



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

CARRERA DE ANÁLISIS DE SISTEMAS

MEJORAMIENTO DEL PROCESO DE GESTIÓN DE FICHAS MÉDICAS
VETERINARIAS MEDIANTE UN APLICATIVO WEB PARA LA CLÍNICA
VETERINARIA HUSKY

Proyecto de investigación previo a la obtención del título de:

Tecnólogo en Análisis de Sistemas

Autor: Damián Benjamín Navarrete Chávez

Tutor: Lic. Wilson Núñez

Quito, Octubre 2015

DECLARATORIA DE AUTORÍA

Declaro que los contenidos de este Trabajo de Titulación previo a la obtención del grado son absolutamente originales, auténticos, personales, las ideas, doctrinas, resultados y conclusiones a los que he llegado son de mi absoluta y exclusiva responsabilidad.

Damián Benjamín Navarrete Chávez

CI. 1719458075

CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, la estudiante **NAVARRETE CHÁVEZ DAMIÁN BENJAMÍN**, por sus propios y personales derechos, a quien en lo posterior se le denominará el “**CEDENTE**”; y, por otra parte, el **INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA**, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el “**CESIONARIO**”. Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “**MEJORAMIENTO DEL PROCESO DE GESTIÓN DE FICHAS MÉDICAS VETERINARIAS MEDIANTE UN APLICATIVO WEB PARA LA CLÍNICA VETERINARIA HUSKY**”, el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. b) Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

SEGUNDA: CESIÓN Y TRANSFERENCIA.- Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

TERCERA: OBLIGACIÓN DEL CEDENTE.- El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

CUARTA: CUANTIA.- La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

QUINTA: PLAZO.- La vigencia del presente contrato es indefinida.

SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.- Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvención, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

SÉPTIMA: ACEPTACIÓN.- Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los 26 días del mes de Octubre del dos mil quince.

f) _____

C.I. 1719458075

CEDENTE

f) _____

Instituto Tecnológico Superior Cordillera

CESIONARIO

AGRADECIMIENTOS

Primeramente agradezco a Dios por darme la vida y la oportunidad de estudiar sustentando a mis padres en todo lo necesario para llegar a donde estoy. Y mi más sincero agradecimiento al Ing. Hugo Heredia y al Lic. Wilson Núñez por haberme asesorado y orientado profesionalmente y de la mejor manera, aportando con sus conocimientos y críticas no solo a lo largo del desarrollo del proyecto, sino en todo el transcurso de este semestre que estoy a punto de culminar,.

Gracias a todos mis maestros quienes han sido mi ejemplo a seguir, a mis compañeros de clase, quienes han sido mi apoyo, a mis amigos y a mi novia que es una gran persona, a todos quienes intervinieron para que esto sea posible. ¡Gracias!

DEDICATORIA

Quiero dedicar este proyecto, resultado de un gran esfuerzo y perseverancia en toda mi carrera, a mis padres Anabel Chávez y Jorge Navarrete, que desde el inicio de mi etapa estudiantil me han apoyado incondicionalmente, dándome aliento para sobresalir de los problemas que se me han presentado, esforzándose para educarme de la mejor manera desde muy pequeño para ser un gran ser humano, llevándome por el camino correcto para ser una persona ejemplar, lo cual me ha permitido llegar hasta esta nueva y crucial etapa de mi vida

Índice General

Contenido	Páginas
DECLARACIÓN DE APROBACIÓN DE TUTOR Y LECTOR.....	¡Error! Marcador no definido.
DECLARATORIA DE AUTORÍA	i
CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL.....	ii
AGRADECIMIENTOS	vi
DEDICATORIA	vii
Resumen Ejecutivo	xvi
Abstract.....	xvii
Introducción	xviii
Capítulo I. Antecedentes.....	1
1.01. Contexto.....	1
1.01. Justificación	2
7.02. Definición del problema central.....	3
Capítulo II. Análisis de involucrados.....	7
2.01. Requerimientos	7
2.01.1 Descripción del sistema actual.....	7
2.01.2 Visión y Alcance	8
Visión.....	8
Alcance	8
2.01.3 Entrevistas.....	9
2.01.4 Matriz de Requerimientos	10
2.01.5 Descripción Detallada.....	12
2.02. Mapeo de Involucrados.....	18

2.03. Matriz de Involucrados	19
Capítulo III. Problemas Y Objetivos.....	20
3.01. Árbol de Problemas.....	20
3.02. Árbol de Objetivos	21
3.03. Diagramas de Casos de Uso.....	22
3.04. Casos de uso de Realización	23
3.05. Diagramas de Secuencia del Sistema.....	27
3.05. Especificación de Casos de Uso.....	31
Capítulo IV. Análisis de Alternativas	34
4.01. Matriz de Análisis de Alternativas.....	34
4.02. Matriz de Impactos de Objetivos.....	34
4.03. Estándares para el Diseño de Clases.	35
4.04. Diagrama de clases.	38
4.05. Modelo Lógico – Físico	40
4.06. Diagrama de Componentes	42
4.07. Diagramas de Estrategias	43
4.08. Matriz de Marco Lógico.	43
4.09. Vistas arquitectónicas	43
4.01.01 Vista Lógica	43
4.01.02 Vista Física.....	44
4.01.03 Vista De Desarrollo.....	45
4.01.04 Vista De Procesos.	46
Capítulo V: Propuesta.....	47
5.01. Especificación de estándares de programación.....	47
5.01.01 Microsoft Visual Studio 2013.	47

5.01.02 ASP.NET	47
5.01.02 MySQL	48
5.02. Diseño de interfaces de usuario.	53
5.03. Especificación de pruebas de unidad	61
5.04. Especificación de pruebas de aceptación	63
5.05. Especificación de pruebas de carga	65
5.06. Configuración del ambiente mínima/ideal	67
Capítulo VI. Aspectos Administrativos.	68
6.01. Recursos.	68
6.02. Presupuesto.	70
6.03. Cronograma.....	71
Capítulo VII. Conclusiones y Recomendaciones.....	72
7.01. Conclusiones.	72
7.02. Recomendaciones.	72
Anexo A.01 Matriz de Involucrados.....	73
Anexo A.02 Matriz de análisis de alternativas.....	74
Anexo A.03 Matriz de impacto de objetivos	75
Anexo A.04 Matriz de análisis de estrategias	77
Anexo A.05 Matriz de marco lógico.....	78
Anexo A.06 Detalle de Estándares de programación.....	81
Anexo A.07 Descripción de configuración de ambiente mínima/ideal	82
Anexo A.08. Manual De Instalación.....	83
Anexo A.09 Manual Técnico	98
Anexo A.10 Manual de Usuario	162
Anexo A.11 Webgrafía	174

Índice de Tablas

Contenido	Páginas
Tabla 1. Matriz de fuerzas T	3
Tabla 2. Matriz de preguntas.....	9
Tabla 3. Requerimientos Funcionales	10
Tabla 4. Requerimientos No Funcionales	11
Tabla 5. Requerimiento funcional 001 (RF001)	11
Tabla 6. Requerimiento funcional 002 (RF002)	13
Tabla 7. Requerimiento funcional 003 (RF003)	14
Tabla 8. Requerimiento no funcional 004 (NRF004)	15
Tabla 9. Requerimiento no funcional 005 (NRF005)	16
Tabla 10. Requerimiento no funcional (NRF006)	17
Tabla 11. Especificación de caso de uso de realización 001.....	23
Tabla 12. Especificación de caso de uso de realización 002.....	24
Tabla 13. Especificación de caso de uso de realización 003.....	25
Tabla 14. Especificación de caso de uso de realización 004.....	26
Tabla 15. Especificación de Caso de Uso 001.	31
Tabla 16. Especificación de Caso de Uso 002.	31
Tabla 17. Especificación de Caso de Uso 003.	32
Tabla 18. Especificación de Caso de Uso 004.	33
Tabla 19. Ingreso al sistema.....	53
Tabla 20. Vista Principal.....	54
Tabla 21. Ingreso de Clientes	54

Tabla 22. Ingreso de Pacientes.....	55
Tabla 23. Asignación de fecha y horario de cita	56
Tabla 24. Menú de Consultas.....	57
Tabla 25. Vista de nueva consulta médica con cita previa.....	58
Tabla 26. Lista de Pacientes.....	59
Tabla 27. Especificación de prueba de unidad PU001, guardar un registro.	61
Tabla 28. Especificación de prueba de unidad PU002, Seleccionar un registro.....	61
Tabla 29. Especificación de prueba de unidad PU003, Actualizar un registro.....	62
Tabla 30. Especificación de prueba de unidad PU004, Eliminar un registro.....	62
Tabla 31. Especificación de prueba de unidad PU005, Buscar un registro.	63
Tabla 32. Especificación de prueba de aceptación PA001.....	63
Tabla 33. Especificación de prueba de aceptación PA002.....	64
Tabla 34. Especificación de prueba de aceptación PA003.....	64
Tabla 35. Especificación de prueba de aceptación PA004.....	65
Tabla 36. Especificación de prueba de carga PC001.	66
Tabla 37. Especificación de prueba de carga PC002.	66
Tabla 38. Especificación de prueba de carga PC003.	67
Tabla 39. Detalle de Recursos Humanos.	68
Tabla 40. Detalle de recursos materiales utilizados	68
Tabla 41. Detalle de Recursos a nivel de Hardware.	69
Tabla 42. Detalle de Recursos a nivel de Software.....	69
Tabla 43. Detalle del presupuesto y gastos incluidos	70

Índice de Figuras

Contenido	Páginas
<i>Figura 1. Mapeo de Involucrados.....</i>	18
<i>Figura 2. Árbol de problemas</i>	20
<i>Figura 3. Árbol de Objetivos.</i>	21
<i>Figura 4. Diagramas de casos de uso.....</i>	22
<i>Figura 5. Caso de uso de realización 001</i>	23
<i>Figura 6. Caso de uso de realización del proceso de anamnesis.</i>	24
<i>Figura 7. Caso de uso de realización de tratamientos de vacunas.</i>	25
<i>Figura 8. Caso de uso de realización de próximas citas.</i>	26
<i>Figura 9. Diagrama de secuencia de nueva ficha médica.....</i>	27
<i>Figura 10. Diagrama de Secuencia de anamnesis del paciente.</i>	28
<i>Figura 11. Diagrama de secuencia de vacunación.</i>	29
<i>Figura 12. Diagrama de Secuencia de próxima cita</i>	30
<i>Figura 13. Clase Mascota.....</i>	35
<i>Figura 14. Atributos de la Clase Mascota</i>	36
<i>Figura 15. Métodos de la Clase Mascota (Insertar, Modificar, Eliminar, Buscar, Listar).....</i>	36
<i>Figura 16. Clase Cliente</i>	37
<i>Figura17. Atributos</i>	37
<i>Figura 18. Métodos de la clase</i>	38
<i>Figura 19. Diagrama de Clases de la solución</i>	39
<i>Figura 20. Modelo Lógico-Físico del Sistema.....</i>	41

<i>Figura 21. Diagrama de Componentes Del Sistema</i>	<i>42</i>
<i>Figura 22. Vista Lógica Del Sistema desarrollado en Rational Rose.</i>	<i>44</i>
<i>Figura 23. Vista Física..</i>	<i>44</i>
<i>Figura 24. Vista De Desarrollo..</i>	<i>45</i>
<i>Figura 25. Vista de Procesos..</i>	<i>46</i>
<i>Figura 26. Importación de la Librería Mysql.Data para la conexión a la base de datos</i>	<i>48</i>
<i>Figura 27. Conexión a la base de datos..</i>	<i>49</i>
<i>Figura 28. Consulta de datos de usuario para ingresar al sistema declarando las variables pertinentes y obteniendo la conexión.</i>	<i>49</i>
<i>Figura 29. Consulta realizada a la base de datos..</i>	<i>49</i>
<i>Figura 30. Método insertar.....</i>	<i>50</i>
<i>Figura 31. Declaración de variables y Validación de campos.....</i>	<i>50</i>
<i>Figura 32. Código de Insertar..</i>	<i>51</i>
<i>Figura 33. Código de la sentencia Try y Catch.</i>	<i>51</i>
<i>Figura 34. Envío de datos a otro formulario.....</i>	<i>51</i>
<i>Figura 35. Recepción de datos desde otro formulario..</i>	<i>52</i>
<i>Figura 36. Código utilizado en css para el diseño del aplicativo</i>	<i>52</i>
<i>Figura 37. Vista del Login del aplicativo.</i>	<i>53</i>
<i>Figura 38. Vista de la página de inicio del aplicativo.....</i>	<i>54</i>
<i>Figura 39. Vista de registro de Clientes.</i>	<i>55</i>
<i>Figura 40. Vista de registro de Pacientes.....</i>	<i>56</i>
<i>Figura 41. Vista de la pantalla de asignación de citas previas.....</i>	<i>57</i>
<i>Figura 42. Vista del menú de consultas.</i>	<i>58</i>

<i>Figura 43. Vista de Nueva consulta médica.</i>	<i>59</i>
<i>Figura 44. Vista de Búsqueda de pacientes y Fichas médicas..</i>	<i>60</i>
<i>Figura 45. Vista de la Ficha médica.....</i>	<i>60</i>
<i>Figura 46. Vista del cronograma planificado para el proyecto realizado en Microsoft Visio 2010</i>	<i>71</i>

Resumen Ejecutivo

El software de gestión de fichas médicas veterinarias es un proyecto planificado para gestionar de manera innovadora los procesos de atención médica veterinaria dentro del consultorio veterinario HUSKY en la ciudad de Quito, ya que gracias a ello esta empresa obtendrá más accesos y facilidades para gestionar su negocio y manejar su empresa, hará que la atención brindada a sus pacientes sea la más adecuada y ágil, satisfaciendo las necesidades del cliente al cien por ciento.

El software que se ha propuesto desarrollar como proyecto de titulación brindará al usuario las características y herramientas más óptimas para evitar errores al momento de llevar la gestión de atención médica veterinaria, ya que se la realiza de manera manual, con la ayuda de este software se obviará y se corregirá por completo esta desventaja del usuario.

Lo que se propone realizar con el software es unificar y centralizar la información y la historia clínica del paciente en un sistema capaz de llevar varios registros a la vez, con múltiples opciones para el usuario, por medio de las cuales podrá acceder a la información que requiera.

Para su desarrollo se utilizó la plataforma de desarrollo Microsoft Visual Studio 2013 conectado a una base de datos que en este caso es MySQL Workbench.

El proyecto está desarrollado en base a una capa de acceso a datos, la misma que es separada por la capa de presentación al usuario.

Abstract

The software of management of veterinary medical records is a xviiccessxvii planned to manage in an innovative manner the processes of veterinary medical care within the HUSKY veterinary clinic in the city of Quito, since thanks to this business will get more xviiccess and facilities for managing your business and manage your business, will make the care provided to their patients the most appropriate and agile, meeting the needs of the customer to a hundred per cent.

The software that has been proposed to develop such as titling xviiccessxvii will provide the user with the features and xviicces best suited to prevent errors when the management of veterinary medical care in that it is done manually, with the help of this software will be bypassed and will be corrected by complete this disadvantage of the user.

What is going to be done with the software is to standardize and centralize the information and the lane clinic the patient in a system capable of carrying multiple records at once, with multiple options for the user, by means of which you can xviiccess the information they require.

For your development we used the development platform Microsoft Visual Studio 2013 connected to a database which is in this case MySQL Workbench.

The xviiccessxvii is developed on the basis of a data xviiccess layer, the same which is separated by the presentation layer to the user.

INTRODUCCIÓN

El presente proyecto se basa en el estudio de la situación actual que tiene la organización, sus atencedentes, su funcionamiento, etc. Se implementa un diseño metodológico ya que se pretende presentar y describir lo que se observa en la realidad social.

El empleo de sistemas informáticos basados en conocimiento toma lugar en industrias, hospitales, oficinas y muchos otros sitios. Su orientación es la automatización del análisis de problemas, la búsqueda de soluciones, la toma de decisiones y el empleo de conocimiento especializado en un campo específico de aplicación. Entre los productos más significativos de los Sistemas basados en Conocimiento se encuentran los Sistemas Expertos, los cuales están encargados de representar el conocimiento de los especialistas de una rama en la procura de su aprovechamiento para tareas de diagnóstico, enseñanza y control.

En el proyecto desarrollado a continuación, se especifican las características de un sistema desarrollado a tres capas, con los módulos necesarios para poder gestionar de manera prolija los procesos que maneja el especialista para dar diagnósticos, y emitir recomendaciones sobre la atención que debe recibir el paciente.

En la primera parte de este proyecto de tesis, se dará a conocer la situación actual, definiendo las razones que originan la necesidad de contar con una herramienta tecnológica que proporcione elementos para el diagnóstico y recetario realizados a los pacientes, con esto se define el problema central y su respectivo análisis.

En los siguientes capítulos se analizarán los objetivos que impulsarán el desarrollo, identificación de involucrados, el diseño del sistema, estándares utilizados para la programación y métodos, para finalizar con los aspectos administrativos.

Capítulo I. Antecedentes

1.01. Contexto

Las empresas que brindan la atención veterinaria necesaria a sus clientes y mascotas ofrecen cuidado veterinario a cualquier especie de animal doméstico, aplicando los tratamientos adecuados y óptimos para la prevención de enfermedades que se puedan presentar a lo largo de su vida.

La sociedad en general necesita de clínicas especializadas en la veterinaria y zootecnia para la atención a toda clase de animales domésticos, para conservar el ecosistema.

En el Ecuador existen varias instituciones veterinarias que ofrecen su servicio a las comunidades, pero muchas de ellas no poseen de un software ni de un hardware adecuado para manejar y gestionar los procesos médicos determinados, es decir la información que se administra en todas las clínicas y sucursales de las mismas se mantiene de manera descentralizada y manual.

Hoy en día se tiene la necesidad de optar por una metodología adecuada, con el único fin de optimizar las funcionalidades y procesos en general siendo muy útil para la humanidad aventajando la monitorización y gestión de información clínica que se podrá llevar a cabo de una manera más eficaz reduciendo el trabajo manual y ahorrando tareas repetitivas que se puedan agilizar con un nuevo sistema que reemplace todos aquellos procesos obsoletos.

1.01. Justificación

El presente proyecto se orienta al desarrollo de un aplicativo informático en ambiente web para la gestión de la información médica de los animales a lo largo de su vida, lo cual servirá de beneficio para ellos como pacientes, así como para los médicos veterinarios tratantes, permitiéndoles acceder a la información actualizada y disponible en cualquier momento, con el fin de prevenir la posterior pérdida u olvido de la misma y evitar que la historia clínica del paciente se encuentre desorganizada.

El compromiso es mejorar la calidad de atención que prestan los profesionales de la salud de la clínica veterinaria HUSKY brindando una atención eficiente, rápida y segura, obviando la pérdida de tiempo registrando los mismos datos en cada contacto con el especialista veterinario, como ocurre en la actualidad en la mayoría de instituciones.

“La administración de la información que cada entidad médica tiene, origina una duplicación de tiempo y funciones que se pueden optimizar si esta información estuviera disponible para cualquier entidad médica que requiera de la misma en una situación emergente o no”. (Vásquez J.,2012)

De esta forma se busca que todo animal, o mascota paciente puedan tener al alcance su información médica a través de su propietario al momento de consultarlo por cualquier profesional, cuando el paciente padece de alguna enfermedad infectocontagiosa, de alergia o alguna enfermedad terminal, etc.

7.02. Definición del problema central

MATRIZ DE FUERZAS

En la matriz de fuerzas T se realizará la identificación de las fuerzas que impulsan y bloquean la solución del problema central o global el cual es la inadecuada gestión de fichas médicas veterinarias, relacionadas con el proyecto.

Tabla 1

Matriz de fuerzas T

SITUACIÓN EMPEORADA	SITUACIÓN ACTUAL				SITUACIÓN MEJORADA
Duplicidad de documentos médicos de los pacientes tratados	Inadecuada Gestión de fichas médicas veterinarias				Adecuado proceso de gestión de fichas médicas través de herramientas adecuadas y eficaces. Del consultorio veterinario Husky. (Marzo 2015-Octubre-2015)
FUERZAS IMPULSADORAS	I	PC	I	PC	FUERZAS BLOQUEADORAS
Mejorar los procesos de tratamientos veterinarios a través de una aplicación web, que será necesaria para el usuario	3	5	5	1	No cuenta con una aplicación informática eficaz que controle y gestione los procesos necesarios a través del usuario
Socialización de los procesos de tratamiento veterinario que se da a los pacientes	4	5	4	3	Desconocimiento de los procesos de tratamiento veterinario de los pacientes.
Facilitar a los profesionales de la salud que colaboran con el proyecto una herramienta que facilite el registro y el seguimiento de la gestión realizada a las mascotas.	3	5	5	3	Optar por herramientas inadecuadas ocasionando una operación ineficiente de los procesos de tratamiento y atención
Registrar los resultados obtenidos en cada tratamiento, resguardando los datos de cada paciente que asiste mediante el software	2	4	4	2	La obtención de resultados registrados manualmente en cada tratamiento que se aplica, genera inseguridad y desconfiabilidad

NOTA: Determinación de las fuerzas bloqueadoras e impulsadoras

PC: Potencial de Cambio

I: Intensidad

Nota: Análisis de fuerzas impulsadoras y bloqueadoras

Análisis de Fuerzas Impulsadoras

FI 1: Mejorar los procesos de tratamientos veterinarios a través de una aplicación web, que será necesaria para el usuario

I=3: Tiene una escala valorativa de tres ya que si no existe una actualización de información de los tratamientos que se aplica al paciente pueden generar errores en los registro médicos.

PC=5: Tiene una escala valorativa de cinco debido a que si se mantiene una actualización de los tratamientos aplicados, no existirá errores en la entrega de diagnósticos de cada paciente. El potencial de cambio se vería muy notorio en el funcionamiento del software.

FI 2: Socialización de los procesos de gestión y tratamiento veterinario

I=4: Si no existe socialización de los procesos de gestión administrativa, no se podrá conocer con exactitud los tratamientos que se deben llevar a cabo para cada paciente y se desconocerá de los procesos de gestión adecuados.

PC=5: El socializar adecuadamente procesos esenciales de gestión dentro de la veterinaria HUSKY facilitara de gran manera el llevar acabo procesos de organización con respecto a la gestión veterinaria.

FI 3: Facilitar a los profesionales de la salud que colaboran con el proyecto una herramienta que facilite el registro y el seguimiento de la gestión realizada a las mascotas.

I=3: Si no se desarrollaría un software con herramientas necesarias para realizar este tipo de gestiones, los veterinarios tratantes de la clínica HUSKY operarían de manera manual los procesos más fundamentales que se llevan a cabo, retrasando el tiempo en vez de optimizarlo.

$PC=5$: implementar un sistema eficiente brindaría una ayuda muy útil a los profesionales de la clínica veterinaria HUSKY ya que facilitaría los procesos que son repetitivos llevándolo a un sistema informático.

FI 4: Registrar los resultados obtenidos en cada tratamiento, resguardando los datos de cada paciente que asiste mediante el software

$I=2$: Tiene una escala valorativa de 2, porque si existiera un registro de resultados por cada tratamiento no se manejaría un control ordenado de cada tratamiento.

$PC=4$: Tiene una escala valorativa de 4, debido a que al registrar cada resultado de cada consulta médica se obtiene un orden por cada tratamiento que se emplea en cada consulta.

Análisis de fuerzas bloqueadoras

FB 1: No se cuenta con una aplicación informática eficaz que controle y gestione los procesos necesarios a través del usuario

$I=5$: Tiene una escala valorativa de 5, porque afecta directamente con los procesos de gestión, si no se actualiza la información clínica de las consultas médicas y tratamientos que se aplican, se perderían los datos y existiría una pérdida total de toda la información almacenada.

$PC=1$: En caso de que esto llegue a ocurrir, sería muy desastroso y dificultoso ya que conllevaría mucho tiempo para restaurar los datos perdidos de cada paciente.

FB 2: Desconocimiento de los procesos de gestión y tratamiento veterinario.

$I=4$: Tiene una escala valorativa de 4, ya que afectaría de una manera considerable desconocer todos estos procesos fundamentales, para dar un servicio adecuado.

$PC=3$: Tiene una escala valorativa de 3, ya que Se podría solucionar optando por varias alternativas y estrategias la socialización de los procesos de gestión veterinaria por medio de charlas o conferencias que capaciten a los profesionales de la clínica HUSKY.

FB 3: Optar por herramientas inadecuadas ocasionando una operación ineficiente y descontrol de los procesos de tratamiento y atención

$I=5$: Tiene una escala valorativa de 4, ya que optar por herramientas equivocadas sin haber planificado con antelación afectaría al proceso en general, desviándonos a una situación empeorada

$PC=3$: Tiene una escala valorativa de 3, ya que la selección de herramientas equivocadas y sin planificación previa, hará que se torne complicado agilizar el proceso de gestión y tratamiento, para llevar a una situación mejorada.

FB 4: La obtención de resultados registrados manualmente en cada tratamiento que se aplica, genera inseguridad y desconfiabilidad

$I=4$: El registro manual de los resultados obtenidos, genera inseguridad total de la integridad de los datos del paciente, ocasionando la posible pérdida de los mismos, haciendo que el médico veterinario obtenga un diagnóstico poco efectivo.

$PC=2$: En caso de pérdida de registros de resultados, se tendría que volver a realizar un tratamiento médico adecuado al paciente, para obtener unos nuevos resultados y diagnósticos.

Capítulo II. Análisis de involucrados

2.01. Requerimientos

2.01.1 Descripción del sistema actual

Los procesos actuales son:

Los pacientes acuden al consultorio veterinario para realizar sus respectivos chequeos médicos, ingresando para ser atendidos por el doctor veterinario, el mismo que comenzará agendando una cita previa para la consulta según el orden en como visitan los pacientes, a continuación realiza una búsqueda de su historial clínico correspondiente en los archivos manuales que mantiene la institución, verificando posteriormente resultados y diagnósticos anteriores, para proceder a brindar la atención médica adecuada, el tratamiento necesario y determinar el nuevo diagnóstico, finalizando con la asignación de la próxima cita para una determinada fecha disponible, registrando de la misma manera en la historia clínica para luego ser archivado manualmente, de todo este proceso se determina existe pérdida de tiempo por la búsqueda y colocación de los documentos solicitados.

2.01.2 Visión y Alcance

Visión

La visión del proyecto es desarrollar un sistema web de gestión médica veterinaria para las consultas, citas y fichas médicas de los pacientes, lo que permitirá optimizar la atención al cliente en las horas estipuladas, para que de esta forma se dé la satisfacción necesaria a todos los pacientes que acuden a la institución.

Alcance

El presente proyecto tiene como finalidad el desarrollo de:

- Un módulo de ingreso de usuarios con los roles de: Administrador y usuario normal.
- Un formulario de datos para el registro de los clientes y pacientes.
- Asignación de citas verificando la disponibilidad de horarios del médico veterinario.
- Generación de una nueva ficha médica del paciente en caso de que no la tenga, y si la tiene, se realiza una búsqueda para determinar prescripciones médicas anteriores y que dolencias ha tenido, para luego hacer los registros correspondientes de los nuevos hallazgos médicos y prescripciones médicas para al finalizar con una nueva cita en una determinada fecha y hora.
- Un módulo necesario para realizar la gestión y registro de vacunaciones y desparasitaciones del paciente.
- Generación de reportes de: Pacientes activos, citas médicas pendientes por día, citas atendidas por día, resumen de consulta e historial médico.

2.01.3 Entrevistas

En esta sección del proyecto se procederá a redactar cada pregunta de manera detallada, colocando un objetivo claro y su respectivo análisis.

Tabla 2

Matriz de Preguntas

Entrevista realizada a Dra. Cristina Baquero (Médico Veterinario del consultorio)

Nº	PREGUNTA	OBJETIVO	ANÁLISIS
1	¿De qué manera se llevará a cabo las consultas médicas de los pacientes?	Conocer el proceso de gestión de fichas médicas veterinarias.	Se debe llenar la ficha médica por el profesional encargado, tomando nota del propietario y la mascota. La ficha médica estará al alcance del cliente servirá para ser consultada por el especialista cuando el animal padece alguna enfermedad crítica, permitiéndole contar con la información médica y poder tomar decisiones correctas en momentos críticos mediante un diagnóstico. Y posteriormente programar vacunas y desparasitaciones para el paciente
2	¿Cómo obtiene usted la historia clínica de su paciente?	Conocer cuál es la manera más accesible para ingresar a la información del paciente con el fin de ahorrar tiempo y mejorar la atención	Se debe permitir tener acceso a la historia clínica de la mascota por medio de un número de identificación que portará la mascota, o a su vez por el nombre de la persona propietaria.
3	¿Cómo usted receta o da un tratamiento médico a sus pacientes?	Conocer los procesos para Tratar a los pacientes de una manera correcta y más rápida	Primeramente se mantiene un listado periódico de las mascotas que aún no han acudido a su cita agendada, para enviarles un recordatorio de que deben asistir. Después que el paciente ha sido atendido se recopilan los síntomas si los tiene y se procede a realizar un diagnóstico para enviar la medicación y el tratamiento adecuado
4	¿Cuáles son los involucrados en el problema?	Identificar de manera clara los involucrados directos e indirectos del problema	Paciente, propietario, Veterinario, Administrador

Nota: desglosamiento de las preguntas realizadas a la persona entrevistada

Elaborado por: Autor del proyecto

2.01.4 Matriz de Requerimientos

Tabla 3

Requerimientos Funcionales.

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
REQUERIMIENTOS FUNCIONALES						
RF001	Luego de Ingresar los datos del propietario del paciente, el sistema deberá ingresar los datos de la mascota en una nueva historia clínica	Médico Veterinaria	Alto	Software	Revisión	Mascota, Propietario de la Mascota y médico Veterinario
RF002	Luego de registrar los datos del propietario y mascota el sistema deberá acceder al historial clínico de la mascota por medio del nombre del Propietario, nombre del paciente o número.	Médico Veterinaria	Alto	Software	Revisión	Mascota, médico veterinario Propietario
RF003	Se debe gestionar el programa de administración médica del paciente en su ficha para posteriormente programar la agenda de las citas con sus fechas respectivas	Médico Veterinaria	Alto	Usuario	Revisión	Mascota y médico Veterinario

Nota: Requerimientos funcionales para el desarrollo.

Tabla 4

Requerimientos No Funcionales.

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
REQUERIMIENTOS NO FUNCIONALES						
NRF001	Se considerará dos tipos de usuario, que son: Administrador y usuario normal	Médico Veterinaria	Medio	Usuario	Revisión	Médico Veterinaria
NRF002	El aplicativo web se ejecutará en el navegador Google Chrome	Usuario	Alto	Hardware	Revisión	Usuario
NFR003	Se utilizará framework ASP.NET para el desarrollo de la aplicación con lenguaje C#	Usuario	Alto	Hardware	Revisión	Usuario

Nota: Requerimientos no funcionales.

2.01.5 Descripción Detallada

Requerimientos Detallados individualmente.

Tabla 5.

Requerimiento funcional 001 (RF001)

Luego de Ingresar los datos del propietario o dueño, el sistema deberá ingresar los datos de la mascota en una nueva historia clínica		Estado	Revisión
Creado por:	Damián Navarrete	Actualizado por:	Damián Navarrete
Fecha Creación:	11/02/2015	Fecha Actualización	12/02/2015
Identificador	RF001		
Estado de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Datos proporcionados por el cliente de su mascota, al médico.		
Descripción	Se debe ingresar la información personal del cliente y datos generales de la mascota para guardar en una nueva historia clínica.		
Datos de Salida	Obtención de Síntomas, Diagnóstico y tratamiento.		
Resultados Esperados	Almacenamiento de datos.		
Origen	Médico Veterinaria		
Dirigido a	Director Clínica Veterinaria		
Prioridad	Alto		
Requerimientos Asociados	RF001		
ESPECIFICACIÓN			
Precondiciones	Datos necesarios para ingresar en la ficha médica		
Post condiciones	Obtención de un diagnóstico preciso		
Criterios de Aceptación	Proceso coherente		

Nota: RF001..

Tabla 6.

Requerimiento funcional 002 (RF002)

Luego de registrar los datos del propietario y mascota el sistema deberá acceder al historial clínico de la mascota por medio del número de cédula del propietario		Estado	Revisión
Creado por:	Damián Navarrete	Actualizado por:	Damián Navarrete
Fecha Creación:	11/02/2015	Fecha Actualización	12/02/2015
Identificador	RF002		
Estado de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Número de historia clínica o nombre del cliente		
Descripción	Se debe tener el acceso total del tratante al historial clínico de la mascota por medio de un número de identificación que portará o por el nombre del cliente		
Datos de Salida	Historial clínico del paciente		
Resultados Esperados	Optimización del proceso de consulta de la historia clínica		
Origen	Médico Veterinaria		
Dirigido a	Director Clínica Veterinaria		
Prioridad	Alto		
Requerimientos Asociados	RF001-RF002		
ESPECIFICACIÓN			
Precondiciones	Se debe tener historia clínica		
Post condiciones	Encontrar datos por medio del nombre del cliente o historia clínica		
Criterios de Aceptación	Organización		

Nota: RF002.

Tabla 7.

Requerimiento funcional 003 (RF003)

Se debe gestionar el programa de administración médica del paciente en su ficha para posteriormente programar la agenda de las citas con sus fechas respectivas		Estado	Revisión
Creado por:	Damián Navarrete	Actualizado por:	Damián Navarrete
Fecha Creación:	11/02/2015	Fecha Actualización	12/02/2015
Identificador		RF003	
Estado de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Diagnósticos del paciente, Tratamientos, y próxima cita		
Descripción	Se debe gestionar el programa de administración médica del paciente en su ficha para posteriormente programar la agenda de las citas con sus fechas respectivas		
Datos de Salida	Información médica del paciente actualizada		
Resultados Esperados	Optimizar el proceso de consulta médica		
Origen	Médico Veterinaria		
Dirigido a	Director Clínica Veterinaria		
Prioridad	Alto		
Requerimientos Asociados	RF001 – RF002		
ESPECIFICACIÓN			
Precondiciones	Paciente esperando el turno para ser atendido		
Post condiciones	Paciente tratado		
Criterios de Aceptación	Procesos Satisfactorios		

Nota: RF003.

Tabla 8.

Requerimiento no funcional 001 (NRF001)

Se considerará dos tipos de usuario, que son: Administrador y usuario normal		Estado	Revisión
Creado por:	Damián Navarrete	Actualizado por:	Damián Navarrete
Fecha Creación:	11/02/2015	Fecha Actualización	12/02/2015
Identificador	RF004		
Estado de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Usuario administrador o usuario normal		
Descripción	Se considerará dos tipos de usuario, que son: Administrador y usuario normal		
Datos de Salida	Gestión del usuario en el sistema		
Resultados Esperados	Utilización adecuada para la gestión de consultas médicas		
Origen	Médico Veterinaria		
Dirigido a	Director Clínica Veterinaria		
Prioridad	Alto		
Requerimientos Asociados	RF001 – RF002 – RF004		
ESPECIFICACIÓN			
Precondiciones	Contar con un usuario		
Post condiciones	Gestión del sistema		
Criterios de Aceptación	Organización		

Nota: NRF001.

Tabla 9.

Requerimiento no funcional 002 (NRF002)

El aplicativo web se ejecutará en el navegador		Estado	Revisión
Google Chrome			
Creado por:	Damián Navarrete	Actualizado por:	Damián Navarrete
Fecha Creación:	11/02/2015	Fecha Actualización	12/02/2015
Identificador	RF004		
Estado de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Navegador Google Chrome		
Descripción	La aplicación rendirá óptimamente ejecutándola desde el navegador Google Chrome		
Datos de Salida	Gestión del usuario en el sistema		
Resultados Esperados	EJE CUCIÓN DEL APLICATIVO		
Origen	Usuario		
Dirigido a	Director Clínica Veterinaria		
Prioridad	Alto		
Requerimientos Asociados	RF004		
ESPECIFICACIÓN			
Precondiciones	Contar con un usuario		
Post condiciones	Gestión del sistema		
Criterios de Aceptación	Organización		

Nota: NRF002.

Tabla 10.

Requerimiento no funcional (NRF003)

Se utilizará framework ASP.NET para el desarrollo de la aplicación con lenguaje C#		Estado	Revisión
Creado por:	Damián Navarrete	Actualizado por:	Damián Navarrete
Fecha Creación:	11/02/2015	Fecha Actualización	12/02/2015
Identificador		RF004	
Estado de Requerimiento	Critico	Tipo de Requerimiento	Funcional
Datos de Entrada	Visual Studio, framework ASP.NET, lenguaje de programación C#		
Descripción	El aplicativo se desarrollará por medio del lenguaje C# y para su ejecución se utilizará la herramienta ASP.NET framework		
Datos de Salida	Aplicativo construido		
Resultados Esperados	DESARROLLAR EL APLICATIVO ORIENTADO A LA WEB		
Origen	Usuario		
Dirigido a	Usuario		
Prioridad	Alto		
Requerimientos Asociados	RF005-NRF005		
ESPECIFICACIÓN			
Precondiciones	Contar con las herramientas de desarrollo necesarias		
Post condiciones	Aplicación construida en un 100%		
Criterios de Aceptación	Organización		

Nota: NRF003.

2.02. Mapeo de Involucrados

Mediante el mapeo de involucrados se trata de identificar a los grupos que, de manera directa e indirecta están relacionados con el inadecuado proceso registro de fichas médicas veterinarias que se desea solucionar o las situaciones que quieren ser modificadas (Figura 1)

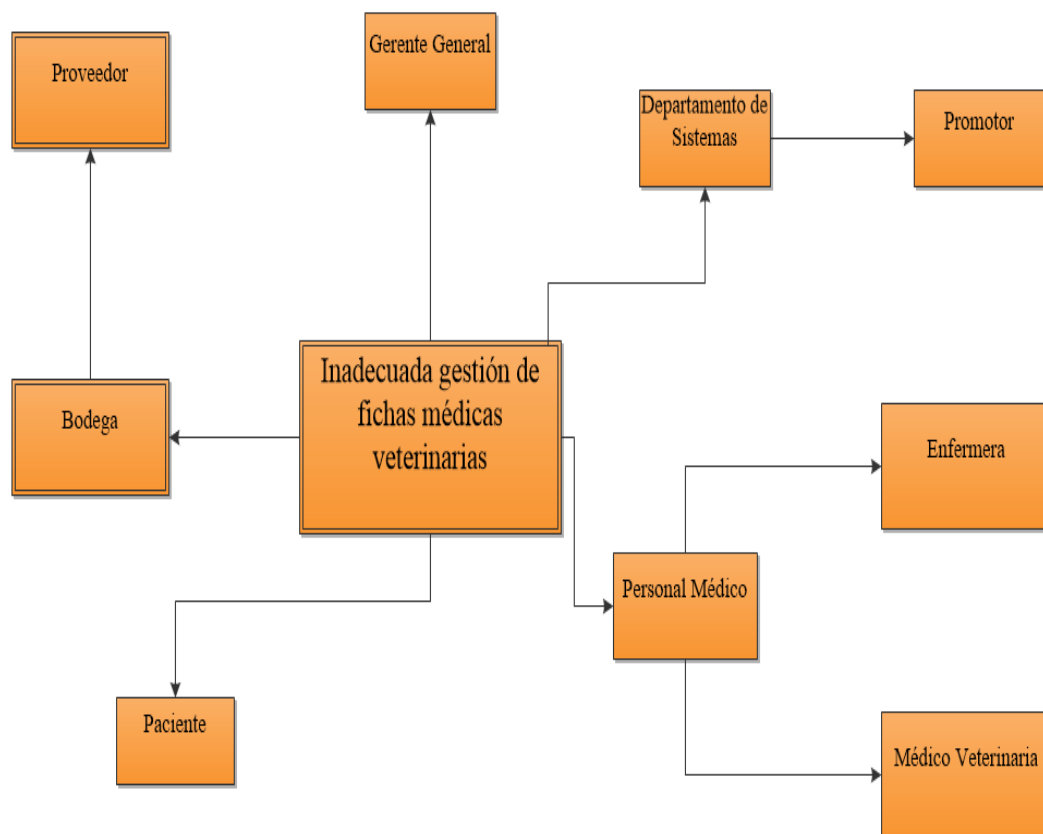


Figura 1 Mapeo de Involucrados En este esquema se organiza los involucrados directos e indirectos que intervienen en el proceso.

2.03. Matriz de Involucrados

La matriz de involucrados viene a ser un cuadro en donde se colocan a todos los participantes o quienes tienen que ver en la problemática analizada y la solución que se esté buscando en este caso para la inadecuada gestión de fichas médicas, detallando sus intereses, problemas, recursos, etc. Esta matriz responde a la pregunta: ¿A quiénes interesa el problema- plan- proyecto?, para desarrollar esta matriz se puede aconsejar desarrollar algún tipo de entrevista o reunión de trabajo con las personas involucradas. O también puede hacer algún tipo de encuestas a los grupos que van a intervenir en el plan o proyecto. En resumen, la “matriz de involucrados se puede usar para sintetizar los resultados de una investigación inicial con las partes interesadas.

(ANEXO A.01).

Capítulo III. Problemas Y Objetivos

3.01. Árbol de Problemas

En esta sección se procederá a elaborar el árbol de problemas, el cual consiste en elaborar una lluvia de ideas en función de las causas que dan origen al problema central, para determinar con efectividad los efectos a los cuales nos llevan dichas causas. Las causas se las esquematiza en la parte inferior del problema central, estas se dividen en causas directas, e indirectas, las causas indirectas son las que promueven a las causas directas, de la misma manera se aplica este concepto para elaborar los efectos que se construyen en la parte superior del problema central, estos se dividen en efectos directos e indirectos.(Figura 2).

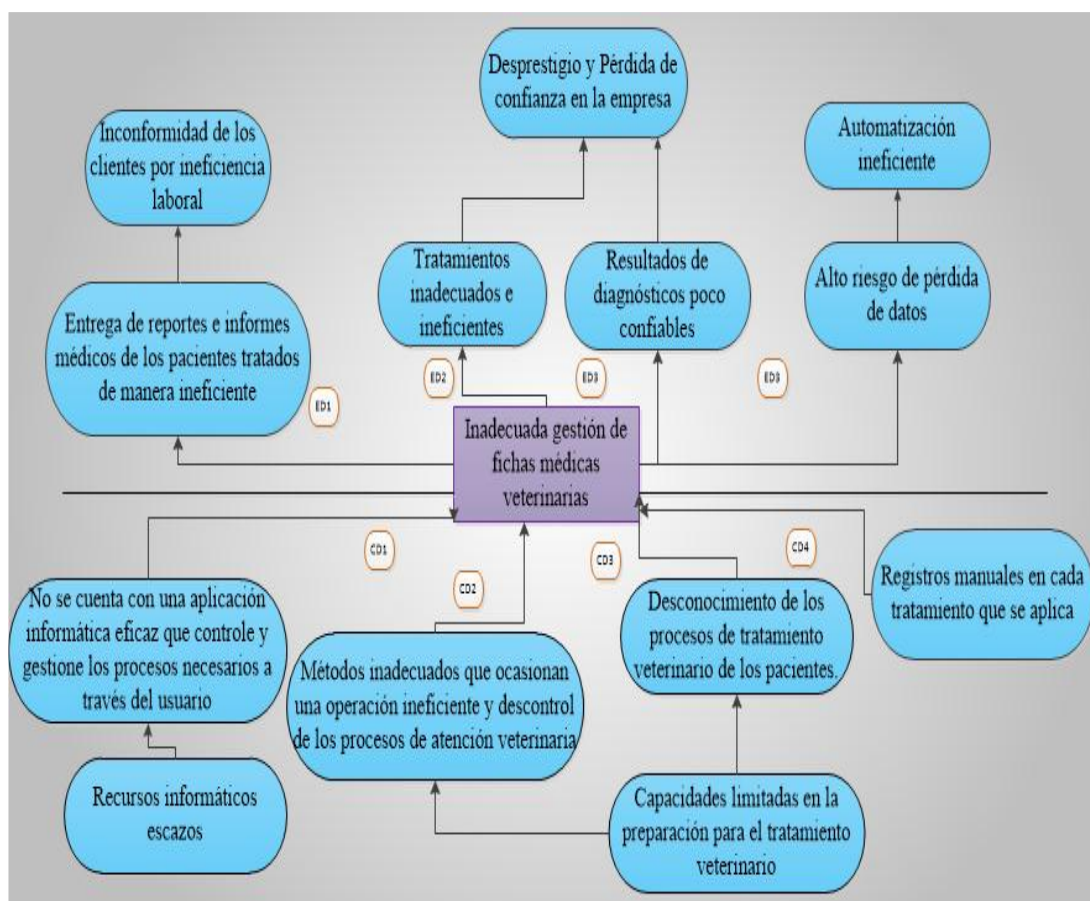


Figura 2 Árbol de problemas En este esquema se organiza detalladamente las causas y efectos que promueve al problema central.

3.02. Árbol de Objetivos

Con el mismo esquema con el que se construyó el árbol de problemas se construirá el árbol de objetivos (Figura 3). Consiste en elaborar un esquema de solución a los problemas que se llevarán a cabo a través de los objetivos a concretar para poder lograr conseguir la situación mejorada.

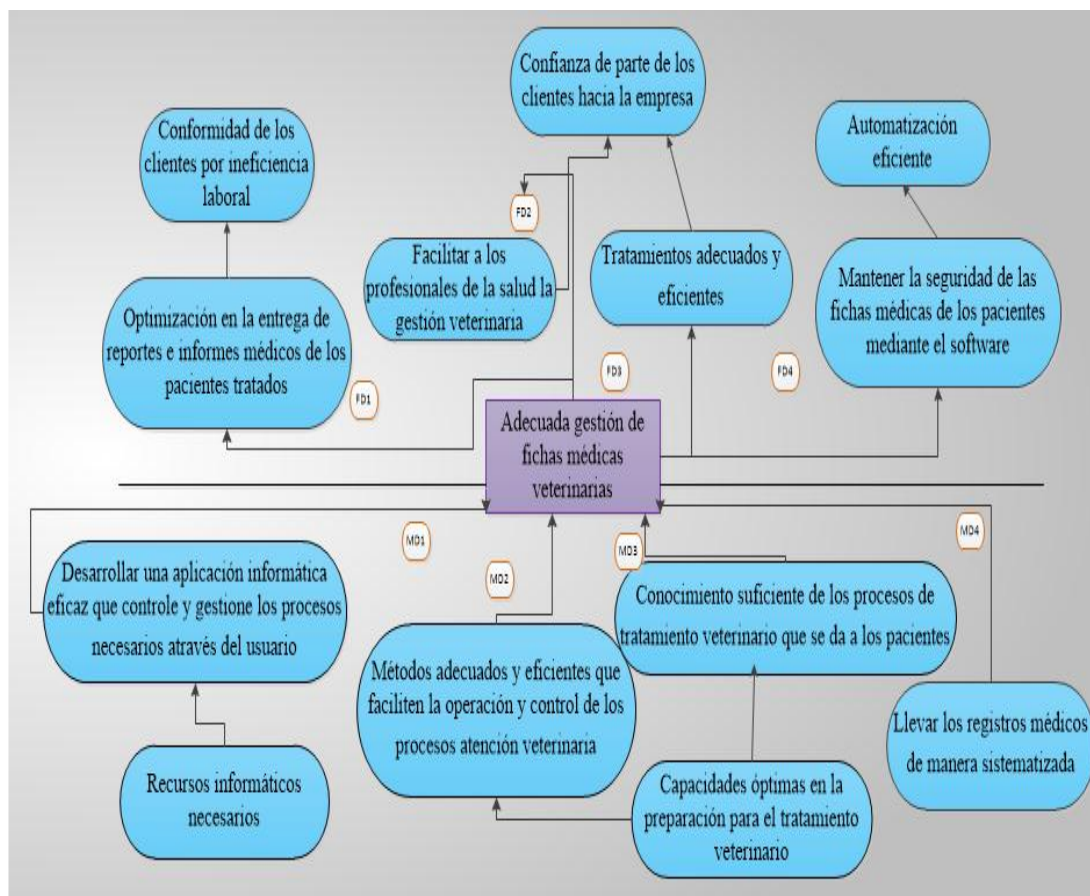


Figura 3 Árbol de Objetivos En este esquema se organiza de forma detallada los medios por los cuales se llegará a los fines, los mismos que se clasifican en Medios directos, indirectos y estructurales al igual que las causas.

3.03. Diagramas de Casos de Uso

Diagrama de caso de uso del sistema (gestión de fichas médicas veterinarias)

(Figura 4)

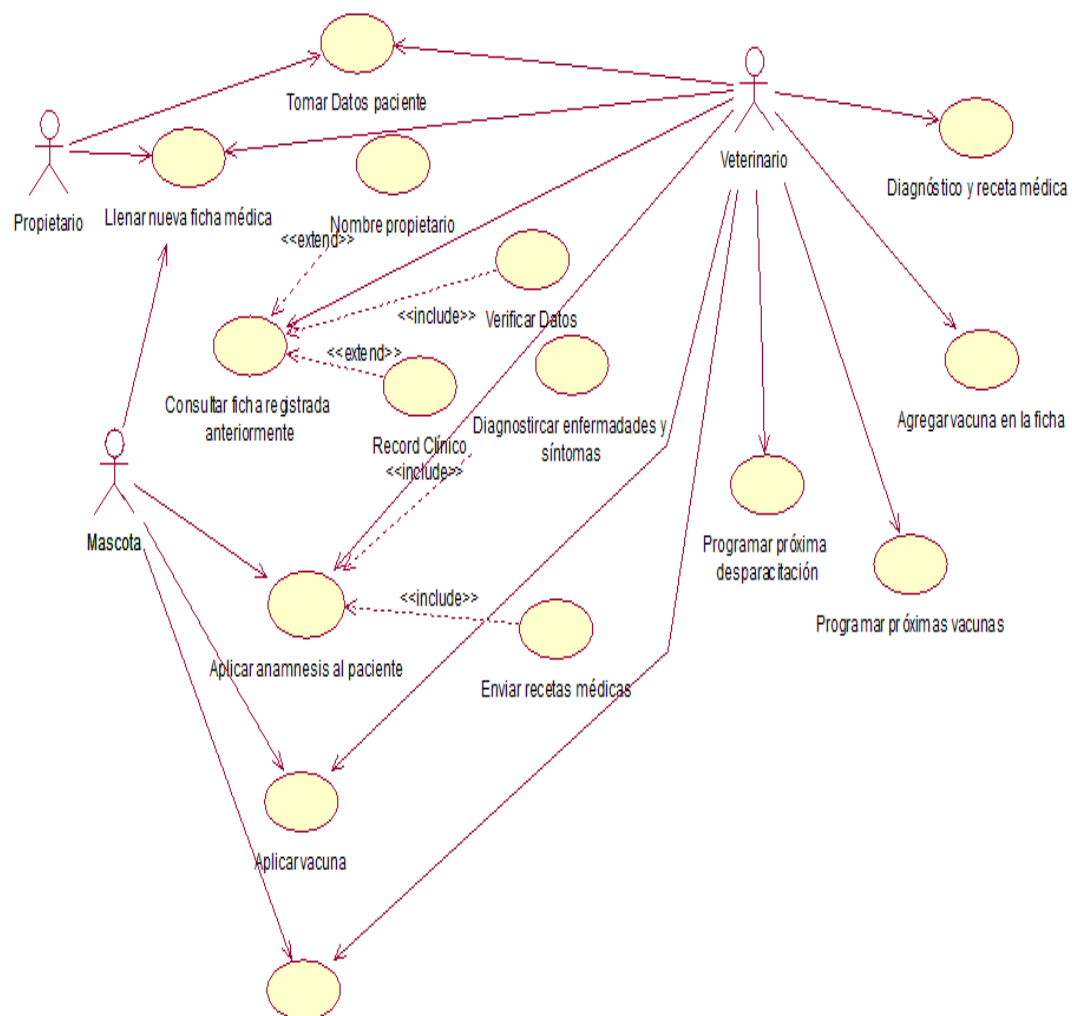


Figura 4 Diagramas de casos de uso En esta figura se esquematiza los procesos que realizará el sistema con sus respectivos actores y casos de uso

3.04. Casos de uso de Realización

Se realizarán los diagramas y especificaciones de casos de uso de realización del sistema.

En la siguiente figura (Figura 5) podemos contemplar el caso de uso de realización, en la que se maneja el proceso que se realiza para una consulta médica que se realizará por primera vez en el consultorio Veterinario.

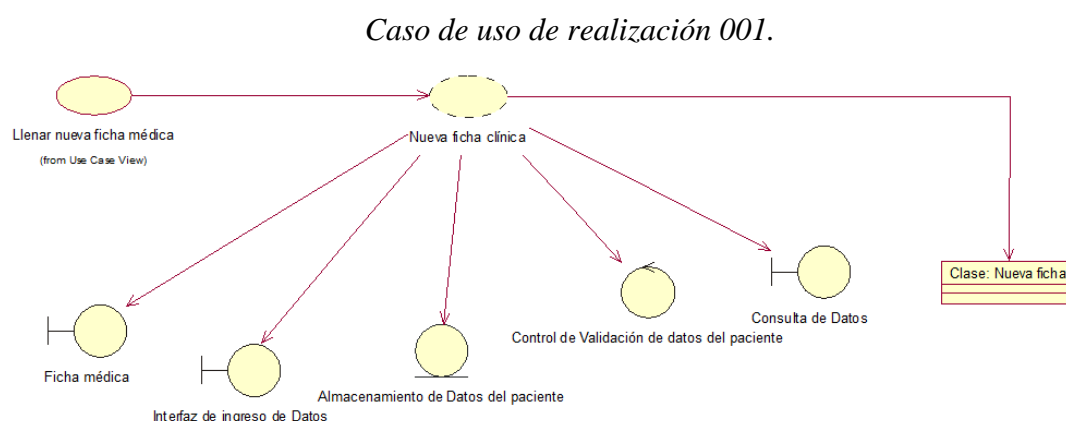


Figura 5 Caso de uso de realización 001 En la figura se esquematiza el proceso de nueva ficha médica.

Tabla 11.

Especificación de caso de uso de realización 001.

Nombre:	Nueva Ficha médica
Identificador:	CUR001
Responsabilidades:	Registrar Nuevos Datos de pacientes
Tipo:	Software
Referencias Casos de uso:	Propietario, Veterinario
Referencias Requisitos:	Obtener Datos necesarios para el nuevo Registro de Datos del paciente y propietario
PRECONDICIONES	
1.- Que el Veterinario (Usuario) Se haya identificado en el sistema como Administrador 2.- Que exista Un módulo para Registrar Datos necesarios del paciente y propietario	
POSCONDICIONES	
1.- Se contará con una nueva ficha médica del paciente registrado	
SALIDAS EN PANTALLA	
Datos del Paciente	

Nota: Especificación CUR001

Caso de uso de realización 002.

En este proceso se describirá esquemáticamente el proceso de examinación y tratamiento del paciente para posteriormente, tomar sus síntomas que lo aquejan, y poder continuar con un diagnóstico y un tratamiento efectivo. (Figura 6)

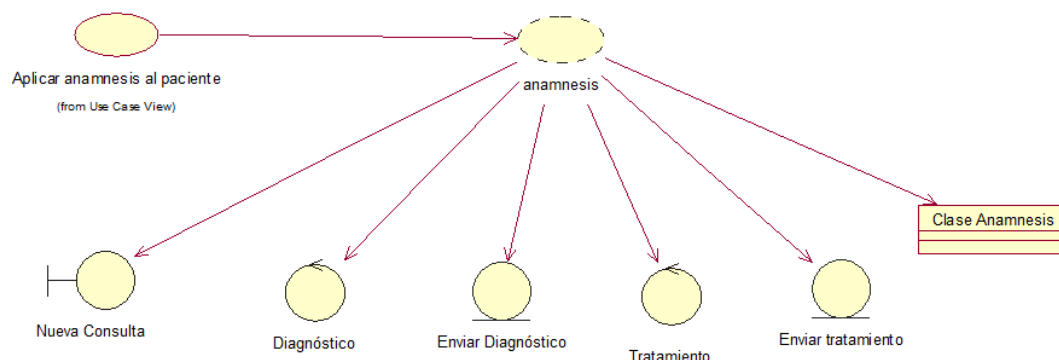


Figura 6 Caso de uso de realización del proceso de anamnesis. En la figura se esquematiza el proceso de anamnesis al paciente.

Tabla 12.

Especificación de caso de uso de realización 002.

Nombre:	Anamnesis
Identificador:	CUR002
Responsabilidades:	Médico Veterinario y paciente
Tipo:	Software
Referencias Casos de uso:	Propietario, Veterinario
Referencias Requisitos:	Atención al paciente para determinar sus síntomas y su respectivo diagnóstico y tratamiento
PRECONDICIONES	
1.- Que el Veterinario (Usuario) Se haya identificado en el sistema como Administrador 2.- Que exista Un módulo para Registrar Datos necesarios del paciente y propietario 3.- Control del programa de medicación	
POSCONDICIONES	
1.- Actualizar Datos del paciente registrando síntomas y tratamientos, para programar una nueva cita	
SALIDAS EN PANTALLA	
Datos	

Nota: Especificación CUR002.

Caso de uso de realización 003.

En la figura que se observa a continuación (Figura 7) Se esquematiza el proceso de tratamiento que se aplica al paciente, principalmente la administración de vacunas que son importantes el llevar un registro y una gestión de los mismos

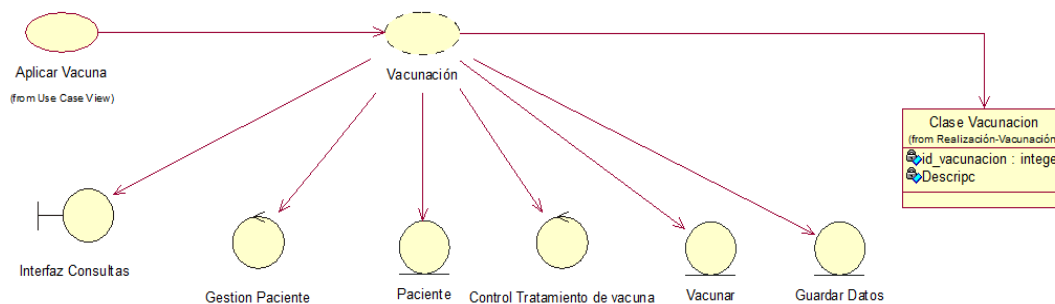


Figura 7 Caso de uso de realización de tratamientos de vacunas. En la figura se esquematiza el proceso de tratamientos de vacunas del paciente.

Tabla 13.

Especificación de caso de uso de realización 003.

Nombre:	Vacunación
Identificador:	CUR003
Responsabilidades:	Médico Veterinario y paciente
Tipo:	Software
Referencias Casos de uso:	Propietario, Veterinario
Referencias Requisitos:	Atención al paciente para determinar sus síntomas y su respectivo diagnóstico y tratamiento
PRECONDICIONES	
1.- Que el Veterinario (Usuario) Se haya identificado en el sistema como Administrador 2.- Que exista Un módulo para almacenar toda la historia clínica del paciente 3.- Cita agendada al paciente para su respectiva consulta médica 4.- Aplicar tratamientos necesarios	
POSCONDICIONES	
1.- Actualizar Datos del paciente registrando tratamientos aplicados 2.- Programar o asignar nueva cita médica para una nueva revisión y nueva vacunación que será necesario	
SALIDAS EN PANTALLA	

Nota: Especificación CUR003

Caso de uso de realización 004.

En la siguiente figura (Figura 8.) Se describen gráficamente el proceso de asignación de citas próximas a las cuales debe asistir el paciente.

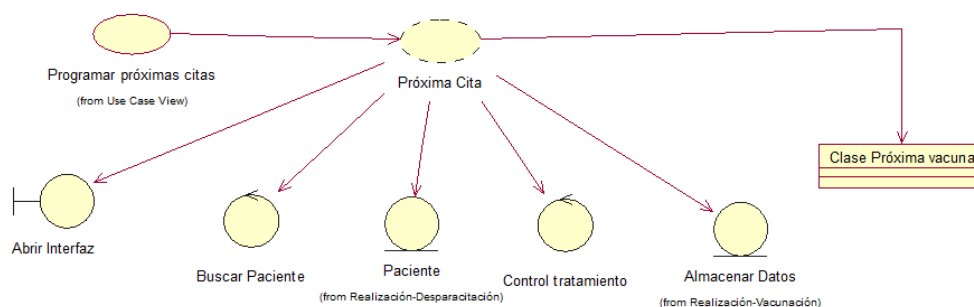


Figura 8 Caso de uso de realización de próximas citas. En la figura se esquematiza el proceso de Citas médicas

Tabla 14.

Especificación de caso de uso de realización 004.

Nombre:	Próxima cita
Identificador:	CUR004
Responsabilidades:	Médico Veterinario y paciente
Tipo:	Software
Referencias Casos de uso:	Propietario, Veterinario
Referencias Requisitos:	Cita médica actual, revisión actual
PRECONDICIONES	
1.- Que el Veterinario (Usuario) Se haya identificado en el sistema como Administrador 2.- Haber asistido a consulta médica para posteriormente programar una nueva cita. 3.- Asignar una fecha necesaria para la próxima cita.	
POSCONDICIONES	
1.- Llevar el control médico del paciente. 1.- Nueva cita médica	
SALIDAS EN PANTALLA	

Nota: Especificación CUR004.

3.05. Diagramas de Secuencia del Sistema

En esta sección se elabora mediante un esquema de secuencia cada uno de los casos de uso, los cuales manejan un proceso determinado dentro del proceso en general, que compete a la gestión de tratamientos, consultas, citas médicas, historias clínicas y fichas médicas.

Diagrama de Secuencia 001.

En la siguiente figura (Figura 9.) se describe el proceso secuencial que se llevará a cabo para realizar el registro y almacenamiento de una nueva ficha clínica, que contiene los datos del propietario y del paciente.

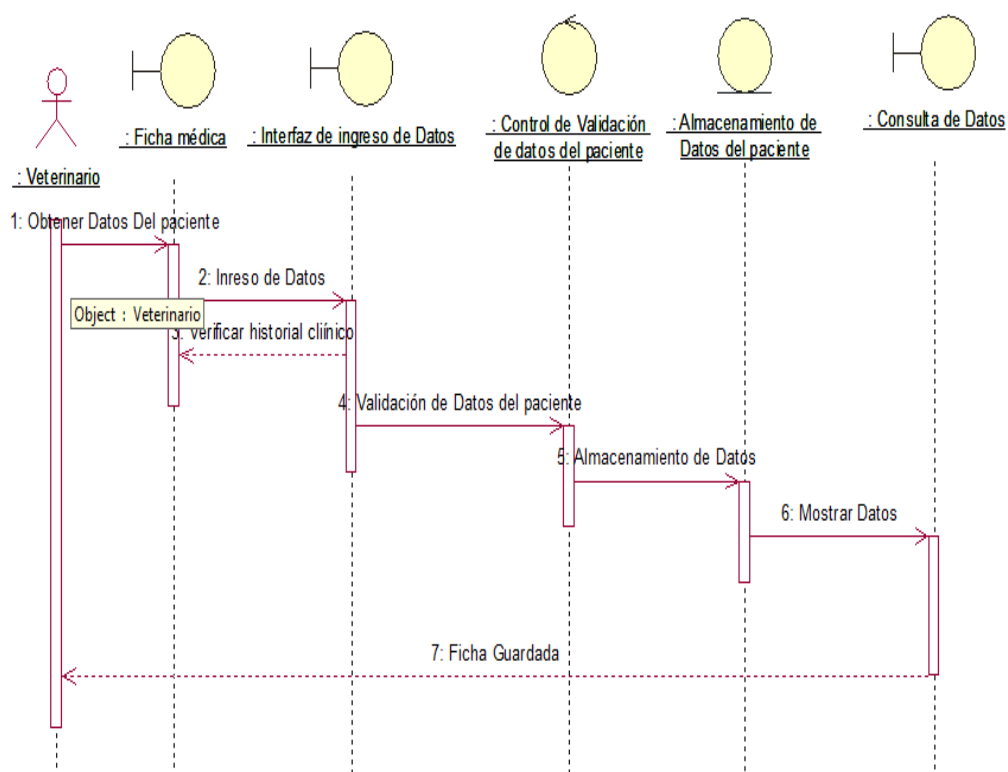


Figura 9 Diagrama de secuencia de nueva ficha médica

Diagrama de Secuencia 002.

Se describe detalladamente los procesos secuenciales que respectan con la anamnesis del paciente, es decir con el proceso de atención médica, revisión, recopilación de síntomas, signos vitales, lo que da lugar a un diagnóstico y un tratamiento lo cual se debe gestionar de manera organizada.(Figura 10)

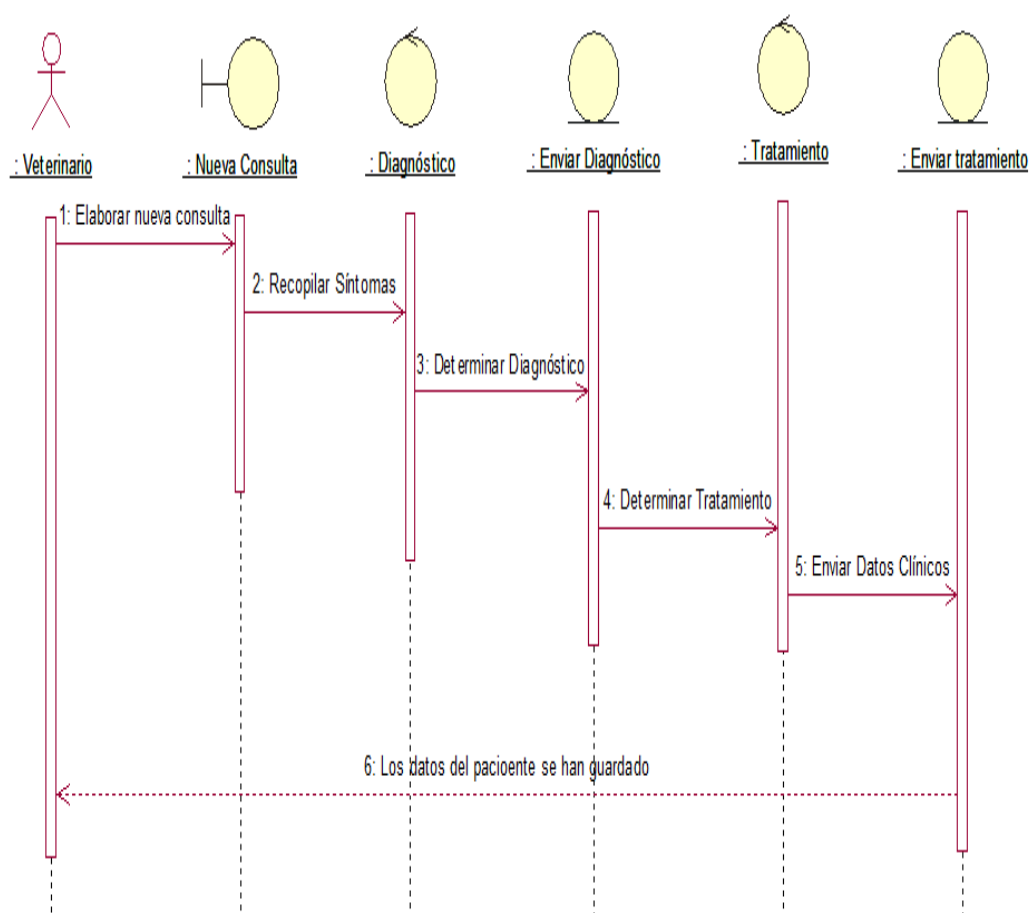


Figura 10 Diagrama de Secuencia de anamnesis del paciente.

Diagrama de Secuencia 003.

En este proceso se describen las secuencias que dan lugar al tratamiento del paciente que son las vacunaciones que se deben aplicar al paciente y a su vez llevar un control de dicho tratamiento. (Figura 11.)

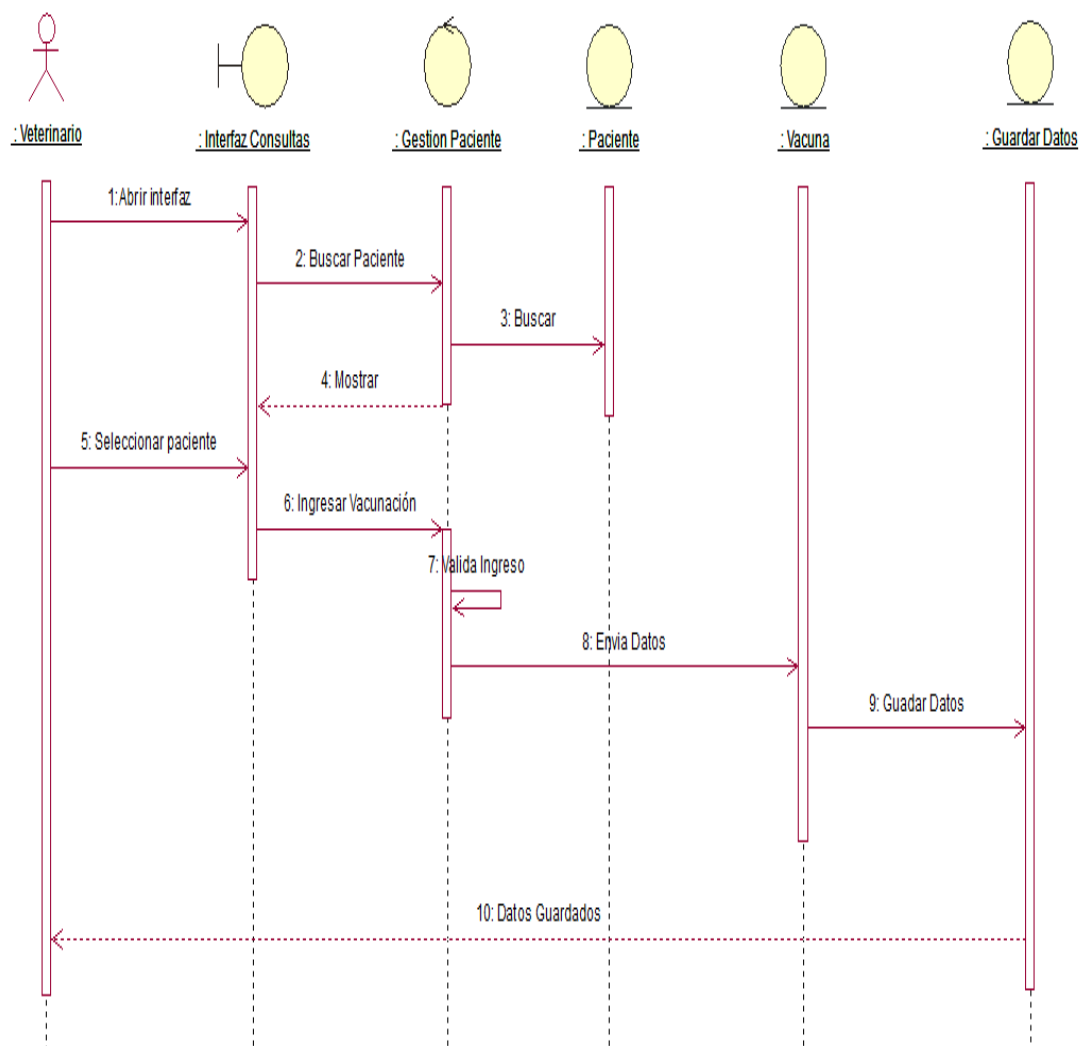


Figura 11 Diagrama de secuencia de vacunación.

Diagrama de Secuencia 004.

En el siguiente esquema se elaboran los procesos secuenciales conforme se realiza la consulta médica del paciente para pasar a asignar una nueva cita y de esta manera culminar el proceso de consulta. (Figura 12.)

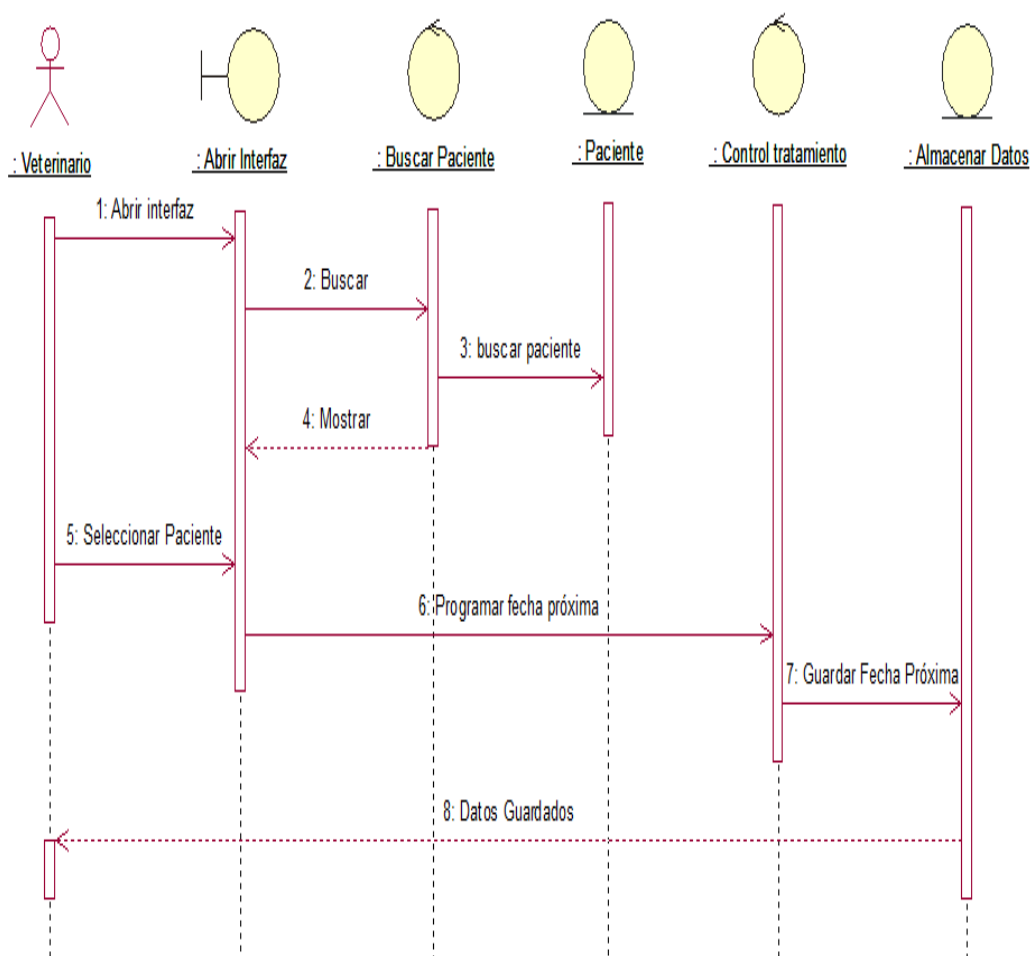


Figura 12 Diagrama de Secuencia de próxima cita

3.05. Especificación de Casos de Uso

Tabla 15.

Especificación de Caso de Uso 001.

CASO DE USO:		Agregar nueva ficha clínica
IDENTIFICADOR: Especifica el flujo para crear una nueva ficha médica en caso que el paciente llegue al consultorio por primera vez		
CURSO TÍPICO DE EVENTOS		
Usuario	Sistema	
Tomar y registrar datos del propietario	Guarda Datos en la base de datos	
Tomar y registrar datos del paciente	Guarda Datos en la base de Datos	
Buscar datos	Muestra datos del paciente	
CURSOS ALTERNATIVOS		
Cliente no está registrado, no puede continuar		
Si no se registra el número de cédula, no se podrá registrar a las mascotas		
Record clínico ya existe		

Nota: Especificación CU001

Tabla 16.

Especificación de Caso de Uso 002.

CASO DE USO:		Realizar Anamnesis
IDENTIFICADOR:	Especifica el flujo para realizar una recopilación de síntomas al paciente para ser registrados posteriormente	
CURSO TÍPICO DE EVENTOS		
Usuario	Sistema	
Consulta Datos registrados	Consulta Datos y muestra en pantalla	
Determinar e ingresar Diagnóstico del paciente	Valida Datos Ingresados	
	Almacena Datos en la base de Datos	
CURSOS ALTERNATIVOS		
Record del paciente no existe, primero cree una nueva ficha médica del paciente para continuar		

Nota: Especificación CU002

MEJORAMIENTO DEL PROCESO DE GESTIÓN DE FICHAS MÉDICAS VETERINARIAS
 MEDIANTE UN APLICATIVO WEB PARA LA CLÍNICA VETERINARIA HUSKY

Tabla 18.

Especificación de Caso de Uso 004.

CASO DE USO:	
Programar próxima cita	
IDENTIFICADOR: Permite al médico veterinario consultar la ficha médica de la mascota paciente para posteriormente controlar las vacunaciones que este lleva en toda su historia clínica y para programar próximas citas para su respectiva vacunación	
CURSO TÍPICO DE EVENTOS	
Usuario	Sistema
Abrir interfaz Donde se encuentra el módulo de procesos	Consulta Datos y muestra en pantalla
Seleccionar opción buscar paciente	Presenta la opción Consultar Datos del paciente por: <ul style="list-style-type: none"> • Cédula del propietario • Nombre de mascota
Seleccionar	Realiza la búsqueda en función al Dato ingresado
Ingresa Dato de Fecha próxima	Sugiere la fecha Indicada para realizar la próxima vacunación, caso contrario el usuario lo hará manualmente si lo requiere
Selecciona Guardar	Valida la información contenida en la ficha Almacena Datos
CURSOS ALTERNATIVOS	
No aplica	

Nota: Especificación CU004

Capítulo IV. Análisis de Alternativas

4.01. Matriz de Análisis de Alternativas.

Una vez establecido el árbol de alternativas se realiza la matriz de alternativas tomando en cuenta las amenazas que se escogieron para tomar las decisiones adecuadas e identificar las posibles soluciones que se pueden llevar a cabo en el proceso valorando las posibilidades de ser llevadas adecuadamente a la práctica y acordar una estrategia a nivel de proyecto, técnicas para identificar niveles de soluciones que puedan llegar a ser la estrategia de dicho proyecto y seleccionar las alternativas, que presentan las mejores estrategias aplicando criterios apropiados o requeridos para aplicar la estrategia. (ANEXO A02). La valoración se hace de la siguiente manera:

GENERAL

- 1 BAJO
- 2 MEDIO BAJO
- 3 MEDIO
- 4 MEDIO ALTO
- 5 ALTO

PRIORIDAD

- 5 BAJO
- 10 MEDIO BAJO
- 15 MEDIO
- 20 MEDIO ALTO
- 25 ALTO

4.02. Matriz de Impactos de Objetivos.

En esta sección del proyecto se procederá a estudiar los impactos y efectos que tendrá el proyecto en sí como los aspectos que se mencionarán a continuación: el impacto ambiental, la sostenibilidad del proyecto, los objetivos que se llevarán a cabo, el impacto de género y la relevancia del proyecto, todos estos aspectos están inmersos muy fundamentalmente dentro de desarrollo del proyecto debido a que nos

da conocer la sustentabilidad y rendimiento que puede llegar a tener en un determinado ámbito laboral. (ANEXO A03).

4.03. Estándares para el Diseño de Clases.

Este es un estudio sobre el Lenguaje de Modelado que abarca desde la definición de sus conceptos hasta su aplicación en un ejemplo práctico, en el mismo veremos cómo UML nos permite experimentar y visualizar un sistema que aún no ha sido codificado.

Clase.

“Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces. Gráficamente se representa como un rectángulo que incluye su nombre, sus atributos y sus operaciones.” (software, 2011)

(Figura 13)

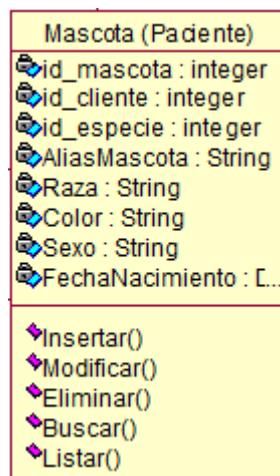


Figura 13 Clase Mascota.

Atributos:

“Los atributos se guardan en variables denominadas de instancia, y cada objeto particular puede tener valores distintos para estas variables.

Las variables de instancia también denominados miembros dato, son declaradas en la clase pero sus valores son fijados y cambiados en el objeto.” (software, 2011)

(Figura 14)

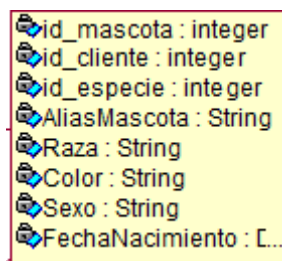


Figura 14 Atributos de la Clase Mascota

Métodos.

Un método es un bloque de código que contiene una serie de instrucciones. “Un programa hace que se ejecuten las instrucciones al llamar al método y especificando los argumentos de método necesarios.” (software, 2011) (Figura 15)

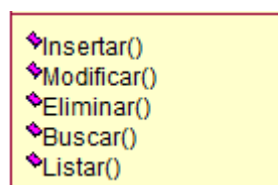


Figura 15 Métodos de la Clase Mascota (Insertar, Modificar, Eliminar, Buscar, Listar)

Clase Cliente

Se visualiza la clase cliente con su respectivo conjunto de objetos que comparte atributos, operaciones, relaciones y semántica, incluye su nombre, sus atributos y sus operaciones. (Figura 16)

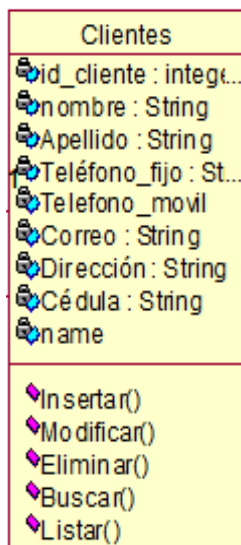


Figura 16 Clase Cliente

Atributos de la clase Cliente

En la figura (Figura 17) se visualiza claramente cada uno de los atributos de la instancia, las variables de instancia son declaradas en la clase pero sus valores son fijados y cambiados en el objeto.

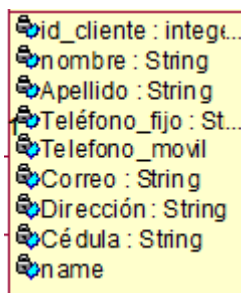


Figura17 Atributos

Métodos de la Clase Cliente (Operaciones)

En la figura (Figura 18) Se observan cada una de las operaciones, que en esencia son los métodos que utiliza la clase Cliente para realizar cada una de las operaciones al ejecutar las instrucciones llamando el método y especificando los argumentos de método necesarios, es decir, cada instrucción se ejecuta al momento de llamar a un método, esto con el fin de manejar y acceder a los datos de la solución

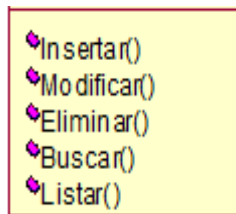


Figura 18 Métodos de la clase

4.04. Diagrama de clases.

“El diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.” (software, 2011)

(Figura 19)

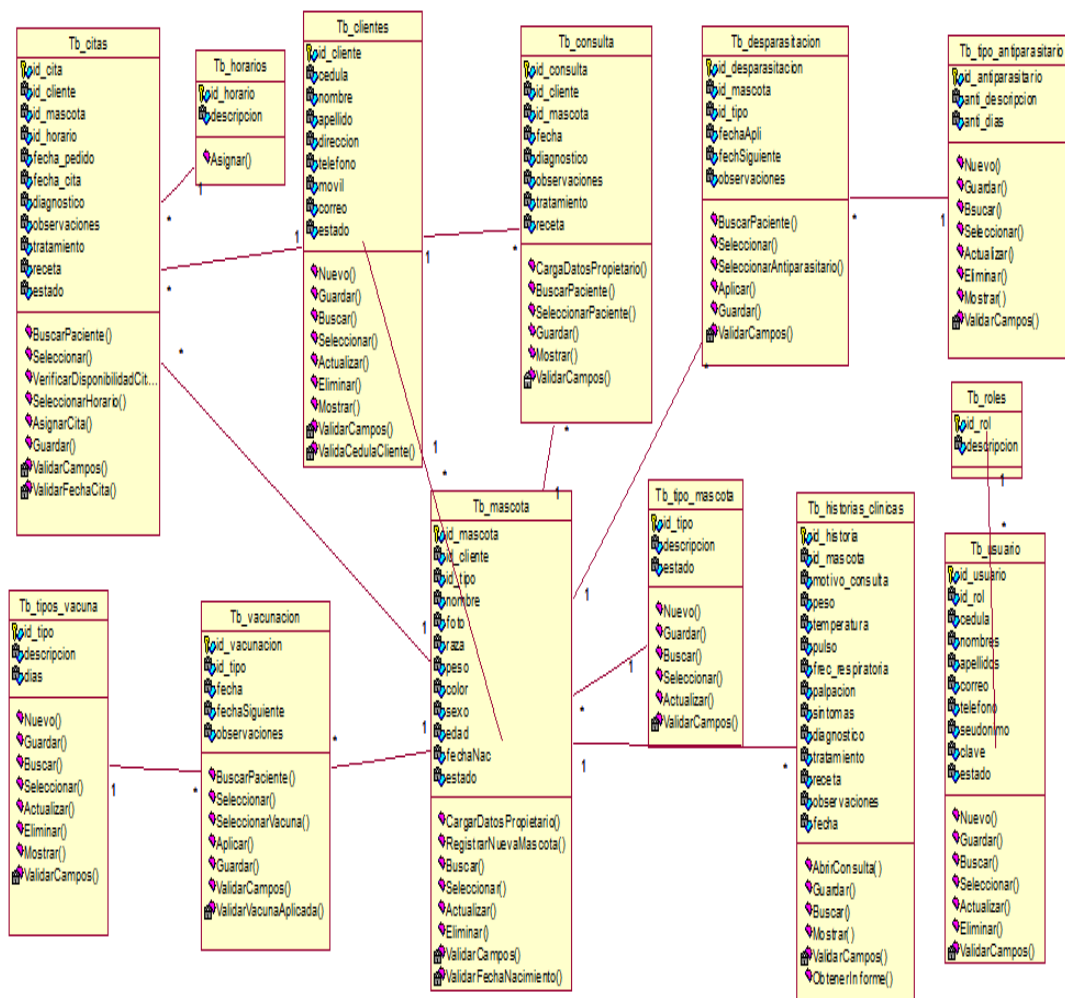


Figura 19 Diagrama de Clases de la solución

4.05. Modelo Lógico – Físico

Modelo Lógico

“Son orientados a las operaciones más que a la descripción de una realidad.

Usualmente están implementados en algún Manejador de Base de Datos. El ejemplo más típico es el Modelo Relacional, que cuenta con la particularidad de contar también con buenas características conceptuales (Normalización de bases de datos).

Este Modelos busca obtener una representación del modelo conceptual que use de forma eficiente las facilidades de estructuración de datos y modelado de restricciones, disponibles en el modelo.” (software, 2011)

Modelo Físico

El diseño físico es el proceso de producir una descripción de la implementación de la base de datos en memoria secundaria. Describe las relaciones base y las estructuras de almacenamiento y métodos de acceso que se utilizarán para acceder a los datos de modo eficiente. El diseño de las relaciones base sólo se puede realizar cuando el diseñador conoce perfectamente toda la funcionalidad que presenta el SGBD que se vaya a utilizar.

Modelo Lógico-Físico.

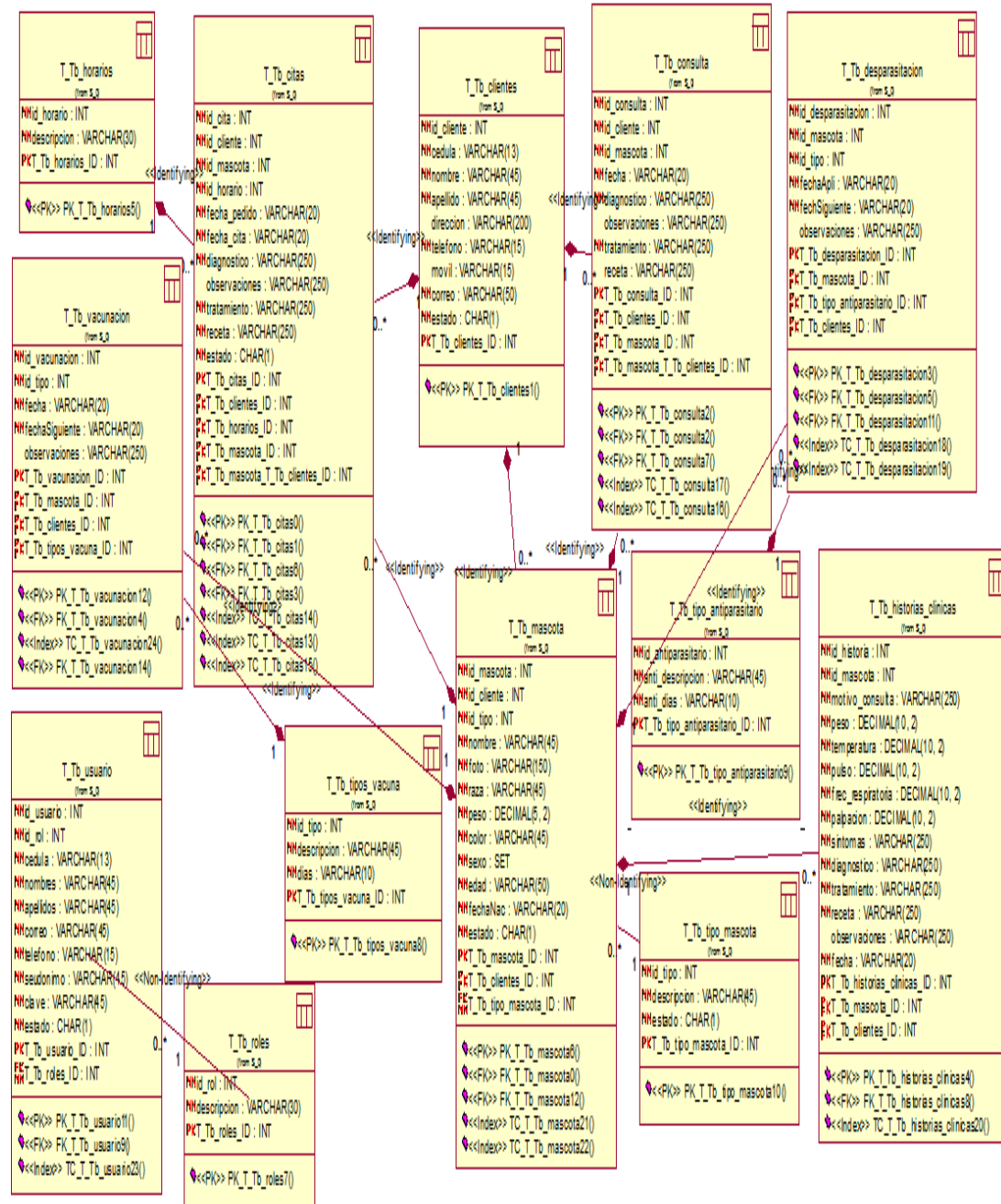


Figura 20 Modelo Lógico-Físico del Sistema

4.06. Diagrama de Componentes

“Un diagrama de componentes muestra los elementos del diseño de un sistema de software. Un diagrama de componentes permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces.” (2012, ASPW)

(Figura 21)

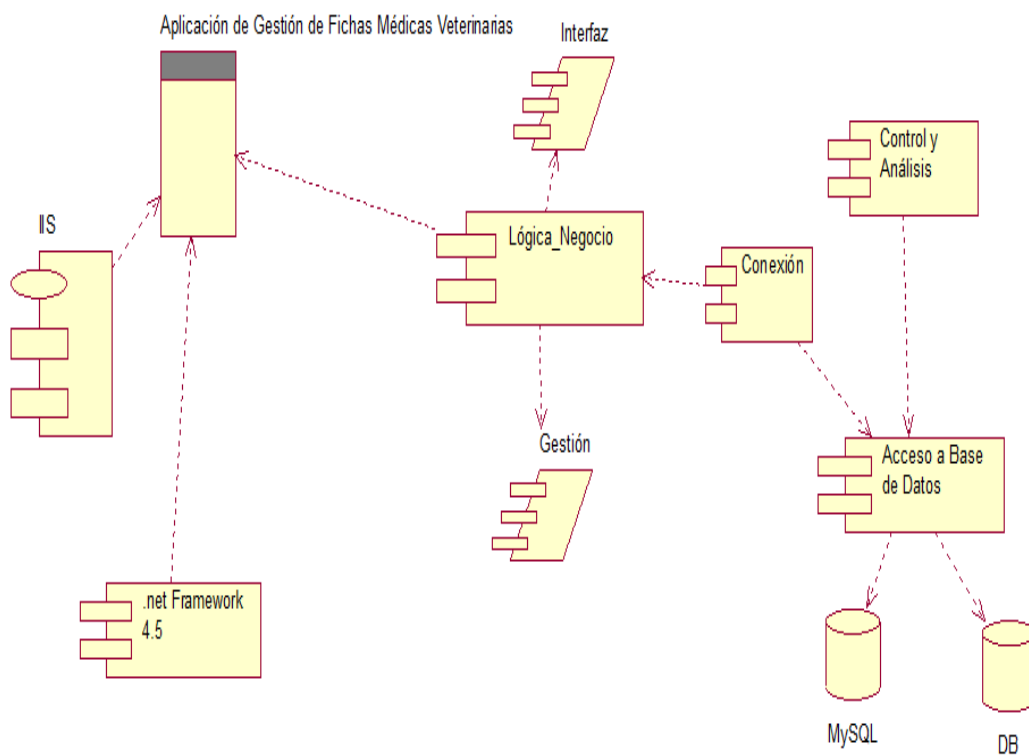


Figura 21 Diagrama de Componentes Del Sistema

4.07. Diagramas de Estrategias

En esta sección del trabajo se elaborará un esquema de forma jerárquica el cual consistirá en desarrollar las actividades que se realizarán para mejorar el proyecto, estas formarán los componentes que serán la solución y que nos abrirá paso para poder llegar al propósito principal que es mejorar el proceso de registro de fichas médicas veterinarias, todo esto contendrá una inversión presupuestaria que se determinará por medio de las actividades desenvueltas y aplicadas, el porcentaje en el cual se cumplieron cada una para determinar con certeza, si las estrategias tomadas fueron eficientes (ANEXO A04.)

4.08. Matriz de Marco Lógico.

Una matriz de marco Lógico es un resumen detallado del desarrollo del proyecto, el cual contiene aspectos financieros del proyecto, presupuestos, costes, y lo que se desea lograr, cómo alcanzar los propósitos establecidos, y también los factores externos indispensables, los objetivos trazados y recursos necesarios (ANEXO A 05)

4.09. Vistas arquitectónicas

4.01.01 Vista Lógica

En el siguiente gráfico se puede observar el desarrollo de la vista lógica del sistema con sus respectivos módulos y componentes necesarios para su establecimiento. (Figura 22)

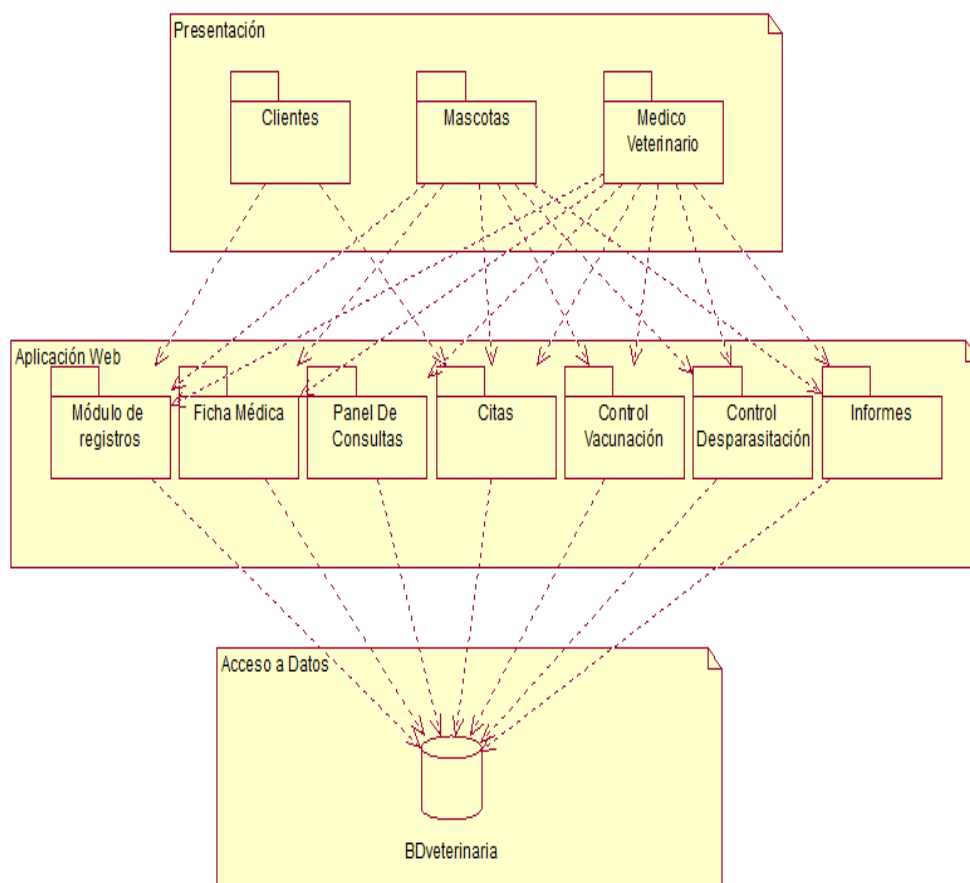


Figura 22 Vista Lógica Del Sistema desarrollado en Rational Rose.

4.01.02 Vista Física.



Figura 23 Vista Física. En el presente gráfico se muestra la Vista Lógica del sistema elaborado en Rational Rose.

4.01.03 Vista De Desarrollo.

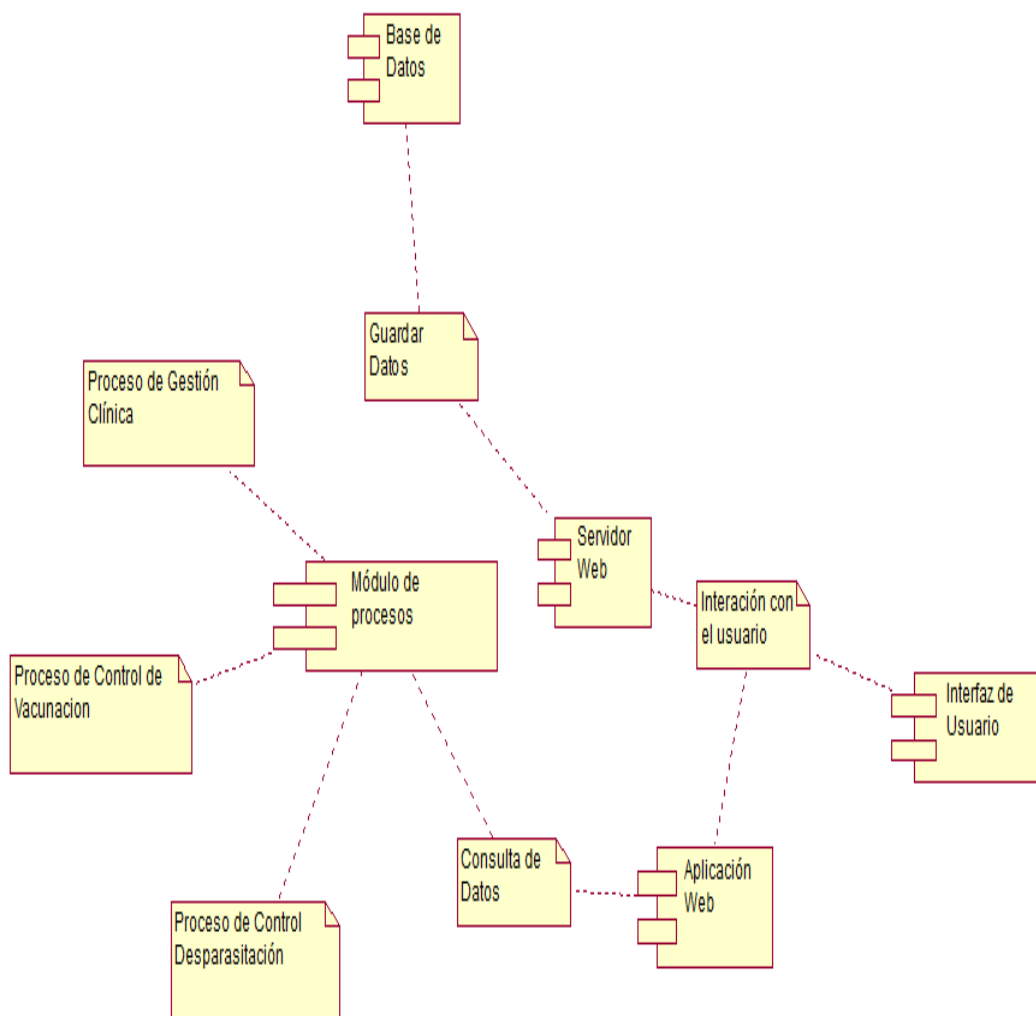


Figura 24 Vista De Desarrollo. En el presente gráfico se puede contemplar la Vista de Desarrollo elaborado en Rational Rose.

4.01.04 Vista De Procesos.

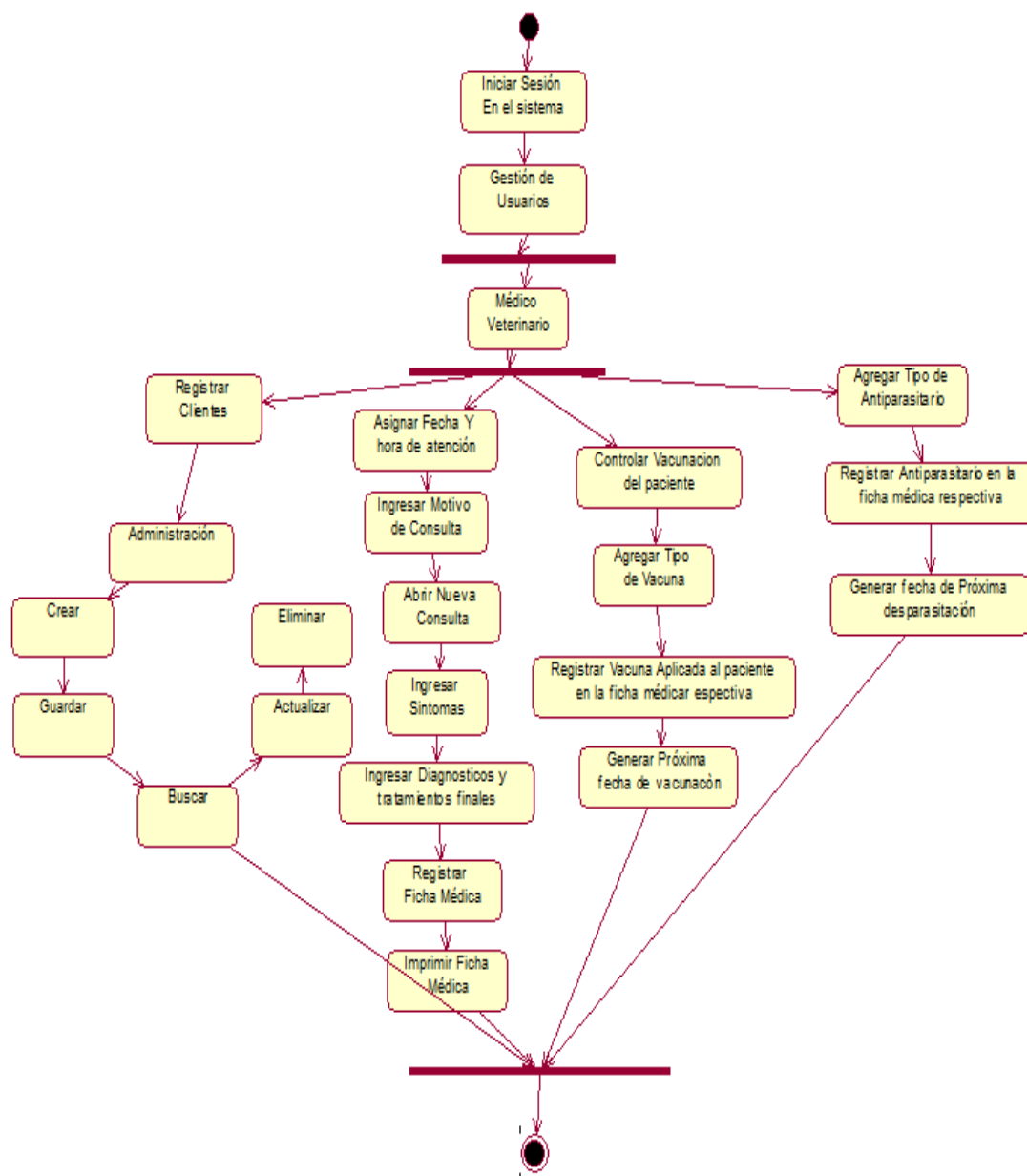


Figura 25 Vista de Procesos. En el presente gráfico se observa el detalle de cada proceso que genera el sistema elaborado en Rational Rose.

Capítulo V: Propuesta

5.01. Especificación de estándares de programación

Los estándares de programación son formas de ordenar y regularizar la programación que se utilizará al desarrollar un proyecto, lo cual hará que cada línea de código digitada sea mucho más entendible para los usuarios del sistema.

(ANEXO A.06)

5.01.01 Microsoft Visual Studio 2013.

Visual Studio 2013 es un IDE (entorno de desarrollo integrado) de plataforma.NET capaz de soportar variados lenguajes de programación como Java, C++, Visual Basic, PHP, C#, etc.

Para el desarrollo del presente proyecto se ha utilizado el lenguaje de programación C#, el cual se ejecutará trabajando sobre el framework de .NET versión 4.5.1., utilizado para el desarrollo de aplicaciones en ambiente web, Utilizando el tipo de desarrollo a tres capas, con una vista lógica del modelo del negocio y clases con acceso a datos, la nomenclatura que se utilizó para establecer los estándares de programación se basa en el control correspondiente que se utilizará para la codificación abreviando el nombre del controlador y a continuación el nombre en el cual lo utilizaremos.

5.01.02 ASP.NET

ASP.NET es el fuerte principal para el desarrollo de todo tipo de aplicaciones, con la facilidad y manejabilidad que propone y gracias a su framework hace que el trabajo y el entorno de programación sea mucho más agradable y fácil de utilizar, ASP.NET es el motor de ejecución de la aplicación, debido a que en base a esta están estructurados los datos y los componentes necesarios para llevar a cabo el

funcionamiento, a más de esto posee controles web donde se coloca el código de tipo ASP: y HTML para el diseño y presentación de la página web.

5.01.02 MySQL

MySQL Workbench es un gestor de base de datos en el cual se procede a carga la base de datos de la aplicación a desarrollar, se toma en consideración la creación del modelo, la base de datos con su respectivo nombre, la creación de las diferentes tablas que contiene, sus nombres y dentro de ellas sus atributos con sus respectivos nombres, un prefijo el cual nos dará la orientación de la tabla y a continuación del prefijo se colocará el nombre del atributo que contiene dicha tabla. Ejemplo: (ClienteId, Cli_id), para la conexión a la base de datos desde el aplicativo, se utilizará el conector de MySQL, importando la librería respectiva para realizar la conexión con éxito como muestra la siguiente figura. (Figura 27)

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Veterinaria
{
    66 referencias
    public class modelo
    {
        Conexion conex = new Conexion();
    }
}
```

Figura 26 Importación de la Librería *MySql.Data* para la conexión a la base de datos

```
public class Conexion
{
    public MySqlConnection myCon;
    47 referencias
    public MySqlConnection ObtenerConexion()
    {
        try
        {
            myCon = new MySqlConnection("server=localhost; database=veterinaria; Uid=root; pwd=");
            myCon.Open();
            return myCon;
        }
        catch (Exception ex)
        {
            return null;
        }
    }
}
3 referencias
public void CerrarConexion()
{
    myCon.Close();
}
```

Figura 27 Conexión a la base de datos. Se observa el código utilizado para realizar la conexión mediante el nombre de la base de datos, usuario y contraseña.

```
public MySqlDataReader consultar_datos_usuario(String cedula,String clave)
{
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    MySqlDataReader resultData = null;
```

Figura 28 Consulta de datos de usuario para ingresar al sistema declarando las variables pertinentes y obteniendo la conexión.

```
try
{
    SecuenciaSQL = string.Format(@"SELECT * FROM
                                |usuario where seudonimo='"+seudonimo+" ' and clave='"+clave+" ' and estado='A'");
    sqlcom = new MySqlCommand(SecuenciaSQL, con);
    resultData = sqlcom.ExecuteReader();
    return resultData;
}
catch (Exception)
{
    return null;
}
finally
{
}
}
```

Figura 29 Consulta realizada a la base de datos. Se seleccionan todos los campos de la tabla usuario mediante el seudónimo del usuario y la clave para poder acceder, se escribe el código sqlcom.ExecuteReader(); necesario para la lectura de datos.

```

Conexion conex = new Conexion();
1 referencia
public int registrar_cliente_(String cedula, String nombre, String apellido, String direccion,
                             String telefono, String movil, String correo)
{
    int filas = 0;
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    try
    {
        SecuenciaSQL = string.Format(@"INSERT INTO veterinaria.clientes
                                     VALUES (null, '{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', 'A');",
                                     cedula, nombre, apellido, direccion, telefono, movil, correo);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        filas = sqlcom.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
    }
    return filas;
}

```

Figura 30 Método insertar. En el gráfico se observa la codificación utilizada para realizar la inserción de un registro hacia la base de dato, se coloca dentro de una clase, que en este caso se llama ModeloCliente.

```

protected void bt_regCliente_Click(object sender, EventArgs e)
{
    if (!ValidarCampos()) return;

    String cedula = tx_cedula.Text;
    String nombres = tx_nombres.Text;
    String apellidos = tx_apellidos.Text;
    String direccion = tx_direccion.Text;
    String telefonoFijo = tx_fijo.Text;
    String telefonoMovil = tx_movil.Text;
    String correo = tx_email.Text;
}

```

Figura 31 Declaración de variables y Validación de campos. En el gráfico se observa las variables declaradas con su tipo de dato respectivo, y los controles los cuales van a obtener el valor de dichas variables.

```

modeloCliente model = new modeloCliente();
try
{
    bool cedRepetida = false;
    var cedula_cliente = model.verificar_cliente_con_cedula_repetida(cedula);
    if (cedula_cliente.Read())
    {
        cedRepetida = true;
    }

    if (cedRepetida)
    {
        mostrarAlerta_Jscript("Cliente ya fue registrado con este número de Cédula");
        tx_cedula.Focus();
    }
    else
    {
        int c = model.registrar_cliente_(cedula, nombres, apellidos, direccion, telefonoFijo, telefonoMovil, correo);
        if (c > 0)
        {
            mostrarAlerta_Jscript("Registro Agregado");
            LimpiarC();
        }
    }
}

```

Figura 32 Código de Insertar. En el gráfico se observa el código utilizado para hacer un llamado al modelo (ModeloCliente) en donde se encuentra el método insertar, se envían los valores de las variables a cada control de objeto para guardarlos, en caso de que exista un error en ingresar el número de cédula, y esta se repita, nos mostrará un mensaje de error, y no permitirá guardar el registro.

```

try
{
}
catch (Exception)
{
    throw;
}

```

Figura 33 Código de la sentencia Try y Catch. Estas sentencias son aplicadas para atrapar posibles excepciones al momento de realizar un proceso, es decir, cuando el sistema falle por alguna causa, esta sentencia nos hará conocer en que parte del proceso se generó la excepción o el error, en el bloque try se ejecuta el proceso que se quiere realizar y en la sentencia catch se mostrará la excepción

```

protected void bt_seleccionar_Click(object sender, ImageClickEventArgs e)
{
    int id = int.Parse((sender as ImageButton).CommandArgument);
    Response.Redirect("A_ActualizarDatosClientes.aspx?encrypt=" + (id * 777));
}

```

Figura 34 Envío de datos a otro formulario. En el gráfico se muestra el código utilizado para enviar datos de un formulario a otro en este caso de formulario de clientes.

```

if (!Page.IsPostBack)
{
    String id_encrypt = Request.QueryString["encrypt"];
    Int32 id = System.Convert.ToInt32(id_encrypt) / 777;
    modeloCliente model = new modeloCliente();
    MySqlDataReader client = model.consultar_clientes_id(id);
    try
    {
        if (client.Read())
        {
            tx_codigo.Text = client.GetInt32("id_cliente").ToString();
            tx_cedula.Text = client.GetString("cedula");
            tx_nombres.Text = client.GetString("nombre");
            tx_apellidos.Text = client.GetString("apellido");
            tx_direccion.Text = client.GetString("direccion");
            tx_fijo.Text = client.GetString("telefono");
            tx_movil.Text = client.GetString("movil");
            tx_email.Text = client.GetString("correo");
        }
    }
    catch (Exception)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this, this.GetType(),
            "", "alert('Error al completar acción');", true);
    }
}

```

Figura 35 Recepción de datos desde otro formulario. En el gráfico se muestra utilizado para mostrar los datos que han sido enviados desde otro formulario en este caso del formulario de Clientes.

```

.templatemo-block { display: block; }
.templatemo-position-relative { position: relative; }
.blue-text { color: #39ADB4; }
.white-text { color: white; }
.green-bg { background-color: #39ADB4; }
.yellow-bg { background-color: #D8D138; }
.white-bg { background-color: white; }
.white-bg .fa-times { background-color: #d7d7d7; }
.white-bg .fa-times:hover { background: #83ccd1; }
.orange-bg {
    background-color: #F17A54;
    color: white;
}

```

Figura 36 Código utilizado en css para el diseño del aplicativo

5.02. Diseño de interfaces de usuario.

Tabla 19.

Ingreso al sistema.

Ítem	Representación	Descripción
A	Image	Logo
B	Label	Nombre empresa
C	Label	Iniciar sesión
D	TextBox	Nombre de Usuario
E	TextBox	Password
F	Button	Ingresar

Nota: Componentes utilizados para la interfaz de ingreso al sistema.



Figura 37. Vista del Login del aplicativo. Se visualiza el ingreso de nombre de usuario y contraseña

Tabla 20.

Vista Principal

Ítem	Representación	Descripción
A	HyperLink	Módulos Horizontal
B	HyperLink	Módulos Vertical
C	Label	Título
D	Image	Imagen

Nota: Componentes utilizados para la interfaz de la vista principal.

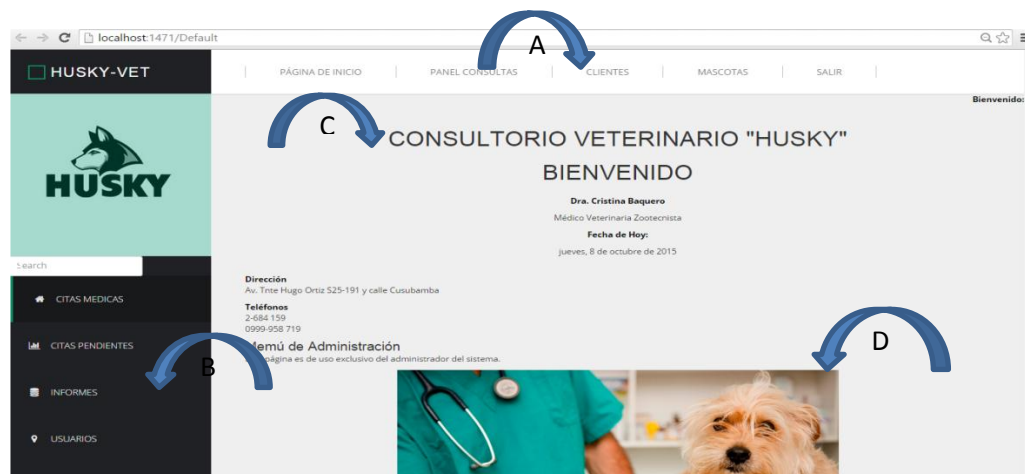


Figura 38. Vista de la página de inicio del aplicativo. Se visualiza la página principal que contiene los diferentes módulos del sistema.

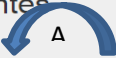
Tabla 21.

Ingreso de Clientes.



Ítem	Representación	Descripción
A	Label	Titulo
B	TextBox	Ingreso
C	Label	Etiqueta de ingreso
D	Button	Enviar

Nota: Componentes utilizados para la interfaz de ingreso de clientes.

Registrar Clientes


DATOS DE CLIENTES: 

Favor llene todos los campos, son requeridos
[Ver Listado>>>](#)

Número de Cédula:   Teléfono Convencional:

Nombres: Teléfono Móvil:

Apellidos: E-mail:

Dirección: 




Figura 39 Vista de registro de Clientes.

Tabla 22.

Ingreso de Pacientes.

Ítem	Representación	Descripción
A	TextBox	Ingreso
B	Label	Datos
C	DropDownList	Tipos
D	Button	Calcular
E	Button	Guardar

Nota: Componentes utilizados para la interfaz de la vista de ingreso de pacientes.

Figura 40. Vista de registro de Pacientes. Luego de haber hecho el registro correspondiente del propietario del paciente o cliente, se procede a tomar los datos del paciente perteneciente al mismo para registrarlo en el sistema.

Tabla 23.

Asignación de fecha y horario de cita.

Ítem	Representación	Descripción
A	Button	Buscar
B	Label	Datos
C	Button	Disponibilidad
D	Button	Horas
E	Label	Horarios

Nota: Componentes utilizados para la interfaz de la vista de Asignación de fecha y hora de cita.

INGRESE EL N° DE CÉDULA DEL PROPIETARIO DEL PACIENTE PARA REALIZAR LA BÚSCA

Cedula del Cliente: **A**

DATOS GENERALES DEL PROPIETARIO (VERIFIQUELOS ANTES DE CONTINUAR CON EL PROCESO.) **B**

Nombre del Cliente: Angélica Janeth Barreiro Cedeño Dirección: Chillogallo Teléfono: 3054457 E-mail: j.anggie@outlook.es

INGRESE LA FECHA PARA VER LA DISPONIBILIDAD DE HORARIOS **C**

Fecha: **C**

HORARIOS DISPONIBLES PARA LA FECHA INGRESADA.

No.	Horario	Estado	No.	Horario	Estado
1	08:00 - 09:00	<input type="button" value="DISPONIBLE"/> D	2	09:00 - 10:00	<input type="button" value="ASIGNADO"/> E
3	10:00 - 11:00	<input type="button" value="DISPONIBLE"/>	4	11:00 - 12:00	<input type="button" value="DISPONIBLE"/>
5	12:00 - 13:00	<input type="button" value="DISPONIBLE"/>	6	14:00 - 15:00	<input type="button" value="DISPONIBLE"/>
7	15:00 - 16:00	<input type="button" value="DISPONIBLE"/>	8	16:00 - 17:00	<input type="button" value="DISPONIBLE"/>

Figura 41. Vista de la pantalla de asignación de citas previas. Se visualiza los horarios disponibles en la fecha ingresada para proceder a asignar la cita al paciente perteneciente al propietario con el número de cédula respectivo.

Tabla 24.

Menú de Consultas.

Ítem	Representación	Descripción
A	LinkButton	Consultas
B	LinkButton	Cita Previa
C	LinkButton	Consulta Pendiente
D	LinkButton	Citas Atendidas

Nota: Componentes utilizados para la interfaz de la vista del menú de consultas.

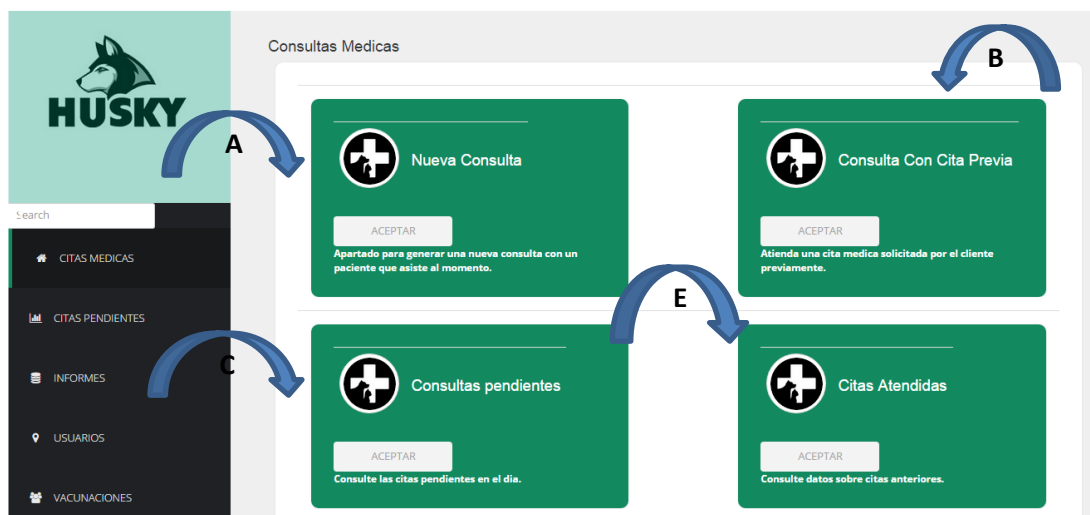


Figura 42. Vista del menú de consultas. Se visualiza las múltiples opciones que tiene el usuario en el menú de consultas.

Tabla 25.

Vista de Nueva consulta médica con cita previa.

Ítem	Representación	Descripción
A	Image	Paciente
B	Label	Datos
C	TextBox	Ingreso
D	Button	Enviar

Nota: Componentes utilizados para la interfaz de la vista de nueva consulta médica con cita previa.

The screenshot shows a web form for creating a new medical consultation. It is divided into several sections: 'Datos del Propietario', 'Datos de la mascota tratada', 'Datos Médicos', and 'Observaciones'. Annotations A, B, C, and D point to specific elements: A points to the pet image, B points to the owner's ID field, C points to the 'Motivo de Consulta' text area, and D points to the 'Finalizar Consulta' button.

Datos del Propietario:

Nombre del propietario: Damian Benjamin Navarrete Chávez Cédula: 1719458075 Dirección: La Gonzáles Suárez

Datos de la mascota tratada:

Código: 4 Fecha de Nacimiento: 04/04/2015 Fecha de atención:
 Nombre de la(s) mascota(s): Blanca Especie: Canino Raza: French Poodle

Datos Médicos

Peso (Kg): Temperatura: Frecuencia Cardíaca:
 Palpación: Frecuencia respiratoria:

Motivo de Consulta: Ingrese el motivo por el que asiste

Síntomas: Ingrese detalladamente los síntomas del paciente tratado...

Diagnostico: Ingrese el diagnóstico del paciente tratado...

Tratamiento: Ingrese el tratamiento que se enviará al paciente tratado...

Receta: Ingrese la receta médica que será prescrita para el paciente tratado...

Observaciones: Ingrese las observaciones adicionales, de la consulta realizada al paciente tratado...

Buttons: Finalizar Consulta, Imprimir Historia, Cancelar

Figura 43 Vista de Nueva consulta médica. Se visualiza los campos necesarios que contiene la vista de nueva consulta

Tabla 28.

Lista de Pacientes.

Ítem	Representación	Descripción
A	Label	Título
B	DropDownList	Tipo
C	TextBox	Ingreso
D	Button	Buscar
E	GridView	Datos
F	ImageButton	Ficha Médica

Nota: Componentes utilizados para la interfaz de la vista de lista de pacientes.

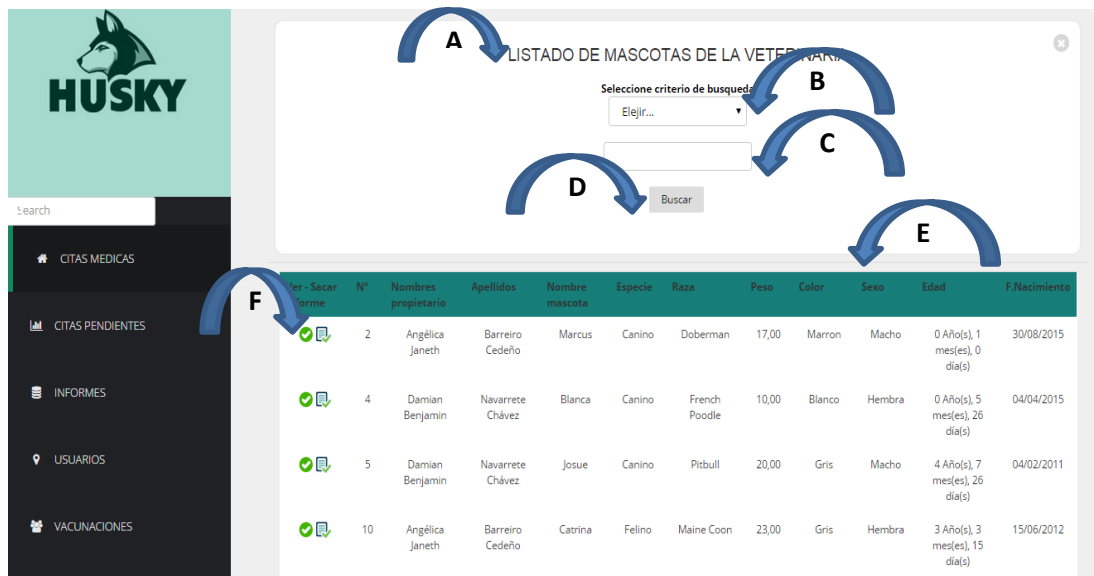


Figura 44. Vista de Búsqueda de pacientes y Fichas médicas. En la opción de registros médicos se visualizan los todos pacientes de la Clínica, en donde se puede buscar algún registro deseado y de esta poder obtener su ficha médica.

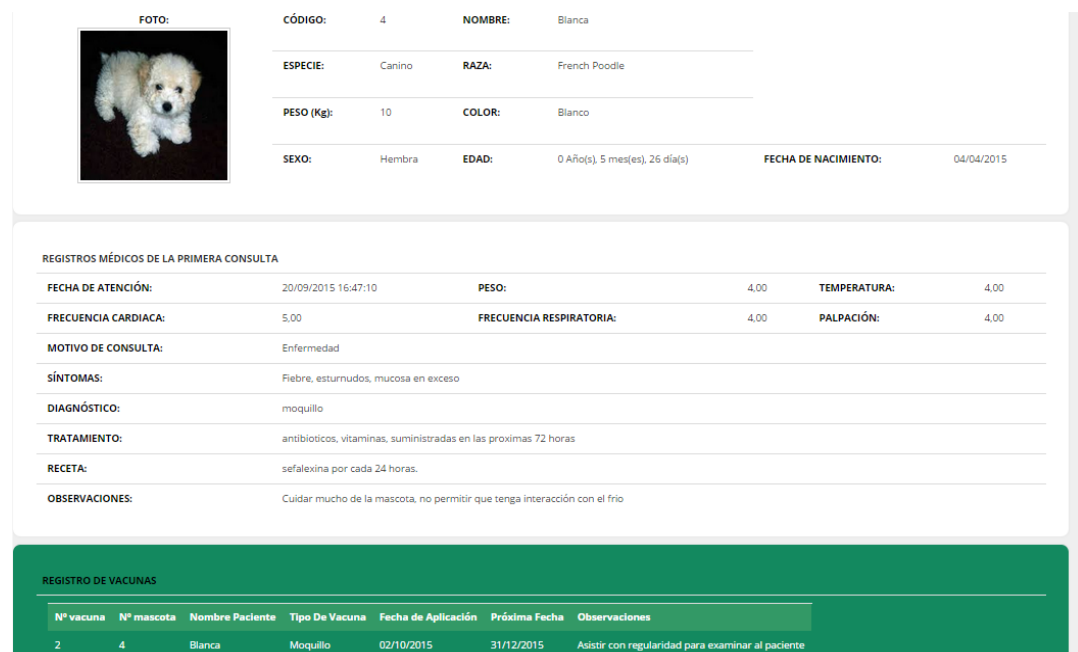


Figura 45 Vista de la Ficha médica. Se visualiza la ficha médica del paciente seleccionado de los registros médicos

5.03. Especificación de pruebas de unidad

Tabla 27.

Especificación de prueba de unidad PU001, guardar un registro.

Identificador de la prueba:	PU001
Método a probar	Guardar
Objetivo de la prueba:	Se realizara para observar el registro óptimo de los pacientes
Datos de Entrada	
Se Deberán ingresar los datos correspondientes en los campos de texto de los diferentes formularios	
Resultado Esperado	
Visualizar los pacientes registrados	
Comentarios	
El método guardar se debe tener en cuenta todos los campos a ingresar	

Nota: Especificación de prueba de unidad PU001

Tabla 28.

Especificación de prueba de unidad PU002, Seleccionar un registro.

Identificador de la prueba:	PU002
Método a probar	Seleccionar
Objetivo de la prueba:	Se realizara para escoger el registro que va a ser modificado o eliminado.
Datos de Entrada	
Deberán elegir los datos del campo de texto existentes en la pagina	
Resultado Esperado	
Visualizar que los campos estén remplazados	
Comentarios	
El método seleccionar, selecciona uno por uno y refleja los datos en los respectivos campos de texto	

Nota: Especificación de prueba de unidad PU002

Tabla 29.

Especificación de prueba de unidad PU003, Actualizar un registro.

Identificador de la prueba:	PU003
Método a probar	Actualizar
Objetivo de la prueba:	Se realizara para una revisión de los datos de todas las paginas
Datos de Entrada	
Selecciona el método actualizar y verifica los datos	
Resultado Esperado	
Visualizar los datos de los campos actualizados	
Comentarios	
El método actualizar ,actualiza según los cambios que exista en cada pagina	
<i>Nota: Especificación de prueba de unidad PU003</i>	

Tabla 30.

Especificación de prueba de unidad PU004, Eliminar un registro.

Identificador de la prueba:	PU004
Método a probar	Eliminar
Objetivo de la prueba:	Se realizara para eliminar registros no deseados
Datos de Entrada	
Deberán elegir los datos a eliminar	
Resultado Esperado	
Visualizar que los campos estén descartados en el sistema	
Comentarios	
El método eliminar ,antes de completar la acción muestra la confirmación de eliminación porque no existe una opción para recuperar lo eliminado	
<i>Nota: Especificación de prueba de unidad PU004</i>	

Tabla 31.

Especificación de prueba de unidad PU005, Buscar un registro.

Identificador de la prueba:	PU005
Método a probar	Buscar
Objetivo de la prueba:	Se realizara para poder encontrar los registros que se va a gestionar en el sistema
Datos de Entrada	
Se Deberá ingresar el criterio de búsqueda para obtener los datos.	
Resultado Esperado	
Visualizar el registro que se está buscando	
comentarios	
El método buscar, muestra en los campos la información correspondiente a la búsqueda realizada.	

Nota: Especificación de prueba de unidad PU005

5.04. Especificación de pruebas de aceptación

Tabla 32.

Especificación de prueba de aceptación PA001

Identificador de la prueba:	PA001
Caso de Uso:	CU001-Nueva ficha clínica
Tipo de usuario:	Cliente
Objetivo de la Prueba:	Analizar los requerimientos necesarios para el registro de antecedentes médicos del paciente.
Secuencia de eventos:	
Se tomará los datos correspondientes del paciente	
Resultados esperados:	
Registro del paciente en el sistema	
Comentarios:	
El registro debe realizarse con éxito	
Estado: Aceptado	

Nota: Especificación de prueba de aceptación PA001 Caso de uso 001.

Tabla 33.

Especificación de prueba de aceptación PA002

Identificador de la prueba:	PA002
Caso de Uso:	CU002-Anamnesis
Tipo de usuario:	Administrador
Objetivo de la Prueba:	Analizar los requerimientos necesarios para la realización de recopilación de síntomas del paciente
Secuencia de eventos:	Se realizará la recopilación de datos médicos del paciente.
Resultados esperados:	Registro de una nueva ficha clínica del paciente respectivo en el sistema
Comentarios:	El registro debe realizarse con éxito
Estado: Aceptado	

Nota: Especificación de prueba de aceptación PA002 Caso de uso 002.

Tabla 34.

Especificación de prueba de aceptación PA003.

Identificador de la prueba:	PA003
Caso de Uso:	CU003-Vacunación
Tipo de usuario:	Administrador
Objetivo de la Prueba:	Analizar los requerimientos necesarios para la realización del proceso de vacunación que se aplicará al paciente.
Secuencia de eventos:	Se realizará la recopilación de información, y antecedentes médicos
Resultados esperados:	Registro de vacunación aplicada al paciente en el sistema
Comentarios:	El proceso debe realizarse con éxito
Estado : Aceptado	

Nota: Especificación de prueba de aceptación PA003 Caso de uso 003.

Tabla 35.

Especificación de prueba de aceptación PA004.

Identificador de la prueba:	PA004
Caso de Uso:	CU003-Próxima Cita
Tipo de usuario:	Administrador
Objetivo de la Prueba:	Analizar los requerimientos necesarios para la realización del proceso asignación de citas.
Secuencia de eventos:	Después de haber realizado los registros respectivos de gestión médica, se asigna una nueva cita médica al paciente
Resultados esperados:	Registro de una nueva cita médica asignada al paciente de manera obligatoria.
Comentarios:	El proceso debe realizarse con éxito.
Estado: Aceptado	

Nota: Especificación de prueba de aceptación PA004 Caso de uso 004.

5.05. Especificación de pruebas de carga

En esta sección lo que se procederá a realizar es verificar el funcionamiento y el rendimiento del sistema sea el más apto para ponerlo en marcha, también se verificará el tiempo de respuesta que tiene el sistema para la ejecución de las diferentes transacciones que debe realizar. Para esto se utilizará un software adecuado llamado LoadUIWeb, el cual es especializado para utilizarlo en este tipo de pruebas, ayudará a facilitar la comprobación y servirá para poner a prueba el rendimiento del aplicativo que se desarrolló, se realizará la prueba empezando por un usuario, para posteriormente hacerlo con múltiples usuarios, de esta manera se comprobará el funcionamiento del aplicativo.

Tabla 36.

Especificación de prueba de carga PC001.

Identificador de la prueba:	PC001
Tipo de Prueba:	Simulación de Ingreso al sistema y realización de gestiones aplicadas al software
Objetivo de la Prueba:	Comprobar el rendimiento del software y verificar que se haya completado sin errores.
Descripción	
La prueba se realizó empezando con un usuario.	
Resultados esperados:	
Rendimiento del sistema en buenas condiciones sin presentar errores.	
Comentarios:	
La prueba de carga no ha presentado errores graves.	

Nota: Especificación de prueba de carga PC001.

Tabla 37.

Especificación de prueba de carga PC002.

Identificador de la prueba:	PC002
Tipo de Prueba:	Simulación de Ingreso al sistema y realización de gestiones aplicadas al software
Objetivo de la Prueba:	Comprobar el rendimiento del software con 10 usuarios virtuales, verificar que se haya completado sin errores
Descripción	
La prueba se realizó con 15 usuarios virtuales.	
Resultados esperados:	
Rendimiento del sistema en buenas condiciones sin presentar errores.	
Comentarios:	
La prueba de carga no ha presentado errores graves.	

Nota: Especificación de prueba de carga PC002.

Tabla 38.

Especificación de prueba de carga PC003.

Identificador de la prueba:	PC003
Tipo de Prueba:	Simulación de Ingreso al sistema y realización de gestiones aplicadas al software
Objetivo de la Prueba:	Comprobar el rendimiento del software con 20 usuarios virtuales, verificar que se haya completado sin errores
Descripción	
La prueba se realizó con 20 usuarios virtuales, los cuales realizan el mismo escenario, realizan el proceso de guardado, modificado, eliminado y búsqueda, en un formulario web.	
Resultados esperados:	
Rendimiento del sistema en buenas condiciones sin presentar errores.	
Comentarios:	
La prueba de carga no ha presentado errores graves.	

Nota: Especificación de prueba de carga PC003.

5.06. Configuración del ambiente mínima/ideal.

La mayoría de las aplicaciones tienen requisitos previos: algunos componentes como .NET Framework en tiempo de ejecución deben estar disponibles en un equipo de destino para que se pueda ejecutar la aplicación. Las herramientas de implementación de Visual Studio incluyen la capacidad de detectar automáticamente la existencia de componentes durante la instalación e instalan un conjunto de requisitos previos predeterminado. Este proceso recibe el nombre de arranque y el componente que lo realiza se denomina arranque genérico. El arranque utiliza paquetes de programa previo para instalar los componentes de requisito previo en el equipo de destino (Microsoft, 2015). (ANEXO A.07)

Capítulo VI. Aspectos Administrativos.

6.01. Recursos.

En esta sección se describirá detalladamente los recursos necesarios los cuales fueron utilizados para el desarrollo del proyecto, que son materiales, humanos y tecnológicos.

Tabla 39.

Detalle de Recursos Humanos.

Descripción	Nombre	Actividad	Responsabilidad
Tutor	Lic. Wilson Núñez	Tutor del proyecto de titulación	Responsable de brindar su ayuda y orientación para el desarrollo del trabajo escrito del proyecto de titulación
Autor del Proyecto, Programador	Alumno Damián Navarrete	Autor del proyecto de titulación	Responsable de llevar a cabo el trabajo completo, tanto escrito como práctico del proyecto de titulación.

Nota: Se visualizan los Recursos humanos.

Tabla 40.

Detalle de Recursos Materiales utilizados.

Descripción	Cantidad	Valor Unitario	Valor Total
Ordenador	1	800,00	800,00
Hojas de papel bond	500	0,06	30,00
Impresiones	120	0,15	18,00
Transporte	70	0.25	17,50
Internet	7 Meses	24,00	168,00
Alimentación	70	2,50	175,00
Luz eléctrica	7 Meses	18,00	126,00
			TOTAL: 1334,50

Nota: Se visualiza los Recursos materiales utilizados.

Tabla 41.

Detalle de Recursos a nivel de Hardware.

Cantidad	Tipo	Descripción
1	Ordenador	Marca: Lenovo Procesador: Intel™ Core™ i5 CPU 2.60GHz RAM: 8Gb Capacidad : 550Gb

Nota: Se visualiza los Recursos a nivel de Hardware.

Tabla 42.

Detalle de Recursos a nivel de Software.

Recurso
Sistema Operativo de 64 Bits, Windows 8 Pro
Visual Studio 2013
WampServer 2.2
MySQL Workbench 6.3

Nota: Se visualiza los Recursos a nivel de Software.

6.02. Presupuesto.

Tabla 43.

Detalle del presupuesto y gastos incluidos.

Descripción	Cantidad	Valor Unitario	Valor Total
Ordenador	1	800,00	800,00
Hojas de papel bond	500	0,06	30,00
Impresiones	120	0,15	18,00
Transporte	70	0.25	17,50
Internet	7 Meses	24,00	168,00
Alimentación	70	2,50	175,00
Luz eléctrica	7 Meses	18,00	126,00
Taller de Profesionalización	1	780,15	780,15
Asignación de Tutor	1	150	150,00
Anillados	1	2,50	2,50
Empastados	1	6	6,,00
			TOTAL: 2273,15

Nota: Se visualiza el Presupuesto y gastos incluidos.

6.03. Cronograma.

Se mostrará el cronograma elaborado, el cual muestra los días de duración de cada actividad, los recursos utilizados y el tiempo total establecido para el desarrollo del proyecto.

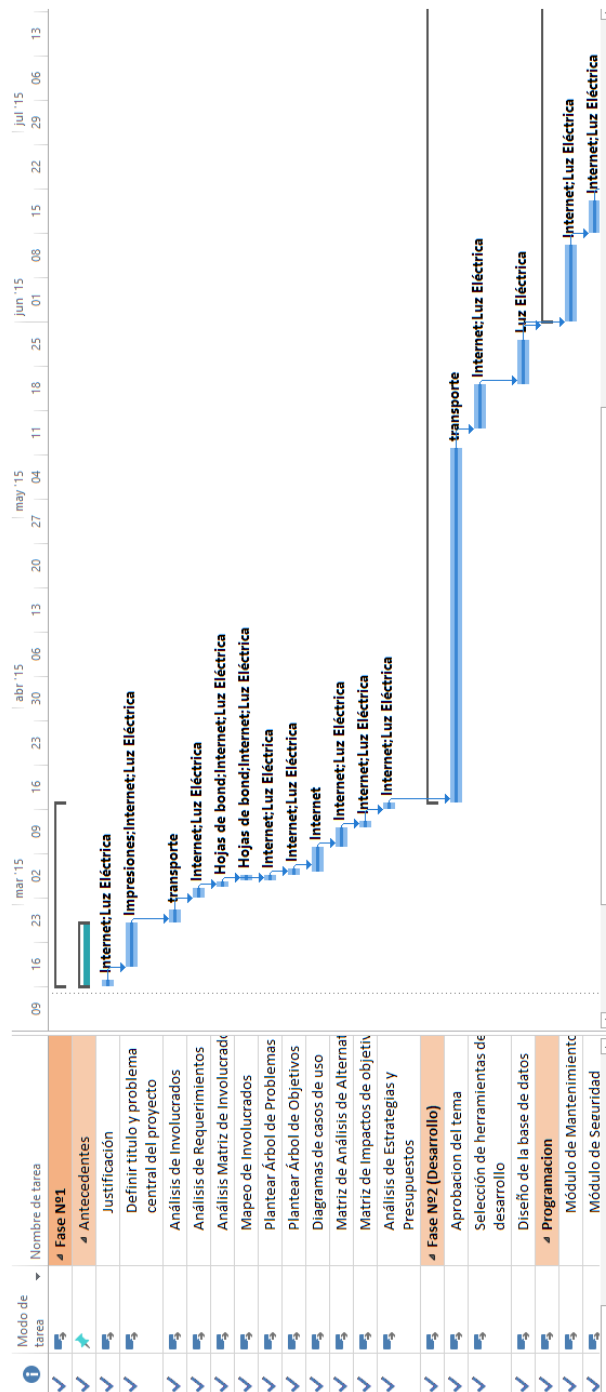


Figura 46 Vista del cronograma planificado para el proyecto realizado en Microsoft Visio 2010

Capítulo VII. Conclusiones y Recomendaciones.

7.01. Conclusiones.

- Culminado el presente proyecto de titulación, se pretende mejorar la calidad del servicio prestado por la empresa para la cual se está desarrollando el software, innovando los procesos principales los cuales lleva a cabo la empresa de una manera no sistematizada.
- El objetivo por el cual fue creado este software, es aportar y solucionar los problemas y la situación actual en la que se encuentra la empresa, y para ello fue necesario el compartimiento de información necesaria para que se haya podido llevar a cabo de manera exitosa.
- Con el desarrollo del proyecto, se ha venido adquiriendo mucho conocimiento y se ha aplicado muchas destrezas las cuales se van desarrollando mientras se realiza el trabajo.

7.02. Recomendaciones.

- Una vez implementado el software de gestión de fichas médicas veterinarias, se debe llevar un cuidado periódico del sistema evitando su mal uso
- Se debe dar el uso adecuado del software para fines de mejora para la empresa, mas no para otro tipo de fines, el objetivo es mejorar la calidad para que los clientes se sientan satisfechos.
- Para conveniencia del usuario del sistema, es primordial que la persona que administre el software tenga claro que no se debe administrar información adicional sobre el mismo y se tomen las debidas precauciones para evitar vulnerabilidades que puedan afectar al software.

Anexos.

Anexo A.01 Matriz de Involucrados

ANÁLISIS DE MATRIZ DE INVOLUCRADOS

Involucrados	Interés en el problema	Problemas Percibidos	Recursos y Mandatos	Interés en el proyecto	Conflictos y Cooperaciones
Médico Veterinaria	Realizar los procesos de consulta médica de manera sistematizada sin pérdida de tiempo	Equivocación y duplicidad de las fichas médicas veterinarias	Técnicos y tecnológicos	Promover un mejor estilo de vida para sus clientes y mascotas	Cooperación entre los actores que se involucran en el proyecto
	Es la interesada en guardar los registros de sus clientes y mascotas de manera más ágil.	Desconocimiento del proceso de registro y gestión veterinaria	Métodos adecuados para realizar el proceso		
Mascota Paciente	La mascota del propietario, es el actor principal mediante el se podrán obtener la información médica para posteriormente llevar el registro médico con su respectiva ficha médica.	Inasistencia a la consulta médica	Asistir periódicamente a su respectiva consulta médica	Ser el objetivo esencial para la solución al problema	Cooperación por parte del médico veterinario y propietario para cumplir la consulta médica
Bodeguero	Se encarga de recibir los productos médicos de la veterinaria para almacenarlos en la bodega y administrarlos.	Provisión escasa de productos	técnicos	Colaborar en el aspecto de la salud dentro de la clínica Veterinaria para llevar a cabo una gestión veterinaria adecuada	Cooperación entre los actores que se involucran en el proyecto

Nota: Se visualiza el interés en el problema, problemas percibidos, recursos y mandatos, interés en el proyecto, conflictos y cooperaciones de cada involucrado.

Anexo A.02 Matriz de análisis de alternativas

ESTRATEGIAS	VARIABLES	NECESIDAD DE RECURSOS			FACTIBILIDAD POLITICA	DURACION DE PROYECTO	PUNTAJE	PRIORIDAD
		HUMANOS	TECNICOS	FINACIEROS				
Desarrollar una aplicación informática eficaz, que controle y gestione los procesos necesarios a través del usuario, para que de esta manera se pueda alcanzar una unificación y optimización de los informes médicos que se almacenan, cada vez que un paciente asiste a consulta								
		4	5	4	5	4	22	ALTO
Llevar un control adecuado, conociendo con plenitud los procesos de tratamientos médicos y de gestión veterinaria, para obtener resultados confiables, y beneficiando de gran manera a los pacientes que acuden a la clínica.								
		4	5	4	4	4	20	MEDIO-ALTO
Mantener una organización adecuada, para la socialización de los procesos de registro y de gestión veterinaria que se llevan a cabo en la clínica, impulsando a mejorar la atención que se brinda a los pacientes obteniendo resultados verídicos y que proporcionen confianza a los clientes.								
		4	2	3	3	2	14	MEDIO BAJO – MEDIO
Proporcionar de manera efectiva a los profesionales de la salud de la clínica, una herramienta eficiente, rápida y segura capaz de reducir y facilitar el tiempo de proceso de registro y el seguimiento de la gestión realizada a sus pacientes.								
		3	1	2	4	2	12	MEDIO BAJO – MEDIO

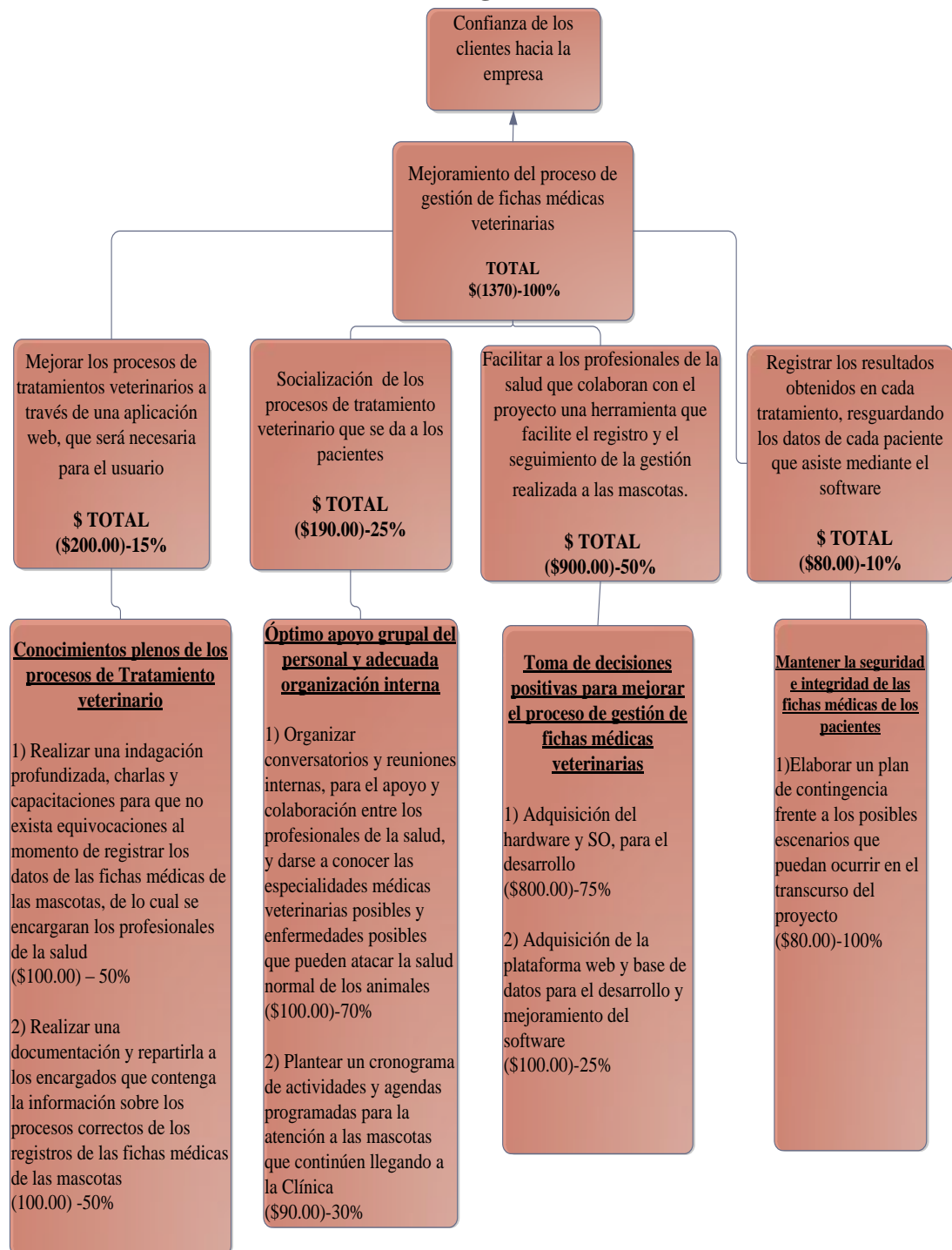
Nota: Se visualiza el desarrollo de la matriz de alternativas.

Anexo A.03 Matriz de impacto de objetivos

OBJETIVOS	FACTIBILIDAD A LOGRARSE	IMPACTO DE GÉNERO	IMPACTO AMBIENTAL	RELEVANCIA	SOSTENIBILIDAD
Llevar un control adecuado, conociendo con plenitud los procesos de registros médicos y de gestión veterinaria, para obtener resultados confiables, beneficiando de gran manera a los pacientes que acuden a la clínica.	Es de gran beneficio y un punto muy eficaz conocer los procesos de registros médicos y gestión veterinaria, para poder lograr un control adecuado de las historias clínicas de los pacientes y su tratamiento a aplicarse.	Intervendrán en el proceso hombres y mujeres	EL desarrollo y la ejecución del proceso dependerá del uso de material reciclable y biodegradable, dando una utilidad responsable sin exceso de material que	Conocimientos plenos de los procesos de registros médicos y gestión veterinaria para obtener resultados que sean verídicos, y diagnosticar a los pacientes.	Para mantener un proceso óptimo los profesionales de la salud involucrados en el proceso, deben reflejar unos conocimientos totales de los procesos de gestión veterinaria y control de fichas médicas correspondientes.
Mantener una organización adecuada, para la socialización de los procesos de registro y de gestión veterinaria que se llevan a cabo en la clínica.	Una organización apropiada para aplicar los procedimientos necesarios para el proceso de registro y gestión veterinaria, impulsará a mejorar la atención brindada por la clínica a los pacientes, obteniendo resultados verídicos y que proporcionen confianza a los clientes	Intervendrán en el proceso hombres y mujeres	EL desarrollo y la ejecución del proceso dependerá del uso de material reciclable y biodegradable, dando una utilidad responsable sin exceso de material que afecte al medio ambiente	Mantener la organización interna de la clínica, adecuando correctamente los procesos de gestión veterinaria y control de fichas médicas de los pacientes correctamente	El personal que participa en el proceso de gestión de fichas médicas veterinarias contará con una planificación adecuada que permita la actualización permanente de la información de los pacientes para generar los resultados necesarios.

<p>Proporcionar de manera efectiva a los profesionales de la salud de la clínica, una herramienta eficiente, rápida y segura capaz de reducir y facilitar el tiempo de proceso de registro y el seguimiento de la gestión realizada a sus pacientes.</p>	<p>Facilitar a los profesionales de la salud de la clínica, un sistema con las herramientas necesarias que les permita llevar el control de todos los procesos de gestión veterinaria, obviando todos los procesos manuales y agilitándolos, reduciendo el tiempo de los procesos mediante esta aplicación.</p>	<p>Intervendrán en el proceso hombres y mujeres</p>	<p>EL desarrollo y la ejecución del proceso dependerá del uso de material reciclable y biodegradable, dando una utilidad responsable sin exceso de material que</p>	<p>Mejorar la calidad de vida de los pacientes y clientes, y a su vez mejorando la atención que se da en la clínica</p>	<p>Con el desarrollo y el mejoramiento de los procesos de gestión de fichas médicas veterinarias, mejorará sustentablemente y ayudará de gran manera a las entidades médicas que prestan sus servicios de atención médica, satisfaciendo las necesidades que tienen los clientes.</p>
--	---	---	---	---	---

Anexo A.04 Matriz de análisis de estrategias



Descripción: En este esquema se analiza de manera detallada, los presupuestos que serán invertidos durante todo el desarrollo del proyecto.

Anexo A.05 Matriz de marco lógico

OBJETIVO	INDICADOR	MEDIOS DE VERIFICACION	SUPUESTOS
Administrar el programa de administración de la medicación que se encuentra en la ficha médica para curar cualquier tipo de enfermedad de la mascota y aún mejor poder rescatar su vida	1.- Alcanzar un nivel Óptimo de profesionalismo 2.- Buen prestigio de la Clínica Veterinaria	1.- Se verifica en el porcentaje de mascotas que han sido sanadas y rescatadas de enfermedades catastróficas 2.- Excelente sistematización e innovación del sistema utilizado para agilizar el proceso de registro de fichas médicas veterinarias	Confianza de los usuarios hacia la clínica veterinaria, Nivel óptimo de profesionalismo, brindando la mejor atención a los pacientes con rapidez y seguridad.
Mejorar el proceso de registro de fichas médicas veterinarias	1.-Realizar el programa de administración de la medicación de la ficha médica de manera adecuada 2.- Resultados de los diagnósticos totalmente confiables 3.- Gestión de registro y seguimiento eficiente, rápido y seguro	1.- Se verifica en el porcentaje de mascotas que han sido sanadas y rescatadas por medio del programa de administración de medicación de la ficha 2.- Herramientas que faciliten el registro y el seguimiento de la gestión al 100%	
Tener el apoyo grupal del personal y conocimientos plenos para la toma de decisiones correctas y así poder mejorar el proceso de registro de fichas médicas	1.- Apoyo grupal 2.- Conocimientos plenos de los procesos de registro médico 3.- toma de decisiones correcta para mejorar el proceso de registro de fichas médicas veterinarias	Estadística de las mascotas rescatadas de enfermedades catastróficas Lista de las mascotas no rescatadas de enfermedades catastróficas	
<u>Conocimientos plenos de los procesos de registros médicos</u>	<u>Coste Total (\$550.00)</u>	Profesionales de la salud que se encargan de administrar el programa de medicación a las mascotas	
1.- Realizar una indagación profundizada, charlas y capacitaciones para	Coste (\$300.00) – 80%		

que no exista equivocaciones al momento de registrar los datos de las fichas médicas de las mascotas, de lo cual se encargaran los profesionales de la salud		
2.- Realizar una documentación y repartirla a los encargados que contenga la información sobre los procesos correctos de los registros de las fichas médicas de las mascotas	Coste (\$250.00) – 20%	
<u>Óptimo apoyo grupal del personal y adecuada organización interna</u>	<u>Coste total (\$830.00)</u>	Organización de manera adecuada de cada área de la clínica, mejorando la atención y llevando los registros correctos de cada paciente para un correcto control de la información de los mismos.
1.- Organizar conversatorios y reuniones internas, para el apoyo y colaboración entre los profesionales de la salud, y darse a conocer las especialidades médicas veterinarias posibles y enfermedades posibles que pueden atacar la salud normal de los animales	Coste (\$600.00)-70%	
2.- Plantear un cronograma de actividades y agendas programadas para la atención a las mascotas que continúen llegando a la Clínica	Coste (\$230.00)-30%	

<u>Toma de decisiones positivas para mejorar el proceso de registro de fichas médicas veterinarias y gestión veterinaria</u>	<u>Coste Total (\$2000.00)</u>	Obtención de fichas médicas veterinarias de manera óptima y rápida al alcance del médico veterinario y contar con la información necesaria para la toma de decisiones adecuadas.	
1.- Adquisición del Hardware y SO, para el desarrollo	Coste (\$800.00)-25%		
2.- Adquisición de la plataforma web y base de datos para el desarrollo y mejoramiento del software	Coste (\$1200)-75%		

Nota: Se visualiza el objetivo, identificador, medios de verificación y supuestos para el proyecto.

Anexo A.06 Detalle de Estándares de programación

Tipo de Controlador	Prefijo	Ejemplo	Descripción
Label	lb	lb_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
Textbox	tx	Tx_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto el control.
DropDownList	ddl	ddl_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
Button	bt	bt_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
LinkButton	lnk	lnk_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
ImageButton	bt	bt_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
HyperLink	hpk	Hpk_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
Image	img	img_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
FileUpload	flp	flp_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
Table	tb	tb_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.
GridView	grd	grd_ejemplo	La nomenclatura implementada empieza con letra minúscula o mayúscula seguido de la abreviación del objeto.

Nota: Se visualiza los estándares de programación utilizados.

Anexo A.07 Descripción de configuración de ambiente mínima/ideal

Recurso	Ambiente Mínimo	Ambiente Ideal
SERVIDOR	Microprocesador: Intel(R) Core(TM) 2 DUO (Microprocesador de doble núcleo, mínimo para la implementación del aplicativo)	Microprocesador: Intel(R) Core(TM) i5 CPU 2.60GHz, (Microprocesador de 4 núcleos, ideal para la implementación y puesta en marcha del sistema)
	Memoria RAM: 4GB, 2GB es útil para la velocidad en los procesos que realice	Memoria RAM: 6GB, 8GB, ideal para la velocidad y agilidad entre los procesos que se desarrollan
	Disco Duro: Disco Duro SAS con un mínimo de 20 GB libres, se utilizan estas características debido al espacio que ocupa la base de datos y la plataforma de desarrollo.	Disco Duro: Disco Duro SAS con un espacio de 1Tb, ideal para el almacenamiento.
	Monitor: SVGA(1028 x 768)	Monitor: SVGA(1028 x 1024)
	Sistema Operativo: Windows 2008 server R2, Windows, Windows XP, Windows 7, Windows 8. Lo más conveniente es utilizar Windows 2008 Server de 32 Bits.	Sistema Operativo: Windows 2008 Server R2, 64 Bits, sería el más óptimo para un servidor en donde la magnitud de alojamiento de datos es extensa.
	Tarjeta de red: 100Mb	Tarjeta de red : 400Mb. 1Gb
COMPUTADOR	Microprocesador: Intel Pentium IV 2.0GHz	Microprocesador: Intel(R) Core(TM) i5/i7 AMD A10 7400p.
	Memoria RAM: 2Gb, 4Gb	Memoria RAM: 6Gb, 8Gb
	Disco Duro: Disco Duro SAS o SATA con un mínimo de 20 GB	Disco Duro: Disco Duro SAS con un espacio de 1Tb
	Sistema Operativo: Windows XP/7. 32 Bits	Sistema Operativo: Windows XP/7/8 64 Bits
	Tarjeta de Red: 100Mb	Tarjeta de Red: 1Gb

Nota: Descripción del ambiente mínima/ideal.

Anexo A.08.

Manual De Instalación

Instalación de MySQL Workbench 6.3.

Para realizar la instalación de MySQL Workbench primero nos dirigiremos a la página principal de MySQL en donde descargaremos el producto para su instalación.

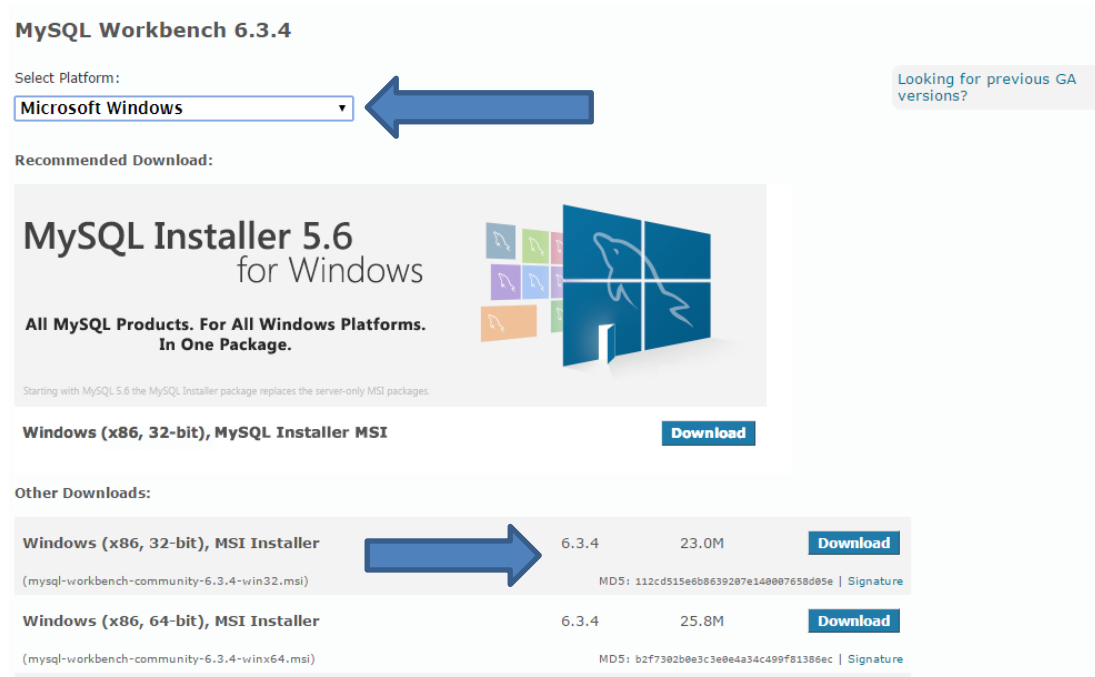


Figura 46. Descarga de MySQL Workbench desde la página oficial de MySQL. Primero se debe seleccionar la plataforma para la que necesitamos y se procede a descargar.

Una vez descargado el producto, el siguiente paso que realizaremos es ejecutarlo.

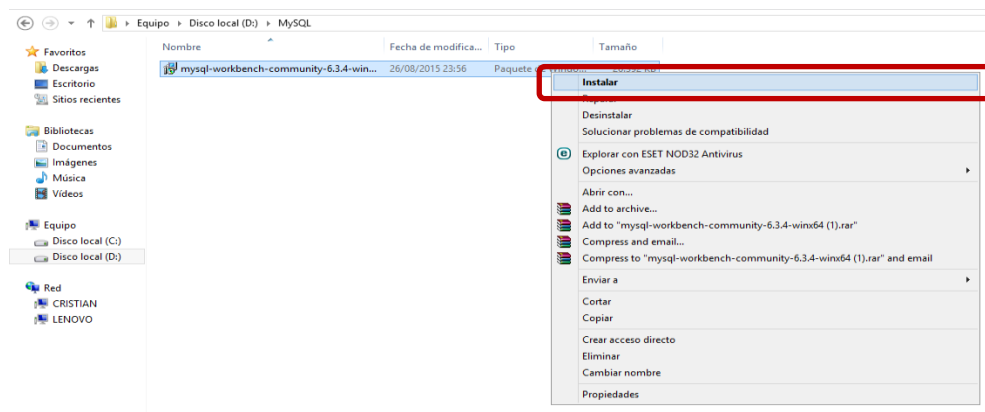


Figura 47. Ejecución del programa para su instalación.

Se abrirá un mensaje en el cuál confirmará si deseamos ejecutar el archivo, presionamos en aceptar.

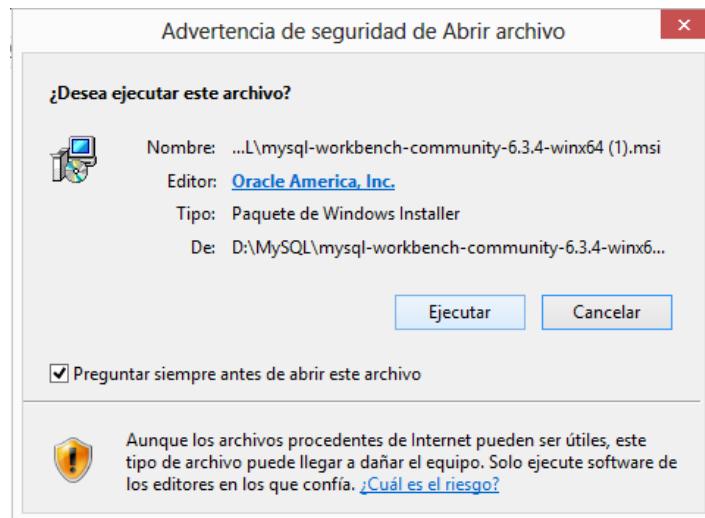


Figura 48. Mensaje de confirmación para proceder con la instalación.

Después de aceptar la instalación, se abrirá el programa para iniciar la instalación correspondiente, presionamos en siguiente.



Figura 49. Pantalla de inicio para la instalación de MySQL Workbench.

Escogemos la carpeta de destino en donde se instalará el programa, en este caso se mantendrá la configuración por defecto (C:\Program Files\MySQL\MySQL Workbench 6.3 CE).

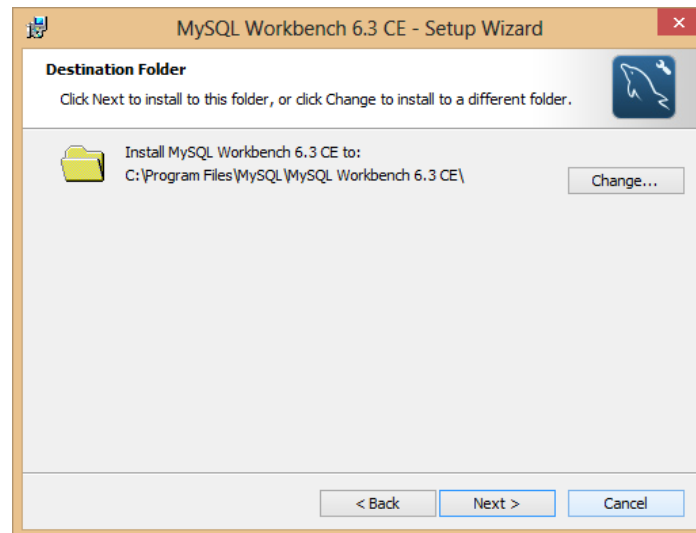


Figura 50. Carpeta de destino de instalación del producto.

Escogemos el tipo de instalación que se requiere para el usuario, en este caso se escogerá la versión completa.

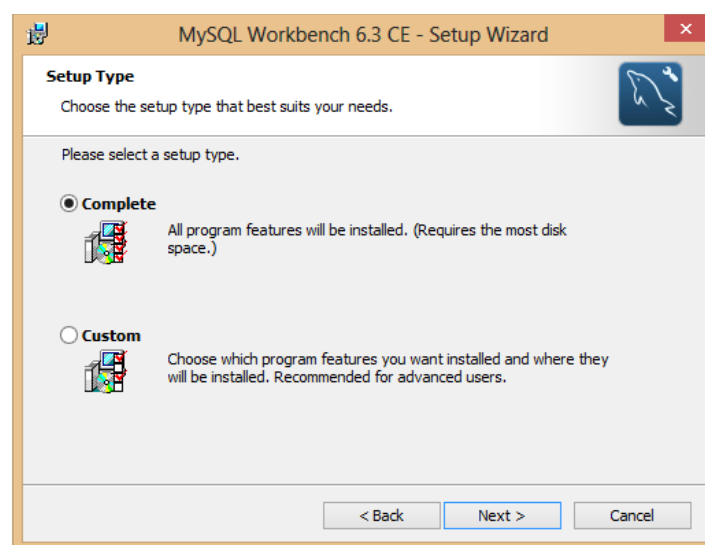


Figura 50. Carpeta de destino de instalación del producto.

A continuación presionamos en Instalar para proceder.

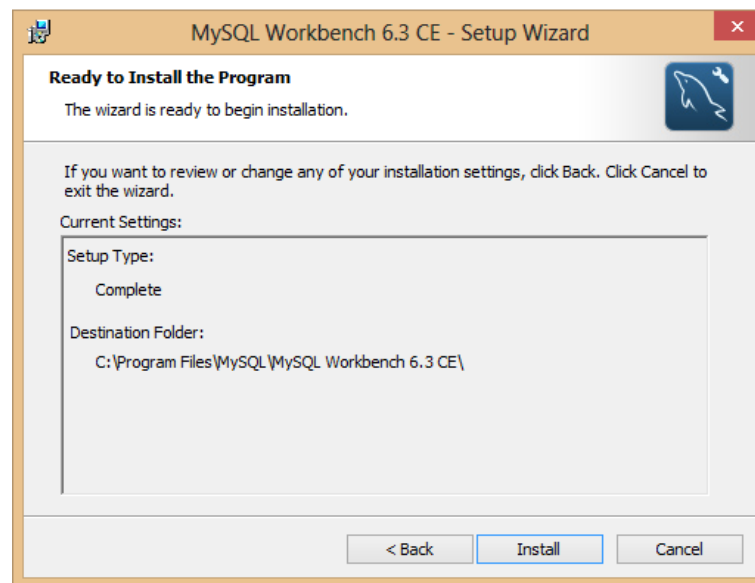


Figura 51. Instalación del producto.

En el siguiente gráfico se observa el proceso de instalación y todos los componentes necesarios para completar la instalación.

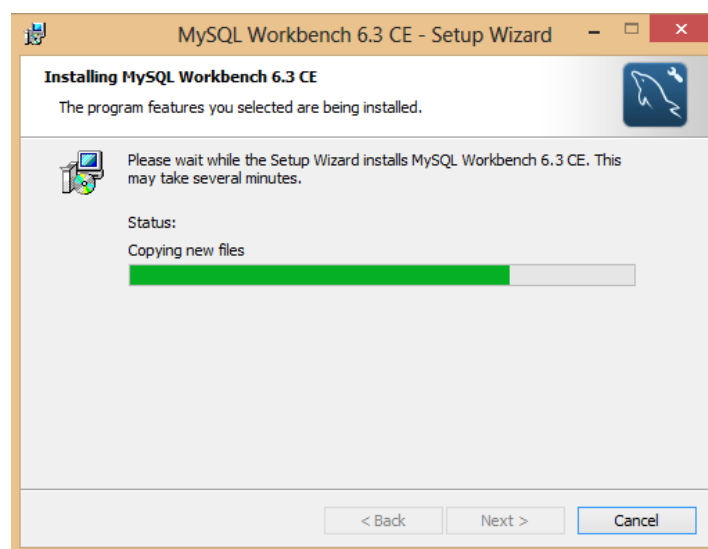


Figura 52. Proceso de instalación.

Seguido de esto continuamos presionando en el botón finalizar para terminar con la instalación.

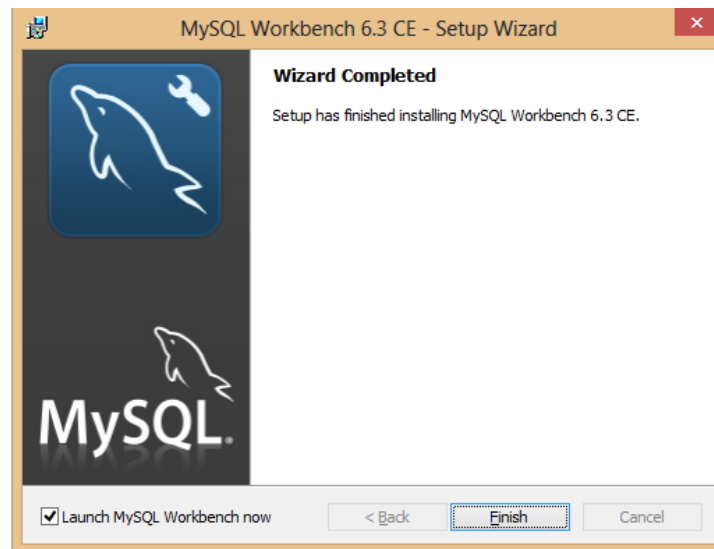


Figura 53. Finalización de la instalación.

Producto listo para ser utilizado por la persona que lo requiera.

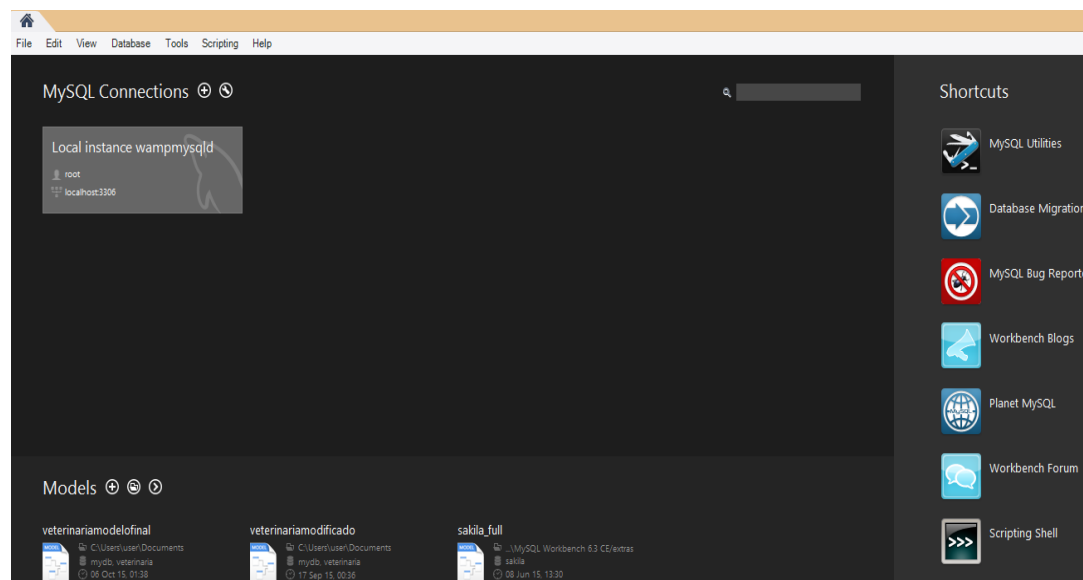


Figura 54. Pantalla de inicio de MySQL Workbench.

Instalación de WampServer 2.2.

Para realizar la instalación de WampServer hacemos click derecho sobre el ícono y a continuación en “Ejecutar como Administrador”

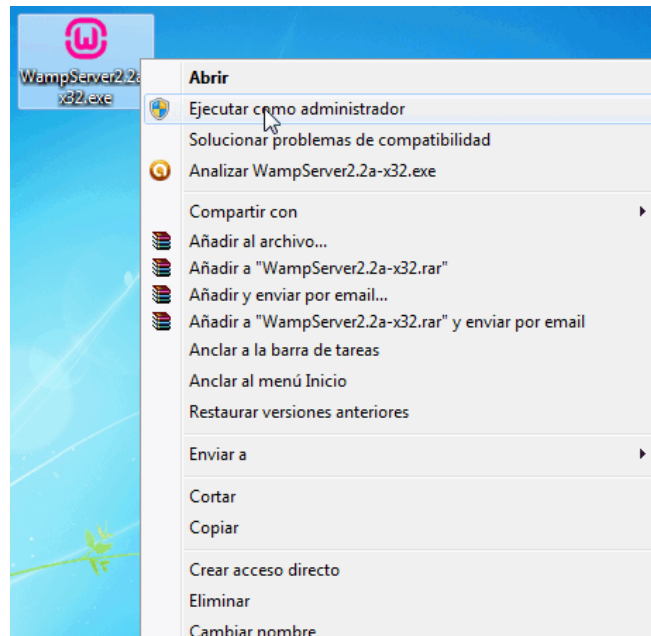


Figura 55. Ejecución del programa para la instalación.

Hecho esto se abrirá la pantalla de bienvenida del programa de instalación, y presionamos en el botón **Next >** para continuar con el proceso.

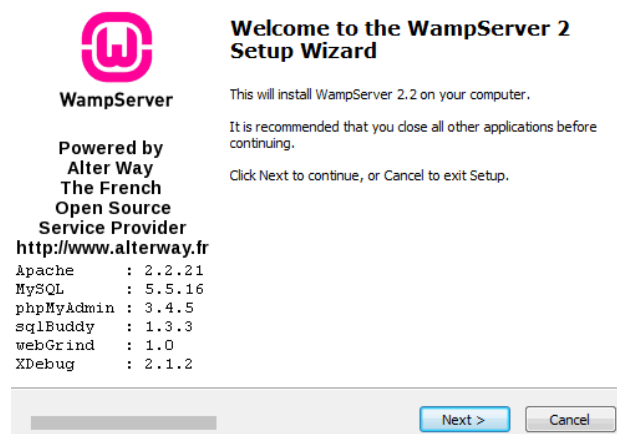



Figura 56. Pantalla de bienvenida.

Nos aparecerá la ventana para aceptar los términos de licencia, escogemos la primera opción, y presionamos en el botón .

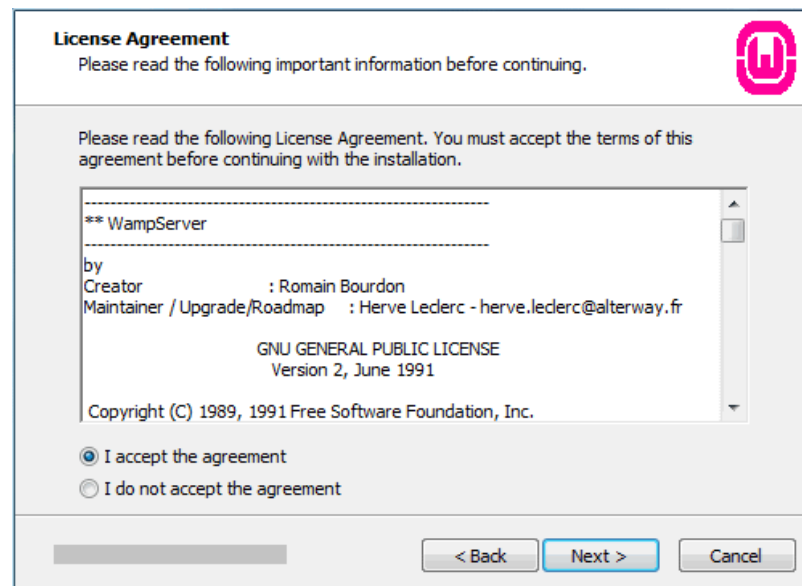
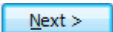


Figura 57. Aceptar términos de licencia.

Se mostrará la carpeta de destino en donde se instalará el programa, podemos mantener la configuración por defecto, o a su vez podemos cambiar la carpeta de destino para la instalación y a continuación presionamos en el botón .

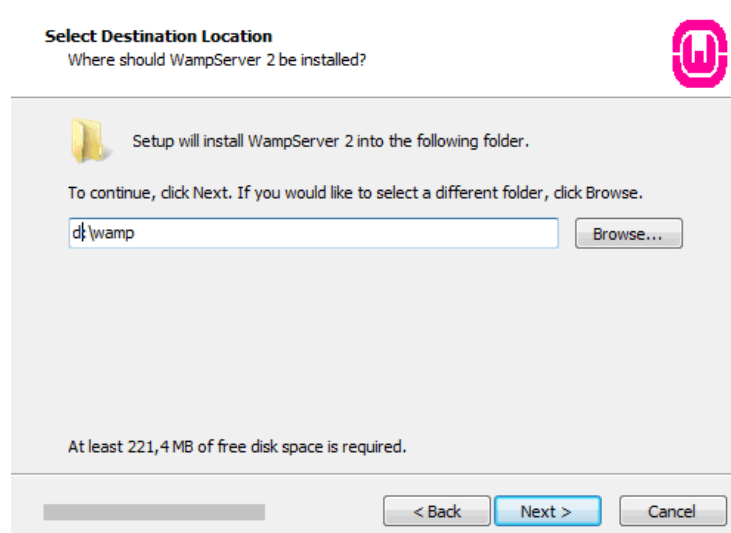
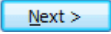


Figura 58. Instalación en la carpeta de destino.

Seleccionamos si deseamos crear un ícono de acceso directo en el escritorio de Windows y también en el menú inicio, en este caso seleccionaremos ambas opciones y presionamos en el botón .

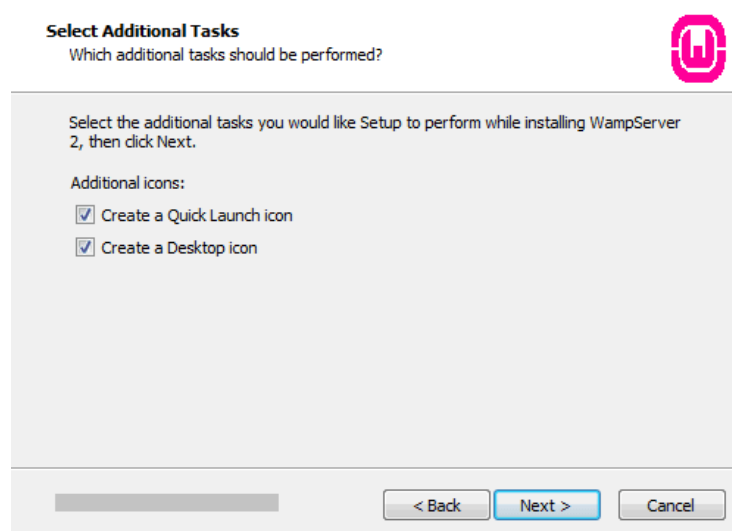


Figura 59. Selección de opciones para crear ícono de acceso directo.

Posteriormente se realizará el proceso de instalación del programa, esperamos unos minutos.

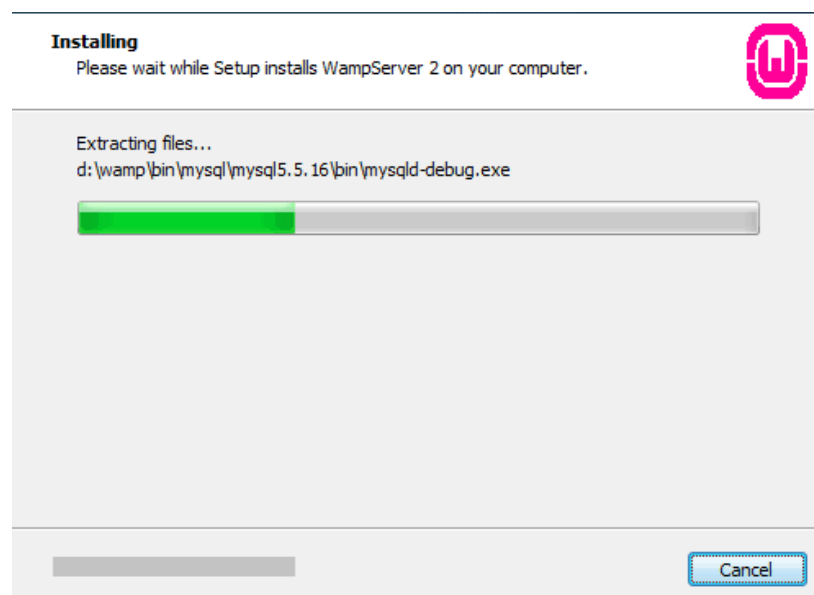


Figura 60. Proceso de instalación del programa.

Una vez finalizada la instalación aparecerá el explorador de archivos para seleccionar el navegador que utilizaremos, por defecto está Internet Explorer, pero podemos modificar y seleccionar el navegador que más nos parezca conveniente haciendo clic en el botón abrir y buscando el ejecutable del navegador.

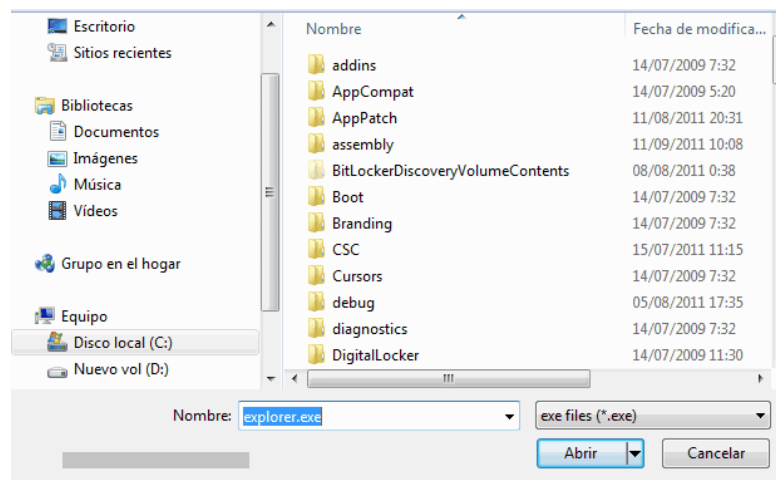
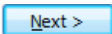


Figura 61. Selección de navegador.

Procederemos a configurar el servidor SMTP en caso de tener montado un servidor de correo, se introduce el nombre del servidor de correo en el campo SMTP, y una dirección de correo electrónico que sea real en el campo de Email, si no estamos seguros de tener un servidor de correo instalado, dejamos los valores que vienen por defecto del programa de instalación y presionamos en el botón .

PHP mail parameters

Please specify the SMTP server and the adresse mail to be used by PHP when using the function mail(). If you are not sure, just leave the default values.

SMTP:
localhost

Email:
tuemail@tudominio

Next >

Figura 62. Configuración del servidor SMPT.

Una vez terminada la instalación por completo ejecutamos el programa si lo deseamos, caso contrario desmarcamos la casilla de “Launch WampServer now”

Y presionamos en el botón **Finish**

WampServer

Powered by
Alter Way
The French
Open Source
Service Provider
<http://www.alterway.fr>

Apache : 2.2.21
MySQL : 5.5.16
phpMyAdmin : 3.4.5
sqlBuddy : 1.3.3
webGrind : 1.0
XDebug : 2.1.2

Completing the WampServer 2 Setup Wizard

Setup has finished installing WampServer 2 on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☒ Launch WampServer 2 now

< Back **Finish**

Figura 63. Finalización de la instalación del programa.

Instalación de Microsoft Visual Studio 2013.

Una vez obtenido el .ISO podemos darle doble clic para montar el Cd virtual y explorar los archivos que este contiene.

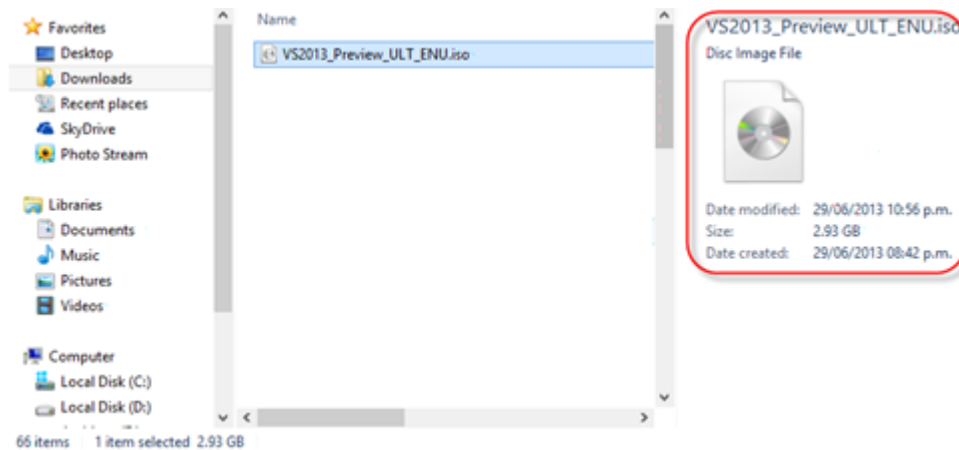


Figura 64. Imagen .ISO para la instalación de Microsoft Visual Studio 2013.

Una vez realizado el paso anterior, a continuación observamos los archivos y componentes necesarios que contiene el archivo .ISO, hacemos doble clic al instalador *vs_ultimate.exe* para empezar.

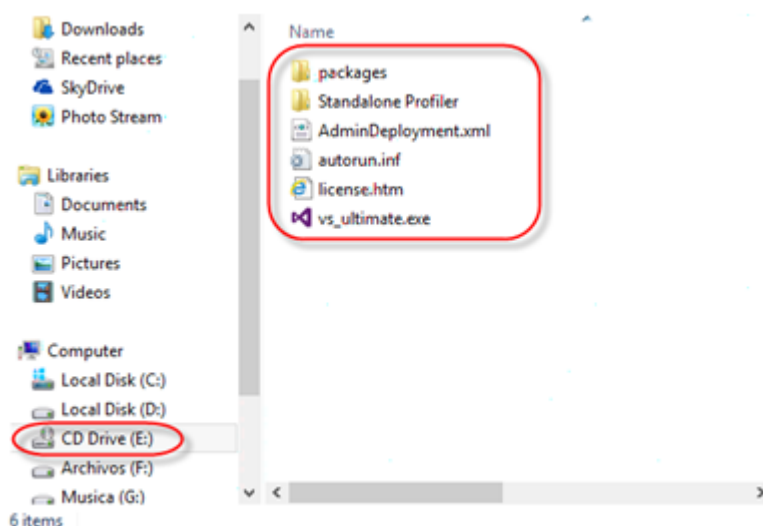


Figura 65. Archivos contenidos dentro del archivo .ISO.

Una vez que ejecutamos el instalador, se abrirá la pantalla de inicio de instalación del Visual Studio, y también la ruta de instalación en el equipo, podemos modificar a otra ruta o dejar los valores por defecto, seguido damos clic en el botón **Next**.

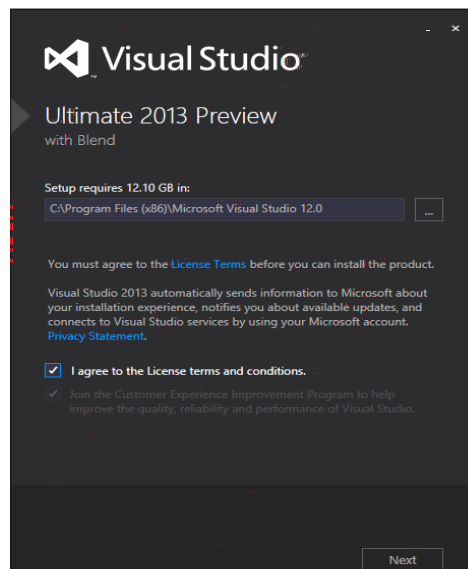


Figura 66. Inicio de la instalación.

A continuación nos desplegará la lista de herramientas u opciones que sean de nuestra utilidad para procederlas a instalar, si deseamos que alguna opción de estas no sea instalada, desmarcamos la casilla correspondiente y damos clic en **INSTALL**.

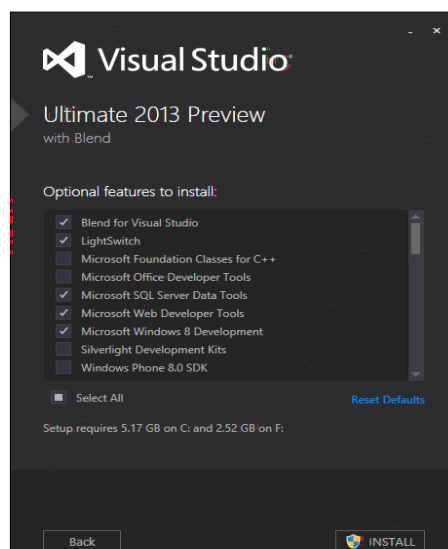



Figura 67. Lista de herramientas a instalar.

La instalación resulta con éxito y para culminar la instalación es necesario reiniciar el equipo, una vez reiniciado nos aparecerá una nueva interfaz, la cual nos pedirá iniciar sesión, a continuación hacemos clic en .

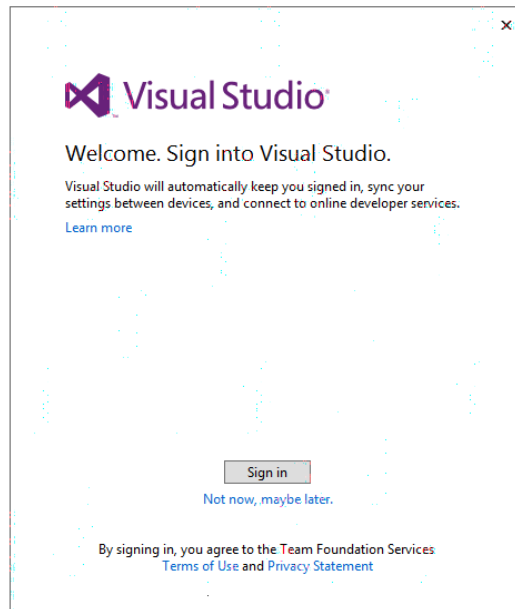


Figura 68. Interfaz de inicio.

Procedemos a iniciar sesión con nuestras credenciales.

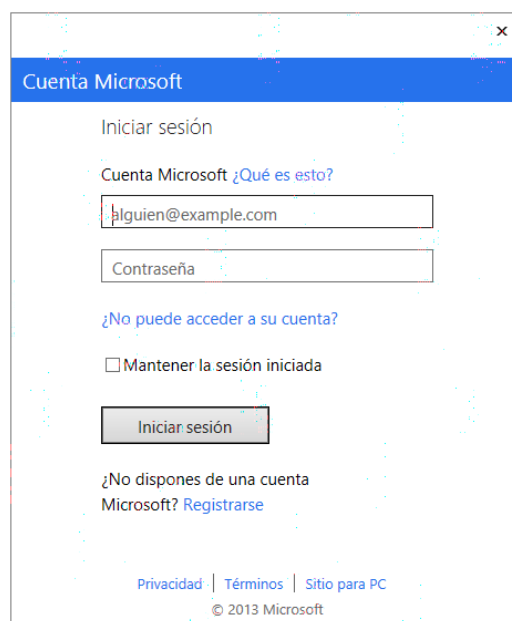


Figura 69. Inicio de sesión en Microsoft.

Hecho esto nos pedirá escoger nuestro ambiente de desarrollo y un tema de nuestro agrado para comenzar a dar uso del software.

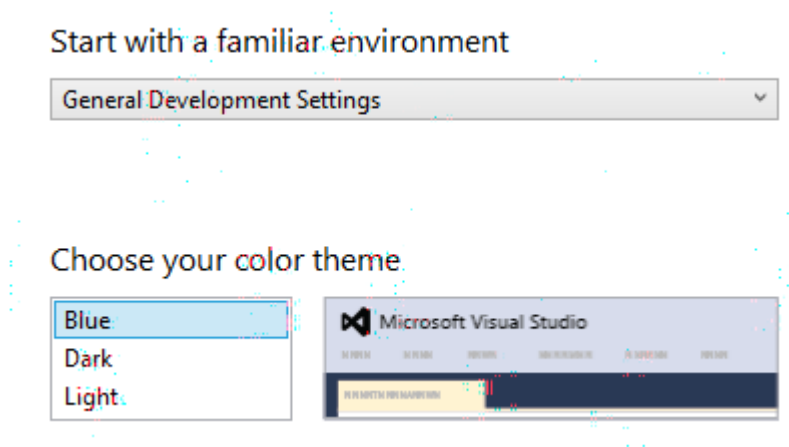


Figura 70. Selección de tema para la interfaz de Visual Studio.

Configurando lo anterior procedemos a abrir nuestro Visual Studio el cual se muestra con la Interfaz que seleccionamos anteriormente.

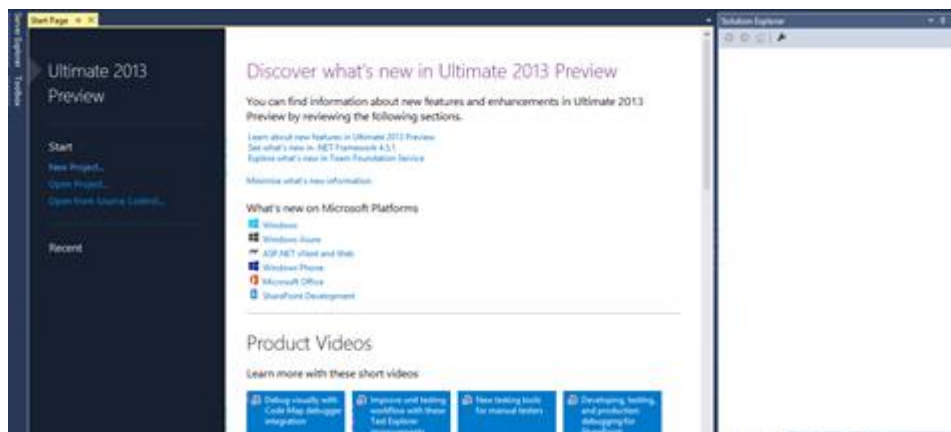


Figura 71. Interfaz principal de Microsoft Visual Studio 2013.

Anexo A.09

Manual Técnico

Para realizar el manual técnico, se tomó en consideración la arquitectura en la que se encuentra desarrollada la aplicación, que en este caso consiste en separar la capa de datos de la capa de presentación al usuario, la capa de presentación es lo que ve el usuario, es decir es la presentación de la interfaz gráfica del sistema, la capa de datos es en donde se encuentran alojados los datos y es la encargada de acceder a los mismos, aquí se realizan los procesos de almacenamiento de datos, recibe solicitudes de almacenamiento o recuperación de cualquier instrucción realizada.

Para el software de control médico veterinario se utilizó el software Microsoft Visual Studio 2013 y para realizar la conexión con la base de datos se utilizó el software MySQL Workbench 6.3.

Vista del formulario de Clientes.

Para la realización de la vista del formulario de clientes se utilizó el siguiente código en HTML, en esta se encuentra también el código utilizado para validar el campo de la cédula, realizado en JavaScript.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="AdminClientes.aspx.cs"
Inherits="Veterinaria.Formulario_web1" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <script>
        $(function () {
            $('#MainContent_tx_cedula').keyup(function () {
                var cedula = $('#MainContent_tx_cedula').val();
                if (cedula.length == 10) {

                    //Obtenemos el digito de la region que son los dos primeros
                    var digito_region = cedula.substring(0, 2);

                    //Pregunto si la region existe ecuador se divide en 24
                    if (digito_region >= 1 && digito_region <= 24) {

                        // Extraigo el ultimo digito
                        var ultimo_digito = cedula.substring(9, 10);

                        //Agrupo todos los pares y los sumo
```

```

        var pares = parseInt(cedula.substring(1, 2)) +
        parseInt(cedula.substring(3, 4)) + parseInt(cedula.substring(5, 6)) +
        parseInt(cedula.substring(7, 8));

        //Agrupo los impares, los multiplico por un factor de
2, si la resultante es > que 9 le restamos el 9 a la resultante
        var numero1 = cedula.substring(0, 1);
        var numero1 = (numero1 * 2);
        if (numero1 > 9) { var numero1 = (numero1 - 9); }

        var numero3 = cedula.substring(2, 3);
        var numero3 = (numero3 * 2);
        if (numero3 > 9) { var numero3 = (numero3 - 9); }

        var numero5 = cedula.substring(4, 5);
        var numero5 = (numero5 * 2);
        if (numero5 > 9) { var numero5 = (numero5 - 9); }

        var numero7 = cedula.substring(6, 7);
        var numero7 = (numero7 * 2);
        if (numero7 > 9) { var numero7 = (numero7 - 9); }

        var numero9 = cedula.substring(8, 9);
        var numero9 = (numero9 * 2);
        if (numero9 > 9) { var numero9 = (numero9 - 9); }

        var impares = numero1 + numero3 + numero5 + numero7 +
numero9;

        //Suma total
        var suma_total = (pares + impares);

        //extraemos el primero digito
        var primer_digito_suma =
String(suma_total).substring(0, 1);

        //Obtenemos la decena inmediata
        var decena = (parseInt(primer_digito_suma) + 1) * 10;

        //Obtenemos la resta de la decena inmediata - la
suma_total esto nos da el digito validador
        var digito_validador = decena - suma_total;

        //Si el digito validador es = a 10 toma el valor de 0
        if (digito_validador == 10)
            var digito_validador = 0;

        //Validamos que el digito validador sea igual al de la
cedula
        if (digito_validador == ultimo_digito) {
            $('#notifica').removeAttr('style');
            $('#notifica').attr('style', 'color: green;');
            $('#notifica').text('Cedula Correcta');

            $('#MainContent_bt_regCliente').removeAttr('disabled');

        } else {
            $('#notifica').removeAttr('style');
            $('#notifica').attr('style', 'color: red;');
            $('#notifica').text('Cedula Incorrecta');
        }
    }
}

```

```

        $('#MainContent_bt_regCliente').attr('disabled',
'disabled');
    }
} else {
    // imprimimos en consola si la region no pertenece
    $('#notifica').removeAttr('style');
    $('#notifica').attr('style', 'color: red;');
    $('#notifica').text('Cedula Incorrecta');
    $('#MainContent_bt_regCliente').attr('disabled',
'disabled');
}
} else {
    //imprimimos en consola si la cedula tiene mas o menos de
10 digitos
    $('#notifica').removeAttr('style');
    $('#notifica').attr('style', 'color: red;');
    $('#notifica').text('Cedula Incorrecta');
    $('#MainContent_bt_regCliente').attr('disabled',
'disabled');
}
});
</script>

<h1>Registrar Clientes</h1>
<br />
<h2>DATOS DE CLIENTES:</h2>
<p>
    &nbsp;
</p>
<div class="templatemo-content-widget white-bg templatemo-overflow-
hidden">
    <table class="table table-responsive">
        <tr>
            <td style="height: 20px;" colspan="2">
                <p style="color: #CC0000; width: 279px">Favor llene todos
los campos, son requeridos</p>
                <p style="color: #CC0000; width: 279px">
                    <strong>
                        <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/A_MuestraClientes.aspx">Ver Listado>></asp:HyperLink></strong>
                    </p>
                    <p style="color: #CC0000; width: 279px">
                        <strong>
                            <asp:Label ID="lb_warning"
runat="server"></asp:Label></strong>
                        </p>
                    </td>
                    <td style="height: 20px">&nbsp;</td>
                <td></td>
            </tr>
            <tr>
                <td style="width: 11px"><b>Número de Cédula:</b></td>
                <td style="width: 367px">
                    <asp:TextBox ID="tx_cedula" runat="server" CssClass="form-
control" Width="158px"></asp:TextBox>
                    &nbsp;<strong id="notifica"></strong></td>
                <td style="width: 103px">
                    <b>Teléfono Convencional:</b></td>

```



```

        <td>
            <asp:TextBox ID="tx_fijo" runat="server" CssClass="form-
control" Placeholder="Ej: 2676441" Width="123px"></asp:TextBox>
            <strong>
                <asp:RangeValidator ID="RangeValidator1"
runat="server" ErrorMessage="Número Telefónico inválido"
MaximumValue="7999999" MinimumValue="2000000" ControlToValidate="tx_fijo"
Type="Integer" Style="color: #CC0000"></asp:RangeValidator></strong>
            </td>
        </tr>
        <tr>
            <td style="width: 11px"><b>Nombres:</b></td>
            <td style="width: 367px">
                <asp:TextBox ID="tx_nombres" runat="server" Width="280px"
CssClass="form-control"></asp:TextBox>
                <strong>
                    <asp:RegularExpressionValidator
ID="RegularExpressionValidator2" runat="server" ControlToValidate="tx_nombres"
ErrorMessage="Por favor Ingrese solo letras"
ValidationExpression="^[a-zñ-ñ-á-é-í-ó-úA-Z ]*$" ForeColor="Red" Style="color:
#CC0000"></asp:RegularExpressionValidator></strong>
                </td>
            <td style="width: 103px">
                <b>Teléfono Móvil:</b></td>
            <td>
                <asp:TextBox ID="tx_movil" runat="server" Placeholder="Ej:
0987396075" CssClass="form-control" Width="125px"></asp:TextBox>
                <strong>
                    <asp:RangeValidator ID="RangeValidator2"
runat="server" ErrorMessage="teléfono móvil inválido"
MaximumValue="0999999999" MinimumValue="0900000000"
ControlToValidate="tx_movil" Type="Integer" Style="color:
#CC0000"></asp:RangeValidator></strong>
                </td>
            </tr>
            <tr>
                <td style="width: 11px"><b>Apellidos:</b></td>
                <td style="width: 367px">
                    <asp:TextBox ID="tx_apellidos" runat="server"
Width="280px" CssClass="form-control"></asp:TextBox>
                    <strong>
                        <asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server"
ControlToValidate="tx_apellidos"
ErrorMessage="Por favor Ingrese solo letras"
ValidationExpression="^[a-zñ-ñ-á-é-í-ó-úA-Z ]*$" ForeColor="Red" Style="color:
#CC0000"></asp:RegularExpressionValidator></strong>
                    </td>
                <td style="width: 103px">
                    <b>E-mail:</b></td>
                <td>
                    <asp:TextBox ID="tx_email" runat="server" Width="290px"
CssClass="form-control" TextMode="Email"></asp:TextBox>
                    <strong>
                        <asp:RegularExpressionValidator
ID="RegularExpressionValidator3" runat="server" ControlToValidate="tx_email"
ErrorMessage="E-mail invalido "
ValidationExpression="\w+([-+.' ]\w+)*@\w+([-.\w+([-+.' ]\w+)*"
ForeColor="Red" Style="color:
#CC0000"></asp:RegularExpressionValidator></strong>

```

```

        </td>
    </tr>
    <tr>
        <td style="width: 11px; height: 16px;">
            <b>Dirección:</b></td>
        <td style="height: 16px" colspan="3">
            <asp:TextBox ID="tx_direccion" TextMode="MultiLine"
runat="server" Width="280px" CssClass="form-control"
Height="99px"></asp:TextBox>
        </td>

    </tr>

    <tr>
        <td style="width: 11px">
            <asp:Button ID="bt_regCliente" runat="server"
Text="Registrar Cliente" CssClass="templatemo-fore-button"
OnClick="bt_regCliente_Click" OnClientClick="return confirm('¿Está seguro que
desea Guardar este registro?')" Width="169px" />
            &nbsp;</td>
            <td style="width: 367px">&nbsp;<asp:Button ID="bt_limpiar0"
runat="server" Text="Limpiar" CssClass="btn btn-primary"
OnClick="bt_limpiar_Click" />
            &nbsp;&nbsp;&nbsp;</td>
            <td>
                &nbsp;<asp:Button ID="Button1" runat="server" Text="Volver"
PostBackUrl="~/A_ListaClientes.aspx" CssClass="btn btn-primary" />
            </td>
        </tr>

    </table>
</div>

</asp:Content>

```

Registrar nuevo cliente y validaciones de campos.

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Veterinaria
{
    public partial class Formulario_web1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void bt_regCliente_Click(object sender, EventArgs e)
        {

```

```
if (!ValidarCampos()) return;

String cedula = tx_cedula.Text;
String nombres = tx_nombres.Text;
String apellidos = tx_apellidos.Text;
String direccion = tx_direccion.Text;
String telefonoFijo = tx_fijo.Text;
String telefonoMovil = tx_movil.Text;
String correo = tx_email.Text;

modeloCliente model = new modeloCliente();
try
{
    bool cedRepetida = false;
    var cedula_cliente =
model.verificar_cliente_con_cedula_repetida(cedula);
    if (cedula_cliente.Read())
    {
        cedRepetida = true;
    }

    if (cedRepetida)
    {
        mostrarAlerta_Jscript("Cliente ya fue registrado con este
número de Cédula");
        tx_cedula.Focus();
    }
    else
    {
        int c = model.registrar_cliente_(cedula, nombres,
apellidos, direccion, telefonoFijo, telefonoMovil, correo);
        if (c > 0)
        {
            mostrarAlerta_Jscript("Registro Agregado");
            LimpiarC();
        }
    }
}
catch (Exception)
{
    mostrarAlerta_Jscript("Error al insertar");
}

finally
{
    model.CerrarConexion();
}
lb_warning.Visible = false;

public void mostrarAlerta_Jscript(string Mensaje)
{
    System.Web.UI.ScriptManager.RegisterStartupScript(this,
this.GetType(), "", "alert('" + Mensaje + "');", true);
}

protected void bt_limpiar_Click(object sender, EventArgs e)
{
    LimpiarC();
}
```

```
public void LimpiarC()
{
    tx_apellidos.Text = string.Empty;
    tx_cedula.Text = string.Empty;
    tx_nombres.Text = string.Empty;
    tx_movil.Text = string.Empty;
    tx_fijo.Text = string.Empty;
    tx_direccion.Text = string.Empty;
    tx_email.Text = string.Empty;
}
private bool ValidarCampos()
{
    if (tx_cedula.Text == string.Empty)
    {
        lb_warning.Text = "¡Debe ingresar el número de cédula!";
        tx_cedula.Focus();
        return false;
    }
    else { lb_warning.Visible = false; }

    if (tx_cedula.Text != null)
    {
        lb_warning.Visible = false;
    }
    if (tx_nombres.Text == string.Empty)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Debe ingresar un nombre');", true);
        tx_nombres.Focus();
        return false;
    }
    if (tx_nombres.Text == string.Empty)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Debe ingresar un nombre');", true);
        tx_nombres.Focus();
        return false;
    }
    if (tx_apellidos.Text == string.Empty)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Debe ingresar un apellido');", true);
        tx_apellidos.Focus();
        return false;
    }
    if (tx_direccion.Text == string.Empty)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese la dirección');", true);
        tx_direccion.Focus();
        return false;
    }
    if (tx_fijo.Text == string.Empty)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese el numero de teléfono Convencional');",
true);
        tx_fijo.Focus();
        return false;
    }
}
```

```

        if (tx_movil.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese el numero de teléfono Móvil');", true);
            tx_movil.Focus();
            return false;
        }
        if (tx_email.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese un E-mail');", true);
            tx_email.Focus();
            return false;
        }
        return true;
    }
}
}

```

Actualizar Cliente.

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Veterinaria
{
    public partial class Formulario_web16 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            lb_warning.Visible = false;
            if (!Page.IsPostBack)
            {
                String id_encrypt = Request.QueryString["encrypt"];
                Int32 id = System.Convert.ToInt32(id_encrypt) / 777;
                modeloCliente model = new modeloCliente();
                MySqlDataReader client = model.consultar_clientes_id(id);
                try
                {
                    if (client.Read())
                    {
                        tx_codigo.Text =
client.GetInt32("id_cliente").ToString();
                        tx_cedula.Text = client.GetString("cedula");
                        tx_nombres.Text = client.GetString("nombre");
                        tx_apellidos.Text = client.GetString("apellido");
                        tx_direccion.Text = client.GetString("direccion");
                        tx_fijo.Text = client.GetString("telefono");
                        tx_movil.Text = client.GetString("movil");
                        tx_email.Text = client.GetString("correo");
                    }
                }
            }
        }
    }
}

```

```
        catch (Exception)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this, this.GetType()
                , "", "alert('Error al completar acción');", true);
        }
    }

    protected void bt_limpiar0_Click(object sender, EventArgs e)
    {
        LimpiarCampos();
    }

    protected void bt_regCliente_Click(object sender, EventArgs e)
    {
        if (!ValidarC()) return;

        int id = System.Convert.ToInt32(tx_codigo.Text);
        String nombres = tx_nombres.Text;
        String cedula = tx_cedula.Text;
        String apellidos = tx_apellidos.Text;
        String direccion = tx_direccion.Text;
        String telefono = tx_fijo.Text;
        String movil = tx_movil.Text;
        String correo = tx_email.Text;
        modeloCliente model = new modeloCliente();

        int actualiza = model.actualizar_cliente(id, nombres, apellidos,
            direccion, telefono, movil, correo);
        if (actualiza > 0)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
                this.GetType(), "", "alert('Los Datos se han Actualizado');", true);

            LimpiarCampos();
        }

        else
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
                this.GetType(), "", "alert('Error al actualizar');", true);
        }
    }

    public void LimpiarCampos()
    {
        tx_nombres.Text = string.Empty;
        tx_apellidos.Text = string.Empty;
        tx_direccion.Text = string.Empty;
        tx_fijo.Text = string.Empty;
```

```

        tx_movil.Text = string.Empty;
        tx_email.Text = string.Empty;
        tx_cedula.Text = string.Empty;
        tx_codigo.Text = string.Empty;
    }
    private bool ValidarC()
    {
        if (tx_nombres.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Debe ingresar un nombre');", true);
            tx_nombres.Focus();
            return false;
        }
        if (tx_apellidos.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Debe ingresar un apellido');", true);
            tx_apellidos.Focus();
            return false;
        }
        if (tx_direccion.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese la dirección');", true);
            tx_direccion.Focus();
            return false;
        }
        if (tx_fijo.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese el numero de teléfono Convencional');",
true);
            tx_fijo.Focus();
            return false;
        }
        if (tx_movil.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese el numero de teléfono Móvil');", true);
            tx_movil.Focus();
            return false;
        }
        if (tx_email.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Ingrese un E-mail');", true);
            tx_email.Focus();
            return false;
        }
        return true;
    }
}
}

```

Eliminar Cliente

```
protected void bt_eliminar_Click(object sender, ImageClickEventArgs e)
```

```
{  
  
    try  
    {  
        int id = int.Parse((sender as ImageButton).CommandArgument);  
        modeloCliente model = new modeloCliente();  
  
        int cliente = model.eliminar_cliente_(id);  
        if (cliente > 0)  
        {  
            mostrarAlerta_Jscript("Cliente Eliminado exitosamente");  
            GridViewClientes.DataSource = model.consultar_clientes();  
            GridViewClientes.DataBind();  
        }  
    }  
    catch (Exception ex)  
    {  
        mostrarAlerta_Jscript("Error al Eliminar");  
    }  
}
```

Buscar Cliente

```
protected void bt_buscar_Click(object sender, EventArgs e)  
{  
    modeloCliente model = new modeloCliente();  
    String nombreCliente = tx_nombre.Text;  
    try  
    {  
        GridViewClientes.DataSource =  
model.consultar_clientesPorNombre(nombreCliente);  
        GridViewClientes.DataBind();  
    }  
    catch (Exception ex)  
    {  
    }  
}
```

Seleccionar Cliente

```
protected void bt_seleccionar_Click(object sender, ImageClickEventArgs e)  
{  
    int id = int.Parse((sender as ImageButton).CommandArgument);  
    Response.Redirect("A_ActualizarDatosClientes.aspx?encrypt=" + (id  
* 777));  
}
```

Acceso a Datos

```
using MySql.Data.MySqlClient;  
using System;
```



```
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Veterinaria
{
    public class modeloCliente
    {
        Conexion conex = new Conexion();
        public int registrar_cliente_(String cedula, String nombre, String
apellido, String direccion,
                                String telefono, String movil, String
correo)
        {
            int filas = 0;
            String SecuenciaSQL;
            MySqlConnection con = conex.ObtenerConexion();
            MySqlCommand sqlcom;
            try
            {
                SecuenciaSQL = string.Format(@"INSERT INTO
veterinaria.clientes
                                VALUES (null, '{0}', '{1}', '{2}',
'{3}', '{4}', '{5}', '{6}', 'A');",
                                cedula, nombre, apellido, direccion, telefono, movil,
correo);
                sqlcom = new MySqlCommand(SecuenciaSQL, con);
                filas = sqlcom.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                throw ex;
            }
            return filas;
        }
        public int eliminar_cliente_(int id_cliente)
        {
            int filas = 0;
            String SecuenciaSQL;
            MySqlConnection con = conex.ObtenerConexion();
            MySqlCommand sqlcom;
            try
            {
                SecuenciaSQL = string.Format(@"UPDATE clientes SET estado='I'
WHERE clientes.id_cliente={0}",
                                id_cliente);
                sqlcom = new MySqlCommand(SecuenciaSQL, con);
                filas = sqlcom.ExecuteNonQuery();
            }
            catch (Exception e)
            {
                throw e;
            }
            return filas;
        }

        public MySqlDataReader consultar_clientes()
        {
            String SecuenciaSQL;
            MySqlConnection con = conex.ObtenerConexion();
```

```
        MySqlCommand sqlcom;
        MySqlDataReader resultData = null;
        try
        {
            SecuenciaSQL = string.Format(@"SELECT * FROM clientes WHERE
estado='A';");
            sqlcom = new MySqlCommand(SecuenciaSQL, con);
            resultData = sqlcom.ExecuteReader();
            return resultData;
        }
        catch (Exception ex)
        {
            return null;
            throw ex;
        }
        finally
        {
        }
    }

    public MySqlDataReader consultar_clientes_id(int id_cliente)
    {
        String SecuenciaSQL;
        MySqlConnection con = conex.ObtenerConexion();
        MySqlCommand sqlcom;
        MySqlDataReader resultData = null;
        try
        {
            SecuenciaSQL = string.Format(@"SELECT * FROM clientes WHERE
id_cliente={0} ;", id_cliente);
            sqlcom = new MySqlCommand(SecuenciaSQL, con);
            resultData = sqlcom.ExecuteReader();
            return resultData;
        }
        catch (Exception ex)
        {
            return null;
            throw ex;
        }
        finally
        {
        }
    }

    public int actualizar_cliente(Int32 id_cliente, String nombres, String
apellidos, String direccion, String telefono, String movil, String correo)
    {
        int filas = 0;
        String SecuenciaSQL;
        MySqlConnection con = conex.ObtenerConexion();
        MySqlCommand sqlcom;
        try
        {
            SecuenciaSQL = string.Format(@"UPDATE `veterinaria`.`clientes`
set nombre='{1}', apellido='{2}', direccion='{3}', telefono='{4}',
movil='{5}', correo='{6}' where id_cliente={0}",
id_cliente, nombres, apellidos, direccion, telefono,
movil, correo);
```

```

        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        filas = sqlcom.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        throw e;
    }
    finally
    {
        CerrarConexion();
    }
    return filas;
}
public MySqlDataReader consultar_clientesPorNombre(String
nombre_cliente)
{
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    MySqlDataReader resultData = null;
    try
    {
        SecuenciaSQL = string.Format(@"SELECT * FROM clientes WHERE
clientes.estado='A' AND clientes.nombre like '{0}%',", nombre_cliente);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        resultData = sqlcom.ExecuteReader();
        return resultData;
    }
    catch (Exception ex)
    {
        return null;
        throw ex;
    }
    finally
    {
    }
}
public MySqlDataReader verificar_cliente_con_cedula_repetida(String
cedula)
{
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    MySqlDataReader resultData = null;
    try
    {
        SecuenciaSQL = string.Format(@"SELECT * FROM `clientes` where
cedula='{0}' ;", cedula);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        resultData = sqlcom.ExecuteReader();
        return resultData;
    }
    catch (Exception ex)
    {
        return null;
        throw ex;
    }
    finally
    {

```

```

    }
}
public void CerrarConexion()
{
    conex.CerrarConexion();
}
}
}

```

Vista del formulario de Mascota

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="M_NuevaMascota.aspx.cs"
Inherits="Veterinaria.M_NuevaMascota" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit"
TagPrefix="cc1" %>
<%@ Register assembly="DevExpress.Web.v13.2, Version=13.2.5.0,
Culture=neutral, PublicKeyToken=b88d1754d700e49a"
namespace="DevExpress.Web.ASPxEditors" tagprefix="dx" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

    <script type="text/javascript">

        function validarnum(e) { // 1

            tecla = (document.all) ? e.keyCode : e.which; // 2

            if (tecla == 8) return true; // 3
            patron = /[0-1-2-3-4-5-6-7-8-9\s]/; // 4
            te = String.fromCharCode(tecla); // 5
            return patron.test(te); // 6

        }

    </script>

    <script>
        $(function () {
            $('#MainContent_tx_cedula').keyup(function () {
                var cedula = $('#MainContent_tx_cedula').val();
                if (cedula.length == 10) {

                    //Obtenemos el digito de la region que son los dos primeros
                    var digito_region = cedula.substring(0, 2);

                    //Pregunto si la region existe ecuador se divide en 24
                    if (digito_region >= 1 && digito_region <= 24) {

                        // Extraigo el ultimo digito
                        var ultimo_digito = cedula.substring(9, 10);

                        //Agrupo todos los pares y los sumo

```

```

        var pares = parseInt(cedula.substring(1, 2)) +
        parseInt(cedula.substring(3, 4)) + parseInt(cedula.substring(5, 6)) +
        parseInt(cedula.substring(7, 8));

        //Agrupo los impares, los multiplico por un factor de
        2, si la resultante es > que 9 le restamos el 9 a la resultante
        var numero1 = cedula.substring(0, 1);
        var numero1 = (numero1 * 2);
        if (numero1 > 9) { var numero1 = (numero1 - 9); }

        var numero3 = cedula.substring(2, 3);
        var numero3 = (numero3 * 2);
        if (numero3 > 9) { var numero3 = (numero3 - 9); }

        var numero5 = cedula.substring(4, 5);
        var numero5 = (numero5 * 2);
        if (numero5 > 9) { var numero5 = (numero5 - 9); }

        var numero7 = cedula.substring(6, 7);
        var numero7 = (numero7 * 2);
        if (numero7 > 9) { var numero7 = (numero7 - 9); }

        var numero9 = cedula.substring(8, 9);
        var numero9 = (numero9 * 2);
        if (numero9 > 9) { var numero9 = (numero9 - 9); }

        var impares = numero1 + numero3 + numero5 + numero7 +
        numero9;

        //Suma total
        var suma_total = (pares + impares);

        //extraemos el primero digito
        var primer_digito_suma =
        String(suma_total).substring(0, 1);

        //Obtenemos la decena inmediata
        var decena = (parseInt(primer_digito_suma) + 1) * 10;

        //Obtenemos la resta de la decena inmediata - la
        suma_total esto nos da el digito validador
        var digito_validador = decena - suma_total;

        //Si el digito validador es = a 10 toma el valor de 0
        if (digito_validador == 10)
            var digito_validador = 0;

        //Validamos que el digito validador sea igual al de la
        cedula
        if (digito_validador == ultimo_digito) {
            $('#notifica').removeAttr('style');
            $('#notifica').attr('style', 'color: green;');
            $('#notifica').text('Cedula Correcta');

            $('#MainContent_bt_registrar').removeAttr('disabled');

        } else {
            $('#notifica').removeAttr('style');
            $('#notifica').attr('style', 'color: red;');
            $('#notifica').text('Cedula Incorrecta');
        }
    
```

```

        $('#MainContent_bt_registrar').attr('disabled',
'disabled');
    }
} else {
    // imprimimos en consola si la region no pertenece
    $('#notifica').removeAttr('style');
    $('#notifica').attr('style', 'color: red;');
    $('#notifica').text('Cedula Incorrecta');
    $('#MainContent_bt_registrar').attr('disabled',
'disabled');
}
} else {
    //imprimimos en consola si la cedula tiene mas o menos de
10 digitos
    $('#notifica').removeAttr('style');
    $('#notifica').attr('style', 'color: red;');
    $('#notifica').text('Cedula Incorrecta');
    $('#MainContent_bt_registrar').attr('disabled',
'disabled');
}
});
});
</script>

<meta charset="utf-8">

<h1>REGISTRAR MASCOTA</h1>
<asp:Panel runat="server" ID="panel_1">
    <div>
<table style="width: 100%" class="table table-responsive">
    <thead>
    </thead>
    <tbody>
        <tr>
            <td colspan="10">
                <h4>INGRESE EL NÚMERO DE CÉDULA DEL PROPIETARIO</h4>
            </td>

        </tr>
        <tr>
            <td style="width: 84px" >

                <b>Cedula:</b>

            </td>
            <td colspan="9">

                <asp:TextBox ID="tx_cedula" CssClass="form-control"
runat="server" Width="146px"></asp:TextBox>
                <strong id="notifica"></strong>

                <br />
                <asp:Button ID="bt_buscarCliente" runat="server" Text="Ver
datos del propietario" CssClass="templatemo-edit-btn"
OnClick="bt_buscarCliente_Click" Width="172px" />

            </td>
        </tr>
    </tbody>
</table>
    </div>

```

```

        <td colspan="10" runat="server" id="tr_labeldatosp"
visible="false">DATOS PROPIETARIO</td>
    </tr>
    <tr id="tr_datProp" runat="server" visible="false">
        <td style="width: 84px">

            <b>Nombres:</b>

        </td>
        <td>
            <asp:Label ID="lb_nombreCliente" runat="server"
Height="17px"></asp:Label>
            <br />
            <asp:Label ID="lb_apellidos" runat="server"></asp:Label>
        </td>
        <td>
            <b id="ape0" visible="false">Dirección:</b></td>
        <td>
            <asp:Label ID="lb_direccion" runat="server"></asp:Label>
        </td>
        <td>
            <b>Teléfono:</b></td>
        <td>
            <asp:Label ID="lb_telefono" runat="server"></asp:Label>
        </td>
        <td>
            <b>Móvil:</b></td>
        <td>
            <asp:Label ID="lb_movil" runat="server"></asp:Label>
        </td>
        <td><b>E-mail:</b></td>
        <td><asp:Label ID="lb_email" runat="server"></asp:Label></td>
    </tr>
</tbody>
</table>
</div>
</asp:Panel>
<asp:Panel ID="panel_2" runat="server" BorderColor="#808080">
    <div class="templatemo-content-widget white-bg templatemo-overflow-
hidden">
        <table style="width: 101%" class="table table-responsive">

            <tbody>

                <tr>
                    <td colspan="6">
                        DATO DE LA MASCOTA
                    </td>

                </tr>

                <tr>
                    <td style="width: 117px"><b>Especie:</b></td>
                    <td style="width: 473px">
                        <asp:DropDownList ID="drp_tipo" runat="server"
CssClass="form-control" Width="117px">
                            <asp:ListItem Text="Elija..." ></asp:ListItem>
                            <asp:ListItem Value="1" Text="Canino" ></asp:ListItem>
                            <asp:ListItem Value="2" Text="Felino"></asp:ListItem>
                        </asp:DropDownList>

```

```

        </td>
        <td style="width: 121px">
            <b>Peso: (Kg)</b></td>
        <td colspan="3">
            <asp:TextBox ID="tx_peso" runat="server" CssClass="form-
control" TextMode="Number" Width="162px" Placeholder="Peso
(Kg)"></asp:TextBox>
            <strong><asp:RangeValidator ID="RangeValidator1"
runat="server" MaximumValue="200" MinimumValue="1" ErrorMessage="Valor no
permitido" style="color: #CC0000" ControlToValidate="tx_peso"
Type="Integer"></asp:RangeValidator></strong>
        </td>

    </tr>
    <tr>
        <td style="width: 117px"><b>Sexo:</b></td>
        <td style="width: 473px">
            <asp:DropDownList ID="drp_sexo" runat="server"
CssClass="form-control" Width="119px">
                <asp:ListItem Text="Elija..." ></asp:ListItem>
                <asp:ListItem Value="Macho" Text="Macho"
></asp:ListItem>
                <asp:ListItem Value="Hembra" Text="Hembra"
></asp:ListItem>
            </asp:DropDownList>
        </td>
        <td>
            <b>Color:</b></td>
        <td colspan="3">
            <asp:TextBox ID="tx_color" runat="server" CssClass="form-
control" Height="31px" Width="158px" ></asp:TextBox> </td>

    </tr>
    <tr>
        <td style="width: 117px"><b>Nombre:
</b></td>

        <td><asp:TextBox ID="tx_nombre" runat="server" CssClass="form-
control" Width="187px"></asp:TextBox> <strong><asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server" ControlToValidate="tx_nombre"
ErrorMessage="Por favor ingrese solo letras"
ValidationExpression="^[a-zñ-ñ-á-é-í-ó-úA-Z ]*$" ForeColor="Red" style="color:
#CC0000"></asp:RegularExpressionValidator>

        </strong>

    </td>

    <td>
        <b>Fecha de Nacimiento:</b></td>

        <td colspan="3">
            <asp:TextBox ID="tx_fechanac" CssClass="form-control"
runat="server" Height="32px" Width="159px" ></asp:TextBox>
            <strong><asp:RangeValidator ID="RangeValidator2"
runat="server" ControlToValidate="tx_fechanac" ErrorMessage="Fecha ingresada
no válida" Type="Date" style="color: #CC0000"></asp:RangeValidator></strong>
            <cc1:calendarextender ID="CalendarExtender1"
Format="dd/MM/yyyy" TargetControlID="tx_fechanac" runat="server"
PopupPosition="TopLeft" Animated="true"/>

```



```

        </td>

    </tr>

    <tr>
        <td>

            <b>Raza:</b>

        </td>
        <td>
            <asp:TextBox ID="tx_raza" runat="server"
            CssClass="form-control" Width="187px" ></asp:TextBox>
            <strong><asp:RegularExpressionValidator
            ID="RegularExpressionValidator2" runat="server" ControlToValidate="tx_raza"
            ErrorMessage="Por favor ingrese solo letras"
            ValidationExpression="^[a-zñ-ñ-á-é-í-ó-úA-Z ]*$" ForeColor="Red" style="color:
            #CC0000"></asp:RegularExpressionValidator>
            </strong>

        </td>
        <td>
            <asp:Button ID="bt_obtener" runat="server"
            Text="Calcular Edad" CssClass="templatemo-edit-btn"
            OnClick="bt_obtener_Click"/><br /><br />

        </td>
        <td>
            <asp:TextBox ID="tx_edad"
            runat="server" ReadOnly="true" Height="29px" Width="215px" CssClass="form-
            control"></asp:TextBox>

        </td>
    </tr>

    <tr>

        <td colspan="4">
            <asp:Button ID="bt_registrar" runat="server"
            CssClass="templatemo-fore-button" OnClick="bt_registrar_Click" Text="Registrar
            Mascota" Width="240px" />
            &nbsp;
            <asp:Button ID="Button1" runat="server"
            CssClass="templatemo-fore-button"PostBackUrl="~/M_ListaMascotas.aspx"
            Text="Ver Listado" />
            &nbsp;
            <asp:Button ID="bt_limpiar" runat="server"
            CssClass="templatemo-fore-button" OnClick="bt_limpiar_Click" Text="Limpiar"
            Width="112px" />
            &nbsp;
            <asp:Button ID="bt_cancelar" runat="server"
            CssClass="templatemo-fore-button"PostBackUrl="~/M_ListaMascotas.aspx"
            Text="Volver" />
        </td>
    </tbody>
</table>
</div>
</asp:Panel>

</asp:Content>

```

Registrar nueva mascota y validar campos.

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Veterinaria
{
    public partial class M_NuevaMascota : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if(!Page.IsPostBack)
            {
                validarFecha();
            }
            lb_nombreCliente.Visible = false;
        }
        protected void bt_registrar_Click(object sender, EventArgs e)
        {
            if (!ValidarC()) return;

            String cedula = tx_cedula.Text;
            String nombre = tx_nombre.Text;
            String raza = tx_raza.Text;
            String color = tx_color.Text;
            String peso = tx_peso.Text;
            String edad = tx_edad.Text;
            String fechanac = tx_fechanac.Text;
            modelo model = new modelo();
            try
            {
                Int32 Id_cliente = 0;
                String Nombre = string.Empty;
                String Apellido = string.Empty;
                String Direccion = string.Empty;
                String sexo = drp_sexo.SelectedValue.ToString();
                Int32 tipo =
                System.Convert.ToInt32(drp_tipo.SelectedValue.ToString());

                MySqlDataReader Cliente =
                model.consultar_datos_cliente_por_cedula(cedula);
                if (cedula != null && cedula != "" && nombre != null && nombre
                != "")
                {
                    if (Cliente.Read())
                    {
                        Id_cliente = Cliente.GetInt32("id_cliente");
                        Nombre = Cliente.GetString("nombre");
                        Apellido = Cliente.GetString("apellido");
                        Direccion = Cliente.GetString("direccion");
                    }
                }
            }
            catch { }
        }
    }
}
```

```
int guardado = model.nueva_mascota(Id_cliente, tipo,
nombre, raza, peso, color, sexo, edad, fechanac);
if(guardado>0)
{
    mostrarAlerta_Jscript("Registro Agregado");
    LimpiarC();
}
}
}

}
catch (Exception)
{
    mostrarAlerta_Jscript("Error al insertar");
}
finally
{
    model.CerrarConexion();
}
}

protected void bt_buscarCliente_Click(object sender, EventArgs e)
{
    tr_datProp.Visible = true;
    tr_labeldatosp.Visible = true;
    modelo model = new modelo();
    try
    {
        String cedula = tx_cedula.Text;
        MySqlDataReader Datacliente =
model.consultar_datos_cliente_por_cedula(cedula);
        if (Datacliente.Read())
        {
            lb_nombreCliente.Text = Datacliente.GetString("nombre");
            //+ " " + Datacliente.GetString("apellido");
            lb_apellidos.Text = Datacliente.GetString("apellido");
            lb_direccion.Text = Datacliente.GetString("direccion");
            lb_telefono.Text = Datacliente.GetString("telefono");
            lb_email.Text = Datacliente.GetString("correo");
            lb_movil.Text = Datacliente.GetString("movil");
            int id_cliente = Datacliente.GetInt32("id_cliente");
            lb_nombreCliente.Visible = true;

        }
        else
        {
            lb_nombreCliente.Text = "Cliente no encontrado";
            lb_nombreCliente.Visible = true;
            tx_cedula.Text = string.Empty;
            tx_nombre.Text = string.Empty;
            tx_fechanac.Text = string.Empty;
            tx_color.Text = string.Empty;
            tx_peso.Text = string.Empty;
            tx_raza.Text = string.Empty;
            drp_sexo.SelectedIndex = 0;
            drp_tipo.SelectedIndex = 0;
            tr_datProp = null;
        }
    }
}
```

```
        lb_telefono.Text = string.Empty;
        lb_apellidos.Text = string.Empty;
        lb_direccion.Text = string.Empty;
        lb_movil.Text = string.Empty;
        lb_email.Text = string.Empty;
    }
}
catch (Exception)
{
    mostrarAlerta_Jscript("Error al completar Acción");
}
}

public void mostrarAlerta_Jscript(string Mensaje)
{
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('" + Mensaje + "');", true);
}

private bool ValidarC()
{
    if (tx_cedula.Text == string.Empty)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Debe ingresar el numero de cédula del cliente');",
true);
        tx_cedula.Focus();
        return false;
    }
    if (tx_nombre.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese el nombre de la mascota");
        tx_nombre.Focus();
        return false;
    }
    if (tx_raza.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese la raza de la mascota");
        tx_raza.Focus();
        return false;
    }
    if (tx_peso.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese el peso de la mascota");
        tx_peso.Focus();
        return false;
    }
    if (tx_color.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese el color de la mascota");
        tx_color.Focus();
        return false;
    }
    if (tx_fechanac.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese la fecha de nacimiento de la
mascota");
    }
}
```

```

        tx_fechanac.Focus();
        return false;
    }
    if (drp_sexo.SelectedIndex == 0)
    {
        mostrarAlerta_Jscript("Elija el sexo de la mascota");
        drp_sexo.Focus();
        return false;
    }
    if (drp_tipo.SelectedIndex == 0)
    {
        mostrarAlerta_Jscript("Elija la especie de la mascota");
        drp_tipo.Focus();
        return false;
    }
    if (tx_edad.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Proporcione la edad de la mascota");
        tx_edad.Focus();
        return false;
    }

    return true;
}
public void LimpiarC()
{
    tx_cedula.Text = string.Empty;
    tx_nombre.Text = string.Empty;
    tx_fechanac.Text = string.Empty;
    tx_color.Text = string.Empty;
    tx_peso.Text = string.Empty;
    tx_raza.Text = string.Empty;
    drp_sexo.SelectedIndex = 0;
    drp_tipo.SelectedIndex = 0;
    tr_datProp = null;
    lb_telefono.Text = string.Empty;
    lb_apellidos.Text = string.Empty;
    lb_direccion.Text = string.Empty;
    lb_nombreCliente.Text = string.Empty;
    tx_edad.Text = string.Empty;
}

protected void bt_limpiar_Click(object sender, EventArgs e)
{
    LimpiarC();
}
public String DiferenciaFechas(DateTime pFechaUno, DateTime pFechaDos)
{
    lb_nombreCliente.Visible = true;
    try
    {
        int[] vDiasMes = new int[12] { 31, -1, 31, 30, 31, 30, 31, 31, 30,
31, 30, 31 };

        DateTime vFechaOrigen;
        DateTime vFechaDestino;
        int vAnio;
        int vMes;
        int vDia;
    }
    catch { }
}

```

```
int vIncremento;

//Determinamos cual es la fecha menor
if (pFechaUno > pFechaDos)
{
    vFechaOrigen = pFechaDos;
    vFechaDestino = pFechaUno;
}
else
{
    vFechaOrigen = pFechaUno;
    vFechaDestino = pFechaDos;
}

// Calculamos los dias
vIncremento = 0;

if (vFechaOrigen.Day > vFechaDestino.Day)
{
    vIncremento = vDiasMes[vFechaOrigen.Month - 1];
}
if (vIncremento == -1)
{
    if (DateTime.IsLeapYear(vFechaOrigen.Year))
    {
        // Para los años bisiestos
        vIncremento = 29;
    }
    else
    {
        vIncremento = 28;
    }
}
if (vIncremento != 0)
{
    vDia = (vFechaDestino.Day + vIncremento) - vFechaOrigen.Day;
    vIncremento = 1;
}
else
{
    vDia = vFechaDestino.Day - vFechaOrigen.Day;
}

//Calculamos los meses
if ((vFechaOrigen.Month + vIncremento) > vFechaDestino.Month)
{
    vMes = (vFechaDestino.Month + 12) - (vFechaOrigen.Month +
vIncremento);
    vIncremento = 1;
}
else
{
    vMes = (vFechaDestino.Month) - (vFechaOrigen.Month +
vIncremento);
    vIncremento = 0;
}
//Calculamos los años
vAnio = vFechaDestino.Year - (vFechaOrigen.Year + vIncremento);
```

```

        return vAnio + " Año(s), " + vMes + " mes(es), " + vDia + "
día(s)";
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

protected void bt_obtener_Click(object sender, EventArgs e)
{
    tr_datProp.Visible = true;
    DateTime inicio = Convert.ToDateTime(tx_fechanac.Text + " 00:00:00");
    DateTime final = DateTime.Today;
    tx_edad.Text = DiferenciaFechas(inicio, final);
}

public void validarFecha()
{
    DateTime fechoy = DateTime.Today;
    DateTime fechalimite = DateTime.Today.AddYears(-18);
    RangeValidator2.MaximumValue = fechoy.ToString().Substring(0, 10);
    RangeValidator2.MinimumValue = fechalimite.ToString().Substring(0,
10);
}
}
}
}

```

Seleccionar Mascota

```

protected void bt_seleccionar_Click(object sender, ImageClickEventArgs e)
{
    int id = int.Parse((sender as ImageButton).CommandArgument);
    Response.Redirect("M_ActualizarDatosMascota.aspx?encrypt=" + (id *
777));
}

```

Buscar Mascota

```

protected void bt_buscar_Click(object sender, EventArgs e)
{
    if (!ValidarCampos()) return;

    String cedula = tx_cedula.Text;
    modelo model = new modelo();
    Int32 Id_cliente = 0;
    String Nombre = string.Empty;
    String Apellido = string.Empty;
    String Direccion = string.Empty;
    try
    {
        MySqlDataReader Cliente =
model.consultar_datos_cliente_por_cedula(cedula);
        if (Cliente.Read())
        {
            Id_cliente = Cliente.GetInt32("id_cliente");

```

```

        Nombre = Cliente.GetString("nombre");
        lb_nomcliente.Text = Cliente.GetString("nombre") + "<br />"
" + Cliente.GetString("apellido");
        lb_direccion.Text = Cliente.GetString("direccion");
        lb_telf_fijo.Text = Cliente.GetString("telefono");
        lb_telfmovil.Text = Cliente.GetString("movil");
        lb_email.Text = Cliente.GetString("correo");
        Apellido = Cliente.GetString("apellido");
        Direccion = Cliente.GetString("direccion");
        lb_mensaje.Visible = false;

    }

    else
    {
        lb_mensaje.Text = "No se ha encontrado el cliente, intente
de nuevo";

        lb_mensaje.Visible = true;
        tbl_datoscliente.Visible = false;
        grid_mascotas.Visible = false;
    }

    MySqlDataReader Mascotas;
    if (Id_cliente > 0)
    {

        Mascotas =
model.consultar_mascotas_por_cliente(Id_cliente);
        grid_mascotas.DataSource = Mascotas;
        grid_mascotas.DataBind();
        tbl_datoscliente.Visible = true;
        grid_mascotas.Visible = true;

    }
    else
    {

    }
}
catch (Exception)
{
    mostrarAlerta_Jscript("Error al completar acción");
}
finally
{
    model.CerrarConexion();
}
}

```

Eliminar Mascota

```

protected void bt_eliminar_Click(object sender, ImageClickEventArgs e)
{

    int id = int.Parse((sender as ImageButton).CommandArgument);
    modelo model = new modelo();

```



```

int mascota = model.eliminar_mascota(id);
if (mascota > 0)
{
    mostrarAlerta_Jscript("Mascota Eliminada exitosamente");
    String cedula = tx_cedula.Text;

    Int32 Id_cliente = 0;
    String Nombre = string.Empty;
    String Apellido = string.Empty;
    String Direccion = string.Empty;
    try
    {
        MySqlDataReader Cliente =
model.consultar_datos_cliente_por_cedula(cedula);
        if (Cliente.Read())
        {
            Id_cliente = Cliente.GetInt32("id_cliente");
            Nombre = Cliente.GetString("nombre");
            lb_nomcliente.Text = Cliente.GetString("nombre") +
"<br />" + Cliente.GetString("apellido");
            Apellido = Cliente.GetString("apellido");
            Direccion = Cliente.GetString("direccion");

        }
        else
        {
            lb_mensaje.Text = "No se ha encontrado el cliente,
intente de nuevo";
            lb_mensaje.Visible = true;
        }

        MySqlDataReader Mascotas;
        if (Id_cliente > 0)
        {
            Mascotas =
model.consultar_mascotas_por_cliente(Id_cliente);
            grid_mascotas.DataSource = Mascotas;
            grid_mascotas.DataBind();
            lb_mostrarmascotas.Visible = true;
            grid_mascotas.Visible = true;

        }
        else
        {

        }
    }
    catch (Exception)
    {
        mostrarAlerta_Jscript("Error al completar acción");
    }
    finally
    {
        model.CerrarConexion();
    }
}
}

```

Acceso a Datos

```
public int nueva_mascota(Int32 id_cliente, Int32 id_tipo, String nombre, String
raza, String peso, String color, String sexo, String edad, String fechanac)
{
    int filas = 0;
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    try
    {
        SecuenciaSQL = string.Format(@"INSERT INTO
`veterinaria`.`mascota` VALUES (NULL, '{0}', '{1}', '{2}', 'No Disponible',
'{3}', '{4}', '{5}', '{6}', '{7}', 'A', '{8}');"
, id_cliente, id_tipo, nombre, raza, peso, color, sexo,
edad, fechanac);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        filas = sqlcom.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        throw e;
    }
    return filas;
}

public int actualizarFoto_mascota(Int32 id_mascota, String dir)
{
    int filas = 0;
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    try
    {
        SecuenciaSQL = string.Format(@" UPDATE mascota set foto='{1}'
where id_mascota={0} ;",
id_mascota, dir);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        filas = sqlcom.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        throw e;
    }
    return filas;
}
```

Vista del formulario de Usuario

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="AdminUsuario.aspx.cs"
Inherits="Veterinaria.Formulario_web13" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <script>
        $(function () {
            $('#MainContent_tx_cedula').keyup(function () {
```

```

var cedula = $('#MainContent_tx_cedula').val();
if (cedula.length == 10) {

    //Obtenemos el digito de la region que son los dos primeros
    digitos
    var digito_region = cedula.substring(0, 2);

    //Pregunto si la region existe ecuador se divide en 24
    regiones
    if (digito_region >= 1 && digito_region <= 24) {

        // Extraigo el ultimo digito
        var ultimo_digito = cedula.substring(9, 10);

        //Agrupo todos los pares y los sumo
        var pares = parseInt(cedula.substring(1, 2)) +
        parseInt(cedula.substring(3, 4)) + parseInt(cedula.substring(5, 6)) +
        parseInt(cedula.substring(7, 8));

        //Agrupo los impares, los multiplico por un factor de
        2, si la resultante es > que 9 le restamos el 9 a la resultante
        var numero1 = cedula.substring(0, 1);
        var numero1 = (numero1 * 2);
        if (numero1 > 9) { var numero1 = (numero1 - 9); }

        var numero3 = cedula.substring(2, 3);
        var numero3 = (numero3 * 2);
        if (numero3 > 9) { var numero3 = (numero3 - 9); }

        var numero5 = cedula.substring(4, 5);
        var numero5 = (numero5 * 2);
        if (numero5 > 9) { var numero5 = (numero5 - 9); }

        var numero7 = cedula.substring(6, 7);
        var numero7 = (numero7 * 2);
        if (numero7 > 9) { var numero7 = (numero7 - 9); }

        var numero9 = cedula.substring(8, 9);
        var numero9 = (numero9 * 2);
        if (numero9 > 9) { var numero9 = (numero9 - 9); }

        var impares = numero1 + numero3 + numero5 + numero7 +
        numero9;

        //Suma total
        var suma_total = (pares + impares);

        //extraemos el primero digito
        var primer_digito_suma =
        String(suma_total).substring(0, 1);

        //Obtenemos la decena inmediata
        var decena = (parseInt(primer_digito_suma) + 1) * 10;

        //Obtenemos la resta de la decena inmediata - la
        suma_total esto nos da el digito validador
        var digito_validador = decena - suma_total;

        //Si el digito validador es = a 10 toma el valor de 0
        if (digito_validador == 10)
    }
}

```

```

        var digito_validador = 0;

        //Validamos que el digito validador sea igual al de la
cedula

        if (digito_validador == ultimo_digito) {
            $('#notifica').removeAttr('style');
            $('#notifica').attr('style', 'color: green;');
            $('#notifica').text('Cedula Correcta');

$('##MainContent_bt_regUusuario').removeAttr('disabled');

            } else {
                $('#notifica').removeAttr('style');
                $('#notifica').attr('style', 'color: red;');
                $('#notifica').text('Cedula Incorrecta');
                $('#MainContent_bt_regUusuario').attr('disabled',
'disabled');
            }
        } else {
            // imprimimos en consola si la region no pertenece
            $('#notifica').removeAttr('style');
            $('#notifica').attr('style', 'color: red;');
            $('#notifica').text('Cedula Incorrecta');
            $('#MainContent_bt_regUusuario').attr('disabled',
'disabled');
        }
    } else {
        //imprimimos en consola si la cedula tiene mas o menos de
10 digitos

        $('#notifica').removeAttr('style');
        $('#notifica').attr('style', 'color: red;');
        $('#notifica').text('Cedula Incorrecta');
        $('#MainContent_bt_regUusuario').attr('disabled',
'disabled');
    }
});
});
</script>
<h1>Registro de Usuarios </h1><br />
<h2>Datos de Usuario:</h2><br />
<p>
    <strong><asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/A_MuestraUsuarios.aspx">Ver Listado></asp:HyperLink></strong>
</p>
<p>
    <strong><asp:Label ID="lb_msjcedula" runat="server" style="color:
#CC0000" ></asp:Label></strong>
</p>
<table class="table table-responsive">
    <tr>
        <td style="width: 98px"><b>Nº Cédula:</b></td>
        <td style="width: 431px">
            <asp:TextBox ID="tx_cedula" runat="server" CssClass="form-
control" Width="200px"></asp:TextBox>
            <strong id="notifica"></strong>
        </td>
        <td style="width: 101px"><b>E-mail:</b></td>
    </tr>

```

```

        <asp:TextBox ID="tx_correo" runat="server" Width="238px"
        CssClass="form-control"></asp:TextBox><strong><asp:RegularExpressionValidator
        ID="RegularExpressionValidator1" runat="server" ControlToValidate="tx_correo"
        ErrorMessage="E-mail invalido "
        ValidationExpression="\w+([-+.' ]\w+)*@\w+([-.\ ]\w+)*\.\w+([-.\ ]\w+)*"
        ForeColor="Red" style="color:
        #CC0000"></asp:RegularExpressionValidator></strong>
    </td>
</tr>
<tr>
    <td style="width: 98px"><b>Rol:</b></td>
    <td style="width: 431px">
        <asp:DropDownList ID="drp_rol" runat="server" CssClass="form-
        control">
            <asp:ListItem Text="Elija..."></asp:ListItem>
            <asp:ListItem Value="1">Normal</asp:ListItem>
            <asp:ListItem Value="2">Administrador</asp:ListItem>
        </asp:DropDownList>

    </td>
    <td style="width: 101px"><b>Teléfono:</b></td>
    <td>
        <asp:TextBox ID="tx_telefono" runat="server" CssClass="form-
        control">
        </asp:TextBox>
        <strong><asp:RangeValidator ID="RangeValidator1" runat="server"
        ErrorMessage="Número Telefónico inválido" MaximumValue="9999999"
        MinimumValue="2000000" ControlToValidate="tx_telefono" Type="Integer"
        style="color: #CC0000"></asp:RangeValidator></strong>
    </td>
</tr>
<tr>
    <td style="width: 98px"><b>Nombres:</b></td>
    <td style="width: 431px">
        <asp:TextBox ID="tx_nombre" runat="server" Width="297px"
        CssClass="form-control"></asp:TextBox><strong><asp:RegularExpressionValidator
        ID="RegularExpressionValidator2" runat="server" ControlToValidate="tx_nombre"
        ErrorMessage="Por favor Ingrese solo letras"
        ValidationExpression="^[a-zñ-ñ-á-é-í-ó-úA-Z ]*$" ForeColor="Red" style="color:
        #CC0000"></asp:RegularExpressionValidator></strong>
    </td>
    <td style="width: 101px"><b>Seudónimo:</b></td>
    <td>
        <asp:TextBox ID="tx_seudonimo" runat="server" Width="118px"
        CssClass="form-control"></asp:TextBox>
    </td>
</tr>
<tr>
    <td style="width: 98px"><b>Apellidos:</b></td>
    <td style="width: 431px">
        <asp:TextBox ID="tx_apellido" runat="server" Width="296px"
        CssClass="form-control"></asp:TextBox><strong><asp:RegularExpressionValidator
        ID="RegularExpressionValidator3" runat="server"
        ControlToValidate="tx_apellido"
        ErrorMessage="Por favor Ingrese solo letras"
        ValidationExpression="^[a-zñ-ñ-á-é-í-ó-úA-Z ]*$" ForeColor="Red" style="color:
        #CC0000"></asp:RegularExpressionValidator></strong>
    </td>
    <td style="width: 101px"><b>Contraseña:</b></td>
    <td>

```

```
<asp:TextBox ID="tx_contra" runat="server" Width="187px"  
CssClass="form-control" TextMode="Password"></asp:TextBox><strong></strong>  
</td>  
</tr>  
<tr>  
    <td style="width: 98px">&nbsp;</td>  
    <td style="width: 431px">  
        &nbsp;</td>  
  
    <td style="width: 101px"><b>Confirmar Contraseña:</b></td>  
    <td>  
        <asp:TextBox ID="tx_ccontra" runat="server" Width="203px"  
CssClass="form-control" TextMode="Password"></asp:TextBox>  
        &nbsp;<strong><asp:Label ID="lb_confirm" runat="server"  
style="color: #CC0000"></asp:Label></strong></td>  
  
</tr>  
<tr>  
    <td style="height: 39px;" colspan="4">  
        <asp:Button ID="bt_regUusuario" runat="server" Text="Agregar Usuario"  
CssClass="templatemo-fore-button" Width="209px" OnClick="bt_regUusuario_Click"/>  
  
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <asp:Button ID="bt_limpiar" runat="server" Text="Limpiar"  
CssClass="btn btn-primary" OnClick="bt_limpiar_Click"/>  
  
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <asp:Button ID="Button4" runat="server" Text="Volver"  
CssClass="btn btn-primary" PostBackUrl=~ /A_MenuUsuarios.aspx"/>  
  
</td>  
  
</tr>  
<tr>  
    <td colspan="4">  
        &nbsp;</td>  
  
</tr>  
</table>  
</asp:Content>
```

Registro de un nuevo Usuario

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MySql.Data.MySqlClient;

namespace Veterinaria
{
    public partial class Formulario_web13 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void bt_limpiar_Click(object sender, EventArgs e)
        {
            LimpiarC();
        }
    }
}
```

```

    }
    public void LimpiarC()
    {
        tx_apellido.Text = string.Empty;
        tx_nombre.Text = string.Empty;
        tx_cedula.Text = string.Empty;
        tx_contra.Text = string.Empty;
        tx_correo.Text = string.Empty;
        tx_seudonimo.Text = string.Empty;
        tx_telefono.Text = string.Empty;
        tx_ccontra.Text = string.Empty;
        lb_confirm.Visible = false;

        lb_msjcedula.Visible = false;
        drp_rol.SelectedIndex = 0;
    }
    public void mostrarAlerta_Jscript(string Mensaje)
    {
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('" + Mensaje + "');", true);
    }

    protected void bt_regUusuario_Click(object sender, EventArgs e)
    {
        if (!VlidarC()) return;

        String cedula = tx_cedula.Text;
        String nombres = tx_nombre.Text;
        String apellidos = tx_apellido.Text;
        String correo = tx_correo.Text;
        String telefono = tx_telefono.Text;
        String seudonimo = tx_seudonimo.Text;
        String clave = tx_contra.Text;
        ModeloUsuario model = new ModeloUsuario();
        try
        {
            if(tx_contra.Text == tx_ccontra.Text)
            {
                Int32 rol =
System.Convert.ToInt32(drp_rol.SelectedValue.ToString());
                int u = model.registrar_usuario(rol, cedula, nombres,
apellidos, correo, telefono, seudonimo, clave);
                if (u > 0)
                {
                    mostrarAlerta_Jscript("Usuario Registrado con Éxito");
                    LimpiarC();
                }
            }
            //else
            //{
            //    mostrarAlerta_Jscript("Las contraseñas no coinciden");
            //    lb_confirm.Text = "Las contraseñas no coinciden";
            //}

        }
        catch (Exception ex)
        {

```

```
        mostrarAlerta_Jscript("Error al insertar");
    }

    finally
    {
    }

}

private bool VlidarC()
{
    if (tx_cedula.Text == string.Empty)
    {
        lb_msjcedula.Text= "¡Debe ingresar el número de cédula!";
        tx_cedula.Focus();
        return false;
    }
    else { lb_msjcedula.Visible = false; }

    if (drp_rol.SelectedIndex == 0)
    {
        mostrarAlerta_Jscript("Elija el perfil de usuario");
        drp_rol.Focus();
        return false;
    }

    if (tx_nombre.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese Los Nombres Completos");
        tx_nombre.Focus();
        return false;
    }

    if (tx_apellido.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese Los Apellidos");
        tx_apellido.Focus();
        return false;
    }
    if (tx_correo.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese un E-mail");
        tx_correo.Focus();
        return false;
    }

    if (tx_telefono.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese un número telefónico");
        tx_telefono.Focus();
        return false;
    }

    if(tx_seudonimo.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese un UserName o Seudónimo");
        tx_seudonimo.Focus();
        return false;
    }
}
```



```

    }

    if (tx_contra.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese una contraseña");
        tx_contra.Focus();
        return false;
    }
    if (tx_contra.Text != tx_ccontra.Text)
    {
        lb_confirm.Text = "Las contraseñas no coinciden";
        tx_contra.Focus();
        return false;
    }
    if (tx_contra.Text == tx_ccontra.Text)
    {
        lb_confirm.Text = "";
    }

    return true;
}
}
}
}

```

Actualizar Usuario

```

using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Veterinaria
{
    public partial class Formulario_web110 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                String id_encrypt = Request.QueryString["encrypt"];
                Int32 id = System.Convert.ToInt32(id_encrypt) / 777;
                ModeloUsuario model = new ModeloUsuario();
                MySqlDataReader client = model.consultar_usuarios_id(id);
                try
                {
                    if (client.Read())
                    {
                        tx_codigo.Text =
client.GetInt32("id_usuario").ToString();
                        tx_cedula.Text = client.GetString("cedula");
                        tx_nombre.Text = client.GetString("nombres");
                        tx_apellido.Text = client.GetString("apellidos");
                        tx_contra.Text = client.GetString("clave");
                    }
                }
            }
        }
    }
}

```

```
        tx_correo.Text = client.GetString("correo");
        tx_telefono.Text = client.GetString("telefono");
        tx_seudonimo.Text = client.GetString("seudonimo");
        drp_rol.SelectedValue = client.GetString("rol");
    }
}
catch (Exception ex)
{
}

}

}

protected void bt_limpiar_Click(object sender, EventArgs e)
{
    LimpiarC();
}

protected void bt_regUsuario_Click(object sender, EventArgs e)
{
    if (!validarC()) return;

    int id = System.Convert.ToInt32(tx_codigo.Text);
    Int32 rol =
System.Convert.ToInt32(drp_rol.SelectedValue.ToString());
    String nombres = tx_nombre.Text;
    String apellidos = tx_apellido.Text;
    String clave = tx_contra.Text;
    String telefono = tx_telefono.Text;
    String seudonimo = tx_seudonimo.Text;
    String correo = tx_correo.Text;
    ModeloUsuario model = new ModeloUsuario();

    int actualiza = model.actualizar_usuario(id, rol, nombres,
apellidos, correo, telefono, seudonimo, clave);
    if (actualiza > 0)
    {
        mostrarAlerta_Jscript("Los Datos se han Actualizado");
        //Response.Redirect("A_MuestraClientes.aspx");
        LimpiarC();
    }

    else
    {
        mostrarAlerta_Jscript("Error al actualizar");
    }
}

public void LimpiarC()
{
    drp_rol.SelectedIndex = 0;
    tx_nombre.Text = string.Empty;
    tx_apellido.Text = string.Empty;
    tx_telefono.Text = string.Empty;
    tx_correo.Text = string.Empty;
```

```
tx_seudonimo.Text = string.Empty;
tx_contra.Text = string.Empty;
tx_ccontra.Text = string.Empty;
lb_confirm.Visible = false;
tx_cedula.Text = string.Empty;
tx_codigo.Text = string.Empty;
}
public void mostrarAlerta_Jscript(string Mensaje)
{
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('" + Mensaje + "');", true);
}
private bool validarC()
{
    if (drp_rol.SelectedIndex == 0)
    {
        mostrarAlerta_Jscript("Elija el perfil de usuario");
        drp_rol.Focus();
        return false;
    }

    if (tx_nombre.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese Los Nombres Completos");
        tx_nombre.Focus();
        return false;
    }

    if (tx_apellido.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese Los Apellidos");
        tx_apellido.Focus();
        return false;
    }
    if (tx_correo.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese un E-mail");
        tx_correo.Focus();
        return false;
    }

    if (tx_telefono.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese un número telefónico");
        tx_telefono.Focus();
        return false;
    }

    if(tx_seudonimo.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese un UserName o Seudónimo");
        tx_seudonimo.Focus();
        return false;
    }

    if (tx_contra.Text == string.Empty)
    {
        mostrarAlerta_Jscript("Ingrese una contraseña");
        tx_contra.Focus();
        return false;
    }
}
```

```
    }  
    if (tx_contra.Text != tx_ccontra.Text)  
    {  
        lb_confirm.Text = "Las contraseñas no coinciden";  
        tx_contra.Focus();  
        return false;  
    }  
    if (tx_contra.Text == tx_ccontra.Text)  
    {  
        lb_confirm.Text = "";  
    }  
  
    return true;  
}  
}  
}
```

Eliminar Usuario.

```
protected void bt_eliminar_Click(object sender, ImageClickEventArgs e)  
{  
    try  
    {  
        int id = int.Parse((sender as ImageButton).CommandArgument);  
        ModeloUsuario model = new ModeloUsuario();  
  
        int mascota = model.eliminar_usuario(id);  
        if (mascota > 0)  
        {  
            mostrarAlerta_Jscript("Usuario Eliminado exitosamente");  
            GridViewUsuarios.DataSource = model.consultar_usuarios();  
            GridViewUsuarios.DataBind();  
        }  
    }  
    catch (Exception ex)  
    {  
        mostrarAlerta_Jscript("Error al Eliminar");  
    }  
}
```

Buscar Usuario.

```
protected void bt_buscar_Click(object sender, EventArgs e)  
{  
    ModeloUsuario model = new ModeloUsuario();  
    String nombre = tx_nombre.Text;  
  
    try  
    {  
        GridViewUsuarios.DataSource =  
model.consultar_UsuariosPorSeudonimoNombre(nombre, nombre);  
        GridViewUsuarios.DataBind();  
    }  
    catch (Exception ex)
```

```

    {
        mostrarAlerta_Jscript("No hay registros");
    }
}

```

Seleccionar Usuario.

```

protected void bt_seleccionar_Click(object sender, ImageClickEventArgs e)
{
    int id = int.Parse((sender as ImageButton).CommandArgument);
    Response.Redirect("A_ActualizarDatosUsuarios.aspx?encrypt=" + (id
* 777));
}

```

Acceso a Datos.

```

using System;
using MySql.Data.MySqlClient;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Veterinaria
{
    public class ModeloUsuario
    {
        Conexion conex = new Conexion();

        public int registrar_usuario(Int32 id_rol, String cedula, String
nombres, String apellidos, String correo, String telefono, String seudonimo,
String clave)
        {
            int filas = 0;
            String SecuenciaSQL;
            MySqlConnection con = conex.ObtenerConexion();
            MySqlCommand sqlcom;
            try
            {
                SecuenciaSQL = string.Format(@"INSERT INTO
`veterinaria`.`usuario` VALUES (NULL, '{0}', '{1}', '{2}', '{3}', '{4}',
'{5}', '{6}', '{7}', 'A');",
                id_rol, cedula, nombres, apellidos, correo, telefono,
seudonimo, clave);
                sqlcom = new MySqlCommand(SecuenciaSQL, con);
                filas = sqlcom.ExecuteNonQuery();
            }
            catch (Exception e)
            {
                throw e;
            }
            return filas;
        }

        public MySqlDataReader consultar_usuarios()
        {
            String SecuenciaSQL;
            MySqlConnection con = conex.ObtenerConexion();
            MySqlCommand sqlcom;
            MySqlDataReader resultData = null;
            try

```

```

        {
            SecuenciaSQL = string.Format(@"SELECT usuario.id_usuario,
usuario.rol, usuario.cedula, usuario.nombres, usuario.apellidos,
usuario.correo, usuario.telefono, usuario.seudonimo, usuario.estado FROM
usuario WHERE estado='A'");
            sqlcom = new MySqlCommand(SecuenciaSQL, con);
            resultData = sqlcom.ExecuteReader();
            return resultData;
        }
        catch (Exception ex)
        {
            return null;
            throw ex;
        }
        finally
        {
        }
    }

    public MySqlDataReader consultar_usuarios_id(int id_usuario)
    {
        String SecuenciaSQL;
        MySqlConnection con = conex.ObtenerConexion();
        MySqlCommand sqlcom;
        MySqlDataReader resultData = null;
        try
        {
            SecuenciaSQL = string.Format(@"SELECT * FROM `usuario`
WHERE usuario.id_usuario={0} ;", id_usuario);
            sqlcom = new MySqlCommand(SecuenciaSQL, con);
            resultData = sqlcom.ExecuteReader();
            return resultData;
        }
        catch (Exception ex)
        {
            return null;
            throw ex;
        }
        finally
        {
        }
    }

    public int eliminar_usuario(int id_usuario)
    {
        int filas = 0;
        String SecuenciaSQL;
        MySqlConnection con = conex.ObtenerConexion();
        MySqlCommand sqlcom;
        try
        {
            SecuenciaSQL = string.Format(@"UPDATE usuario SET
estado='I' WHERE id_usuario={0}", id_usuario);
            sqlcom = new MySqlCommand(SecuenciaSQL, con);
            filas = sqlcom.ExecuteNonQuery();
        }
        catch (Exception e)
        {
            throw e;
        }
    }

```

```

    }
    return filas;
}

public int actualizar_usuario(Int32 id_usuario, Int32 rol, String
nombres, String apellidos, String correo, String telefono, String seudonimo,
String clave)
{
    int filas = 0;
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    try
    {
        SecuenciaSQL = string.Format(@"UPDATE
`veterinaria`.`usuario` set rol='{1}', nombres='{2}', apellidos='{3}',
correo='{4}', telefono='{5}', seudonimo='{6}', clave=MD5('{7}') where
id_usuario={0};",
        id_usuario, rol, nombres, apellidos, correo, telefono,
seudonimo, clave);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        filas = sqlcom.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        throw e;
    }
    finally
    {
        CerrarConexion();
    }
    return filas;
}

public MySqlDataReader consultar_UsuariosPorSeudonimoNombre(String
nombre_usuario, String seudonimo_usuario)
{
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    MySqlDataReader resultData = null;
    try
    {
        SecuenciaSQL = string.Format(@"SELECT * FROM usuario WHERE
usuario.estado='A' AND usuario.seudonimo like '{0}%' OR usuario.nombres
like'{0}%', nombre_usuario, seudonimo_usuario);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        resultData = sqlcom.ExecuteReader();
        return resultData;
    }
    catch (Exception ex)
    {
        return null;
        throw ex;
    }
    finally
    {
    }
}

public void CerrarConexion()
{

```

```

        conex.CerrarConexion();
    }

}

```

Vista del formulario de citas.

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="T_CitaMedica.aspx.cs"
Inherits="Veterinaria.Formulario_web14" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit"
TagPrefix="tool" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

    <asp:Label ID="lb_fecha" runat="server" Width="253px"></asp:Label><br />
    <br />

    <h2>ASIGNACIÓN DE CITAS PREVIAS PARA CONSULTA.</h2>

    <div class="templatemo-content-widget white-bg">
        <table class="table table-responsive">
            <thead>
                <tr>
                    <td colspan="4"><b>INGRESE EL Nº DE CÉDULA DEL PROPIETARIO
DEL PACIENTE PARA REALIZAR LA BÚSQUEDA.</b>
                </td>
            </tr>
            </thead>
            <tbody>
                <tr>
                    <td style="width: 137px">
                        <b>Cedula del Cliente:</b></td>
                    <td style="width: 156px">
                        <asp:TextBox ID="tx_cedulaCliente" runat="server"
CssClass="form-control"></asp:TextBox></td>
                    <td style="width: 199px">
                        <asp:Button ID="btn_buscar" runat="server"
Text="Buscar" CssClass="btn btn-primary" OnClick="btn_buscar_Click"
Height="34px" Width="123px" /></td>
                    <td>
                        <asp:TextBox ID="tx_idCliente" runat="server"
Visible="false" CssClass="form-control"></asp:TextBox>
                    </td>
                </tr>
            </tbody>
        </table>
    </div>
    <div class="templatemo-content-widget white-bg" id="div_datos"
runat="server" visible="false">
        <table class="table table-responsive">
            <thead>
                <tr>
                    <td colspan="8"><b>DATOS GENERALES DEL PROPIETARIO
(VERIFIQUELOS ANTES DE CONTINUAR CON EL PROCESO.)</b>
                </td>
            </thead>
            <tbody>

```



```

<tr id="tr_nombre" runat="server">
    <td style="width: 75px">

        <b>Nombre del Cliente:</b></td>
    <td style="width: 138px">
        <asp:Label ID="lb_nombreCliente"
runat="server"></asp:Label><br />
        <asp:Label ID="lb_apellidos" runat="server"></asp:Label>

    </td>
    <td style="width: 86px">

        <b>Dirección:</b></td>
    <td>

        <asp:Label ID="lb_direccion" runat="server"></asp:Label>

    </td>
    <td>

        <b>Teléfono:</b></td>
    <td>

        <asp:Label ID="lb_telefono" runat="server"></asp:Label>

    </td>
    <td>

        <b>E-mail:</b></td>
    <td>

        <asp:Label ID="lb_email" runat="server"></asp:Label>

    </td>

</tr>
<tr id="tr_1" runat="server">
    <td colspan="8">
        <b>INGRESE LA FECHA PARA VER LA DISPONIBILIDAD DE
HORARIOS.</b>
    </td>
</tr>
<tr id="tr_2" runat="server">
    <td style="width: 52px">
        <b>Fecha:</b></td>

    <td style="width: 138px">
        <asp:TextBox ID="tx_fecha" runat="server" CssClass="form-
control" Width="109px"></asp:TextBox>
        <tool:CalendarExtender ID="CalendarExtender1"
TargetControlID="tx_fecha" runat="server" PopupPosition="Right"
Animated="true" Format="dd/MM/yyyy" />

    </td>
    <td colspan="6">
        <asp:Button ID="btn_buscar_disp" runat="server"
CssClass="templatemo-edit-btn" Text="Ver Disponibilidad"
OnClick="btn_buscar_disp_Click" Width="144px" />
    </td>
</tr>

```

MEJORAMIENTO DEL PROCESO DE GESTIÓN DE FICHAS MÉDICAS VETERINARIAS
MEDIANTE UN APLICATIVO WEB PARA LA CLÍNICA VETERINARIA HUSKY

```

        <tr>
            <td>5</td>
            <td>
                <asp:Label ID="lbl_4" runat="server"></asp:Label></td>
            <td>
                <asp:Button ID="btn_4" runat="server"
                CssClass="templatemo-fore-button" Text="Disponible" OnClick="btn_4_Click"
                /></td>
            <td>6</td>
            <td>
                <asp:Label ID="lbl_5" runat="server"></asp:Label></td>
            <td>
                <asp:Button ID="btn_5" runat="server"
                CssClass="templatemo-fore-button" Text="Disponible" OnClick="btn_5_Click"
                /></td>
        </tr>
        <tr>
            <td>7</td>
            <td>
                <asp:Label ID="lbl_6" runat="server"></asp:Label></td>
            <td>
                <asp:Button ID="btn_6" runat="server"
                CssClass="templatemo-fore-button" Text="Disponible" OnClick="btn_6_Click"
                /></td>
            <td>8</td>
            <td>
                <asp:Label ID="lbl_7" runat="server"></asp:Label></td>
            <td>
                <asp:Button ID="btn_7" runat="server"
                CssClass="templatemo-fore-button" Text="Disponible" OnClick="btn_7_Click"
                /></td>
        </tr>
        <tr>
            <td>9</td>
            <td>
                <asp:Label ID="lbl_8" runat="server"></asp:Label></td>
            <td>
                <asp:Button ID="btn_8" runat="server"
                CssClass="templatemo-fore-button" Text="Disponible" OnClick="btn_8_Click"
                /></td>
            <td>10</td>
            <td>
                <asp:Label ID="lbl_9" runat="server"></asp:Label></td>
            <td>
                <asp:Button ID="btn_9" runat="server"
                CssClass="templatemo-fore-button" Text="Disponible" OnClick="btn_9_Click"
                /></td>
        </tr>
    </tbody>
</table>

    <div class="templatemo-content-widget white-bg" id="div_datos1"
    runat="server" visible="false">
        <table runat="server" id="tabla_mascotas" visible="false" class="table
        table-responsive">
            <tbody>
                <tr>
                    <td>
                        <label id="lb_horarioS">Horario Seleccionado: </label>

```

```

        <asp:TextBox ID="tx_codigo_horario" runat="server"
Visible="false"></asp:TextBox>
        <asp:Label ID="lb_horario_select"
runat="server"></asp:Label>
    </td>
</tr>
<tr runat="server" visible="false" id="grd_mascota">
    <td>
        <b>SELECCIONE EL PACIENTE AL QUE DESEA ASIGNAR EL
HORARIO.</b>
    </td>
</tr>
<tr>
    <td style="text-align:center">
        <asp:GridView CssClass="table table-responsive"
ID="GridViewListaMascotas" runat="server"
        HeaderStyle-BackColor="#167d78"
BorderColor="Transparent" AutoGenerateColumns="false">
            <Columns>
                <asp:BoundField DataField="id_mascota"
HeaderText="Codigo" />
                <asp:BoundField DataField="descripcion"
HeaderText="Especie" />
                <asp:BoundField DataField="nombre"
HeaderText="Nombre" />
                <asp:BoundField DataField="raza" HeaderText="Raza"
/>
                <asp:BoundField DataField="peso" HeaderText="Peso"
/>
                <asp:BoundField DataField="color"
HeaderText="Color" />
                <asp:BoundField DataField="sexo" HeaderText="Sexo"
/>
                <asp:BoundField DataField="edad" HeaderText="Edad"
/>
                <asp:TemplateField>
                    <ItemTemplate>
                        <asp:Button ID="btn_registrarCita"
CommandArgument='<%# Eval("id_mascota") %>' runat="server" Text="Asignar Cita"
CssClass="templatemo-edit-btn" OnClick="btn_registrarCita_Click"
OnClientClick="return confirm('¿Dese asignar este horario al
paciente?')"></asp:Button>
                    </ItemTemplate>
                </asp:TemplateField>
            </Columns>
        </asp:GridView>
        <br />
    </td>
</tr>
<tr>
    <td>
        <asp:Button ID="Salir" runat="server" Text="Volver al
Panel" CssClass="templatemo-fore-button" PostBackUrl="~/T_Consultas.aspx" />
        <asp:Button ID="btn_cancelar" runat="server"
Text="Cancelar" CssClass="templatemo-fore-button"
PostBackUrl="~/T_CitaMedica.aspx" Visible="false" />
    </td>
</tr>
</tbody>
</table>

```

```
</div>  
</asp:Content>
```

Programación

```
using MySql.Data.MySqlClient;  
using System;  
using System.Collections.Generic;  
using System.Drawing;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
  
namespace Veterinaria  
{  
    public partial class Formulario_web14 : System.Web.UI.Page  
    {  
        protected void Page_Load(object sender, EventArgs e)  
        {  
            if(!Page.IsPostBack)  
            {  
                String fecha = DateTime.Today.ToLongDateString();  
                lb_fecha.Text = fecha;  
                lb_fecha.Visible = true;  
  
                CalendarExtender1.StartDate = DateTime.Today.AddDays(1);  
                ValidarFecha();  
            }  
        }  
  
        public void disponibilidadCita()  
        {  
            modelo model = new modelo();  
  
            MySqlDataReader horarios = model.consultar_horarios();  
            String[] MisHorarios = new String[20];  
            int[] Mis_id_Horarios = new int[20];  
            int[] Mis_horarios_cita = new int[20];  
            int i = 0;  
            while (horarios.Read())  
            {  
                Mis_id_Horarios[i] = horarios.GetInt32("id_horario");  
                MisHorarios[i] = horarios.GetString("descripcion");  
                i++;  
            }  
  
            model.CerrarConexion();  
  
            lbl_0.Text = MisHorarios[0];  
            lbl_1.Text = MisHorarios[1];  
            lbl_2.Text = MisHorarios[2];  
            lbl_3.Text = MisHorarios[3];  
            lbl_4.Text = MisHorarios[4];  
            lbl_5.Text = MisHorarios[5];  
            lbl_6.Text = MisHorarios[6];  
            lbl_7.Text = MisHorarios[7];  
            lbl_8.Text = MisHorarios[8];  
        }  
    }  
}
```

```
lbl_9.Text = MisHorarios[9];

i = 0;
MySQLDataReader horariosCita =
model.consultar_horarios_citas_por_fecha(tx_fecha.Text);
while (horariosCita.Read())
{
    Mis_horarios_cita[i] = horariosCita.GetInt32("id_horario");
    this.deshabilita(Mis_horarios_cita[i]);
    i++;
}

}

public void deshabilita(int id_horario){
    switch (id_horario)
    {
        case 1:
            btn_0.Enabled = false;
            btn_0.Text = "Asignado";
            btn_0.CssClass = "templatemo-red-button";
            break;

        case 2:
            btn_1.Enabled = false;
            btn_1.Text = "Asignado";
            btn_1.CssClass = "templatemo-red-button";
            break;

        case 3:
            btn_2.Enabled = false;
            btn_2.Text = "Asignado";
            btn_2.CssClass = "templatemo-red-button";
            break;

        case 4:
            btn_3.Enabled = false;
            btn_3.Text = "Asignado";
            btn_3.CssClass = "templatemo-red-button";
            break;

        case 5:
            btn_4.Enabled = false;
            btn_4.Text = "Asignado";
            btn_4.CssClass = "templatemo-red-button";
            break;

        case 6:
            btn_5.Enabled = false;
            btn_5.Text = "Asignado";
            btn_5.CssClass = "templatemo-red-button";
            break;

        case 7:
            btn_6.Enabled = false;
            btn_6.Text = "Asignado";
            btn_6.CssClass = "templatemo-red-button";
            break;
    }
}
```

```
case 8:
    btn_7.Enabled = false;
    btn_7.Text = "Asignado";
    btn_7.CssClass = "templatemo-red-button";
    break;

case 9:
    btn_8.Enabled = false;
    btn_8.Text = "Asignado";
    btn_8.CssClass = "templatemo-red-button";
    break;

case 10:
    btn_9.Enabled = false;
    btn_9.Text = "Asignado";
    btn_9.CssClass = "templatemo-red-button";
    break;

default:
    break;
    }
}

protected void btn_buscar_Click(object sender, EventArgs e)
{
    div_datos.Visible = true;

    modelo model = new modelo();
    try
    {
        String cedula = tx_cedulaCliente.Text;
        MySqlDataReader Datacliente =
model.consultar_datos_cliente_por_cedula(cedula);
        if(Datacliente.Read())
        {
            lb_nombreCliente.Text = Datacliente.GetString("nombre");
            lb_apellidos.Text = Datacliente.GetString("apellido");
            lb_direccion.Text = Datacliente.GetString("direccion");
            lb_telefono.Text = Datacliente.GetString("telefono");
            lb_email.Text = Datacliente.GetString("correo");

            int id_cliente = Datacliente.GetInt32("id_cliente");
            tx_idCliente.Text = id_cliente + "";
            GridViewListaMascotas.DataSource =
model.consultar_mascotas_por_cliente(id_cliente);
            GridViewListaMascotas.DataBind();

        }
        else
        {
            mostrarAlerta_Jscript("Cliente no Encontrado");
            div_datos.Visible = false;

        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

```

    }
}
public void mostrarAlerta_Jscript(string Mensaje)
{
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('" + Mensaje + "');", true);
}

protected void btn_registrarCita_Click(object sender, EventArgs e)
{
    try
    {
        Int32 id_cliente = System.Convert.ToInt32(tx_idCliente.Text);
        Int32 id_mascota = int.Parse((sender as
Button).CommandArgument);
        String fecha = tx_fecha.Text;
        String fechaHoy = DateTime.Now.ToString().Substring(0,10);
        String hora = tx_codigo_horario.Text;
        modelo model = new modelo();
        int r = model.registrar_cita_medica_(id_cliente, id_mascota,
fechaHoy, fecha, hora);
        if (r > 0)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this, this.GetType(),
"", "Cita asignada exitosamente");", true);
        }
    }
    catch (Exception)
    {
    }
}

protected void btn_buscar_disp_Click(object sender, EventArgs e)
{
    div_datos1.Visible = false;
    if (!ValidarC()) return;

    table_disponibilidad.Visible = true;
    habilitar_todo();
    disponibilidadCita();
}

public void habilitar_todo()
{
    btn_0.Enabled = true;
    btn_0.Text = "Disponible";
    btn_0.CssClass = "templatemo-fore-button";

    btn_1.Enabled = true;
    btn_1.Text = "Disponible";
    btn_1.CssClass = "templatemo-fore-button";
}

```



```
btn_2.Enabled = true;
btn_2.Text = "Disponible";
btn_2.CssClass = "templatemo-fore-button";

btn_3.Enabled = true;
btn_3.Text = "Disponible";
btn_3.CssClass = "templatemo-fore-button";

btn_4.Enabled = true;
btn_4.Text = "Disponible";
btn_4.CssClass = "templatemo-fore-button";

btn_5.Enabled = true;
btn_5.Text = "Disponible";
btn_5.CssClass = "templatemo-fore-button";

btn_6.Enabled = true;
btn_6.Text = "Disponible";
btn_6.CssClass = "templatemo-fore-button";

btn_7.Enabled = true;
btn_7.Text = "Disponible";
btn_7.CssClass = "templatemo-fore-button";

btn_8.Enabled = true;
btn_8.Text = "Disponible";
btn_8.CssClass = "templatemo-fore-button";

btn_9.Enabled = true;
btn_9.Text = "Disponible";
btn_9.CssClass = "templatemo-fore-button";
}

protected void btn_0_Click(object sender, EventArgs e)
{
    div_datos1.Visible = true;
    tx_codigo_horario.Text = "1";
    lb_horario_select.Text = lbl_0.Text;
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion();", true);
    tabla_mascotas.Visible = true;
    table_disponibilidad.Visible = false;
}
```

```
protected void btn_1_Click(object sender, EventArgs e)
{
    div_datos1.Visible = true;
    tx_codigo_horario.Text = ""+2;
    lb_horario_select.Text = lbl_1.Text;
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion()", true);
    tabla_mascotas.Visible = true;
    table_disponibilidad.Visible = false;
}

protected void btn_2_Click(object sender, EventArgs e)
{
    div_datos1.Visible = true;
    tx_codigo_horario.Text = "" + 3;
    lb_horario_select.Text = lbl_2.Text;
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion()", true);
    tabla_mascotas.Visible = true;
    table_disponibilidad.Visible = false;
}

protected void btn_3_Click(object sender, EventArgs e)
{
    div_datos1.Visible = true;
    tx_codigo_horario.Text = "" + 4;
    lb_horario_select.Text = lbl_3.Text;
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion()", true);
    tabla_mascotas.Visible = true;
    table_disponibilidad.Visible = false;
}

protected void btn_4_Click(object sender, EventArgs e)
{
    div_datos1.Visible = true;
    tx_codigo_horario.Text = "" + 5;
    lb_horario_select.Text = lbl_4.Text;
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion()", true);
    tabla_mascotas.Visible = true;
    table_disponibilidad.Visible = false;
}

protected void btn_5_Click(object sender, EventArgs e)
{
    div_datos1.Visible = true;
    tx_codigo_horario.Text = "" + 6;
    lb_horario_select.Text = lbl_5.Text;
    System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion()", true);
    tabla_mascotas.Visible = true;
    table_disponibilidad.Visible = false;
}

protected void btn_6_Click(object sender, EventArgs e)
{
    div_datos1.Visible = true;
    tx_codigo_horario.Text = "" + 7;
```

```

        lb_horario_select.Text = lbl_6.Text;
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion();", true);
        tabla_mascotas.Visible = true;
        table_disponibilidad.Visible = false;
    }

    protected void btn_7_Click(object sender, EventArgs e)
    {
        div_datos1.Visible = true;
        tx_codigo_horario.Text = "" + 8;
        lb_horario_select.Text = lbl_7.Text;
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion();", true);
        tabla_mascotas.Visible = true;
        table_disponibilidad.Visible = false;
    }

    protected void btn_8_Click(object sender, EventArgs e)
    {
        div_datos1.Visible = true;
        tx_codigo_horario.Text = "" + 9;
        lb_horario_select.Text = lbl_8.Text;
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion();", true);
        tabla_mascotas.Visible = true;
        table_disponibilidad.Visible = false;
    }

    protected void btn_9_Click(object sender, EventArgs e)
    {
        div_datos1.Visible = true;
        tx_codigo_horario.Text = "" + 10;
        lb_horario_select.Text = lbl_9.Text;
        System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "cerrarAcorcion();", true);
        tabla_mascotas.Visible = true;
        table_disponibilidad.Visible = false;
    }

    public void ValidarFeHa()
    {
        DateTime fechalimite = DateTime.Today.AddYears(1);
        RangeValidator1.MaximumValue = fechalimite.ToString().Substring(0,
10);

        DateTime fechacita = DateTime.Today.AddDays(1);
        RangeValidator1.MinimumValue = fechacita.ToString().Substring(0,
10);
    }

    private bool ValidarC()
    {
        if (tx_fecha.Text == string.Empty)
        {
            System.Web.UI.ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "", "alert('Por favor Proporcione una fecha');", true);
            tx_fecha.Focus();
            table_disponibilidad.Visible = false;
            return false;
        }
        return true;
    }

```

```
    }
  }
}
```

Acceso a Datos.

```
public int registrar_cita_medica_(Int32 id_cliente, Int32 id_mascota, String
fechaHoy, String fechaCita, String hora)
{
    int filas = 0;
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    try
    {
        SecuenciaSQL = string.Format(@" INSERT INTO citas (`id_cita`,
`id_cliente`, `id_mascota`, `fecha_pedido`, `id_horario`,
`diagnostico`, `observaciones`, `tratamiento`, `receta`,
`estado`, `fecha_cita`)
VALUES (null, '{0}', '{1}', '{2}', '{3}', NULL, NULL, NULL,
NULL, '{4}', '{5}' );",
id_cliente, id_mascota, fechaHoy, hora, "P", fechaCita);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        filas = sqlcom.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        throw e;
    }
    return filas;
}

public MySqlDataReader consultar_mascotas_por_cliente(int id_cliente)
{
    String SecuenciaSQL;
    MySqlConnection con = conex.ObtenerConexion();
    MySqlCommand sqlcom;
    MySqlDataReader resultData = null;
    try
    {
        SecuenciaSQL = string.Format(@"SELECT
mascota.id_mascota,tipo_mascota.descripcion,mascota.nombre,mascota.raza,
mascota.peso,mascota.color,mascota.sexo,mascota.fechaNac, mascota.edad
FROM mascota join tipo_mascota on
mascota.id_tipo=tipo_mascota.id_tipo where mascota.id_cliente={0} and
mascota.estado='A' ";", id_cliente);
        sqlcom = new MySqlCommand(SecuenciaSQL, con);
        resultData = sqlcom.ExecuteReader();
        return resultData;
    }
    catch (Exception)
    {
        return null;
    }
    finally
    {
    }
}
}
```

Script de la base de Datos

```
-- Estructura de tabla para la tabla `citas`
CREATE TABLE IF NOT EXISTS `citas` (
  `id_cita` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `id_cliente` int(10) unsigned NOT NULL,
  `id_mascota` int(10) unsigned NOT NULL,
  `id_horario` int(10) unsigned NOT NULL,
  `fecha_pedido` varchar(50) NOT NULL,
  `fecha_cita` varchar(20) NOT NULL,
  `diagnostico` varchar(250) DEFAULT NULL,
  `observaciones` varchar(250) DEFAULT NULL,
  `tratamiento` varchar(250) DEFAULT NULL,
  `receta` varchar(250) DEFAULT NULL,
  `estado` set('P','A','F') DEFAULT 'P',
  PRIMARY KEY (`id_cita`),
  KEY `rel_cita_cliente_idx` (`id_cliente`),
  KEY `rel_cita_mascota_idx` (`id_mascota`),
  KEY `rel_cita_horario_idx` (`id_horario`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=20 ;

-- Estructura de tabla para la tabla `clientes`
CREATE TABLE IF NOT EXISTS `clientes` (
  `id_cliente` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `cedula` varchar(13) NOT NULL,
  `nombre` varchar(45) NOT NULL,
  `apellido` varchar(45) NOT NULL,
  `direccion` varchar(45) DEFAULT NULL,
  `telefono` varchar(15) NOT NULL,
  `movil` varchar(15) DEFAULT NULL,
  `correo` varchar(50) DEFAULT NULL,
  `estado` set('A','I') NOT NULL DEFAULT 'A',
  PRIMARY KEY (`id_cliente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

-- Estructura de tabla para la tabla `consulta`
CREATE TABLE IF NOT EXISTS `consulta` (
  `id_consulta` int(11) NOT NULL AUTO_INCREMENT,
  `id_cliente` int(10) unsigned NOT NULL,
  `id_mascota` int(10) unsigned NOT NULL,
  `fecha` varchar(50) NOT NULL,
  `diagnostico` varchar(250) NOT NULL,
  `observaciones` varchar(250) DEFAULT NULL,
  `tratamiento` varchar(250) NOT NULL,
  `receta` varchar(250) DEFAULT NULL,
  PRIMARY KEY (`id_consulta`),
  KEY `rel_cliente_mascota_idx` (`id_cliente`),
  KEY `rel_mascota_consulta_idx` (`id_mascota`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;

-- Estructura de tabla para la tabla `desparasitacion`
CREATE TABLE IF NOT EXISTS `desparasitacion` (
  `id_desparasitacion` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `id_mascota` int(10) unsigned NOT NULL,
```

```

`id_tipo` int(10) unsigned NOT NULL,
`fechaAppli` varchar(20) DEFAULT NULL,
`fechSiguiete` varchar(20) NOT NULL,
`observaciones` varchar(250) NOT NULL,
PRIMARY KEY (`id_desparasitacion`),
KEY `rel_desparasitacion_mascota_idx` (`id_mascota`),
KEY `rel_desparasitacion_tipo_idx` (`id_tipo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

-- Estructura de tabla para la tabla `historias_clinicas`
CREATE TABLE IF NOT EXISTS `historias_clinicas` (
  `id_historia` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `id_mascota` int(10) unsigned NOT NULL,
  `motivo_consulta` varchar(250) DEFAULT NULL,
  `peso` decimal(10,2) DEFAULT NULL,
  `temperatura` decimal(10,2) DEFAULT NULL,
  `pulso` decimal(10,2) DEFAULT NULL,
  `frec_respiratoria` decimal(10,2) DEFAULT NULL,
  `palpacion` decimal(10,2) DEFAULT NULL,
  `sintomas` varchar(250) DEFAULT NULL,
  `diagnostico` varchar(250) DEFAULT NULL,
  `tratamiento` varchar(250) NOT NULL,
  `receta` varchar(250) DEFAULT NULL,
  `observaciones` varchar(250) DEFAULT NULL,
  `fecha` varchar(20) NOT NULL,
  PRIMARY KEY (`id_historia`),
  KEY `rel_historial_mascota_idx` (`id_mascota`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

-- Estructura de tabla para la tabla `horarios`
CREATE TABLE IF NOT EXISTS `horarios` (
  `id_horario` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(30) NOT NULL,
  PRIMARY KEY (`id_horario`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;

-- Estructura de tabla para la tabla `mascota`
CREATE TABLE IF NOT EXISTS `mascota` (
  `id_mascota` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `id_cliente` int(10) unsigned NOT NULL,
  `id_tipo` int(10) unsigned NOT NULL,
  `nombre` varchar(45) DEFAULT NULL,
  `foto` varchar(150) DEFAULT NULL,
  `raza` varchar(45) DEFAULT NULL,
  `peso` decimal(5,2) DEFAULT NULL,
  `color` varchar(45) DEFAULT NULL,
  `sexo` set('Macho','Hembra') DEFAULT NULL,
  `edad` varchar(50) NOT NULL,
  `estado` set('A','I') DEFAULT NULL,
  `fechaNac` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`id_mascota`),
  KEY `rel_mascota_cliente_idx` (`id_cliente`),
  KEY `rel_mascota_tipo_idx` (`id_tipo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=25 ;

```

```
-- Estructura de tabla para la tabla `roles`
CREATE TABLE IF NOT EXISTS `roles` (
  `id_rol` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_rol`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

-- Estructura de tabla para la tabla `tipos_vacuna`
CREATE TABLE IF NOT EXISTS `tipos_vacuna` (
  `id_tipo` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(45) DEFAULT NULL,
  `dias` varchar(10) NOT NULL,
  PRIMARY KEY (`id_tipo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- Estructura de tabla para la tabla `tipo_antiparasitario`
CREATE TABLE IF NOT EXISTS `tipo_antiparasitario` (
  `id_antiparasitario` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `anti_descripcion` varchar(50) NOT NULL,
  `anti_dias` varchar(50) NOT NULL,
  PRIMARY KEY (`id_antiparasitario`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- Estructura de tabla para la tabla `tipo_mascota`
CREATE TABLE IF NOT EXISTS `tipo_mascota` (
  `id_tipo` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(45) NOT NULL,
  `estado` set('A','I') NOT NULL,
  PRIMARY KEY (`id_tipo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

-- Estructura de tabla para la tabla `usuario`
CREATE TABLE IF NOT EXISTS `usuario` (
  `id_usuario` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `rol` int(10) unsigned NOT NULL DEFAULT '1',
  `cedula` varchar(10) NOT NULL,
  `nombres` varchar(50) NOT NULL,
  `apellidos` varchar(45) NOT NULL,
  `correo` varchar(45) NOT NULL,
  `telefono` varchar(45) NOT NULL,
  `seudonimo` varchar(45) NOT NULL,
  `clave` varchar(50) NOT NULL,
  `estado` set('A','I') NOT NULL DEFAULT 'A',
  PRIMARY KEY (`id_usuario`),
  UNIQUE KEY `cedula_UNIQUE` (`cedula`),
  KEY `rel_usuario_rol_idx` (`rol`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;

-- Estructura de tabla para la tabla `vacunacion`
CREATE TABLE IF NOT EXISTS `vacunacion` (
  `id_vacunacion` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `id_mascota` int(10) unsigned NOT NULL,
  `id_tipo` int(10) unsigned NOT NULL,
  `fecha` varchar(20) DEFAULT NULL,
```

```
`fechaSiguiete` varchar(20) NOT NULL,  
`observaciones` varchar(250) NOT NULL,  
PRIMARY KEY (`id_vacunacion`),  
KEY `rel_vacunacion_mascota_idx` (`id_mascota`),  
KEY `rel_vacunacion_tipo_idx` (`id_tipo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;  
  
-- Filtros para la tabla `citas`  
ALTER TABLE `citas`  
  ADD CONSTRAINT `rel_cita_cliente` FOREIGN KEY (`id_cliente`) REFERENCES  
`clientes` (`id_cliente`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `rel_cita_horario` FOREIGN KEY (`id_horario`) REFERENCES  
`horarios` (`id_horario`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `rel_cita_mascota` FOREIGN KEY (`id_mascota`) REFERENCES  
`mascota` (`id_mascota`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
-- Filtros para la tabla `consulta`  
ALTER TABLE `consulta`  
  ADD CONSTRAINT `rel_cliente_consulta` FOREIGN KEY (`id_cliente`) REFERENCES  
`clientes` (`id_cliente`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `rel_mascota_consulta` FOREIGN KEY (`id_mascota`) REFERENCES  
`mascota` (`id_mascota`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
-- Filtros para la tabla `desparasitacion`  
ALTER TABLE `desparasitacion`  
  ADD CONSTRAINT `rel_desparasitacion_mascota` FOREIGN KEY (`id_mascota`) REFERENCES  
`mascota` (`id_mascota`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `rel_desparasitacion_tipo` FOREIGN KEY (`id_tipo`) REFERENCES  
`tipo_antiparasitario` (`id_antiparasitario`) ON DELETE NO ACTION ON UPDATE NO  
ACTION;  
  
-- Filtros para la tabla `historias_clinicas`  
ALTER TABLE `historias_clinicas`  
  ADD CONSTRAINT `rel_historial_mascota` FOREIGN KEY (`id_mascota`) REFERENCES  
`mascota` (`id_mascota`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
-- Filtros para la tabla `mascota`  
ALTER TABLE `mascota`  
  ADD CONSTRAINT `rel_mascota_cliente` FOREIGN KEY (`id_cliente`) REFERENCES  
`clientes` (`id_cliente`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `rel_mascota_tipo` FOREIGN KEY (`id_tipo`) REFERENCES  
`tipo_mascota` (`id_tipo`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
-- Filtros para la tabla `usuario`  
ALTER TABLE `usuario`  
  ADD CONSTRAINT `rel_usuario_rol` FOREIGN KEY (`rol`) REFERENCES `roles`  
(`id_rol`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
-- Filtros para la tabla `vacunacion`  
ALTER TABLE `vacunacion`  
  ADD CONSTRAINT `rel_vacunacion_mascota` FOREIGN KEY (`id_mascota`) REFERENCES  
`mascota` (`id_mascota`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `rel_vacunacion_tipo` FOREIGN KEY (`id_tipo`) REFERENCES  
`tipos_vacuna` (`id_tipo`) ON DELETE NO ACTION ON UPDATE NO ACTION;
```


Diccionario de Datos.

“Contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.” (Web, 2015) las especificaciones de cada campo son las siguientes.

PK= Clave primaria

NN=No admite null

UQ= Único

BIN = Binario

UN= Número con signos positivos

ZF=Rellenar con ceros a la izquierda de los números

AI= Autoincremental

Fuente: (2012, ASPW, 2012)

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_cliente	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
cedula	VARCHAR(13)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombre	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
apellido	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
direccion	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
telefono	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
movil	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
correo	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
estado	SET('A','I')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'A'

Figura 72. Diccionario de datos para la tabla Clientes.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_mascota	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
id_cliente	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_tipo	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombre	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
foto	VARCHAR(150)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
raza	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
peso	DECIMAL(5,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
color	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sexo	SET('Macho','Hembra')	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
edad	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
estado	SET('A','I')	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
fechaNac	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figura 73. Diccionario de datos para la tabla Mascota.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_usuario	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
rol	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
cedula	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombres	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
apellidos	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
correo	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
telefono	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
seudonimo	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
clave	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
estado	SET('A','I')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'A'
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 74. Diccionario de datos para la tabla Usuario.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_rol	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
descripcion	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 75. Diccionario de datos para la tabla Roles.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_consulta	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
id_cliente	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_mascota	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
diagnostico	VARCHAR(250)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
observaciones	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tratamiento	VARCHAR(250)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
receta	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 76. Diccionario de datos para la tabla Consulta.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_vacunacion	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
id_mascota	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_tipo	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
fechaSiguiente	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
observaciones	VARCHAR(250)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 77. Diccionario de datos para la tabla Vacunacion.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_desparasitacion	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_mascota	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_tipo	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fechaAppli	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fechaSiguiete	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
observaciones	VARCHAR(250)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 78. Diccionario de datos para la tabla Desparasitación.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_cita	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
id_cliente	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_mascota	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_horario	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha_pedido	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha_cita	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
diagnostico	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
observaciones	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tratamiento	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
receta	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
estado	SET('P','A','F')	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'P'
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 79. Diccionario de datos para la tabla Citas.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_historia	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
id_mascota	INT(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
motivo_consulta	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
peso	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
temperatura	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
pulso	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
frec_respiratoria	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
palpacion	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
sintomas	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
diagnostico	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tratamiento	VARCHAR(250)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
receta	VARCHAR(250)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
observaciones	VARCHAR(250)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
historias_clinicascol	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 80. Diccionario de datos para la tabla historias_clinicas.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_tipo	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
descripcion	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
estado	SET('A','I')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 81. Diccionario de datos para la tabla tipo_mascota.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_antiparasitario	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
anti_descripcion	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
anti_dias	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 82. Diccionario de datos para la tabla tipo_antiparasitario.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_horario	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
descripcion	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 83. Diccionario de datos para la tabla horarios.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_tipo	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
descripcion	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
dias	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 84. Diccionario de datos para la tabla tipos_vacuna.

Anexo A.10 Manual de Usuario

Manual de Usuario

Ingreso al sistema.

Figura 85. Ingreso al sistema.

Instrucciones:


- Se debe ingresar un nombre de usuario ya registrado.
- Se debe ingresar una contraseña
- Ingresar al Sistema pulsando en 



Figura 86. Página Principal del aplicativo.

Indicaciones:

- Podemos ingresar a los diferentes módulos dirigiéndonos al menú vertical o al menú horizontal.
- Si queremos terminar la sesión presionamos **SALIR**

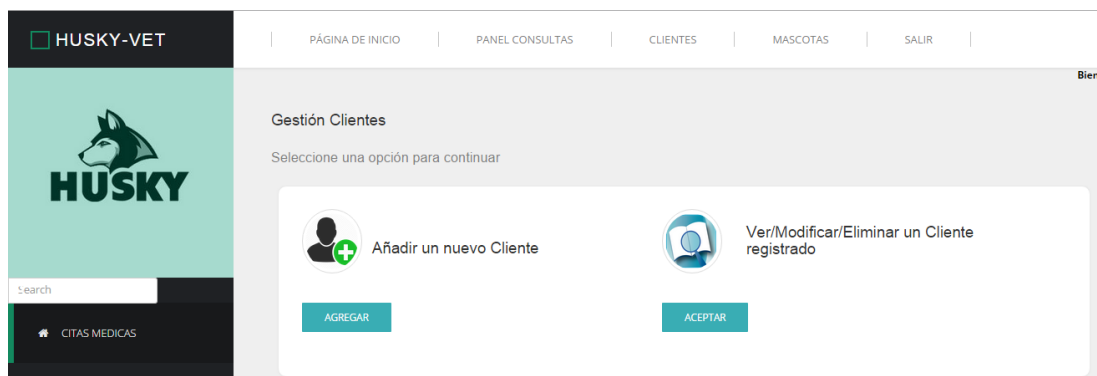


Figura 87. Menú de Gestión de Clientes.

Indicaciones:

- Si deseamos agregar un nuevo cliente presionamos el botón **AGREGAR**
- Si deseamos ver, modificar o eliminar un usuario registrado presionamos el botón **ACEPTAR** al lado derecho.

Favor llene todos los campos, son requeridos
[Ver Listado>>>](#)

Número de Cédula:	<input type="text"/>	Teléfono Convencional:	<input type="text" value="Ej: 2676441"/>
Nombres:	<input type="text"/>	Teléfono Móvil:	<input type="text" value="Ej: 0987396075"/>
Apellidos:	<input type="text"/>	E-mail:	<input type="text"/>
Dirección:	<input type="text"/>		

Figura 88. Ingreso de Clientes.

Indicaciones:

- Primeramente colocamos el número de cédula del cliente el cual debe estar correctamente ingresado
- Continuamos llenando los campos necesarios para poder registrar al cliente
- Una vez llenado todo presionamos el botón .
- Si queremos regresar presionamos el botón .
- Si queremos reestablecer los valores presionamos el botón .
- Si no ha habido errores en el registro nos aparecerá un mensaje de confirmación de guardado con éxito, caso contrario no nos permitirá guardar la información en el sistema.

Escriba el nombre del cliente:

Buscar

Seleccionar Eliminar	id_cliente	cedula	nombre	apellido	direccion	telefono	movil	correo	estado
<input checked="" type="checkbox"/> <input type="checkbox"/>	1	1719458075	Damian Benjamin	Navarrete Chávez	La Gonzáles Suárez	2308776	0987396075	the_forcex@hotmail.c	A
<input checked="" type="checkbox"/> <input type="checkbox"/>	2	1727342444	Angélica Janeth	Barreiro Cedeño	Chillogallo	3054457	0983778788	j.anggie@outlook.es	A
<input checked="" type="checkbox"/> <input type="checkbox"/>	3	1787096658	Renato Peter	Rodríguez Duran	Estadio Del Aucas	2686018	0983222672	piterdo@hotmail.es	A
<input checked="" type="checkbox"/> <input type="checkbox"/>	4	1708082514	Anabela Lady	Chávez Rodríguez	Antonio José de Sucre	2676441	0980796862	anabela.12@hotmail.com	A
<input checked="" type="checkbox"/> <input type="checkbox"/>	5	1722529532	Cristhian Mauricio	Barros	El Placer	2287934	0988776765	barros.cristhian@hotmail.com	A
<input checked="" type="checkbox"/> <input type="checkbox"/>	6	1724141088	Dario Xavier	Yanacallo	Puembo	2556566	0987665544	xavier@hotmail.com	A
<input checked="" type="checkbox"/> <input type="checkbox"/>	7	1711088896	Jose Esteban	Chávez	La Prensa	2667887	0976675545	jose.e@hotmail.com	A

VOLVER AL MENU DE CLIENTES NUEVO CLIENTE

Figura 88. Litado de Clientes activos.

Indicaciones:

- Podemos buscar un cliente por su nombre si el usuario lo requiere, introducimos el nombre y seguido damos clic en el botón buscar
- Para eliminar un cliente presionamos el botón con el ícono
- Para seleccionar un cliente y actualizar sus datos damos clic en el ícono y se abrirá la siguiente interfaz para actualizar los datos. (Figura 89)

ACTUALIZAR DATOS DE CLIENTES

DATOS DE CLIENTES:

Favor llene todos los campos, son requeridos

[Ver Listado>>>](#)

Código:

Número de Cédula: Teléfono Convencional:

Nombres: Teléfono Móvil:

Apellidos: E-mail:

Dirección:

ACTUALIZAR DATOS Cancelar Limpiar

Figura 89. Formulario de actualizar datos de un cliente.

- Después de haber hecho los cambios pertinentes presionamos en el botón

ACTUALIZAR DATOS

REGISTRAR MASCOTA

INGRESE EL NÚMERO DE CÉDULA DEL PROPIETARIO

Cedula:

[Ver datos del propietario](#)

DATO DE LA MASCOTA

Especie:	<input type="text" value="Elija..."/>	Peso: (Kg)	<input type="text" value="Peso (Kg)"/>
Sexo:	<input type="text" value="Elija..."/>	Color:	<input type="text"/>
Nombre:	<input type="text"/>	Fecha de Nacimiento:	<input type="text"/>
Raza:	<input type="text"/>	Calcular Edad	<input type="text"/>

[REGISTRAR MASCOTA](#)
[VER LISTADO](#)
[LIMPIAR](#)
[VOLVER](#)

Figura 90. Formulario para registrar una nueva mascota.

Indicaciones:

- Ingresamos un número de cédula de un cliente registrado para poder visualizar los datos presionando en el botón [Ver datos del propietario](#)
- Llenamos todos los campos los cuales son necesarios para poder completar el registro, para calcular la edad presionamos el botón [Calcular Edad](#)
- Hecho esto, presionamos el botón [REGISTRAR MASCOTA](#)
- Si deseamos ver el listado de todas las mascotas presionamos en el botón

[VER LISTADO](#)

INGRESE EL N° DE CÉDULA DEL PROPIETARIO DEL PACIENTE PARA REALIZAR LA BÚSQUEDA.

Cedula del Cliente:

DATOS GENERALES DEL PROPIETARIO (VERIFIQUELOS ANTES DE CONTINUAR CON EL PROCESO.)

Nombre del Cliente: Damian Benjamin Navarrete Chávez Dirección: La Gonzáles Suárez Teléfono: 2308776 E-mail: the_forcex@hotmail.c

INGRESE LA FECHA PARA VER LA DISPONIBILIDAD DE HORARIOS.

Fecha:

HORARIOS DISPONIBLES PARA LA FECHA INGRESADA.

No.	Horario	Estado	No.	Horario	Estado
1	08:00 - 09:00	<input type="button" value="DISPONIBLE"/>	2	09:00 - 10:00	<input type="button" value="ASIGNADO"/>
3	10:00 - 11:00	<input type="button" value="DISPONIBLE"/>	4	11:00 - 12:00	<input type="button" value="DISPONIBLE"/>
5	12:00 - 13:00	<input type="button" value="DISPONIBLE"/>	6	14:00 - 15:00	<input type="button" value="DISPONIBLE"/>
7	15:00 - 16:00	<input type="button" value="DISPONIBLE"/>	8	16:00 - 17:00	<input type="button" value="DISPONIBLE"/>
9	17:00 - 18:00	<input type="button" value="DISPONIBLE"/>	10	18:00 - 19:00	<input type="button" value="DISPONIBLE"/>

Figura 91. Sistema de agendación con citas previas.

Indicaciones:

- Primero ingresamos el número de cédula del propietario del paciente o mascota
- Una vez visualizados sus datos y confirmados, procedemos a ingresar una fecha para la cita previa presionando en el botón
- Se nos desplegará los horarios disponibles del médico veterinario para la fecha ingresada.
- Verificamos la disponibilidad y seguido damos clic en el botón
- Hecho esto se nos abrirá la lista de mascotas pertenecientes al propietario actual (Figura 92).

INGRESE EL N° DE CÉDULA DEL PROPIETARIO DEL PACIENTE PARA REALIZAR LA BÚSQUEDA.

Cedula del Cliente:

DATOS GENERALES DEL PROPIETARIO (VERIFIQUELOS ANTES DE CONTINUAR CON EL PROCESO.)

Nombre del Cliente: Damian Benjamin Navarrete Chávez Dirección: La Gonzáles Suárez Teléfono: 2308776 E-mail: the_forcex@hotmail.c

INGRESE LA FECHA PARA VER LA DISPONIBILIDAD DE HORARIOS.

Fecha:

Horario Seleccionado: 12:00 - 13:00

Codigo	Especie	Nombre	Raza	Peso	Color	Sexo	Edad	
4	Canino	Blanca	French Poodle	10,00	Blanco	Hembra	0 Año(s), 5 mes(es), 26 día(s)	<input type="button" value="Asignar Cita"/>
5	Canino	Josue	Pitbull	20,00	Gris	Macho	4 Año(s), 7 mes(es), 26 día(s)	<input type="button" value="Asignar Cita"/>
12	Felino	Federico	American Shorthair	9,00	Gris con Blanco	Macho	0 Año(s), 5 mes(es), 29 día(s)	<input type="button" value="Asignar Cita"/>

Figura 92. Asignación de cita al paciente.

- Procedemos a presionar el botón al paciente que queremos.

LISTADO DE CITAS PENDIENTES




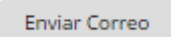
Realiza la búsqueda por:

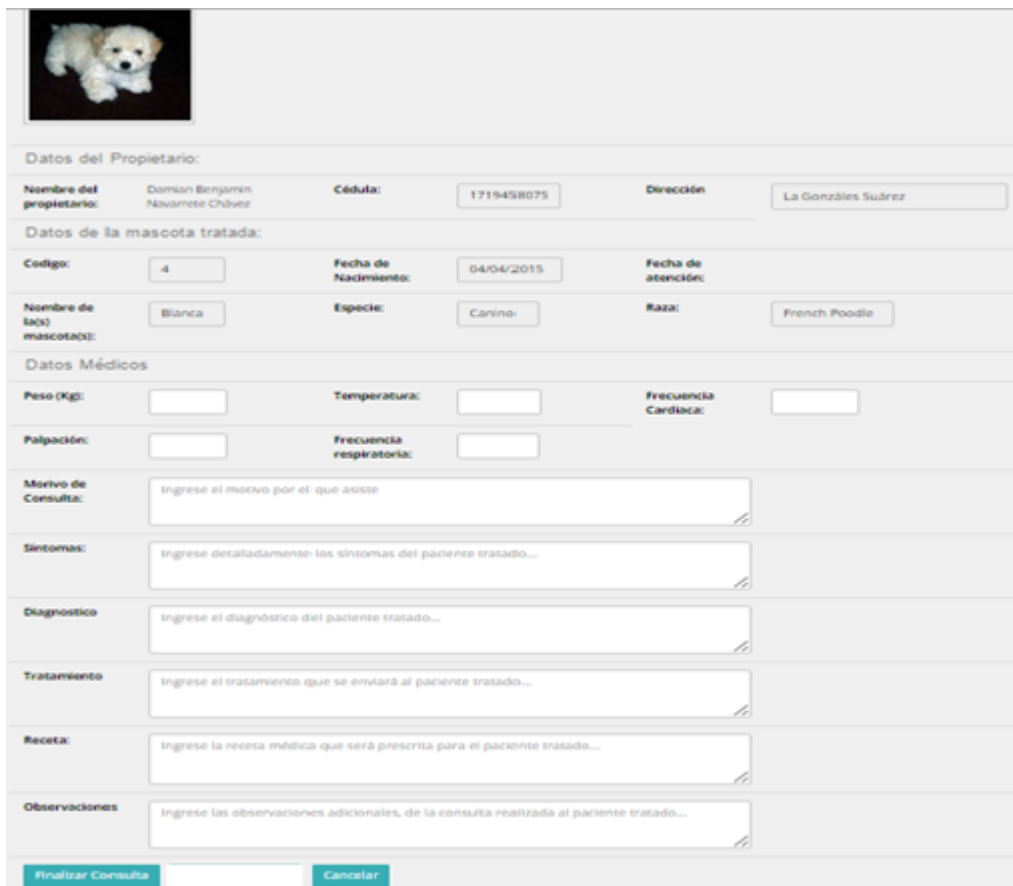
- Elejr...
- Nombre de Paciente
- N° Cita
- Nombre Propietario
- Todos

	N°	Nombre Prop	Apellido Prop	Cedula	Mascota	Raza	Edad	Fecha de registro	Fecha de cita	Hora	Accion
<input type="button" value="Enviar Correo"/>	15	Damian Benjamin	Navarrete Chávez	1719458075	Josue	Pitbull	4 Año(s), 7 mes(es), 26 día(s)	13/09/2015	05/09/2015	10:00 - 11:00	<input type="button" value="Abrir Consulta"/> <input type="button" value="Cancelar Cita"/>
<input type="button" value="Enviar Correo"/>	14	Damian Benjamin	Navarrete Chávez	1719458075	Blanca	French Poodle	0 Año(s), 5 mes(es), 26 día(s)	13/09/2015	05/09/2015	08:00 - 09:00	<input type="button" value="Abrir Consulta"/> <input type="button" value="Cancelar Cita"/>
<input type="button" value="Enviar Correo"/>	8	Damian Benjamin	Navarrete Chávez	1719458075	Federico	American Shorthair	0 Año(s), 5 mes(es), 29 día(s)	08/09/2015	13/09/2015	12:00 - 13:00	<input type="button" value="Abrir Consulta"/> <input type="button" value="Cancelar Cita"/>
<input type="button" value="Enviar Correo"/>	3	Angélica Janeth	Barreiro Cedeño	1727342444	Bruno	Pastor Aleman	4 Año(s), 7 mes(es), 26 día(s)	03/09/2015	13/09/2015	08:00 - 09:00	<input type="button" value="Abrir Consulta"/> <input type="button" value="Cancelar Cita"/>
<input type="button" value="Enviar Correo"/>	12	Damian Benjamin	Navarrete Chávez	1719458075	Blanca	French Poodle	0 Año(s), 5 mes(es), 26 día(s)	05/09/2015	13/09/2015	09:00 - 10:00	<input type="button" value="Abrir Consulta"/> <input type="button" value="Cancelar Cita"/>
<input type="button" value="Enviar Correo"/>	4	Angélica Janeth	Barreiro Cedeño	1727342444	Catrina	Maine Coon	3 Año(s), 3 mes(es), 15 día(s)	03/09/2015	13/09/2015	09:00 - 10:00	<input type="button" value="Abrir Consulta"/> <input type="button" value="Cancelar Cita"/>
<input type="button" value="Enviar Correo"/>	5	Angélica Janeth	Barreiro Cedeño	1727342444	Catrina	Maine Coon	3 Año(s), 3 mes(es), 15 día(s)	05/09/2015	13/09/2015	10:00 - 11:00	<input type="button" value="Abrir Consulta"/> <input type="button" value="Cancelar Cita"/>

Figura 93. Listado de citas pendientes de fechas posteriores.

Indicaciones:


- Podemos elegir el tipo de búsqueda que queremos realizar presionando en 
- Se nos desplegará una lista pequeña, seleccionamos el tipo de búsqueda, y en el campo de texto lo digitamos, y damos clic en Buscar
- Una vez encontrada a cita que queremos, procedemos a abrir la consulta al paciente, presionando en el botón 
- Si deseamos cancelar esa cita, le damos clic en el botón 
- Si queremos enviar un recordatorio al cliente de que debe asistir a consulta presionamos el botón .



Datos del Propietario:			
Nombre del propietario:	Dorian Benjamin Navarrete Chaves	Cédula:	1719458075
Dirección:	La Gonzáles Suárez		
Datos de la mascota tratada:			
Código:	4	Fecha de Nacimiento:	04/04/2015
Fecha de atención:			
Nombre de la(s) mascota(s):	Bianca	Especie:	Canino
Raza:	French Poodle		
Datos Médicos			
Peso (Kg):		Temperatura:	
Frecuencia Cardíaca:		Frecuencia respiratoria:	
Palpación:			
Motivo de Consulta:	Ingrese el motivo por el que asiste		
Síntomas:	Ingrese detalladamente los síntomas del paciente tratado...		
Diagnóstico:	Ingrese el diagnóstico del paciente tratado...		
Tratamiento:	Ingrese el tratamiento que se enviará al paciente tratado...		
Receta:	Ingrese la receta médica que será prescrita para el paciente tratado...		
Observaciones:	Ingrese las observaciones adicionales, de la consulta realizada al paciente tratado...		
Finalizar Consulta		Cancelar	

Figura 94. Pantalla de nueva consulta.

Indicaciones:

- Para registrar una nueva consulta al paciente, es necesario llenar los campos presentados.
- Y para finalizar presionamos el botón Finalizar Consulta.
- Si queremos cancelar el proceso, presionamos en el botón 












LISTADO DE MASCOTAS DE LA VETERINARIA											
<div> Selecione criterio de busqueda <div> Elegir... </div> <div> </div> <div> Buscar </div> </div>											
Ver - Sacar Informe	N°	Nombres propietario	Apellidos	Nombre mascota	Especie	Raza	Peso	Color	Sexo	Edad	F.Nacimiento
 	2	Angélica Janeth	Barreiro Cedeño	Marcus	Canino	Doberman	17,00	Marron	Macho	0 Año(s), 1 mes(es), 0 día(s)	30/08/2015
 	4	Damian Benjamin	Navarrete Chávez	Blanca	Canino	French Poodle	10,00	Blanco	Hembra	0 Año(s), 5 mes(es), 26 día(s)	04/04/2015
 	5	Damian Benjamin	Navarrete Chávez	Josue	Canino	Pitbull	20,00	Gris	Macho	4 Año(s), 7 mes(es), 26 día(s)	04/02/2011
 	10	Angélica Janeth	Barreiro Cedeño	Catrina	Felino	Maine Coon	23,00	Gris	Hembra	3 Año(s), 3 mes(es), 15 día(s)	15/06/2012

Figura 95. Listado de pacientes.

Indicaciones:


- Seleccionamos el criterio de búsqueda haciendo clic en 
- Procedemos a digitar la búsqueda y presionamos el botón buscar
- Presionamos en el ícono  para ver la historia clínica del paciente, y a continuación nos aparecerá el siguiente formulario (Figura 96)

FICHA MÉDICA



CONSULTORIO VETERINARIO HUSKY

DATOS GENERALES

FOTO:	CÓDIGO:	4	NOMBRE:	Blanca
	ESPECIE:	Canino	RAZA:	French Poodle
	PESO (Kg):	10	COLOR:	Blanco
	SEXO:	Hembra	EDAD:	0 Años(x), 5 meses(x), 26 días(x)
	FECHA DE NACIMIENTO: 04/04/2015			

REGISTROS MÉDICOS DE LA PRIMERA CONSULTA

FECHA DE ATENCIÓN:	20/09/2015 16:47:10	PESO:	4,00	TEMPERATURA:	4,00
FRECUENCIA CARDÍACA:	5,00	FRECUENCIA RESPIRATORIA:	4,00	PALPACIÓN:	4,00
Motivo de Consulta:	Enfermedad				
Síntomas:	fiebre, estornudos, mucosa en exceso				
Diagnóstico:	moquillo				
Tratamiento:	antibióticos, vitaminas, suministradas en las próximas 72 horas				
Receta:	sefalexina por cada 24 horas.				
Observaciones:	Cuidar mucho de la mascota, no permitir que tenga interacción con el frío				

REGISTRO DE VACUNAS

N°	N° mascota	Nombre Paciente	Tipo De Vacuna	Fecha de Aplicación	Próxima Fecha	Observaciones
2	4	Blanca	Moquillo	02/10/2015	31/12/2015	Asistir con regularidad para examinar al paciente

REGISTRO DE DESPARASITACIONES

N° Antiparasitario	N° mascota	Nombre Paciente	Tipo De Antiparasitario	Fecha de Aplicación	Próxima Desparasitación	Observaciones
1	4	Blanca	Netecan	06/10/2015	07/01/2016	Ninguna
3	4	Blanca	preventif	11/10/2015	09/01/2016	Es necesario aplicar la próxima desparasitación dentro de 90 días.

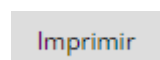
FIRMA DEL MÉDICO VETERINARIO

DATOS:

Imprimir
Volver

Figura 96. Historia clínica del paciente.

- Si deseamos imprimir un reporte de la historia clínica presionamos el botón



- Y nos aparecerá el reporte generado (Figura 97)

Imprimir

Total: 2 páginas

Guardar Cancelar

Destino

Guardar como PDF

Cambiar...

Páginas

Todo

p. ej. 1-5, 8, 11-13

Diseño

Vertical

Tamaño del papel

Letter

Márgenes

Predeterminado

Configuración

Encabezado y pie de página

Gráficos de fondo


19/08/2015

Medicinal 10714_Fichas_medicina/tema/tema10714

FICHA MÉDICA

CONSULTORIO VETERINARIO HUSKY

DATOS GENERALES

FOTO: 

EDAD: 4

SEXO: Macho

RAZA: Husky

FECHA DE NACIMIENTO: 04/04/2015

REGISTROS MÉDICOS DE LA PRIMERA CONSULTA

FECHA DE ATENCIÓN: 19/08/2015

PRESENCIA CLÍNICA: 1,00

PRESENCIA RESPIRATORIA: 4,00

PRESENCIA PALPACIÓN: 4,00

Motivo de Consulta: Enfermedad

Síntomas: Fiebre, letargo, vómitos en exceso

Diagnóstico: infección

Tratamiento: antibióticos, vitaminas, sueros en las próximas 72 horas

Receta: sulfonamida por cada 12 horas

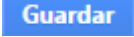
Observaciones: Cuidar mucho de la mascota, no permitir que tenga interacción con el frío

REGISTRO DE VACUNAS

MP	MP	Nombre	Tipo de	Fecha de	Próxima Fecha	Observaciones
vacuna	muerte	Paciente	Vacuna	Aplicación		
2	4	Blanca	Muñeja	02/10/2015	31/10/2015	Asesor con regularidad para evaluar al paciente

107

Figura 96. Historia clínica del paciente.

- Para guardar el documento generado, presionamos el botón 
- Y tendremos nuestro reporte generado en formato PDF.

Anexo A.11 Webgrafía

- 2012, C. d. (2012). *ASPW*. Obtenido de ASPW :
<https://aspw2012.wordpress.com/2012/04/02/tipos-de-datos-en-mysql/>
- 2012, C. d. (s.f.). *ASPW*. Obtenido de
<https://aspw2012.wordpress.com/2012/04/02/tipos-de-datos-en-mysql/>
- Grado, A. d. (s.f.). *Asesoría de Tesis y Trabajos de Grado*. Obtenido de Asesoría de Tesis y Trabajos de Grado:
<http://asesoriatensis1960.blogspot.com/2010/12/antecedentes-de-la-investigacion.html>
- Microsoft. (2015). *Visual Studio*. Obtenido de Visual Studio:
[https://msdn.microsoft.com/es-es/library/vstudio/7eh4aaa5\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/vstudio/7eh4aaa5(v=vs.100).aspx)
- Web, D. (2015). *Definiciones Web*. Obtenido de Definiciones Web:
https://www.google.com.ec/?gfe_rd=cr&ei=pPUfVr6YMeSp8wfmr5vIDg&gws_rd=ssl#q=diccionario+de+datos
- Vásconez, J. C. (Marzo de 2012). *Marketing para desarrolladores de software, PDF 2012*:
<https://www.dspace.espol.edu.ec/bitstream/123456789/14896/1/Sistema%20de%20Registro%20Unicos%20de%20Historias%20Clinicas.pdf>
- Zumba P. (Octubre de 2011). *Ingeniería de software, PDF 2011*:
<https://www.dspace.uazuay.edu.ec/bitstream/datos/2429/1/07897.pdf>
- Wikipedia 2015. (Enero 2015). *Medicina Veterinaria y Zootecnia.*:
<http://es.wikipedia.org/wiki/Veterinaria>