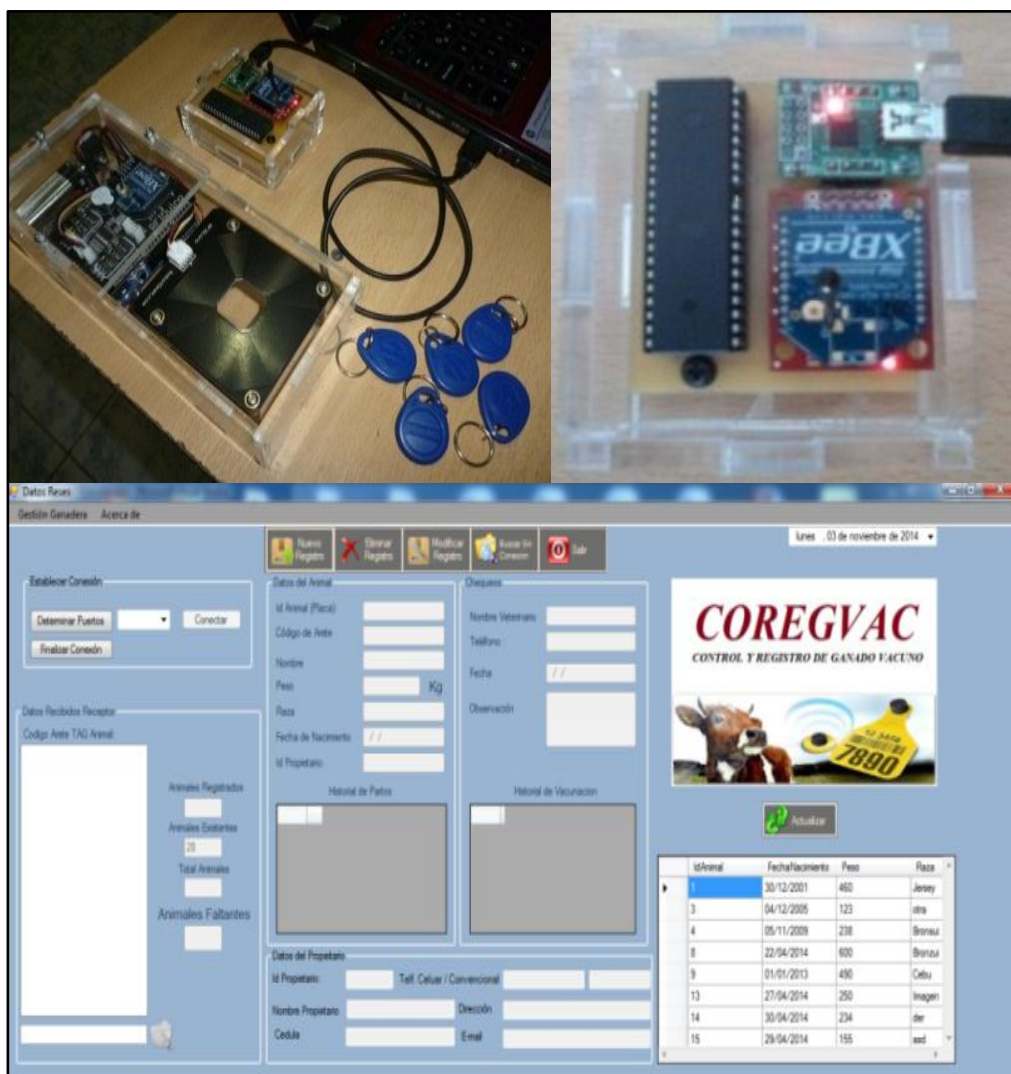


Figura 69 Sistema COREGVAC  
Fuente: El Autor

## **6.13 CONCLUSIONES Y RECOMENDACIONES**

### **6.13.2 Conclusiones**

- El resultado de la creación del prototipo ha logrado cumplir con los objetivos de la propuesta, es decir se diseñó e implemento un sistema de control y registro computacional de ganado vacuno utilizando como base la tecnología RFID. Al utilizar tarjetas de desarrollo en lugar de microcontroladores y tarjetas de fabricación artesanal, en la mayoría de este proyecto es por la buena estabilidad y bajo ruido en la transmisión y recepción de los datos ya que la tecnología RFID como ya se vio utiliza la propagación de ondas electromagnéticas lo que conlleva a que en una tarjeta artesanal no se cuenta con un material antiestático y de bajo ruido y el proceso de transición tenga interferencias e influya en un registro incorrecto de datos.
- Esta herramienta integrada con su hardware y su software ofrece una ayuda importante en el control y registro de ganado vacuno en un hato específico de producción, ofrece valor agregado y seguridad sanitaria al utilizarlo, además se puede implementar en otros hatos de producción. La creación y aplicación del sistema prototipo CORECVAG, nos permitió tener mayor certeza en el tratamiento de la información, su diseño busca satisfacer todas las necesidades de un ganadero, permitiendo tener integridad y automatización en los datos de los animales, presentando además interfaces amigables y de fácil operación para el usuario.
- Al utilizar transmisores de RF tanto en el lector emisor como en el receptor hace que esta herramienta sea autónoma y convirtiéndose en un dispositivo robusto y modular para la transmisión y recepción de datos sin que intervengan otros canales de transmisión convencionales además trabaja en un espectro de frecuencia gratuito.



### 6.13.3 Recomendaciones

- Si el sistema prototipo se llegara a implantar en lugares con mayor área de cobertura sería necesario cambiar los transmisores para tener mayor alcance entre emisor y receptor y obviamente hacer las pruebas de conectividad estableciendo líneas de vista para la comunicación.
- Cuando se implanta este tipo de herramienta es necesario brindar una capacitación a la persona que va a utilizarla para garantizar el óptimo funcionamiento de los módulos y del sistema en general. Puesto que el sistema trabaja con una base de datos es recomendable realizar respaldos periódicos con el fin de garantizar disponibilidad de los mismos y sobre todo tener siempre esta información al alcance y de forma confiable.
- Para realizar este tipo de trabajos donde implica hacer transmisión y obtener datos por radiofrecuencia se debe utilizar módulos que no genere ruidos esto con la finalidad de obtener la información de forma correcta y sin distorsión.

## CAPITULO VII

### 7 BIBLIOGRAFIA

#### BIBLIOGRAFIA

- Axelson, J. (2007). Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems, segunda edición
- Everts, G. (2008). Paseo por Visual Basic 2008.
- Gidekel, A. (2006), Introducción a la identificación por radiofrecuencia RFID
- Pressman, R.S. (2010). Ingeniería de software. Un enfoque Práctico, séptima edición, Pag 351 a la 362.
- Wiley Publishing, Inc.(2008). Professional Visual Basic® 2008, Bill Evjen, Billy Hollis Bill, Sheldon Kent Sharkey.

#### WEBGRAFIA

- Academia.edu, F. c. (11 de 05 de 2014). *Una guía práctica sobre el mundo de Arduino*. Recuperado el 14 de 12 de 2014, de <http://www.academia.edu>:  
[http://www.academia.edu/8875046/Una\\_gu%C3%ADa\\_pr%C3%A1ctica\\_sobre\\_el\\_mundo\\_de\\_Arduino](http://www.academia.edu/8875046/Una_gu%C3%ADa_pr%C3%A1ctica_sobre_el_mundo_de_Arduino)
- Arduino. (12 de 12 de 2013). <http://www.arduino.cc>. Recuperado el 20 de 09 de 2014, de <http://www.arduino.cc>:  
<http://www.arduino.cc/en/guide/introduction>
- Atmel. (01 de 06 de 2014). *Atmel Data Sheet*. Recuperado el 12 de 12 de 2014, de <http://www.atmel.com>: [http://www.atmel.com/images/Atmel-8272-8-bit-AVR-microcontroller-ATmega164A\\_PA-324A\\_PA-644A\\_PA-1284\\_P\\_datasheet.pdf](http://www.atmel.com/images/Atmel-8272-8-bit-AVR-microcontroller-ATmega164A_PA-324A_PA-644A_PA-1284_P_datasheet.pdf)
- Digi. (02 de 03 de 2013). *XCTU Next generation configuration platform for XBee*. Recuperado el 12 de 01 de 2014, de <http://www.digi.com>:

<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu>

INTECO, Instituto Nacional de Tecnologías de la Comunicación Española. (06 de 03 de 2010). *Guía sobre seguridad y privacidad de la tecnología RFID*.

Recuperado el 20 de 01 de 2014, de <https://www.agpd.es>:

[https://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Guias/Guia\\_RFID.pdf](https://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Guias/Guia_RFID.pdf)

LIBERA. (27 de 12 de 2010). *whitepaper RFID: Tecnología, Aplicaciones y Perspectivas*. Recuperado el 04 de 08 de 2014, de Libera Networks:

[http://www.libera.net/uploads/documents/whitepaper\\_rfid.pdf](http://www.libera.net/uploads/documents/whitepaper_rfid.pdf)

MCI Electronics, A. O. (17 de 10 de 2010). *Guía del Usuario XBEE Series I*.

Recuperado el 16 de 09 de 2014, de <http://www.olimex.cl>:

[http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia\\_Usuario.pdf](http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf)

Microsoft. (1 de 11 de 2007). *Visual Studio*. Recuperado el 14 de 09 de 2014, de

Visual Estudio 2008: <https://msdn.microsoft.com/es-es/library/52f3sw5c%28v=vs.90%29.aspx>

Microsoft. (31 de 05 de 2014). *Tareas básicas en Access 2010*. Recuperado el 14 de 09 de 2014, de Tareas básicas en Access 2010:

<https://support.office.com/es-es/article/Tareas-b%C3%A1sicas-en-Access-2010-268acfed-2484-4822-acb3-c30e58045588?ui=es-ES&rs=es-ES&ad=ES>

Telectrónica. (25 de 08 de 2006). *Introducción a la Identificación por Radio*

*Frecuencia - RFID*. Recuperado el 12 de 12 de 2013, de

<http://www.telectronica.com/>:

<http://www.telectronica.com/index.php/libro-rfid-telectronica>

Xbee.cl Ingeniería MCI Ltda. (01 de 01 de 2014). *¿QUÉ ES XBEE?* Recuperado

el 14 de 08 de 2014, de <http://xbee.cl>: <http://xbee.cl/que-es-xbee/>

# **ANEXOS**

# MANUAL DE USUARIO

## **COREGVAC** CONTROL Y REGISTRO DE GANADO VACUNO Versión 1.0



## SISTEMA DE CONTROL Y REGISTRO DE GANADO VACUNO

## 1. Descripción del Sistema

El sistema llamado COREGVAC, es un sistema robusto, confiable que mejora la producción y la productividad dentro de un hato ganadero, es una aplicación diseñada para el control y el registro de Ganado Vacuno brindando al usuario ventajas tales como:

- Una forma muy simple de manejar la información ya que provee de interfaces fáciles de interpretar y de interactuar,
- Mantiene de forma ordenada, segura y precisa la información de un animal características propias como peso raza, foto etc.

Este software permite gestionar las siguientes tareas

- Ingreso al sistema
- Visualizar toda la información directa en base a un código recibido
- Agregar Registros
  - Nuevo animal
  - Nuevo propietario
  - Nuevas Vacunas
  - Nuevos Partos
  - Nuevos chequeos
- Eliminar Registros
- Modificar registros
- Búsquedas en una base de datos con y sin conexión a su dispositivo electrónico
- Establecer una conexión mediante el puerto USB con el hardware de recepción

## 2. Especificaciones técnicas

El sistema de control y registro de ganado vacuno de sus siglas COREGVAC debe tener las siguientes especificaciones técnicas mínimas para su funcionamiento.

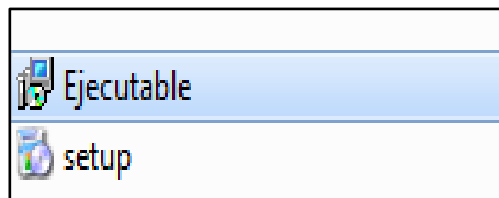
Software	Hardware
<ul style="list-style-type: none"> <li>• Sistema Operativo Windows vista, 7</li> <li>• FrameWork3.5 o superior</li> </ul>	<ul style="list-style-type: none"> <li>• Microprocesador Dual core o superior</li> <li>• Disco duro 160 Gb</li> <li>• Memoria Ram 1GB</li> <li>• Puertos de comunicación USB necesarios</li> </ul>

**Tabla 22** Especificaciones técnicas mínimas para el funcionamiento de COREGVAC  
Fuente: El Autor

## 3. Guía de Instalación

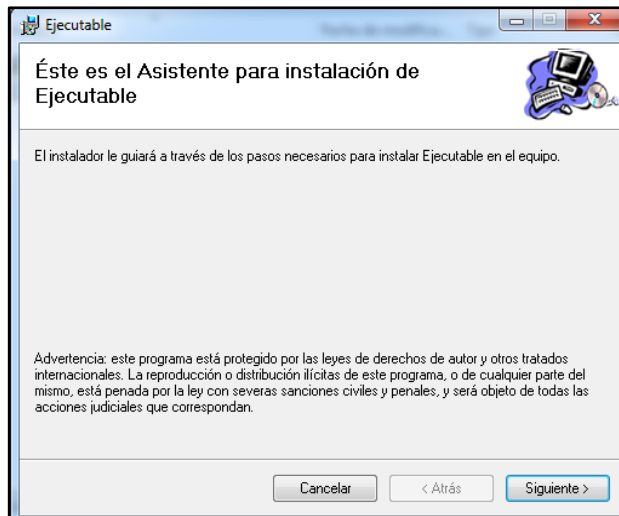
A continuación se detalla como instalar el software necesario del sistema COREGVAC todo el software necesario se encuentra en el CD de instalación.

- Insertamos el cd de la aplicación
- Dos clics en el archivo ejecutable



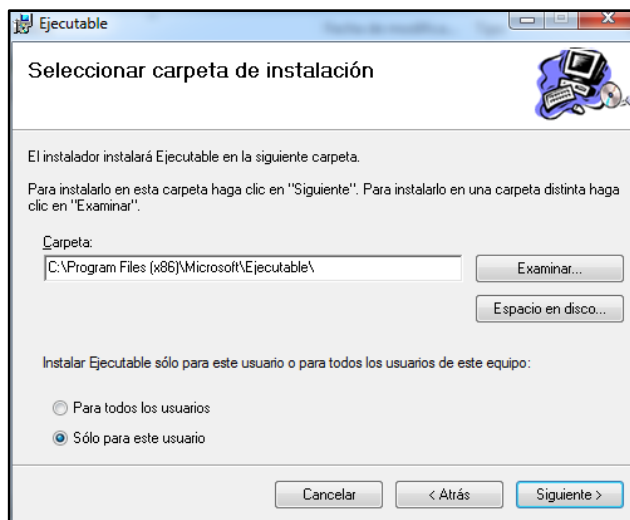
**Figura 70** Ejecutar programa de instalación  
Fuente: El Autor

- Click en siguiente



**Figura 71** Asistente de instalación  
Fuente: El Autor

- Seleccionamos la carpeta de instalación luego en siguiente y confirmamos la instalación



**Figura 72** Selección carpeta de instalación  
Fuente: El Autor

- El sistema se instala y en el escritorio aparecerá el ejecutable.

**Archivo base de Datos.-** El sistema cuenta con una base de datos la cual se va a integrar y la misma que llevara toda la información. Para esto hay que copiar el



archivo que se encuentra en el cd de instalación para esto seguimos los siguientes pasos.

- ✓ Ir a la carpeta base y copiarla en la unidad c:/ y luego ejecutar la aplicación

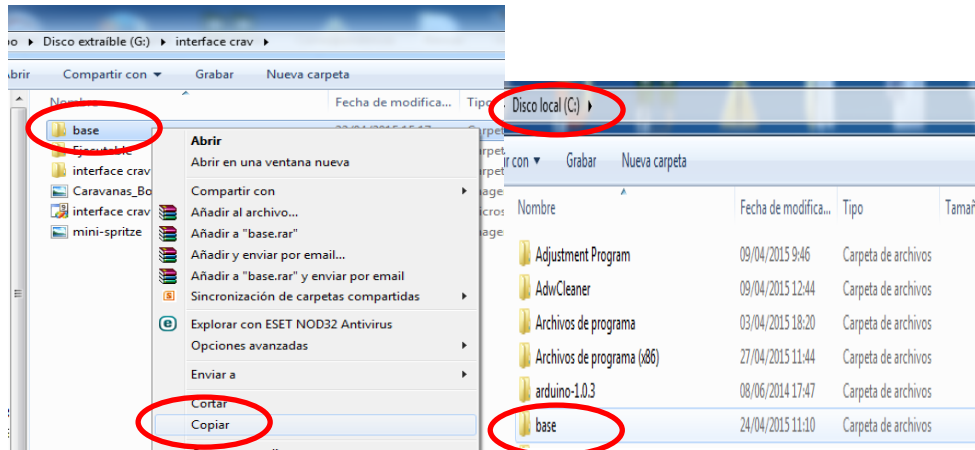


Figura 73 archivo de Base de datos  
Fuente: El Autor

#### 4. Ingreso al sistema

En la pantalla de acceso se requiere contraseña y el usuario ingresa al sistema al dar click en ingresar



Figura 74 Ingreso al sistema  
Fuente: El Autor

#### 1. Ventana principal de información

En esta pantalla se muestran todas las características propias del animal a consultar



**Figura 75** Ventana principal de Información  
**Fuente:** El Autor

- Panel de conexión y determinación de puertos existentes para empezar a recibir los código por medio del dispositivo de recepción



**Figura 76** Panel de Conexión con le Receptor  
**Fuente:** El Autor

## 2. Cinta de opciones

La cinta de opciones es una manera de organizar ventanas relacionadas para que sean fáciles de encontrar.

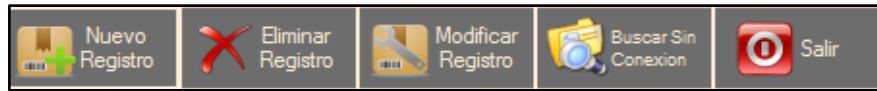


Figura 77 Cinta de opciones

Fuente: El Autor

### 4.1 Nuevo Registro

Permite crear nuevos registros individuales tiene 5 opciones. Nuevo animal, Nuevo propietario Nuevas Vacunas, Nuevos Partos, Nuevos chequeos veterinarios

Todas las ventanas tienen esta cinta de opciones



- **Nuevo Animal.-** Permite crear un nuevo registro individual de cada animal con todas sus características propias

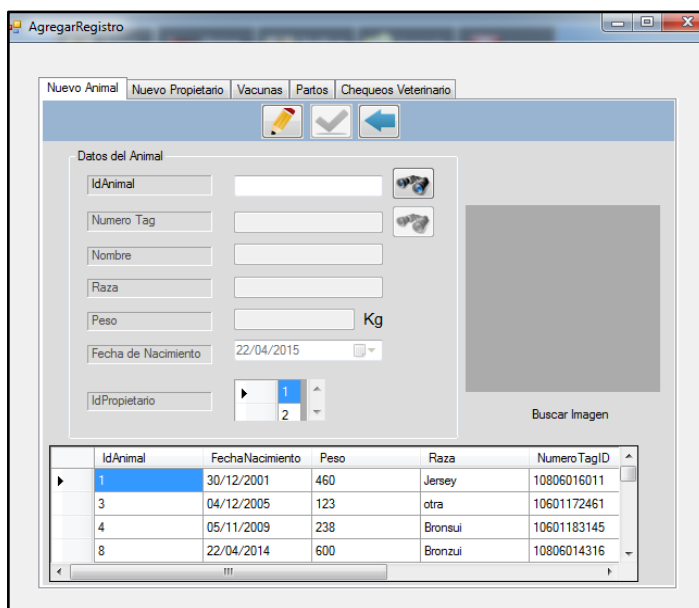


Figura 78 Ventana para Nuevo Registro Animal

Fuente: El Autor

- **Nuevo Propietario.-** Permite crear un nuevo registro individual del propietario anterior al que se le adquirió un animal llenando sus campos como se muestra en la figura.

	IdPropietario	Nombre	Cedula	Celular	Direccion
▶	1	Juan perez	0602926693	995082778	Riobamba
	2	Luis Condo	0602926987	9878589700	Penipe
	3	Luis Cuji	0602926682	989878765	Guquillas
	4	Miguel Villegas	0602926693	995082778	algun lugar
	5	Pedro	Paredes	0989876543	Algun Luqar

*Figura 79 Ventana para Nuevo Registro Propietario*

*Fuente: El Autor*

- **Nuevas Vacunas.-** Permite crear un nuevo registro individual de una vacuna asociada a un animal llenando sus campos como se muestra en la figura.

	IdVacuna	FechaVacunacion	NombreVacuna	IdAnimal
▶	6	28/05/2014	er	1
	7	28/05/2014	Aftosa 500mls	3
	2	21/05/2014	Ivermectina y Ant...	1
	10	26/11/2014	aftosa	4563

*Figura 80 Ventana para Nuevo Registro Vacunas*

*Fuente: El Autor*

- **Nuevos Partos.-** Permite crear un nuevo registro individual de una parto y registrar el sexo de la cría

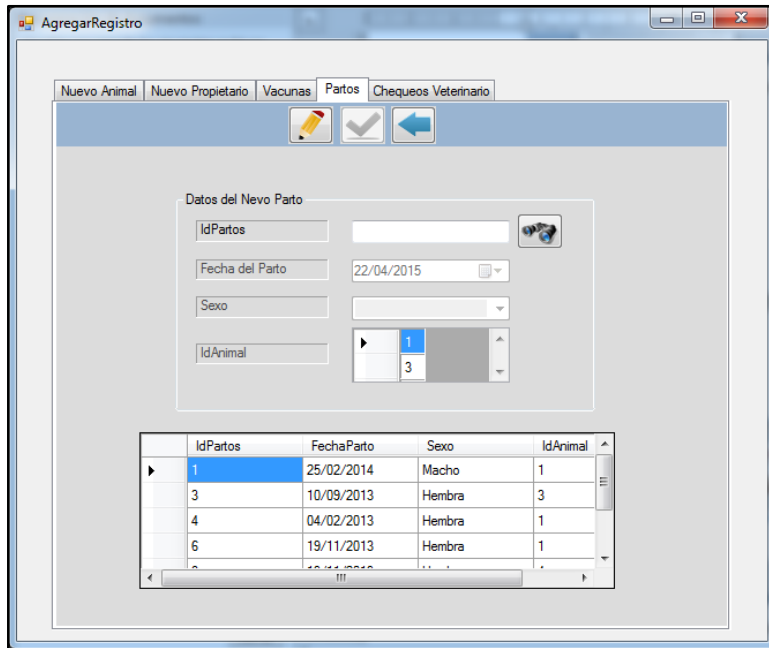


Figura 81 Ventana para Nuevo Registro Partos  
Fuente: El Autor

- **Nuevo Chequeo veterinario.-** Permite crear un nuevo registro individual de un chequeo veterinario que tiene el animal.

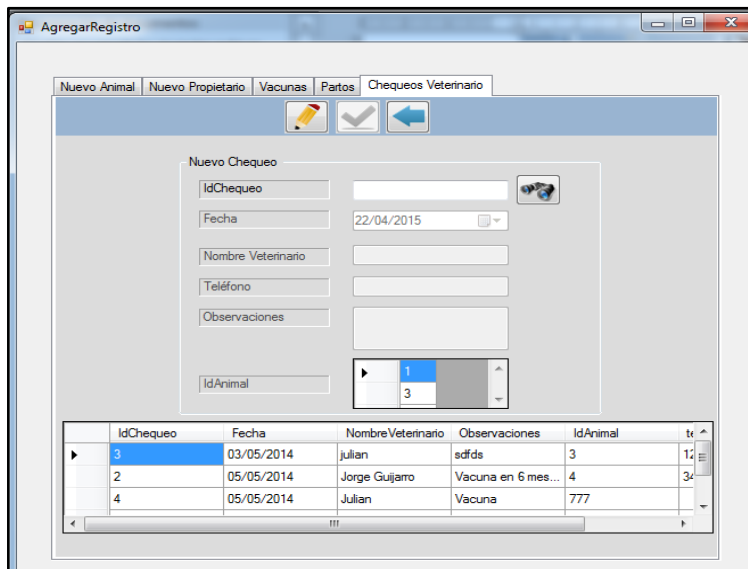
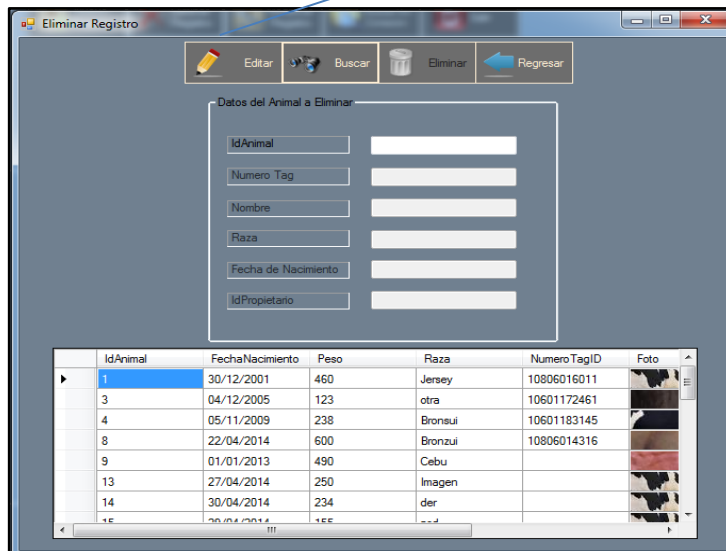


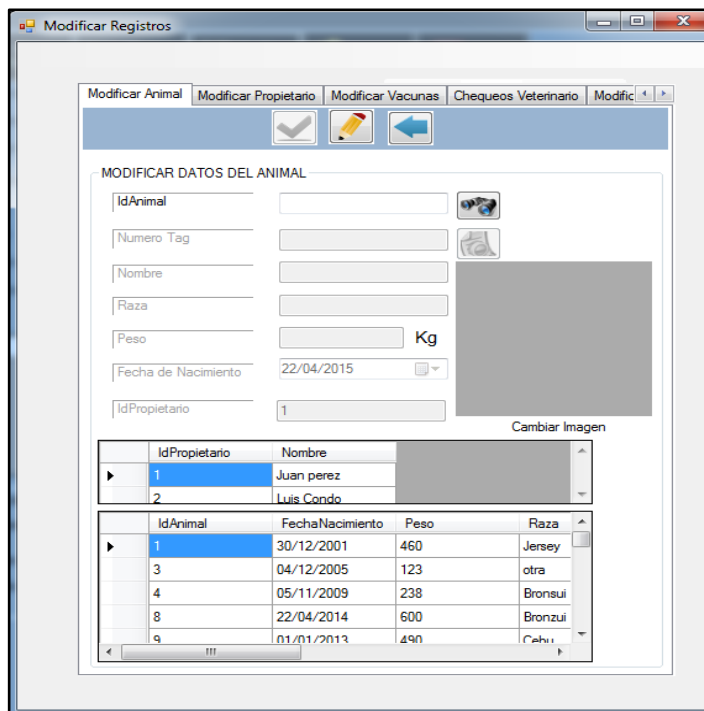
Figura 82 Ventana para Nuevo Registro Chequeos Veterinario  
Fuente: El Autor

**4.2 Eliminar Registro.-** Elimina todo un registro seleccionado; con tan solo ingresar el ID de identificación se realiza la búsqueda y esta ventana muestra si existe el registro a eliminar



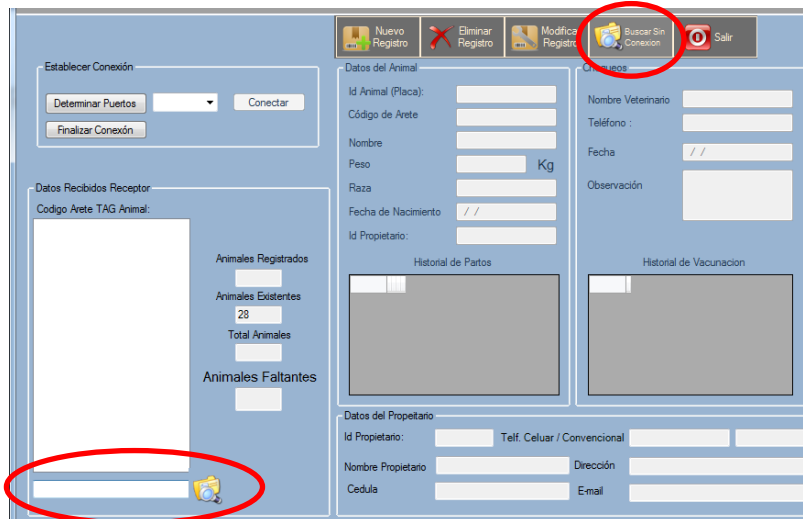
*Figura 83 Ventana para Eliminar Registro*  
Fuente: El Autor

**4.3 Modificar Registro.-** Realiza una modificación de los registros existentes



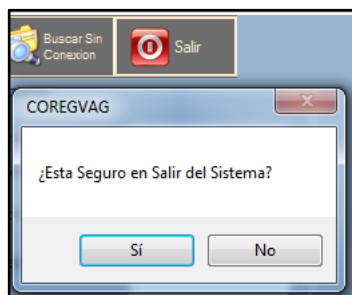
*Figura 84 Ventana para Modificar Registros*  
Fuente: El Autor

**4.4 Buscar sin Conexión.-** Establece una búsqueda sin utilizar el dispositivo receptor.

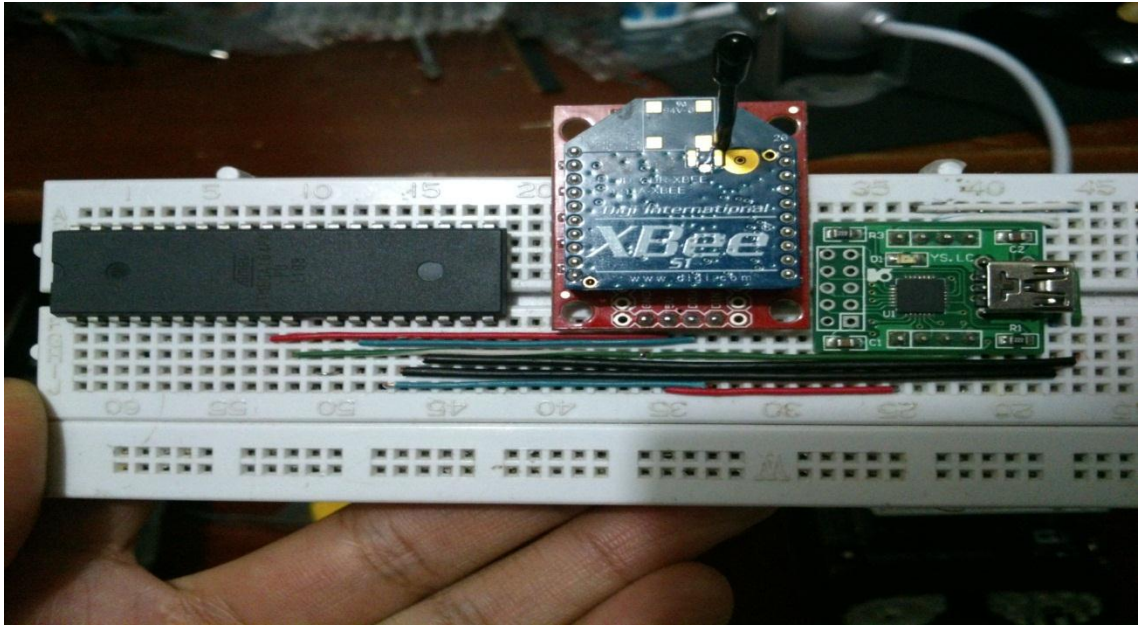


**Figura 85** Modo de búsqueda sin conexión  
Fuente: El Autor

**3.6 Salir .**Finaliza la aplicación y retorna al sistema operativo



**Figura 86** Finalizar Aplicación  
Fuente: El Autor



*Figura 87 Circuito Receptor en el protoboard  
Fuente El Autor*



*Figura 88 Montaje y soldadura de elementos en el circuito impreso del receptor  
Fuente: El Autor*

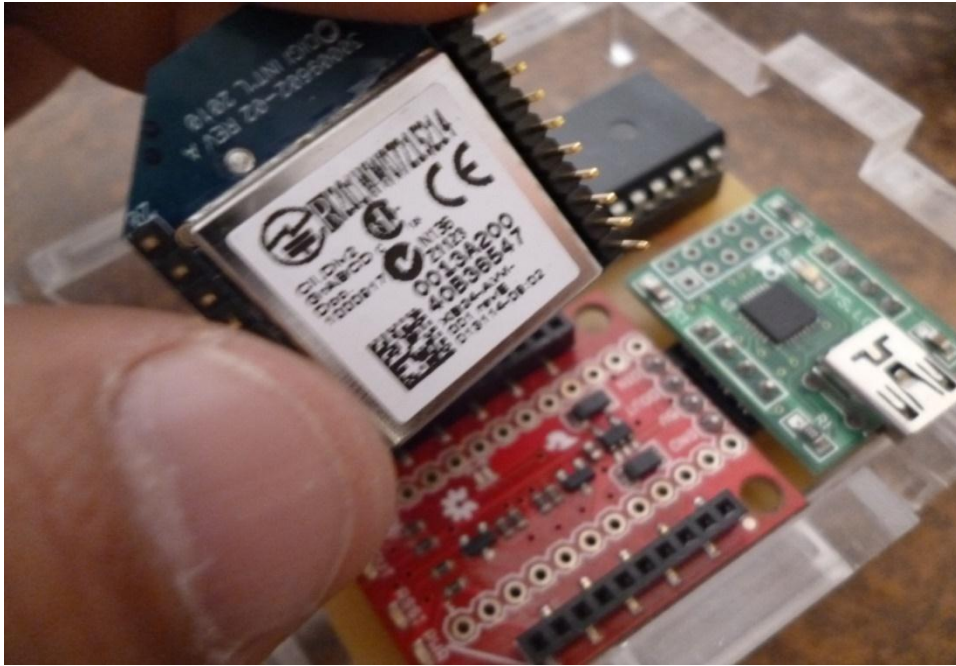




**Figura 89** Revisión suelda de elementos en el circuito impreso del receptor  
**Fuente :**El Autor



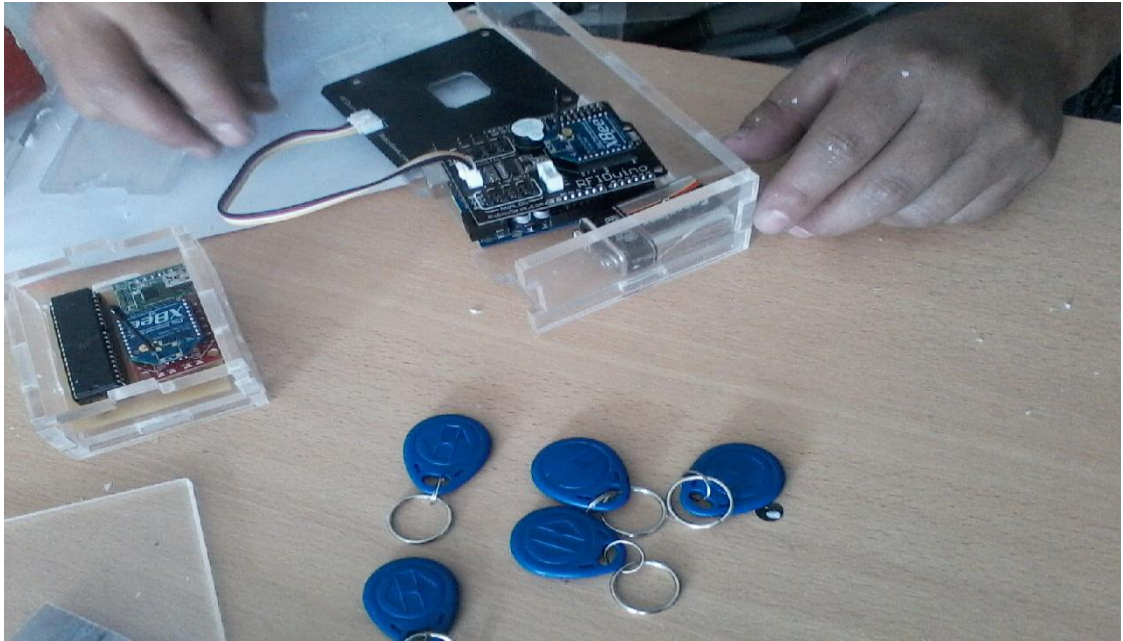
**Figura 90** Revisión final elementos en el circuito impreso del receptor  
**Fuente:** El Autor



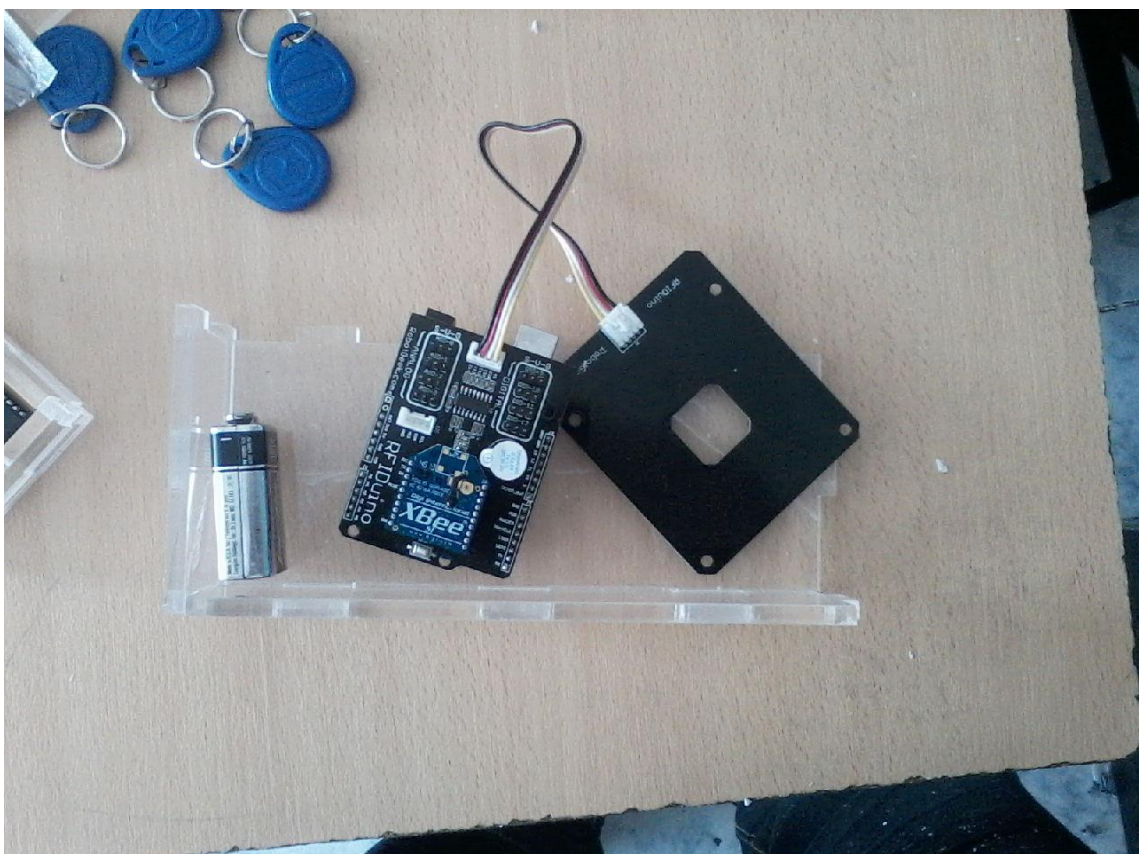
*Figura 91 Montaje módulo Xbee receptor en el módulo*  
*Fuente: El Autor*



*Figura 92 Montaje final del módulo de recepción*  
*Fuente: El Autor*



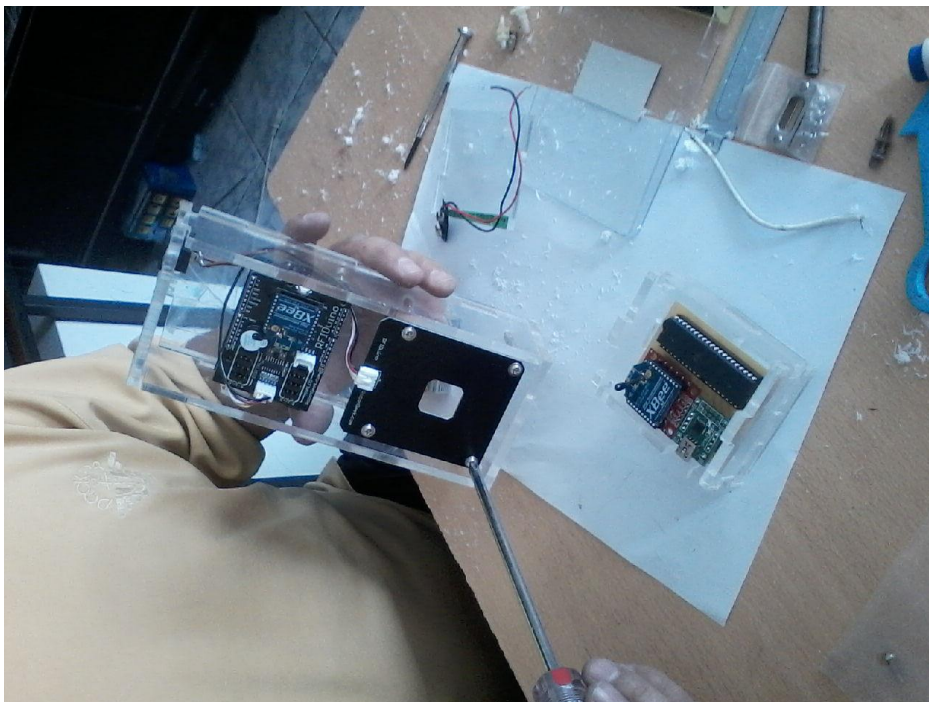
*Figura 93 Montaje Modulo de lectura y transición*  
*Fuente: El Autor*



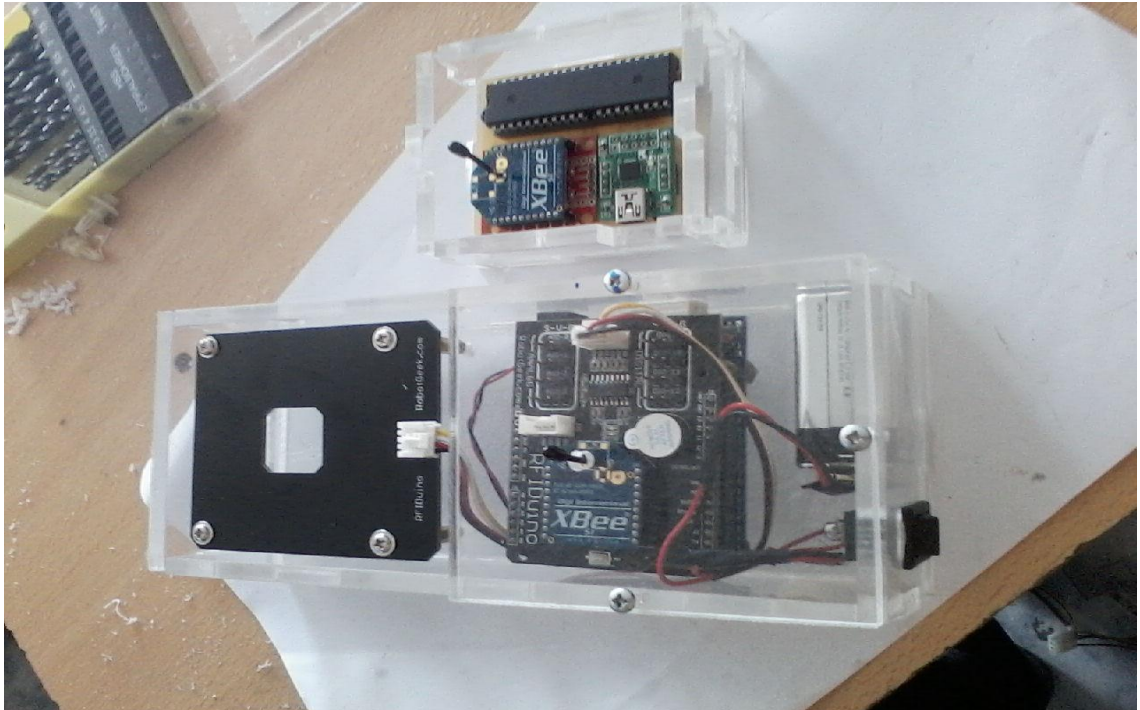
*Figura 94 95 Montaje alimentación Modulo de lectura y transición*  
*Fuente: El Autor*



**Figura 96** Montaje, cableado Modulo de lectura y transición  
**Fuente:** El Autor



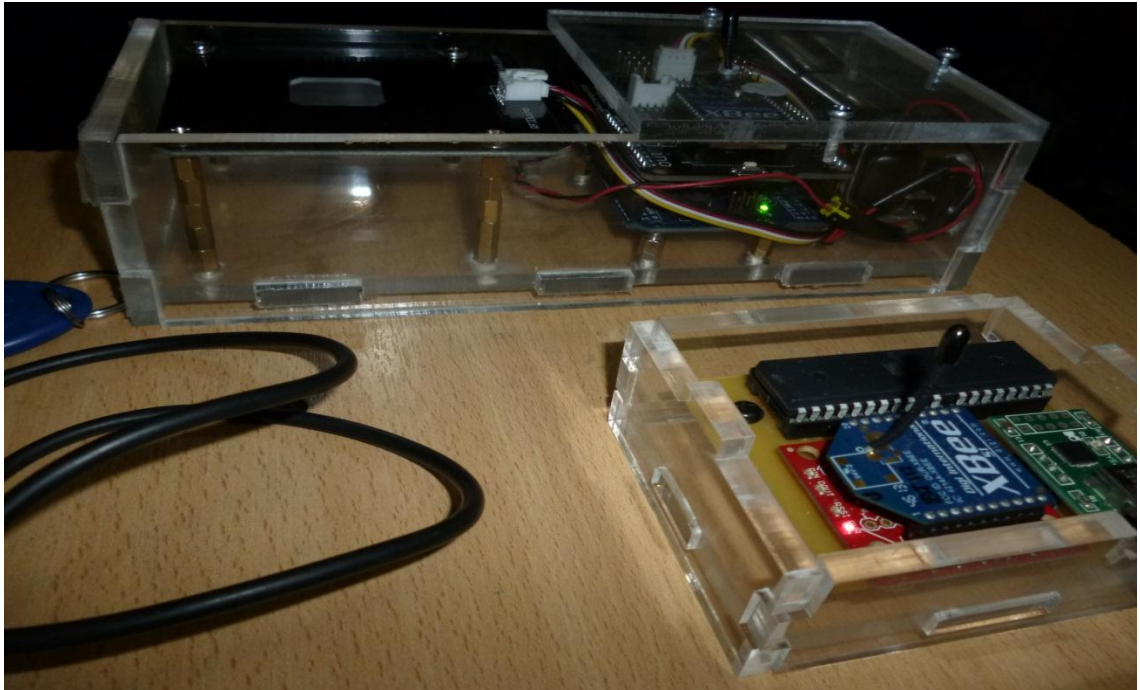
**Figura 97** Aseguramiento de la antena módulo de lectura y transición  
**Fuente:** El Autor



**Figura 98** Módulos de lectura/transmisión y recepción listos  
**Fuente:** El Autor



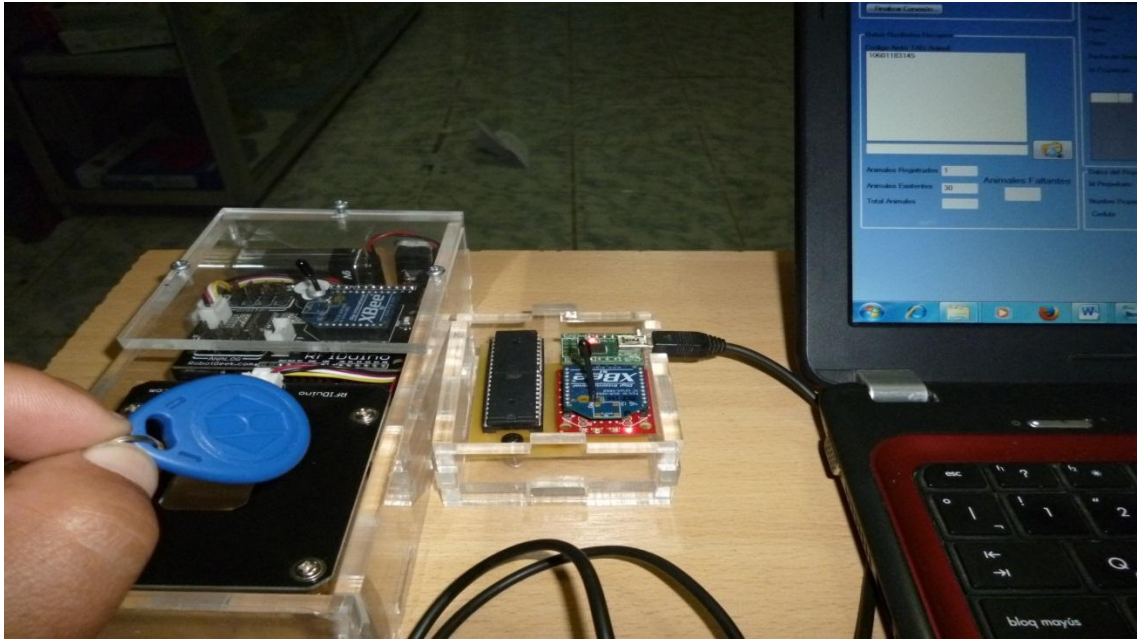
**Figura 99** Pruebas Modulo de Recepción  
**Fuente:** El Autor



*Figura 100 Módulos en operatividad*  
*Fuente: El Autor*



*Figura 101 Pruebas iniciales de lectura y transmisión*  
*Fuente: El Autor*



*Figura 102 Pruebas sistema completo*  
*Fuente: El Autor*



*Figura 103 Presentación de todo el sistema*  
*Fuente: El Autor*



**Figura 104** Animales del Hato ganadero  
**Fuente:** El Autor



**Figura 105** Zona de pastoreo  
**Fuente:** El Autor

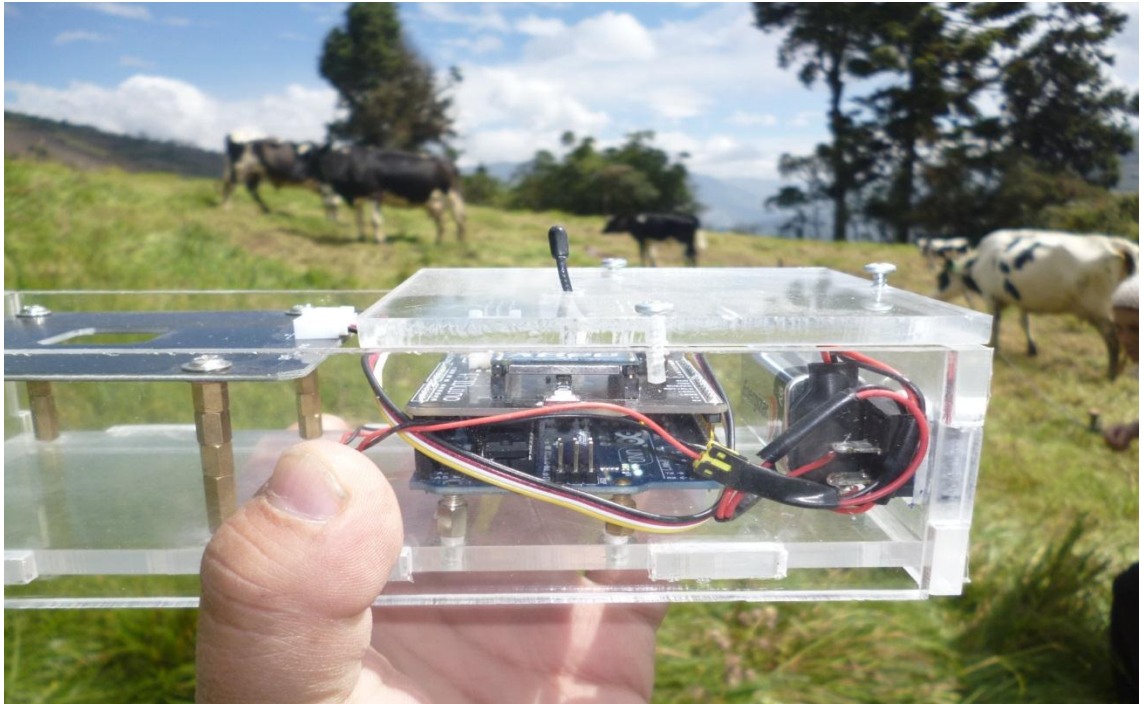




**Figura 106** Distribución de animales en la zona de pastoreo  
**Fuente:** El Autor



**Figura 107** Ubicación de la infraestructura  
**Fuente:** El Autor



**Figura 108** *Modulo de lectura y transición en la zona de pastoreo*  
**Fuente:** *El Autor*



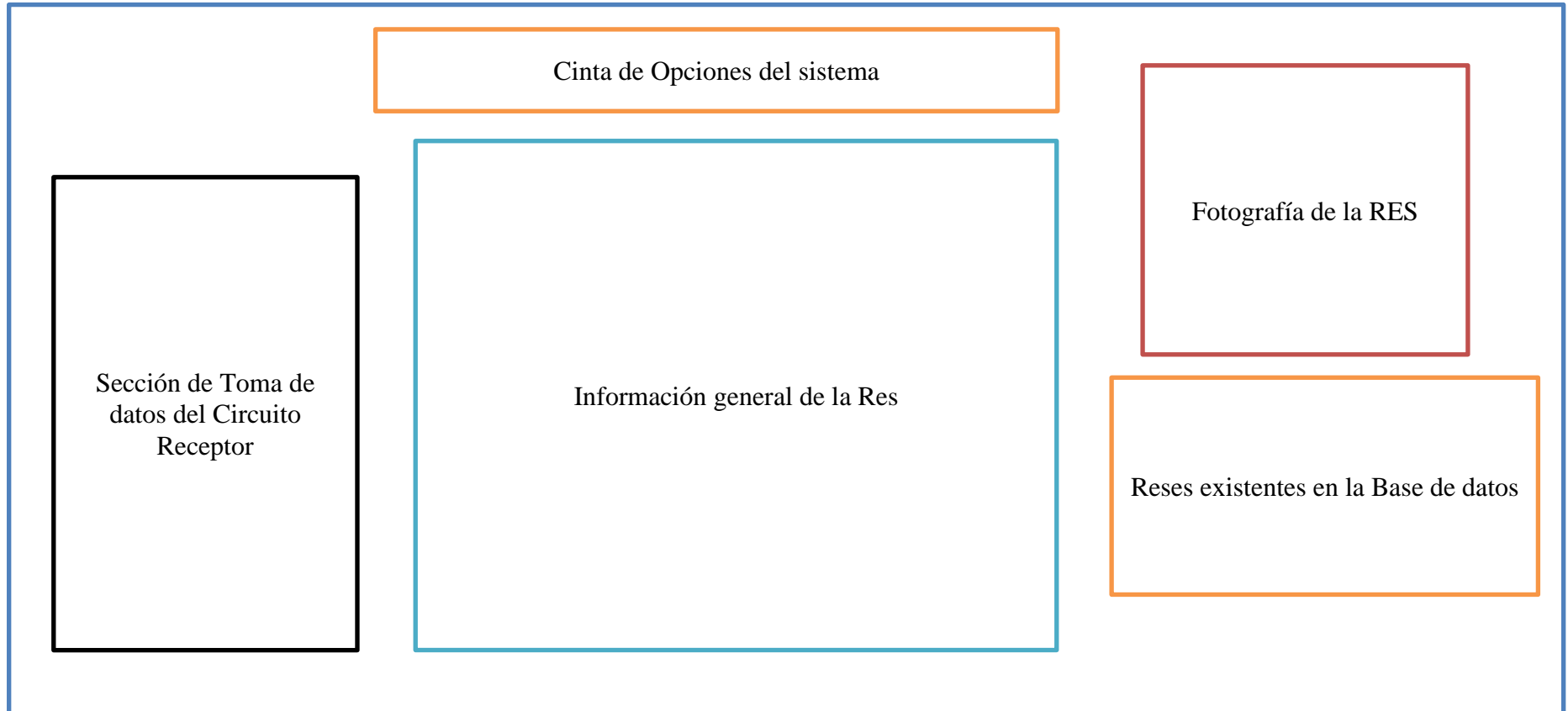
**Figura 109** *Prueba1 de lectura*  
**Fuente:** *El Autor*



**Figura 110** Prueba2 de lectura  
**Fuente:** El Autor

## BOSQUEJO DE LA INTERFAZ DEL SISTEMA

### Pantalla principal

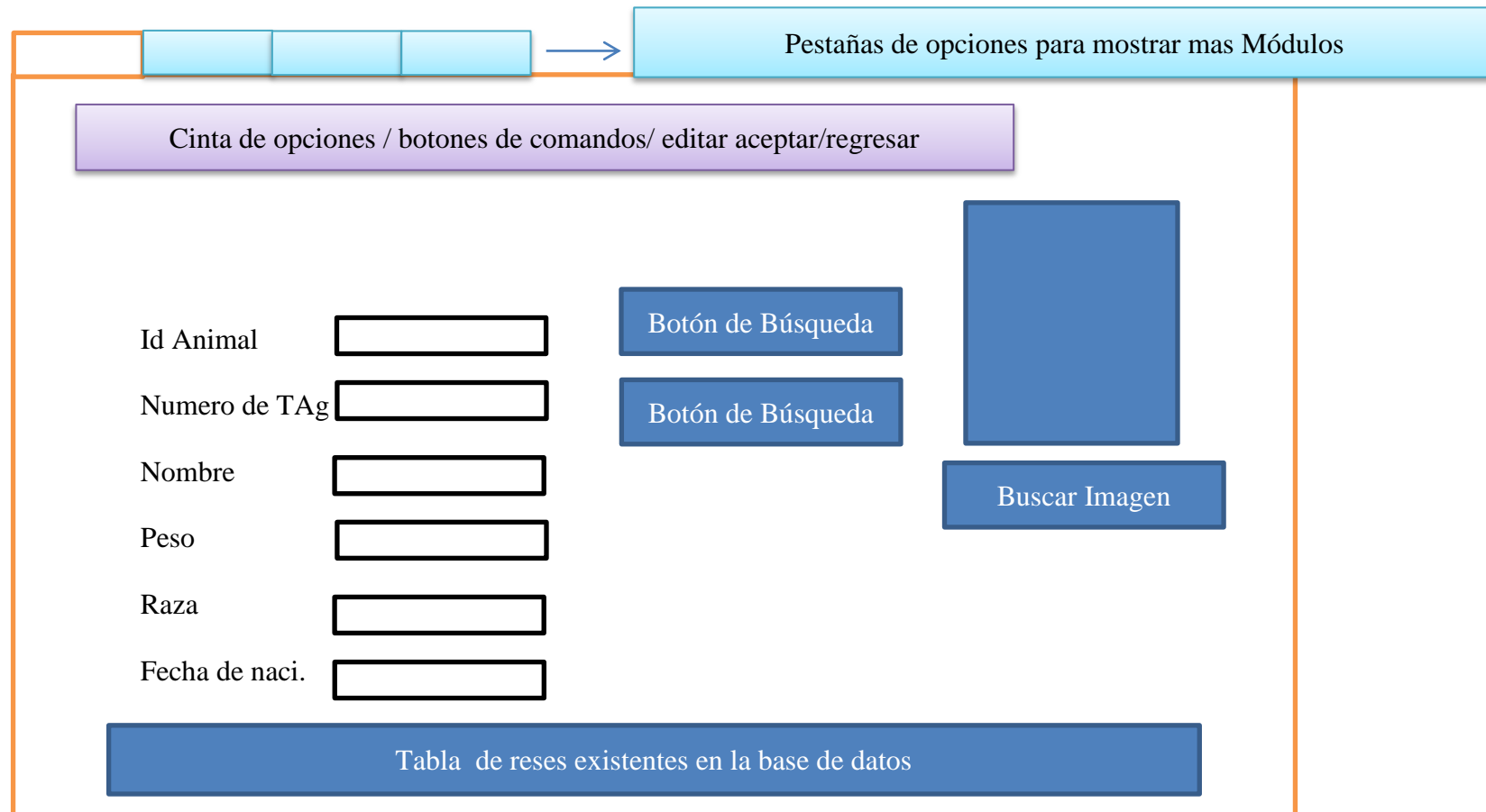


*Figura 111 Bosquejo De Interfaz principal del Sistema*

*Fuente: El Autor*

## BOSQUEJO DE INTERFAZ REGISTROS Agregar, Modificar

Para ingresar nueva RES



*Figura 112 Bosquejo De Interfaz Agregar Modificar*

*Fuente: El Autor*

## Eliminar Registro

The diagram illustrates the 'Eliminar Registro' (Delete Record) interface. It is enclosed in an orange border and contains the following elements:

- Cinta de opciones / botones de comandos/ editar /buscar/eliminar/regresar**: A purple ribbon at the top containing navigation and action buttons.
- Datos eliminar de la res**: A blue header bar for the data entry section.
- Form fields**: Six input fields for data entry, each with a label to its left:
  - Id Animal
  - Numero de TAg
  - Nombre
  - Peso
  - Raza
  - Fecha de naci.
- Tabla de reses existentes en la base de datos**: A large blue rectangular area at the bottom representing the data table.

**Figura 113** Bosquejo De Interfaz Eliminar Registro  
**Fuente:** El Autor

## CÓDIGO EN VISUAL

- PANTALLA PRINCIPAL INFO ANIMAL

```
Imports System.Data.OleDb 'permite el manejo de bases de datos
acces
Imports System.Data
Imports System.IO
Imports System.IO.Ports

Public Class infoanimal
    Dim varconex As New
OleDb.OleDbConnection(My.Settings.COREGVACConnectionString)
    Dim DS As New DataSet
    'variables del puerto serial
    Dim strbufferout As String
    Dim strbufferin As String
    'hastaqui

    Dim r1, r2, r3, con, i, xId As Integer
    Dim recibe As String = ""
    'limpia el formulario
    Private Sub limpiar_Formulario()

        'limpia datos del animal
        txtidanimal.Clear()
        Numero.Clear()
        Nombrr.Clear()
        Peso.Clear()
        Raza.Clear()
        mtb_FechaNacimiento.Clear()
        txtidpropietario.Clear()
        Me.PicPhoto1.Image = Nothing
        H.Refresh()
        H1.ClearSelection()

        'limpia chequeos
        NombreVeterinario.Clear()
        txttelevete.Clear()
        mtb_fecha.Clear()
        Observaciones.Clear()

        'limpia propietario
        Texnompropi.Clear()
        Texcedul.Clear()
        Texnucel.Clear()
        Texteconv.Clear()
        Texidpropie.Clear()
        Texdirecc.Clear()
        Texmail.Clear()

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Btn_Agergar.Click
```

```

        'declramos el objeto FAR = Formulario
        Dim FAR As New AgregarRegistro
        FAR.Show()

    End Sub

    Private Sub inha()
        gbdaosanimal.Enabled = False
        gbchequeos.Enabled = False
        gbdatosre.Enabled = False
    End Sub
    Private Sub habi()
        gbdaosanimal.Enabled = True
        gbchequeos.Enabled = True
        'gbdatosre.Enabled = True
    End Sub
    Private Sub infoanimal_Load(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles MyBase.Load

    Me.ChequeosVeterinarioTableAdapter.Fill(Me.COREGVACDataSet1.Cheque
    osVeterinario)

    Me.ChequeosVeterinarioTableAdapter.Fill(Me.COREGVACDataSet1.Cheque
    osVeterinario)
        Me.AnimalTableAdapter4.Fill(Me.COREGVACDataSet.Animal)
        logo.Visible = True

        ACCESO.Visible = False
        PicPhoto1.Image = Nothing
        limpiar_Formulario()
        DataGridView2.Enabled = False

        Obtenerdato()

        Me.AnimalTableAdapter3.Fill(Me.COREGVACDataSet1.Animal)
        inha()

        'LIMPIAR TODO LO DEL PUERTO SERIAL MIS VARIABLES
        strbufferin = ""
        strbufferout = ""
        Btn_Conectar.Enabled = False
        Timer1.Enabled = False
        'hASTAQUI

        'prueba de lectura de datos com
        '-----

        For Each PuertoSerial As String In
    My.Computer.Ports.SerialPortNames
            Cbpuertos.Items.Add(PuertoSerial)
        Next
        If Cbpuertos.Items.Count > 0 Then
            'Cbpuertos.Text = Cbpuertos.Items(0)
        Else
            'Cbpuertos.Enabled = False
        End If
        H1.DataSource = Nothing
        H1.Refresh()
        H.DataSource = Nothing

```



```

H.Refresh()

TextBox1.Text = Me.AnimalTableAdapter3.conteo()
Me.PartosTableAdapter.Fill(Me.COREGVACDataSet1.Partos)
TextBox6.Focus()
For Each item As Object In Me.ListBox1.Items
    Me.ListBox1.Items.Add(item)
Next
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
    Dim id1 As Integer
    'prueba para cargar imagen ---

    'buscar
    'TextBox6.Focus()
    If TextBox6.Text = "" Then
        MsgBox("DEBE ESCOGER UN DATO PARA AREALIZAR LA
CONSULTA", MsgBoxStyle.Exclamation, "SISTEMA COREGVAC")

        'infoanimal.Visible = False
    Else
        'prueba vizualizar foto

        Dim cn As New OleDb.OleDbConnection
        'cn.ConnectionString =
"Provider=Microsoft.Jet.OleDb.4.0; " & _
        ' " Data Source=" & Application.StartupPath &
"\data.mdb"
        cn.ConnectionString = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"

        cn.Open()

        'Dim arrImage() As Byte
        Dim myMS As New IO.MemoryStream
        Dim da As New OleDbDataAdapter("SELECT * FROM Animal
WHERE NumeroTagID = " & Me.TextBox6.Text, cn)
        Dim dt As New COREGVACDataSet1
        da.Fill(dt)

        Me.PicPhoto1.Image =
System.Drawing.Image.FromStream(myMS)
        ' End If
        'Me.btnAdd.Enabled = False
        ' Else
        MsgBox("Datos No encontrados!")
    End If

    '-----

    If Buscareg(txtidanimal.Text) = True Then
        'enviar mensaje
        MessageBox.Show("Registro Existente")
        id1 = Val(txtidpropietario.Text)
    End If
End Sub

```

```

        Buscaregpropietario(id1)
        If Buscaregl(txtidanimal.Text) = False Then
            'limpia chequeos
            NombreVeterinario.Clear()
            txttelefvete.Clear()
            mtb_fecha.Clear()
            Observaciones.Clear()
        End If
        TextBox6.Focus()
        'Buscaregpropietario(txtidpropietario.Text)
    Else
        MessageBox.Show("El registro no Existe")
        limpiar_Formulario()
        TextBox6.Focus()
    End If

End Sub

Function Buscaregpropietario(ByVal xIdp As String) As Boolean
    'convertir cadena en numero
    Dim Idp As Double
    Idp = Convert.ToInt32(xId)
    'conexion
    Dim Conexion As New OleDbConnection
    Conexion.ConnectionString = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"
    'cadena de conexion ... mensajero
    Dim cadena As String = "SELECT * FROM Propietario WHERE
IdPropietario = " & Val(txtidpropietario.Text)
    'Dim cadenal As String =
    'Adaptador
    Dim Adaptador As New OleDbDataAdapter(cadena, Conexion)
    'Data set .. esqueleto de la base de datos (contiene
registros campos etc
    Dim Ds As New DataSet
    'Llenar el data set... abrir la conexion
    Conexion.Open()
    Adaptador.Fill(Ds) 'llena con datos al data set
    Conexion.Close()
    'contar registro
    If (Ds.Tables(0).Rows.Count = 0) Then
        'NO hay registro
        Return False
    Else
        'Si hay registros
        'mostrar datos
        Texnompropi.Text =
Ds.Tables(0).Rows(0)("Nombre").ToString()
        Texcedul.Text =
Ds.Tables(0).Rows(0)("Cedula").ToString()
        Texnucel.Text =
Ds.Tables(0).Rows(0)("Celular").ToString()
        Texteconv.Text =
Ds.Tables(0).Rows(0)("Telefono").ToString()
        Texidpropie.Text =
Ds.Tables(0).Rows(0)("Idpropietario").ToString()
        Texdirecc.Text =
Ds.Tables(0).Rows(0)("Direccion").ToString()
    End If
End Function

```

```

        Texmail.Text =
Ds.Tables(0).Rows(0) ("Correo").ToString()

        '----
        Ds.Dispose()
        Return True

    End If

End Function

Function Buscareg(ByVal xIdtag As String) As Boolean
    'convertir cadena en numero
    Dim Idtag As Double
    Idtag = Convert.ToInt32(xIdtag)
    'conexion
    Dim Conexion As New OleDbConnection
    Conexion.ConnectionString = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"
    'cadena de conexion ... mensajero
    Dim cadena As String = "SELECT * FROM Animal WHERE
NumeroTagID = " & Idtag
    'Dim cadenal As String =
    'Adaptador
    Dim Adaptador As New OleDbDataAdapter(cadena, Conexion)
    'Data set .. esqueleto de la base de datos (contiene
registros campos etc
    Dim Ds As New DataSet
    'Llenar el data set... abrir la conexion
    Conexion.Open()
    Adaptador.Fill(Ds) 'llena con datos al data set
    Conexion.Close()
    'contar registro
    If (Ds.Tables(0).Rows.Count = 0) Then
        'NO hay registro
        Return False
    Else
        'Si hay registros
        'mostrar datos
        Nombr.Text = Ds.Tables(0).Rows(0) ("Nombre").ToString()
        mtb_FechaNacimiento.Text =
Ds.Tables(0).Rows(0) ("FechaNacimiento").ToString()
        Raza.Text = Ds.Tables(0).Rows(0) ("Raza").ToString()
        Numero.Text =
Ds.Tables(0).Rows(0) ("NumeroTagID").ToString()
        Peso.Text = Ds.Tables(0).Rows(0) ("Peso").ToString()
        txtidanimal.Text =
Ds.Tables(0).Rows(0) ("IdAnimal").ToString()
        txtidpropietario.Text =
Ds.Tables(0).Rows(0) ("IdPropietario").ToString()

        Ds.Dispose() 'liberar memoria del data set

        Return True

    End If

End Function

```

```

Function Buscareg1(ByVal xId As String) As Boolean
    Dim Id As Double
    Id = Convert.ToInt32(xId)
    Dim Conexion As New OleDbConnection
    Conexion.ConnectionString = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"
    Dim cadena As String = "SELECT * FROM ChequeosVeterinario
WHERE IdAnimal = " & Id
    Dim Adaptador As New OleDbDataAdapter(cadena, Conexion)
    Dim Ds As New DataSet
    Conexion.Open()
    Adaptador.Fill(Ds) 'llena con datos al data set
    Conexion.Close()
    If (Ds.Tables(0).Rows.Count = 0) Then
        'no hay registro
        Return False
    Else
        'Si hay registros
        'mostrar datos
        NombreVeterinario.Text =
Ds.Tables(0).Rows(0) ("NombreVeterinario").ToString()
        mtb_fecha.Text =
Ds.Tables(0).Rows(0) ("Fecha").ToString()
        Observaciones.Text =
Ds.Tables(0).Rows(0) ("Observaciones").ToString()
        txttelefvete.Text =
Ds.Tables(0).Rows(0) ("Telefono").ToString()
        Ds.Dispose()
        Return True
    End If
End Function

```

```

Function Buscareg2(ByVal xId As String) As Boolean
    'convertir cadena en numero
    Dim Id As Double
    Id = Convert.ToInt32(xId)
    'conexion
    Dim Conexion As New OleDbConnection
    Conexion.ConnectionString = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"
    'cadena de conexion ... mensajero
    Dim cadena As String = "SELECT * FROM Vacunas WHERE
IdAnimal = " & Id
    'Dim cadenal As String =
'Adaptador
    Dim Adaptador As New OleDbDataAdapter(cadena, Conexion)

    'Data set .. esqueleto de la base de datos (contiene
registros campos etc
    Dim Ds As New DataSet
    'Llenar el data set... abrir la conexion
    Conexion.Open()
    Adaptador.Fill(Ds) 'llena con datos al data set
    Conexion.Close()

    'contar registro
    If (Ds.Tables(0).Rows.Count = 0) Then

```

```

        'no hay registro
        Return False
    Else
        Ds.Dispose()
        Return True
    End If
End Function

'funcion parra llenar las grillas
Function Buscareg5(ByVal xId As String) As Boolean
    Dim Id As Double
    Id = Convert.ToInt32(xId)
    Dim connectionString As String = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"
    Dim ocom As New OleDb.OleDbConnection(connectionString)
    Dim ds As New DataSet
    Dim OleComand As New OleDbCommand
    Dim da As New OleDbDataAdapter
    da.SelectCommand = OleComand
    da.SelectCommand.Connection = ocom
    '-----
    OleComand.CommandText = "SELECT IdAnimal,
FechaVacunacion, NombreVacuna FROM Vacunas WHERE IdAnimal=" & Id
'Consulta SQL
    da.Fill(ds, "Vacunas") 'Llena el Dataset...
    H.DataSource = ds.Tables("Vacunas") 'Muestra el contenido
del Dataset dentro de la grilla
End Function

Function Buscareg6(ByVal xId As String) As Boolean
    Dim Id As Double
    Id = Convert.ToInt32(xId)
    Dim connectionString As String = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"
    Dim ocom As New OleDb.OleDbConnection(connectionString)
    Dim ds As New DataSet
    Dim OleComand As New OleDbCommand
    Dim da As New OleDbDataAdapter
    da.SelectCommand = OleComand
    da.SelectCommand.Connection = ocom
    '-----
    OleComand.CommandText = "SELECT IdAnimal, FechaParto,
Sexo FROM Partos WHERE IdAnimal=" & Id 'Consulta SQL
    da.Fill(ds, "Partos")
    H1.DataSource = ds.Tables("Partos")

End Function

Private Sub Button6_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Salir.Click
    ' MsgBox("SEGURO QUE DESEA SALIR!",
MsgBoxStyle.DefaultButton3, "SISTEMA COREGVAC")
    Dim resp As DialogResult = MessageBox.Show("¿Esta Seguro
en Salir del Sistema?", " COREGVAG ", MessageBoxButtons.YesNo)
    'evalua el si o el no del usuario
    If resp = Windows.Forms.DialogResult.Yes Then
        End
    End

```

```

Else
    'no... no elimina el registro
    'But_Eliminar.Enabled = False
    'Limpiar_formulario()
    'Tex_Ide.Focus()
End If

End Sub
Sub Generar()
    Dim i As Integer
    For i = 0 To ListBox1.Items.Count - 1
        con = i + 1
    Next
    TextBox5.Text = con
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button3.Click

    Dim i As Integer
    For i = 0 To ListBox1.Items.Count - 1
        'for j = ListBox1.Items.Count - 1
        If TextBox14.Text = ListBox1.Items(i).ToString Then
            MsgBox("YA ESTA REGISTRADO ESTE DATO",
MsgBoxStyle.Exclamation, "SISTEMA COREGVAC")
            ListBox1.Items.Remove(i)
            Control.CheckForIllegalCrossThreadCalls = False
            TextBox14.Clear()
            Generar()
            'ElseIf TextBox4.Text <> ListBox1.Items.ToString
Then
                'ListBox1.Items.Add(TextBox4.Text)
            End If

        Next
        If TextBox14.Text = "" Then
            MsgBox("NO EXISTEN DATOS", MsgBoxStyle.Exclamation,
"SISTEMA COREGVAC")
            'ListBox1.Items.Clear()
            Generar()
        ElseIf TextBox14.Text <> "" Then
            'Else If Val(
            ListBox1.Items.Add(TextBox14.Text)
            Control.CheckForIllegalCrossThreadCalls = False
            TextBox14.Clear()
            Generar()
        End If

        'If TextBox4.Text = ListBox1.Item Then

    End Sub

Private Sub ListBox1_MouseClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
ListBox1.MouseClick
    TextBox6.Capture = True
    TextBox6.Text = ListBox1.SelectedItem

```

```

End Sub
Private Sub Obtenerdato()

    'obtener el numerode animales existentes en la base de
datos
    Dim connection As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\base.accdb;Persist Security Info=False;")
    Dim tabla As New DataTable
    connection.Close()
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button4.Click
    'Dim w As Integer
DataGridView2.Enabled = True
habi()
'cerrar puerto
Btn_Conectar.Text = "Conectar"
'btnenviardnabled = False
txtbuferin.Enabled = False
Txtbufferout.Enabled = False
Timer1.Enabled = False
SerialPort1.Close()
'hastaqui
Button3.Enabled = False
TextBox14.Clear()
TextBox14.Focus()
TextBox14.Enabled = False

If Val(TextBox5.Text) > Val(TextBox1.Text) Then
    MsgBox("NO PUEDEN EXISTIR MAS ANIMALES QUE LOS
REGISTRADOS", MsgBoxStyle.Exclamation, "SISTEMA COREGVAC")
    TextBox3.Text = ""
    TextBox2.Text = ""
End If

If Val(TextBox5.Text) < Val(TextBox1.Text) Then
    'MsgBox("RECEPCIÓN FINALIZADA",
MsgBoxStyle.Information, "SISTEMA COREGVAC")
    MsgBox("HAY ANIMALES FALTANTES EN LA ZONA DE
PASTORAJE", MsgBoxStyle.Information, "SISTEMA COREGVAC")
    r1 = Val(TextBox1.Text) - Val(TextBox5.Text)

    TextBox2.BackColor = Color.Red

    TextBox2.Text = r1
    TextBox3.Text = Val(TextBox5.Text)
End If
If Val(TextBox2.Text) = 0 Then
    MsgBox("REGISTRO SATISFACTORIO NO FALTAN ANIMALES",
MsgBoxStyle.Information, "SISTEMA COREGVAC")
    TextBox2.Text = "NINGUNO"

    'If TextBox2.Text = "" Then

End If
TextBox3.Text = Val(TextBox5.Text)
'End If

```

```

End Sub

Private Sub Borrar_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Borrar.Click
    Dim FER As New Eliminar
    FER.Show()
End Sub

Private Sub AgregarToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AgregarToolStripMenuItem.Click
    'declramos el objeto FAR = Formulario
    Dim FAR As New AgregarRegistro
    FAR.Show()
End Sub

Private Sub ModificarToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ModificarToolStripMenuItem.Click
    'declramos el objeto FAR = Formulario
    Dim FMR As New Modificar
    FMR.Show()
End Sub

Private Sub EliminarToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
EliminarToolStripMenuItem.Click
    'declramos el objeto FER = formulario registro
    Dim FER As New Eliminar
    FER.Show()
End Sub

Private Sub Actualizar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Actualizar.Click
    'declramos el objeto FAR = Formulario
    Dim FMR As New Modificar
    FMR.Show()
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Cbpuertos.SelectedIndexChanged

End Sub
'manejo puerto serial
Private Sub Btn_Conectar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Btn_Conectar.Click
    'para conectarse al puerto
    If Btn_Conectar.Text = "Conectar" Then
        SerialPort1.PortName = Cbpuertos.Text
        Btn_Conectar.Text = "Desconectar"
        gbdatosre.Enabled = True
        'btnenviardato.Enabled = True
        txtbufferin.Enabled = True
        Txtbufferout.Enabled = True
        Timer1.Enabled = True
        SerialPort1.Open()

```



```

        'ElseIf Btn_Conectar.Text = "desconectar" Then
    Else : Btn_Conectar.Text = "Conectar"
        'btnenviarnabled = False
        txtbuferin.Enabled = False
        Txtbufferout.Enabled = False
        Timer1.Enabled = False
        SerialPort1.Close()

    End If
    'hastaqui

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Timer1.Tick
    'otraprueba
    strbufferin = SerialPort1.ReadExisting
    If strbufferin <> "" Then
        'ListBox1.Items.Add(strbufferin)
        txtbuferin.Text = strbufferin
        'ListBox1.Items .
        strbufferin = ""
        SerialPort1.DiscardInBuffer()

    End If

End Sub

Private Sub Btn_determinarconexion_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn_determinarconexion.Click
    gbdatosre.Enabled = False
    TextBox6.Clear()
    Cbpuertos.Items.Clear()
    Button4.Enabled = True
    For Each puertodisponible As String In
My.Computer.Ports.SerialPortNames
        Cbpuertos.Items.Add(puertodisponible)
    Next
    If Cbpuertos.Items.Count > 0 Then
        Cbpuertos.Text = Cbpuertos.Items(0)
        MessageBox.Show("Puertos COM Encontrados SELECCIONE UN
PUERTO")
        Btn_Conectar.Enabled = True
    Else
        MessageBox.Show("Ningun Puerto Seleccionado ")
        Btn_Conectar.Enabled = False
        'btnenviarnabled = False
        Cbpuertos.Items.Clear()
        Cbpuertos.Text = ""

    End If
End Sub

Private Sub Txtbufferout_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Txtbufferout.TextChanged
    SerialPort1.DiscardOutBuffer()
    strbufferout = Txtbufferout.Text
    SerialPort1.Write(strbufferout)

```

```

End Sub

Private Sub txtbuferin_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtbuferin.TextChanged
    Dim i As Integer

    For i = 0 To ListBox1.Items.Count - 1
        'for j = ListBox1.Items.Count - 1
        If txtbuferin.Text = ListBox1.Items(i).ToString Then
            'MsgBox("YA ESTA REGISTRADO ESTE DATO",
MsgBoxStyle.Exclamation, "SISTEMA COREGVAC")
            ListBox1.Items.Remove(i)
            Control.CheckForIllegalCrossThreadCalls = False
            txtbuferin.Clear()
            Generar()
            'ElseIf TextBox4.Text <> ListBox1.Items.ToString
Then
                'ListBox1.Items.Add(TextBox4.Text)
            End If

        Next
        If txtbuferin.Text = "" Then
            'MsgBox("NO EXISTEN DATOS", MsgBoxStyle.Exclamation,
"SISTEMA COREGVAC")
            'ListBox1.Items.Clear()
            Generar()
        ElseIf txtbuferin.Text <> "" Then
            'Else If Val(
            ListBox1.Items.Add(txtbuferin.Text)
            Control.CheckForIllegalCrossThreadCalls = False
            txtbuferin.Clear()
            Generar()
        End If

    End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
    Dim FAp As New agregarprueba
    FAp.Show()
End Sub

Private Sub Btn_actualizarinfo_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Btn_mostrar.Click
    'Obtenerdato()
    TextBox1.Text = Me.AnimalTableAdapter3.conteo()
    Me.AnimalTableAdapter3.Fill(Me.COREGVACDataSet1.Animal)
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button5.Click
    Dim arrImage() As Byte
    Dim myMS As New IO.MemoryStream
    Dim con As New OleDbConnection

```

```

        con.ConnectionString = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"
        con.Open()
        '-----
        'declaro una variable para almacenar el nombre del campo
        Dim ncampo As String
        Try
            If con.State = ConnectionState.Closed Then
                con.Open()
            End If

            ncampo = "NumeroTagID"

            If TextBox6.Text = "" Then
                MsgBox("DEBE ESCOGER UN DATO PARA AREALIZAR LA
CONSULTA", MsgBoxStyle.Exclamation, "SISTEMA COREGVAC")

                'infoanimal.Visible = False
            Else
                Dim datos As New OleDbDataAdapter("SELECT * FROM
Animal WHERE NumeroTagID = '" & Val(TextBox6.Text) & "'", con)
                Dim dt As New DataTable
                datos.Fill(dt)
                Me.DataGridView1.DataSource = dt

                If dt.Rows.Count > 0 Then
                    logo.Visible = False
                    logol.Visible = False
                    H1.Visible = True
                    H.Visible = True
                    MsgBox("Registro Existente!")
                    DataGridView2.Enabled = True
                    Me.txtidanimal.Text =
dt.Rows(0).Item("IdAnimal") & ""
                    Me.Numero.Text =
dt.Rows(0).Item("NumeroTagID") & ""
                    Me.Nombr.Text = dt.Rows(0).Item("Nombre") & ""
                    Me.Peso.Text = dt.Rows(0).Item("Peso") & ""
                    Me.Raza.Text = dt.Rows(0).Item("raza") & ""
                    Me.mtb_FechaNacimiento.Text =
dt.Rows(0).Item("FechaNacimiento") & ""
                    Me.txtidpropietario.Text =
dt.Rows(0).Item("idPropietario") & ""

                    If Not IsDBNull(dt.Rows(0).Item("Foto")) Then
                        arrImage = dt.Rows(0).Item("Foto")
                        For Each ar As Byte In arrImage
                            myMS.WriteByte(ar)
                        Next
                        Buscareg6(txtidanimal.Text)
                        Buscareg5(txtidanimal.Text)
                        Buscareg1(txtidanimal.Text)
                        Buscaregpropietario(txtidpropietario.Text)
                        'Buscareg2(txtidanimal.Text)
                        Me.PicPhoto1.Image =
System.Drawing.Image.FromStream(myMS)
                    End If
                    'Me.btnAdd.Enabled = False
                End If
            End If
        End Try
    End Sub

```

```

        Else
            logo.Visible = True
            logo1.Visible = True
            MsgBox("Registro no Encontrado!")
            limpiar_Formulario()
            TextBox6.Focus()
            TextBox6.Clear()
            H1.Visible = False
            H.Visible = False
        End If
    End If
    con.Close()
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try
End Sub

Private Sub TextBox6_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox6.TextChanged
    Dim vIn As Decimal = 0.0
    Dim vOut As String = vIn.ToString(vIn)
    'Dim v As d
End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SalirToolStripMenuItem.Click

    Dim resp As DialogResult = MessageBox.Show("¿Esta Seguro
en Salir del Sistema?", " COREGVAG ", MessageBoxButtons.YesNo)
    'evalua el si o el no del usuario
    If resp = Windows.Forms.DialogResult.Yes Then
        End

    Else

    End If
End Sub

Private Sub AcercaDeToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AcercaDeToolStripMenuItem.Click
    'declramos el objeto FAR = Formulario
    Dim FAR As New Form3
    FAR.Show()

End Sub

Private Sub Button6_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button6.Click
    gbdatosre.Enabled = True
    TextBox6.Focus()
    TextBox6.ReadOnly = False
    TextBox6.Visible = True
    TextBox6.Enabled = True
    DataGridView2.Enabled = True
End Sub

```

```

    Private Sub Button7_Click_1 (ByVal sender As System.Object,
ByVal e As System.EventArgs)
        ListBox1.Items.Clear()
        TextBox6.Clear()
    End Sub

End Class

```

- **AGREGAR REGISTROS**

```

Imports System.Data.OleDb 'permite el manejo de bases de datos
acces
Imports System.Data
Imports System.IO
Public Class AgregarRegistro

    Private Sub AgregarRegistro_Load (ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Me.AnimalTableAdapter1.Fill (Me.COREGVACDataSet.Animal)

Me.PropietarioTableAdapter1.Fill (Me.COREGVACDataSet.Propietario)
    Me.PartosTableAdapter.Fill (Me.CoregvacDataSet1.Partos)

Me.ChequeosVeterinarioTableAdapter.Fill (Me.CoregvacDataSet1.Cheque
osVeterinario)
    Me.VacunasTableAdapter.Fill (Me.CoregvacDataSet1.Vacunas)

Me.PropietarioTableAdapter.Fill (Me.CoregvacDataSet1.Propietario)

Me.PropietarioTableAdapter.Fill (Me.CoregvacDataSet1.Propietario)

Me.AnimalTableAdapter.Fill (Me.CoregvacDataSet1.Animal)

        Dim Cnn As New OleDb.OleDbConnection()
        'conexion
        Cnn.ConnectionString = "Provider =
Microsoft.ACE.OLEDB.12.0;Data Source =
C:\\base\\COREGVAC.accdb;Persist Security Info=False"

        Cnn.Open()
        MsgBox ("Connection Establecida con la base de datos!")

        Me.AnimalTableAdapter.Fill (Me.CoregvacDataSet1.Animal)
        'TODO: esta línea de código carga datos en la tabla
'COREGVACDataSet1.Propietario' Puede moverla o quitarla según sea
necesario.
        picphoto.Image = Nothing
        Interface_Entrada()
        Interface_Entradaprop()
        Interface_Entradavac()
        Interface_Entradapartos()
        Interface_Entradavet()
    End Sub

    Private Sub Interface_Entrada()

```

```
'Se ejecuta al cargar el formulario
'habilitados
Lab_Idanimal.Enabled = True
Tex_Id.Enabled = True
But_Verificar.Enabled = True
But_Guardar.Enabled = True
Lab_Idpropietario.Enabled = True

'desabilitados animal
Lblnumerotag.Enabled = False
Tex_Numerotag.Enabled = False
Lab_FechaNacimiento.Enabled = False
dtfechanacimiento.Enabled = False
Lab_raza.Enabled = False
Tex_Raza.Enabled = False
lblpeso.Enabled = False
Tex_Peso.Enabled = False
Lab_Nombre.Enabled = False
Tex_Nombrea.Enabled = False
But_Guardar.Enabled = False
LblIdpropietario.Enabled = False
tex_Idpropietario.Enabled = False
btn_Verificartag.Enabled = False
End Sub
Private Sub Interface_Entradaprop()
'Se ejecuta al cargar el formulario
'habilitados
Lab_Idpropietario.Enabled = True
Tex_propietarioid.Enabled = True
Butverificarp.Enabled = True

Butguardar.Enabled = False
LblNombrepro.Enabled = False
Tex_Nombrepropietario.Enabled = False
Lab_cedula.Enabled = False
Tex_cedula.Enabled = False
Lab_direccion.Enabled = False
Tex_direccion.Enabled = False
Lab_correo.Enabled = False
Tex_correo.Enabled = False
Lab_celular.Enabled = False
Tex_celular.Enabled = False

End Sub
Private Sub Interface_Entradavac()
'Se ejecuta al cargar el formulario
'habilitados
lbl_idvacunas.Enabled = True
Tex_idvacunas.Enabled = True
Butverificarvac.Enabled = True
But_guardarvac.Enabled = False
lbl_fechavacunacion.Enabled = False
dt_fechadevacunacion.Enabled = False
Lbl_nombrevacuna.Enabled = False
Tex_NombreVacuna.Enabled = False
lbl_idanimalvac.Enabled = False
Tex_idanimalvac.Enabled = False

End Sub
```

```
Private Sub Interface_Entradapartos()  
    'Se ejecuta al cargar el formulario  
    'habilitados  
    Lab_idpartos.Enabled = True  
    Tex_idpartos.Enabled = True  
    But_verificarp.Enabled = True  
    But_guaradp.Enabled = False  
    Lab_fechaparto.Enabled = False  
    Dt_fechaparto.Enabled = False  
    Lab_sexo.Enabled = False  
    Comb_sexo.Enabled = False  
    Lab_idanimalp.Enabled = False  
    Tex_idanimalp.Enabled = False  
  
End Sub  
Private Sub Interface_Entradavet()  
    'Se ejecuta al cargar el formulario  
    'habilitados  
    Lab_idchequeo.Enabled = True  
    Tex_idchequeo.Enabled = True  
    But_buscarv.Enabled = True  
    But_guardarv.Enabled = False  
    Lab_fechachequeo.Enabled = False  
    Dt_fechachequeo.Enabled = False  
    Lab_nombrevete.Enabled = False  
    Tex_nombrevete.Enabled = False  
    Lab_telefvete.Enabled = False  
    Tex_telefvete.Enabled = False  
    Lab_observaciones.Enabled = False  
    Tex_observaciones.Enabled = False  
    Lab_idanimalvete.Enabled = False  
    Tex_idanimalvete.Enabled = False  
  
End Sub  
  
Private Sub Interface_Datos()  
    'Se ejecuta al ingresar nuevos datos  
    'deshabilitados  
    Lab_Idanimal.Enabled = False  
    Tex_Id.Enabled = False  
    Lblnumerotag.Enabled = False  
    Tex_Numerotag.Enabled = False  
    But_Verificar.Enabled = False  
    But_Guardar.Enabled = False  
    LblIdpropietario.Enabled = False  
    tex_Idpropietario.Enabled = False  
    'mtb_FechaNacimiento.Enabled = False  
    lblpeso.Enabled = False  
    Tex_Peso.Enabled = False  
    'habilitados animal  
    Lblnumerotag.Enabled = True  
    Tex_Numerotag.Enabled = True  
    Lab_FechaNacimiento.Enabled = True  
    dtfechanacimiento.Enabled = True  
  
    'mtb_FechaNacimiento.Enabled = True  
    Lab_raza.Enabled = True  
    Tex_Raza.Enabled = True
```

```
Lab_Nombre.Enabled = True
Tex_Nombrea.Enabled = True
But_Guardar.Enabled = True
LblIdpropietario.Enabled = True
tex_Idpropietario.Enabled = True
lblpeso.Enabled = True
Tex_Peso.Enabled = True

End Sub
Private Sub Interface_Datosprop()
    'Se ejecuta al ingresar nuevos datos
    'deshabilitados
    Lab_Idpropietario.Enabled = False
    Tex_propietarioid.Enabled = False
    Butverificarp.Enabled = False

    Butguardar.Enabled = True
    LblNombrepro.Enabled = True
    Tex_Nombrepropietario.Enabled = True
    Lab_cedula.Enabled = True
    Tex_cedula.Enabled = True
    Lab_direccion.Enabled = True
    Tex_direccion.Enabled = True
    Lab_correo.Enabled = True
    Tex_correo.Enabled = True
    Lab_celular.Enabled = True
    Tex_celular.Enabled = True

End Sub
Private Sub Interface_Datosvac()
    'Se ejecuta al ingresar nuevos datos
    'deshabilitados
    lbl_idvacunas.Enabled = False
    Tex_idvacunas.Enabled = False
    Butverificarvac.Enabled = True
    But_guardarvac.Enabled = True
    lbl_fechavacunacion.Enabled = True
    dt_fechadevacunacion.Enabled = True
    Lbl_nombrevacuna.Enabled = True
    Tex_NombreVacuna.Enabled = True
    lbl_idanimalvac.Enabled = True
    Tex_idanimalvac.Enabled = True

End Sub
Private Sub Interface_Datospartos()
    'Se ejecuta al ingresar nuevos datos
    'deshabilitados
    Lab_idpartos.Enabled = False
    Tex_idpartos.Enabled = False
    But_verificarp.Enabled = False
    But_guaradp.Enabled = True
    Lab_fechaparto.Enabled = True
    Dt_fechaparto.Enabled = True
    Lab_sexo.Enabled = True
    Comb_sexo.Enabled = True
    Lab_idanimalp.Enabled = True
    Tex_idanimalp.Enabled = True

End Sub
```



```

Private Sub Interface_Datosvet ()
    'Se ejecuta al ingresar nuevos datos
    'deshabilitados
    Lab_idchequeo.Enabled = False
    Tex_idchequeo.Enabled = False
    But_buscarv.Enabled = False
    But_guardarv.Enabled = True
    Lab_fechachequeo.Enabled = True
    Dt_fechachequeo.Enabled = True
    Lab_nombrevete.Enabled = True
    Tex_nombrevete.Enabled = True
    Lab_telefvete.Enabled = True
    Tex_telefvete.Enabled = True
    Lab_observaciones.Enabled = True
    Tex_observaciones.Enabled = True
    Lab_idanimalvete.Enabled = True
    Tex_idanimalvete.Enabled = True

End Sub

'Verificar Registros
Private Sub But_Verificar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles But_Verificar.Click

    'Interface_Datos ()

    If Tex_Id.Text = "" Then
        MessageBox.Show("Debe ingresar un valor..")
        Limpiar_formulario()
        Interface_Entrada()
        Tex_Id.Focus()

    ElseIf Buscareg(Tex_Id.Text) = True Then
        MessageBox.Show("El registro ya existe")
        Tex_Id.Focus()
        Limpiar_formulario()
        Interface_Entrada()

        'But_Guardar.Enabled = False
    Else
        MessageBox.Show("ingrese los nuevos valores")
        'Interface_Datos ()

        Lab_Idanimal.Enabled = False
        Tex_Id.Enabled = False
        Tex_Numerotag.Focus()
        Lblnumerotag.Enabled = True
        Tex_Numerotag.Enabled = True
        But_Verificar.Enabled = False
        btn_Verificartag.Enabled = True
        PropietarioDataGridView1.Enabled = True

    End If
    'End If
    'para la foto -----

```

