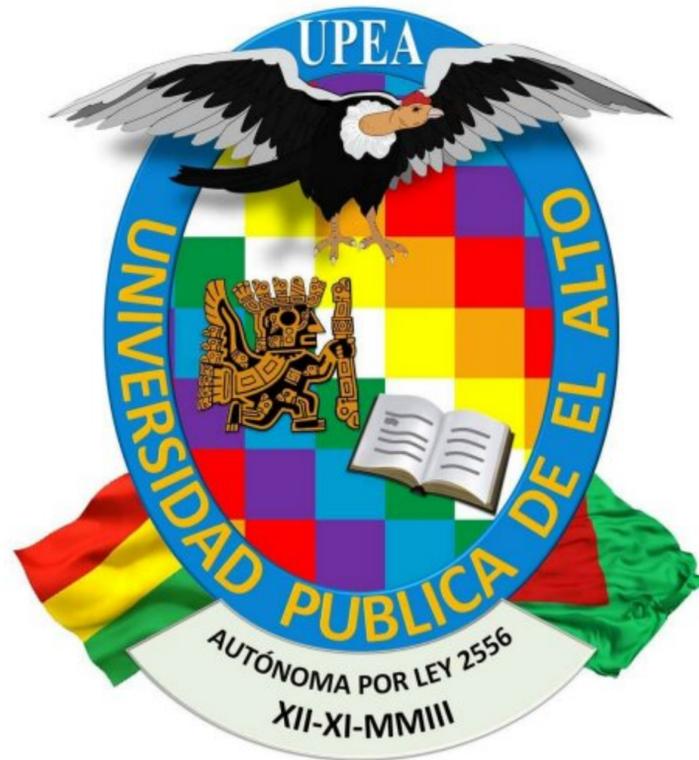


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES”

CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, CARRERA MEDICINA VETERINARIA
Y ZOOTECNIA, CLÍNICA VETERINARIA U.P.E.A.

Para Optar al Título de Licenciatura en Ingeniería de Sistemas
Mención: MENCIÓN INFORMÁTICA Y COMUNICACIONES

Postulante: Dan Cruz Mamani Casas

Tutor Metodológico: Ing. Maricel Yarari Mamani

Tutor Especialista: Ing. Yolanda Escobar Mancilla

Tutor Revisor: Ing. Gabriel Reynaldo Sirpa Huayhua

EL ALTO – BOLIVIA

2021

DEDICATORIA

Dedico este proyecto a:

A Dios que me dio la vida y fortaleza.

A mis padres que siempre me apoyaron y me enseñaron el valor del esfuerzo y el trabajo para lograr el objetivo deseado.

AGRADECIMIENTOS

Agradecer a mis distinguidos tutores:

A mi tutor metodológico Ing. Maricel Yarari Mamani, por brindarme su tiempo, apoyo, motivación, paciencia y conocimiento brindada a mi persona.

A mi tutor especialista Ing. Yolanda Escobar Mancilla, por el apoyo constante, por guiarme con sus conocimiento para realizar este proyecto.

A mi tutor revisor Ing. Gabriel Reynaldo Sirpa Huayhua, por brindarme su tiempo, paciencia y conocimiento brindada a mi persona.

A la Universidad Pública de El Alto, La Carrera Ingeniería de Sistemas por ser como mi segundo hogar en estos años de estudio.

Al Director de la carrera Medicina Veterinaria y Zootecnia al Lic. MVZ R. Efrain Berdeja Ovidio, por la confianza en este proyecto. También al Lic. MVZ Cecilio Coaricona Nina por la predisposición para coordinar con el presente proyecto.

A mis compañeros de la universidad, por la amistad brindada en los años de estudios que llevamos juntos.

ÍNDICE GENERAL

CAPITULO I

MARCO PRELIMINAR	Pág.
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES DEL PROYECTO	2
1.2.1 Antecedentes Institucionales	2
1.2.2 Antecedentes Académicos.....	3
1.3 PLANTEAMIENTO DEL PROBLEMA	5
1.3.1 Problema Principal	5
1.3.2 Problemas Secundarios.....	6
1.4 OBJETIVOS	7
1.4.1 Objetivo General	7
1.4.2 Objetivos Específicos.....	7
1.5 JUSTIFICACIÓN	8
1.5.1 Justificación Técnica	8
1.5.2 Justificación Económica.....	8
1.5.3 Justificación Social.....	9
1.6 METODOLOGÍA.....	9
1.7 HERRAMIENTAS.....	10
1.8 LIMITES Y ALCANCES.....	11
1.8.1 Limites.....	11
1.8.2 Alcances	11
1.9 APORTES	12

CAPITULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN	14
2.2 SISTEMA	14
2.3 WEB	14
2.4 GESTIÓN DE HISTORIAL CLÍNICA	14

2.5 SISTEMA WEB DE GESTIÓN	15
2.6 HERRAMIENTA DE DESARROLLO.....	15
2.6.1 Gestor de Base de Datos	15
2.6.2 Lenguaje de Programación	16
2.6.3 Framework	17
2.6.4 Bootstrap.....	18
2.6.5 Livewire	18
2.6.6 Servidor	19
2.7 INGENIERIA DE SOFTWARE	19
2.7.1 Modelos y Metodologías	20
2.7.2 Métodos de Desarrollo Ágil	23
2.8 METÓDOLOGIA UWE	24
2.8.1 Fases de la Metodología UWE.....	25
2.9 ARQUITECTURA DE SOFTWARE.....	30
2.9.1 Patrón M.V.C.	30
2.9.2 Ciclo de Vida del MVC	31
2.9.3 Ventajas y Desventajas.....	32
2.10 CALIDAD DE SOFTWARE	33
2.10.1 Normas de Calidad ISO 9000	33
2.10.1 Factores de la Calidad ISO 9126	34
2.11 ANÁLISIS DE COSTOS DE SOFTWARE COCOMO	44
2.11.1 COCOMO II.....	45
2.12 SEGURIDAD DEL SISTEMA	49
2.12.1 Norma ISO 27001	49
2.13 PRUEBAS DE ACEPTACIÓN.....	50
CAPITULO III	
MARCO APLICATIVO	
3.1 INTRODUCCIÓN	59
3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL	59
3.3 ANÁLISIS DE REQUERIMIENTO	59

3.3.1 Obtención de Requerimientos.....	59
3.3.2 Identificación de Actores	60
3.3.3 Identificación de Requerimientos	61
3.3.4 Diagrama de Caso de Uso en General.....	63
3.4 MODELO DE CONTENIDO	72
3.4.1 Diagrama de Clases.....	72
3.5 MODELO DE NAVEGACIÓN.....	74
3.6 MODELO DE PRESENTACIÓN:.....	77
3.7 MODELO DE PROCESO.....	80
3.8 IMPLEMENTACIÓN DEL SISTEMA	83
3.8.1 Interfaz de Inicio de Sesión	83
3.8.2 Módulos que Integran el Sistema.....	84
3.9 PRUEBAS DE SOFTWARE.....	96
3.9.1 Pruebas de Caja Blanca.....	96
3.3.4 Pruebas de Caja Negra	98
3.10 CALIDAD.....	101
3.10.1 Normas ISO 9126	101
3.11 SEGURIDAD	109
3.11.1 ISO 27001	109
3.12 COSTOS	110
3.12.1 COCOMO II	110

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES.....	114
4.2 RECOMENDACIONES	115
BIBLIOGRAFÍA	116
ANEXOS.....	138
ANEXO A: Árbol de Problemas	
ANEXO B: Árbol de Objetivos	
ANEXO C: Manual de Usuario	

ANEXO D: Manual Técnico
ANEXO E: Documentación

ÍNDICE DE FIGURAS

CAPITULO I

Figura 1. Organigrama Clínica Veterinaria U.P.E.A.....	3
---	---

CAPITULO II

Figura 2. Modelo Cascada	20
---------------------------------------	----

Figura 3. Modelo en Espiral.....	22
---	----

Figura 4. El Proceso De La Programación Extrema	24
--	----

Figura 5. Modelo de Caso de Uso	26
--	----

Figura 6. Modelo de Contenido	27
--	----

Figura 7. Modelo de Navegación.....	28
--	----

Figura 8. Modelo de Presentación.....	29
--	----

Figura 9. Modelo del Proceso.....	30
--	----

Figura 10. Modelo Vista Controlador	32
--	----

Figura 11. Características ISO/IEC 9126.....	35
---	----

Figura 12. Caja Blanca.....	51
------------------------------------	----

Figura 13. Notación de Grafo de Flujo.....	52
---	----

Figura 14. Bucle Simple	53
--------------------------------------	----

Figura 15. Bucle concatenado	53
---	----

Figura 16. Bucle anidado	54
---------------------------------------	----

Figura 17. Bucle no estructurado	55
---	----

Figura 18. Caja Negra.....	55
-----------------------------------	----

CAPITULO III

Figura 19. Casos de Uso del Sistema	63
--	----

Figura 20. Caso de Uso Ingresar al Sistema Clínico	64
---	----

Figura 21. Caso de Uso: Ingresar al Sistema Publicaciones	65
--	----

Figura 22. Caso de Uso: Registrar Propietario	66
--	----

Figura 23. Caso de Uso: Registrar Mascota	67
--	----

Figura 24. Caso de Uso: Registrar Diagnóstico.....	68
---	----

Figura 25. Caso de Uso: Registrar Tratamiento	69
--	----

Figura 26. Caso de Uso: Publicar Adopción	71
--	----

Figura 27. Diagrama de Clases	72
--	----

Figura 28. Modelo de Datos Relacional.....	73
---	----

Figura 29. Diagrama de Navegación Administrador	74
--	----

Figura 30. Diagrama de Navegación: Usuario.....	75
--	----

Figura 31. Diagrama de navegación: Mostrar Historial	75
Figura 32. Diagrama de navegación: Invitado	76
Figura 33. Diagrama de Navegación: Historial Clínico.....	77
Figura 34. Diagrama de Presentación: Administrador de usuarios	78
Figura 35. Diagrama de Presentación: Usuario	79
Figura 36. Diagrama de Presentación: Mostrar Invitado.....	80
Figura 37. Diagrama de Actividades: Gestión de usuario	81
Figura 38. Diagrama de Actividades: Registro propietario.....	81
Figura 39. Diagrama de Actividades: Gestión Historial	82
Figura 40. Diagrama de Actividades: Gestión de Tratamiento.....	82
Figura 41. Pantalla: Control de Acceso	83
Figura 42. Pantalla de Inicio	84
Figura 43. Pantalla: Lista de Usuarios	85
Figura 44. Formulario: Registro de Usuarios	86
Figura 45. Formulario: Modificar Usuarios.....	87
Figura 46. Formulario: Registro Propietario.....	88
Figura 47. Formulario: Modificar Propietario.....	88
Figura 48. Pantalla: Información de Propietario.....	89
Figura 49. Pantalla: Lista de Mascotas.....	90
Figura 50. Formulario: Modificar Mascota	90
Figura 51. Formulario: Registro de Diagnostico.....	91
Figura 52. Formulario: Registro Terapéutica	91
Figura 53. Pantalla: Historial de Mascota	92
Figura 54. Reporte: Historial Clínico de Mascota.....	93
Figura 55. Formulario de Registro de Publicaciones	94
Figura 56. Interfaz de Publicaciones	95
Figura 57. Pantalla: Estadística de Mascotas.....	96
Figura 58. Caja Blanca.....	97

INDICE DE TABLAS

CAPITULO I

Tabla 1. Requerimientos del equipamiento	8
---	---

CAPITULO II

Tabla 2. Diferencias entre metodologías ágiles y no ágiles	20
---	----

Tabla 3. Dominios de Información de Puntos Función	36
---	----

Tabla 4. Factores de Ponderación	36
---	----

Tabla 5. Valores de Ajuste de la Complejidad	37
---	----

Tabla 6. Métrica de Adecuación	38
---	----

Tabla 7. Métrica de Madurez	39
--	----

Tabla 8. Métrica de Entendibilidad	40
---	----

Tabla 9. Métrica de Comportamiento en el Tiempo	41
--	----

Tabla 10. Métrica de Cambiabilidad.....	43
--	----

Tabla 11. Métrica de Conformidad de Transportabilidad.....	44
---	----

Tabla 12. Constantes de Complejidad	47
--	----

Tabla 13. Variables Factor Ajuste del Esfuerzo	48
---	----

CAPITULO III

Tabla 14. Obtención de Requerimientos	60
--	----

Tabla 15. Identificaciones de actores.....	60
---	----

Tabla 16. Análisis de Requerimientos Funcionales.....	61
--	----

Tabla 17. Análisis de Requerimientos No Funcionales	62
--	----

Tabla 18. Descripción de Caso de Uso: Ingresar al Sistema Clínico.....	64
---	----

Tabla 19. Descripción Caso de Uso: Ingresar al Sistema Publicaciones.....	65
--	----

Tabla 20. Descripción Caso de Uso Registrar Propietario	66
--	----

Tabla 21. Descripción Caso de Uso: Registrar Mascota	67
---	----

Tabla 22. Descripción Caso de Uso: Registrar Diagnóstico	69
---	----

Tabla 23. Descripción Caso de Uso: Registrar Tratamiento.....	70
--	----

Tabla 24. Descripción Caso de Uso: Publicar Adopción	71
---	----

Tabla 25. Caso de Prueba de Ingreso al Sistema.....	99
--	----

Tabla 26. Caso de Prueba Registro de Usuarios	100
--	-----

Tabla 27. Caso de Prueba Registro de Mascota.....	100
--	-----

Tabla 28. Caso de Prueba Registro de Diagnóstico Clínico.....	101
--	-----

Tabla 29. Características de funcionalidad.	102
---	-----

Tabla 30. Parámetros de medición	102
---	-----

Tabla 31. Cálculo de Punto de Función	103
Tabla 32. Valores de Ajuste de Complejidad	103
Tabla 33. Escala de Valoración Para La Evaluación de Uso.....	106
Tabla 34. Cuestionario de Evaluación de Uso	107
Tabla 35. Datos Obtenidos del Sistema.....	108
Tabla 36. Constantes de Complejidad	111
Tabla 37. Ecuaciones del modelo COCOMO II.....	111
Tabla 38. Variables factor ajuste del esfuerzo	111

CAPÍTULO I

MARCO PRELIMINAR

1.1 INTRODUCCIÓN

En la actualidad las instituciones públicas y privadas han comprendido la importancia de implementar un sistema como medio de apoyo a la gestión de historial y control de acceso inmediato a recursos, servicios y precios médicos para la sociedad, en particular con la urbe alteña, tener control sobre los datos y la información de diversos documentos permite mejora en disponibilidad y confiabilidad. Por lo cual se exige que las empresas e instituciones mejoren el uso tecnológico.

La U.P.E.A ¹ cuenta con varias carreras actualmente y cada día va en constante crecimiento tanto a nivel de infraestructura como a nivel académico, el Área de Ciencias Agrícolas Pecuarias y Recursos Naturales cuenta con la carrera de Medicina Veterinaria y Zootecnia, dentro de esta carrera se encuentra la Clínica Veterinaria U.P.E.A., esta se encarga de mantener y gestionar los recursos, con actividades que puedan atender a los animales de diferentes especies en particular a la canina y felina, sin embargo al estar en crecimiento el espacio otorgado se encuentra en una parte de la Universidad, con limitaciones de infraestructura, el manejo de los historiales clínicos se controla de manera manual, cabe recalcar que la Clínica también procede como intermediario entre personas que quieran dar en adopción a sus mascotas y las que quieren adoptar para hacerse cargo de estas. El historial clínico es de constante manejo ya que la comunidad estudiantil de la carrera Medicina Veterinaria y Zootecnia, auxiliares, pasantes y estudiantes los solicitan para sus actividades académicas, lo que hace que los documentos se deterioren e incluso en ocasiones se genera la pérdida de estos documentos.

En la Clínica Veterinaria U.P.E.A. se controla el acceso a la documentación solo al personal autorizado, como en este caso los recursos y servicios, el desarrollo de un sistema web ayudará de gran manera en los registros diarios, permitirá visualizar tratamientos, adopciones, historiales, estadísticas, etc. Así también para aquellas mascotas que están en proceso adopción, también brindará información detallada a los

¹ Universidad Pública de El Alto, ver www.upea.bo

usuarios en general, para el seguimiento de estas.

El presente proyecto de grado, propone solucionar las debilidades en la inexistencia de información respecto a los servicios, ubicación, horarios de atención, etc. La falta de funciones que puedan facilitar las actividades cotidianas en la Clínica, desorganización y falsos anuncios provocan el descontento del usuario, por lo que se desarrollará un portal Web, así mismo el proyecto permitirá al usuario el acceso a toda la información concerniente a la información de los pacientes y sus tratamientos existentes dentro de la Clínica, permitiendo al usuario visualizar los diferentes tipos de archivos, entre ellos reportes, el portal web como plan estratégico para el acceso rápido a recursos y servicios logrando un paso más a la modernización y crecimiento tecnológico en la carrera Medicina Veterinaria y Zootecnia.

1.2 ANTECEDENTES DEL PROYECTO

1.2.1 Antecedentes Institucionales

La Clínica Veterinaria U.P.E.A. dependiente de la carrera Medicina Veterinaria y Zootecnia de la Universidad Pública de El Alto, se encuentra en la Zona Villa Esperanza Avenida Sucre.

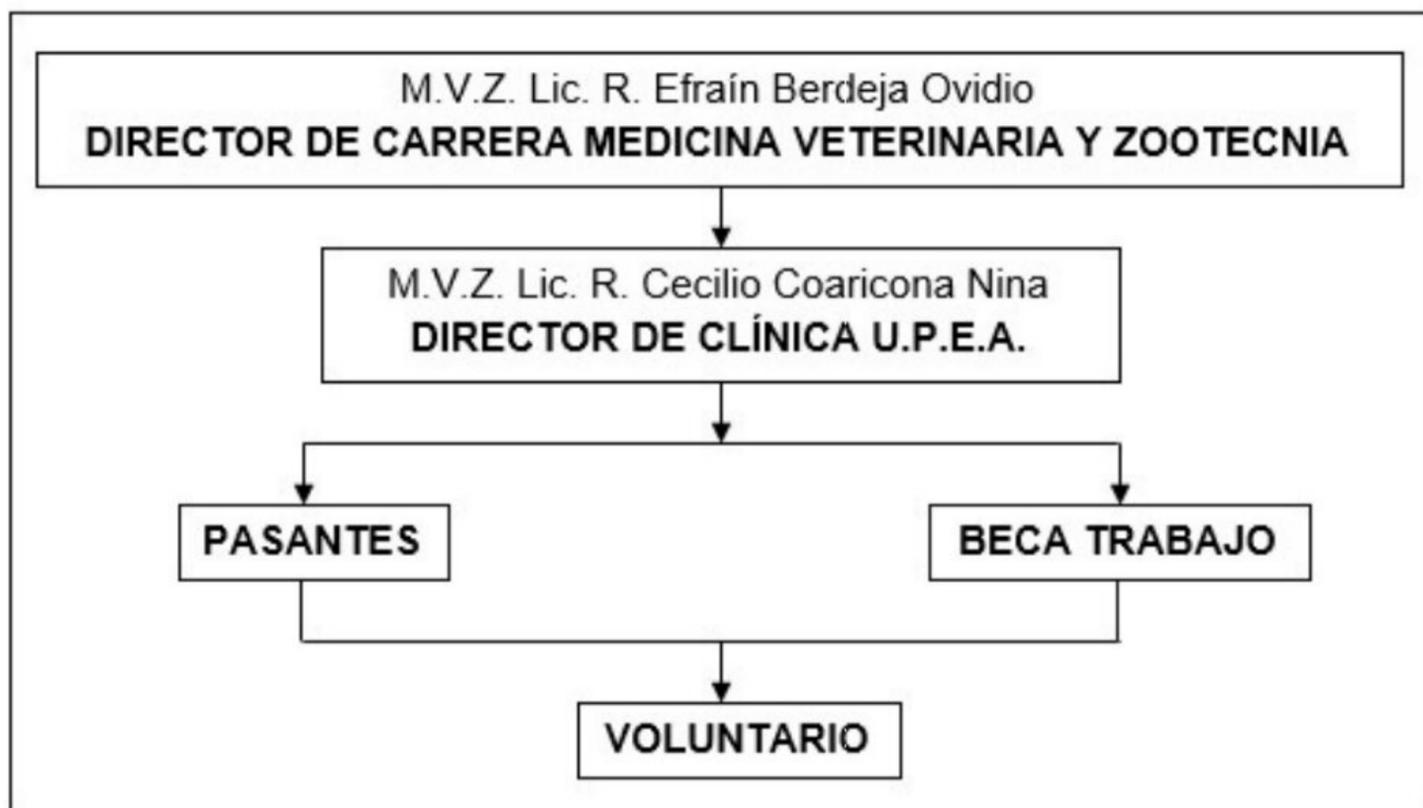
- **Objetivo:** Brindar sus servicios de atención medicas a las mascotas, brindando todo tipo de servicios desde vacunaciones antirrábicas hasta cirugías.
- **Misión:** La Carrera de Medicina Veterinaria y Zootecnia, tiene la noble misión de formar profesionales identificados con la realidad regional, nacional y mundial, capaces de resolver problemas de la pecuaria sostenible.

Con eficiente criterio científico socioeconómico, con alto sentido ético y sensibilidad social, que sustente las actuales políticas globalizantes y de protección del medio ambiente en beneficio de los animales y consecuentemente del hombre.

- **Visión:** La carrera de medicina veterinaria y zootecnia de la U.P.E.A. es una institución académica, social y política, con identidad y esencia propia que rescata la pluriculturalidad y distintos grados de formación cognoscitiva de los productores de animales de interés zootécnico.

La Clínica Veterinaria al servicio de la sociedad en general ofrece sus diferentes servicios como ser: desparasitación, vacuna antirrábica, lavativa general, cirugías en general, etc. A precios económicos comprometidos con la ciudad de El Alto realizando una excelente atención tanto a los propietarios como a las mascotas.

Figura 1. Organigrama Clínica Veterinaria U.P.E.A.



Fuente: Clínica Veterinaria U.P.E.A., 2001

1.2.2 Antecedentes Académicos

Para el presente documento se ha realizado un análisis de trabajos desarrollados sobre este tema, obteniendo información que se relaciona con la investigación.

A continuación, se muestran proyectos realizados parecidos a nuestro trabajo a nivel internacional y nacional las cuales son:

Internacional

- (Vargas, Rodríguez & De la Torre. 2019), “Sistema De Información Para La Gestión Y Control De Citas de la Clínica Veterinaria de la Universidad Cooperativa de Colombia en el Municipio de Arauca”, este sistema fue desarrollado en la Universidad Cooperativa De Colombia Facultad De Ingeniería De Sistemas Programa De Ingeniería De Sistemas Arauca. El proyecto del software se creó con el fin de contribuir a la gestión y control de citas médicas para las mascotas de los clientes de la Universidad Cooperativa de Colombia Programa de Medicina Veterinaria y Zootecnia. Éste facilita el registro y gestión de mascotas, sean felinas, caninas, etc.
- (Alegría, Martínez, Ramos & Gil. 2015), “Sistema Informático para la Gestión y Control de la Clínica Veterinaria de Pequeñas Especies de la Universidad de El Salvador (SIGESCLIVET)”, este sistema fue desarrollado en la Universidad De El Salvador Facultad De Ingeniería Y Arquitectura Escuela De Ingeniería De Sistemas Informáticos, un Sistema Informático para la Gestión y Control de la Clínica Veterinaria de Pequeñas Especies de la Universidad de El Salvador, para brindar mejor atención a los pacientes y sus propietarios.
- (Hernández. 2013), “Sistema de información para el control de expedientes clínicos para médicos veterinarios”, desarrollado en la Universidad Central “Marta Abreu” de Las Villas Facultad de Matemática, Física y Computación Departamento de Ciencia de la Computación, un sistema informático para el trabajo con expedientes clínicos como apoyo la docencia y medio para facilitar el trabajo del médico veterinario mediante el uso del patrón de diseño Modelo Vista Controlador (MVC).

Nacional

- (Condori, Gonzales, Guachalla & Javier. 2016), “Sistema de Gestión para Clínica Veterinaria”, desarrollado en la Universidad Loyola, sistema web para la gestión de clínicas veterinarias, que permite almacenar información por doctores y secretarias, acerca de clientes, pacientes y servicios de manera eficiente y modernizada.

- (Tórrez. 2015), “Sistema Web de gestión de historiales clínicos veterinarios”, desarrollado en la Universidad Mayor de San Andrés, sistema Web de historiales clínicos, basado en la metodología ágil de desarrollo de software XP, que permite realizar el seguimiento y control de los antecedentes clínicos de Animales S.O.S.

1.3 PLANTEAMIENTO DEL PROBLEMA

La Clínica Veterinaria U.P.E.A. perteneciente a la carrera Medicina Veterinaria y Zootecnia de la Universidad Pública de El Alto, dedicada a brindar servicios como desparasitación, vacuna antirrábica, lavativa general, cirugías en general, etc. En la actualidad existen muchos usuarios ya que la Clínica es muy concurrida, que vienen por el buen servicio y precios módicos, realiza su historial de diagnóstico clínico de forma manual, y estos son archivados en carpetas que ocupan mucho espacio, donde además al momento de poder obtener el historial de una mascota este se hace moroso, o al momento de que algún estudiante de la carrera necesite algún historial para realizar sus prácticas, se tiene la amenaza de que el documento sea maltratado o en el peor de los casos sea perdido. Por lo cual la entidad debe ponerse en modernidad, para optimizar el servicio que se brinda a la sociedad en general.

1.3.1 Problema Principal

Luego de realizar un análisis de los problemas, se ha diseñado un árbol de problemas(ver Anexo A), que permitió identificar que no existe una sistematización del movimiento de la información, el cual se lo realiza de forma manual, hace que en la clínica la obtención de información respecto a que si alguna vez la mascota o cliente tenga un registro de que haya visitado la clínica sea morosa e incluso obliga en la mayoría de los casos a llenar una nueva ficha médica, lo cual incurre a duplicidad de datos en los historiales, además no se cuenta con un registro de personas que quieran dejar mascotas en adopción, o un registro de las personas que adoptaron, así no se puede hacer seguimiento al adoptante y no se sabe a qué condiciones fue llevado a la mascota.

La falta de una estrategia de organización establecida en el manejo de la documentación, tiene como consecuencia la deficiente interrelación entre el personal de la Clínica y los clientes, confusión con los registros de los pacientes, duplicidad de datos tanto de la mascota como de los propietarios, todo esto afecta en la eficiencia de los servicios que brinda la clínica.

Por los motivos mencionados anteriormente, se plantea la siguiente pregunta:

¿Es posible generar un portal Web que permita el fácil y rápido acceso a los registros de los pacientes, información de los servicios que ofrece la Clínica Veterinaria U.P.E.A., manteniendo como plan estratégico de promover el uso del portal Web?

1.3.2 Problemas Secundarios

- Falta de disponibilidad de la información del historial clínico de los pacientes, debido a que se encuentra desactualizada.
- No existe una base de datos actualizada ni completa del historial clínico de los pacientes, debido a que no se maneja la información de manera correcta.
- Excesiva documentación, ya que se archivan los documentos en archivadores, los cuales ocupan un espacio considerable en las instalaciones tomando en cuenta el espacio reducido con el que cuenta la clínica.
- Dificultad de seguimiento en los tratamientos, debido a la mala organización en los registros de historial clínico.
- Ningún tipo de seguimiento en las adopciones, no existe registro de usuarios/personas/pacientes que dejan en adopción.
- Falta de información de los servicios, campañas y promociones que solo se muestran a un grupo reducido de personas, ya que no cuenta con una página Web.

1.4 OBJETIVOS

El objetivo del presente proyecto nace como una respuesta a las necesidades de la institución, luego de un análisis de los problemas y su relación causa-efecto, se realizó el diseño del árbol de objetivos (ver anexo B).

1.4.1 Objetivo General

Desarrollar e implementar un Sistema Web de Gestión de Historial Clínico Veterinario y Control de Adopciones, que permita un control eficiente de los procesos de registro, búsqueda y reportes en la Clínica Veterinaria U.P.E.A., para optimizar el nivel de servicio que se brinda a las mascotas.

1.4.2 Objetivos Específicos

- Obtener la información oportuna y en tiempo real del estado en el que se encuentra el historial.
- Diseñar un Sistema Web con una interfaz amigable al usuario, que le permita manipular de manera eficiente la información que se genera en los procesos de la clínica.
- Controlar la actividad de los usuarios del Sistema Web, para evitar incidentes de seguridad, actividades irregulares o problemas operacionales.
- Evitar duplicidad de datos a través de la cédula de identidad del propietario y un código único que se asignará a la mascota, para que de esta forma no se interrumpa los tratamientos.
- Brindar un espacio para que los usuarios/personas/pacientes puedan realizar sus anuncios de adopción, para que otras personas puedan acceder a la información de las mascotas en adopción.
- Proporcionar información de los servicios, ubicación, y horarios de atención de la Clínica Veterinaria para mejorar la interacción entre clientes y la clínica, mediante la página Web.

- Brindar reportes estadísticos para un control de la cantidad de pacientes registrados en la Clínica Veterinaria U.P.E.A.

1.5 JUSTIFICACIÓN

1.5.1 Justificación Técnica

Con el avance de la tecnología en estos últimos tiempos, hoy en día se hace necesario el conocimiento básico del manejo de la computadora, al ya tener la base para manejar distintos paquetes básicos en las computadoras, con la capacitación adecuada y el manual del software, el usuario no tendrá problemas al momento de usar el software que se desarrollará.

La Clínica Veterinaria U.P.E.A. cuenta con los siguientes componentes con un valor aproximado detallado.

Tabla 1. Requerimientos del equipamiento

Nombre	Especificaciones	Marca
Monitor	19" ,Resolucion 1366x768 Pixeles	Samsung
Cpu	Pentium R Dual Core @ 2.2ghz	Intel
Ram	2 Gb	
Disco Duro	Sata, Capacidad 1tb	Hitachi
Teclado	Multimedia	Delux
Mouse	Óptico	Delux
Impresora	Epson L380, Sistema Continuo	Epson

Fuente: Elaboración Propia

1.5.2 Justificación Económica

La clínica al ser muy concurrida por su buen servicio y precios económicos, genera recursos propios, además al ser parte de la carrera de Medicina Veterinaria y Zootecnia cuenta con recursos externos.

La realización de este proyecto se justifica económicamente, en el material de escritorio, en este caso las hojas de papel y carpetas en donde se procedía a guardar la información manualmente, con esto se garantiza una atención adecuada y rápida en el momento de recepción de un paciente lo cual permitirá que los recursos aumenten en beneficio y crecimiento de la Clínica, además al usarse software libre² hace que solo se tenga que preocupar por el pago del dominio y el servidor.

1.5.3 Justificación Social

La ciudad de El Alto al ser una ciudad en progreso y pobre en su mayoría, necesita de una clínica, para que sus mascotas sean atendidas y que los precios estén al alcance de la población, por lo que se hace muy importante el crecimiento de la clínica para la atención de la ciudad. El presente proyecto beneficiará al personal de la Clínica Veterinaria U.P.E.A. facilitando el trabajo en el manejo y distribución del Historial Clínico.

Director: podrá controlar los registros del personal a su cargo, historial clínico de las mascotas y sus tratamientos.

Estudiantes Medicina Veterinaria y Zootecnia: pasantes, beca trabajo, voluntarios y estudiantes gracias al sistema Web podrán acceder al historial clínico de los pacientes, de esta manera se beneficiaran en su desarrollo académico.

Urbe Alteña: beneficiara aproximadamente a 400 personas en promedio que acuden a esta Clínica Veterinaria por mes, esto obtenido de registro de la clínica.

1.6 METODOLOGÍA

➤ Método de Ingeniería

La metodología de Ingeniería de software, que se aplicará para el desarrollo del proyecto será UWE, "UWE es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML³ para sus modelos y sus métodos, lo que facilita la

² Software Libre, programa gratuito que se puedes modificar y utilizar libremente.

³ Unified Modeling Language (Lenguaje Unificado de Modelado)

transición” (Guerrero, Ucán, & Menéndes, 2014), las fases que comprende esta metodología son:

- Análisis de Requerimientos
- Diseño del Sistema
- Codificación del Software
- Pruebas
- La Instalación o Fase de Implementación
- El Mantenimiento

1.7 HERRAMIENTAS

Las herramientas para el desarrollo del sistema serán los siguientes:

➤ **Hypertext PreProcesor “PHP”**

“PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML” (Gallegos, 2003).

➤ **MySQL**

“MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation” (Gallegos, 2003).

➤ **Laravel**

“Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP, basado en el modelo vista controlador, intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP” (Morales, 2019).

➤ **Métricas de Calidad ISO-9126**

Para el presente proyecto se utilizará el estándar de calidad ISO 9126, “hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se

considere de calidad. El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad” (Figueroa, 2012).

➤ **Costos COCOMO II**

“Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto” (Gómez, López, Migani, & Otazú, COCOMO Un Modelo de Estimación de Costos, 2010).

➤ **Seguridad ISO-27000**

“La serie de normas internacionales ISO/IEC 27000 ofrecen una serie de recomendaciones de mejores prácticas, para la gestión de la seguridad de la información, y esta puede ser aplicada en cualquier organización sin importar el tamaño que tenga” (Regina Baena, Mendoza Mendez, & Dorantes, 2019).

1.8 LÍMITES Y ALCANCES

1.8.1 Limites

La creación de un sistema tecnológico, que permita registrar los datos de los antecedentes médicos de las mascotas y todo aquello vinculado con la salud de los animales que permitirá al médico veterinario, visualizar el historial clínico de la mascota y hacer un seguimiento a su proceso evolutivo, tratamiento y recuperación y dar una mejora en el tiempo debido a que tendrá toda la información y a la vez reducir la movilidad de cada médico para acceder a la información.

1.8.2 Alcances

El departamento de archivo se enfocará en la problemática de la historia clínica en el cual se observa que existe la falta de cuidado en el manejo de la información médica y la falta de control a las adopciones, de esta manera se propone que logre lo siguiente:

- A) Módulo de Administración Del Sistema
 - Gestión de Usuarios
 - Gestión de Roles y Permisos
- B) Módulo de Administración de Propietarios
 - Gestión propietarios
 - Búsqueda de acuerdo al campo que se requiera
- C) Módulo de Gestión de Mascotas
 - Adición mascotas
 - Asignación de un código único de registro
 - Consultas de acuerdo al campo que se requiera
- D) Módulo de Administración de Historial Clínico
 - Asignación de un código único de registro
- E) Módulo de Administración de Adopciones
 - Gestión adopciones
 - Búsqueda de acuerdo al campo que se requiera
 - Publicación actualizada en la página Web
- F) Módulo de Reportes
 - Reportes de personal, propietarios, mascotas, historial clínico y tratamientos
 - Reportes Estadísticos

1.9 APORTES

1.9.1 Aportes

Este proyecto sin duda presenta un gran aporte, desde el aporte social hasta el aporte académico

- Con la implementación del Sistema Web de Gestión de Historial Clínico Veterinario y Control de Adopciones, se podrá ver reducida la carga laboral en procedimientos de la Clínica y presenta una mejora en la atención de los clientes.

- Con el control de adopciones, las personas que deseen dejar en adopción a una o más mascotas por distintos motivos, pueden poner avisos de adopción, de esta forma informar a las personas que quieran adoptar a alguna mascota.
- Los estudiantes que deseen la información clínica, para su desarrollo académico pueden acceder a una copia de este mediante el software.

CAPÍTULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN

El marco teórico que fundamenta este proyecto tiene como objetivo proporcionar una idea más clara respecto a todas las herramientas que se usarán en el proceso del desarrollo del proyecto. Se encontrarán los conceptos muy básicos, los complementarios y específicos.

2.2 SISTEMA

Un sistema es módulo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí. El concepto se utiliza tanto para definir a un conjunto de conceptos como a objetos reales dotados de organización también puede mencionarse la noción del informático, muy común en las sociedades modernas. Este tipo de sistemas denominan al conjunto de hardware, software y soporte humano que forman parte de una empresa u organización. Incluyen ordenadores con los programas necesarios para procesar datos y las personas encargadas de su manejo. (Roja, 2018)

2.3 WEB

La World Wide Web o simplemente WWW o Web es uno de los métodos más importantes de comunicación que existe en Internet. Consiste en un sistema de información basado en Hipertexto (texto que contiene enlaces a otras secciones del documento o a otros documentos). La información reside en forma de páginas Web en ordenadores que se denominan servidores Web y que forman los nodos de esta telaraña. Se denomina páginas Web a documentos que contienen elementos multimedia (imágenes, texto, audio, vídeo y otros) además de enlaces de hipertexto. (EUATM, 1992)

2.4 GESTIÓN DE HISTORIAL CLÍNICA

¿Qué es gestión? Como posible respuesta decimos que es interactuar en todas las áreas de una empresa, organización, unidades, instituciones informativas y otros, entendemos además a toda “actividad dirigida a obtener y asignar los recursos necesarios para el cumplimiento de los objetivos de la organización”. Implica también observar – evaluar sobre las metas alcanzadas con los recursos aplicados.

Entendemos gestión: a la herramienta que engloba fases de la administración siendo la gestión misma un elemento con desarrollo propio de la administración, utilizando nuestra unidad de la información la metodología de gestión, como elemento diferencial para obtener recursos de distintas naturalezas. Solo mencionamos las partes que constituyen la gestión y la posibilidad que nos da este accionar al aplicar a cualquier metodología de trabajo. (Santillan Aldan, 2002, pág. 2)

La gestión de historial, dentro de una clínica es muy importante, dentro del marco de que se encuentran todos los datos de los pacientes, tratamientos, etc. Por lo cual el manejo debe ser eficiente.

2.5 SISTEMA WEB DE GESTIÓN

Un sistema web de gestión es una herramienta que permite controlar, planificar, organizar y automatizar las tareas administrativas de una organización. Un sistema de gestión analiza los rendimientos y los riesgos de una empresa, con el fin de otorgar un ambiente laboral más eficiente y sostenible.

Algunas empresas o PyMEs⁴ cuentan con actividades que no están automatizadas, que con frecuencia se soportan en sistemas departamentales y casi siempre en hojas Excel desarrolladas individualmente por los usuarios implicados en cada una de las funciones. Un software de gestión unifica la operación de todas las áreas del negocio para alinearlas con los objetivos de la empresa. Es importante que cada área tenga claro y definido tanto su objetivo como sus metas. Esto da la garantía de direccionar tu negocio más fácil y eficazmente.

2.6 HERRAMIENTAS DE DESARROLLO

2.6.1 Gestor de Base de Datos

MySQL, el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso

⁴ Empresa pequeña o mediana en cuanto a volumen de ingresos, valor del patrimonio y número de trabajadores.

modelo de negocio.

MySQL es un sistema de gestión de bases de datos. Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compra a una galería de pintura o las más vastas cantidades de información en una red corporativa. Para añadir, acceder, y procesar los datos almacenados en una base de datos, necesita un sistema de gestión de base de datos como MySQL Server. Al ser los computadores muy buenos en tratar grandes cantidades de datos, los sistemas de gestión de bases de datos juegan un papel central en computación, como aplicaciones autónomas o como parte de otras aplicaciones. (Vespito, 2011, pág. 4).

La Ingeniería de Software aplicada en este proyecto es muy importante para el desarrollo del software, este utiliza normas criterios y normas que se aplican al presente proyecto.

La ingeniería de software incluye procesos, métodos y herramientas que permitan elaborar a tiempo y con calidad sistemas complejos basados en computadoras. El proceso de software incorpora cinco actividades estructurales: comunicación, planeación, modelado, construcción y despliegue que son aplicables a todos los proyectos del software. La práctica de la ingeniería de software es una actividad para resolver problemas, que sigue un conjunto de principios fundamentales. (Pressman, 2010, pág. 21)

2.6.2 Lenguaje de Programación

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML.

Comparado con ASP⁵, la principal ventaja de PHP es su carácter multiplataforma. Por otro lado, los programas en ASP resultan más lentos y pesados, y también menos estables. En los entornos Microsoft la ventaja de ASP es que los servidores web de Microsoft soportan directamente ASP sin necesidad de ninguna instalación adicional

Comparando el lenguaje PHP con el lenguaje Perl⁶, utilizado habitualmente en la programación CGI, puede decirse que PHP fue diseñado para desarrollo de scripts orientados a web, mientras que Perl fue diseñado para hacer muchas más cosas y debido a esto, se hace muy complicado. La sintaxis de PHP es menos confusa y más estricta, pero sin perder la flexibilidad. (Cobo, Gómez, Pérez, & Rocha, 2005, pág. 36)

2.6.3 Framework

Laravel es un framework para aplicaciones web con sintaxis expresiva y elegante. Creemos que el desarrollo debe ser una experiencia agradable y creativa para que sea verdaderamente enriquecedora. Laravel busca eliminar el sufrimiento del desarrollo facilitando las tareas comunes utilizadas en la mayoría de los proyectos web, como la autenticación, enrutamiento, sesiones y almacenamiento en caché.

Laravel es un framework para el lenguaje de programación PHP. Aunque PHP es conocido por tener una sintaxis poco deseable, es fácil de usar, fácil de desplegar y se le puede encontrar en muchos de los sitios web modernos que usas día a día. No solo ofrece atajos útiles, herramientas y componentes para ayudarte a conseguir el éxito en tus proyectos basados en web, sino que también intenta arreglar alguna de las flaquezas de PHP.

Laravel tiene una sintaxis bonita, semántica y creativa, que le permite destacar entre la gran cantidad de frameworks disponibles para el lenguaje. Hace que PHP sea un placer, sin sacrificar potencia y eficiencia. Es sencillo de entender, permite mucho la modularidad de código lo cual es bueno en la reutilización de código. (Desconocido, 2015, pág. 3)

⁵ Active Server Pages, lenguaje de scripting del lado del servidor creado por Microsoft.

⁶ Perl, es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto

2.6.4 Bootstrap

Como ya hemos comentado antes, Bootstrap es uno de los frameworks más populares y utilizados del mercado para la creación de páginas responsive, habiendo sido desarrollado por el equipo de Twitter. Entre los navegadores soportados se encuentran Chrome, Firefox, Opera, Safari e Internet Explorer a partir de la versión 8 (aunque en la versión 7 también funciona correctamente).

Está preparado para funcionar tanto en navegadores de PCs y portátiles con cualquier tamaño de pantalla así como para tablets y smartphones de tamaños mucho más reducidos. (Javier, 2018, pág. 8)

2.6.5 Tailwind

Tailwind CSS es una potente herramienta para el desarrollo frontend. Está dentro de la clasificación de los frameworks CSS o también llamados frameworks de diseño. Permite a los desarrolladores y diseñadores aplicar estilos a los sitios web de una manera ágil y optimizada.

Las hermosas interfaces de usuario personalizadas se pueden construir de manera efectiva usando CSS sin realmente poner mucho esfuerzo de codificación. Al ser un framework de primeras utilidades, Tailwind CSS ofrece la ventaja de diseñar cada componente de manera distintiva, de la manera que deseese.

Los sitios web cuidadosamente diseñados seguramente serán atractivos a la vista, pero el tamaño de la aplicación también es de vital importancia. Cuanto más grande sea la aplicación, más énfasis se debe poner en los estilos, lo que crea mucho caos y confusión en las últimas etapas de desarrollo.

2.6.6 Livewire

Livewire es un framework fullstack para el desarrollo de componentes Laravel que pueden comunicarse automáticamente entre la vista y el controlador, de modo que se produzcan comportamientos dinámicos sin usar Javascript.

Livewire es un sistema para desarrollo de componentes dinámicos basados en PHP y con vistas en Blade, que son capaces de reaccionar reactivamente en el lado del cliente con Ajax⁷ a cambios en los datos.

2.6.7 Servidor

El servidor web Apache es un servidor Web gratuito desarrollado por el Apache Server Project (Proyecto Servidor Apache) cuyo objetivo es la creación de un servidor web fiable, eficiente y fácilmente extensible con código fuente abierto gratuito. Este proyecto es conjuntamente manejado por un grupo de voluntarios localizados alrededor del mundo que a través de Internet planean y desarrollan el servidor y la documentación relacionada con éste. Estos voluntarios son conocidos como el grupo Apache.

El servidor web Apache es simplemente una máquina que ejecuta el programa llamado daemon http, httpd. Al igual que otros demonios de red, httpd recibe peticiones de un cliente web, Netscape por ejemplo, y envía el recurso solicitado. Para realizar las configuraciones de red, el servidor http Apache debe iniciarse con permisos de root. Específicamente, necesita enlazarse al puerto 80 para escuchar peticiones y aceptar conexiones. Una vez hecho esto, Apache abandona todos sus derechos y se ejecuta como un usuario distinto de root, como se especifique en sus archivos de configuración. El usuario predeterminado es Apache, que pertenece al grupo Apache. (Sampedro , Marques Díaz, & Vargas, 2002, pág. 11)

2.7 INGENIERIA SOFTWARE

El modelo de desarrollo de software establece el orden en el que realizarán las actividades, establece los requisitos de entrada y salida para cada una de las actividades.

Es necesario destacar el ciclo de vida del proyecto y el modelo de desarrollo, nos ayuda a controlar las actividades del proyecto desde el inicio hasta el fin del mismo. El modelo de desarrollo nos ayuda a la forma en la que vamos a construir el producto, tanto el modelo de desarrollo y el ciclo de vida se complementan para generar el producto

⁷ Ajax (Asynchronous Javascript and XML), Navegador Web de Forma Asincrona

desde el punto de vista técnico y administrativo.

Tabla 2. Diferencias entre metodologías ágiles y no ágiles

M. Tradicional	M. Ágiles
Waterfall (cascada)	Kanban
Prototipado	Scrum
Espiral	Lean
Incremental	Programación extrema (XP)
Diseño Rápido de Aplicaciones (RAD)	

Fuente: Elaboración propia

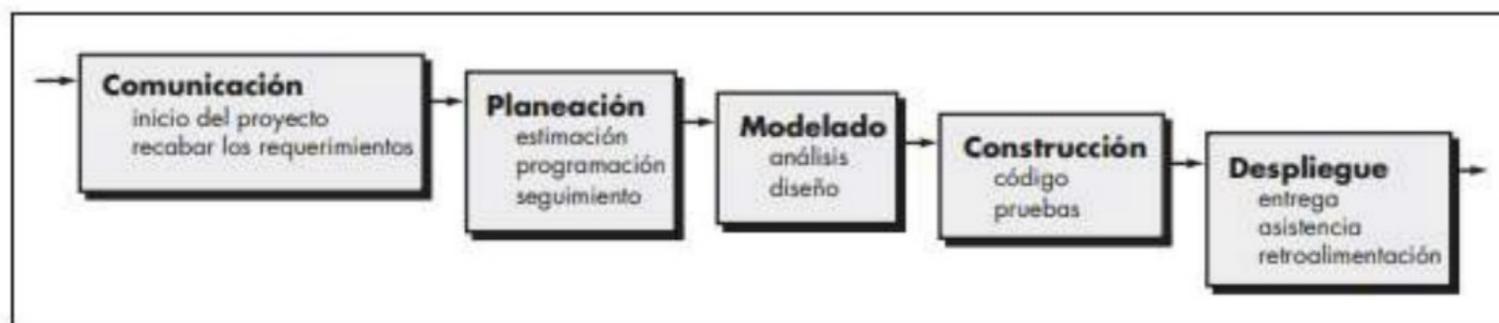
2.7.1 Modelos y Metodologías

Los distintos modelos y metodologías para el desarrollo de software son:

2.7.1.1 Modelo en Cascada

Este es el modelo en el cual se ordenan rigurosamente las etapas del desarrollo del software, de esto se obtiene que el inicio de una etapa de desarrollo deba de esperar el fin de la etapa anterior. De esto se obtiene que cualquier error detectado lleve al rediseño del área de código afectado, lo cual aumenta de costo el desarrollo del proyecto.

Figura 2. Modelo Cascada



Fuente: Pressman, (2010)

2.7.1.2 Modelo en Prototipos

Pertenece a los modelos evolutivos, en el cual el prototipo debe ser construido

rápidamente y con la utilización escasa de recursos. El prototipo es construido para mostrárselo al cliente, obtener críticas y retroalimentación, con lo cual se obtendrán los requisitos específicos para la aplicación a partir de las metas graficas que son mostradas. Las etapas de este modelo son:

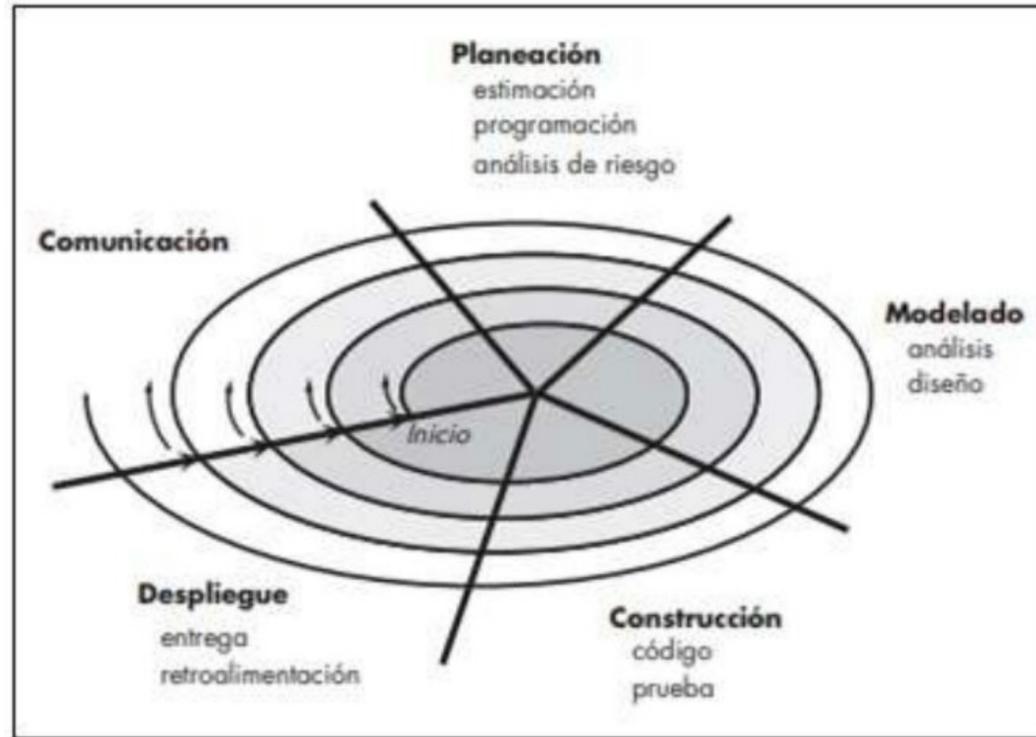
- Plan rápido
- Modelo
- Construcción del prototipo
- Entrega y retroalimentación
- Comunicación
- Entrega del desarrollo final

Entre sus ventajas se encuentra que, es apto para el cliente que conoce a grandes rasgos el objetivo del software y a su vez, al equipo de desarrollo le ofrece una mejor visibilidad de la interacción del cliente con el software y el ambiente en el cual debe de realizarse.

2.7.1.3 Modelo en Espiral

Es el modelo en el cual las actividades se desarrollan en espiral, estas actividades se realizan conforme se van seleccionando de acuerdo al análisis de riesgo. En cada iteración en este modelo, se deberán de tomar en cuenta los objetivos, las alternativas que se deberán de tomar de acuerdo a las características, estas son experiencia personal, requisitos a cumplir, las formas de gestión del sistema, entre otros. Este modelo tiene dos formas en las cuales se debe de planificar el proyecto, la forma angular, la cual indica únicamente el avance del software dentro del proyecto y la forma radial, la cual indica el aumento del costo dado que cada iteración conlleva más tiempo de desarrollo.

Figura 3. Modelo en Espiral



Fuente: Pressman, (2010)

2.7.1.4 Desarrollo Iterativo y Creciente

Resuelve los problemas encontrados en el modelo en cascada, en cual ofrece entornos de trabajo con técnicas para su correcta utilización. Este tipo de modelo es esencial para el método de programación extrema. Este tipo de programación consiste en la realización de programas de manera incremental, la cual sirve para obtener ventaja de lo que se ha realizado a lo largo del proyecto. En este se entran a varias iteraciones con las cuales se obtendrá el software final y a su vez, se le agregaran nuevas funcionalidades a cada etapa. Se puede dividir en los siguientes procesos:

➤ **Etapa de inicialización**

La meta de esta etapa es la construcción de un producto en el cual se pueda obtener retroalimentación de parte del usuario final.

➤ **Etapa de iteración**

Consiste en el análisis, rediseño e implementación del producto de las iteraciones anteriores.

➤ **Lista de control del proyecto**

Son las tareas que se crean que describen las partes que conforman el proyecto, son implementadas y rediseñadas en cada etapa de iteración del producto.

2.7.2 Métodos de Desarrollo Ágil

Para Jacobson (2002), citado por Pressman (2010, pág. 256), sostiene que:

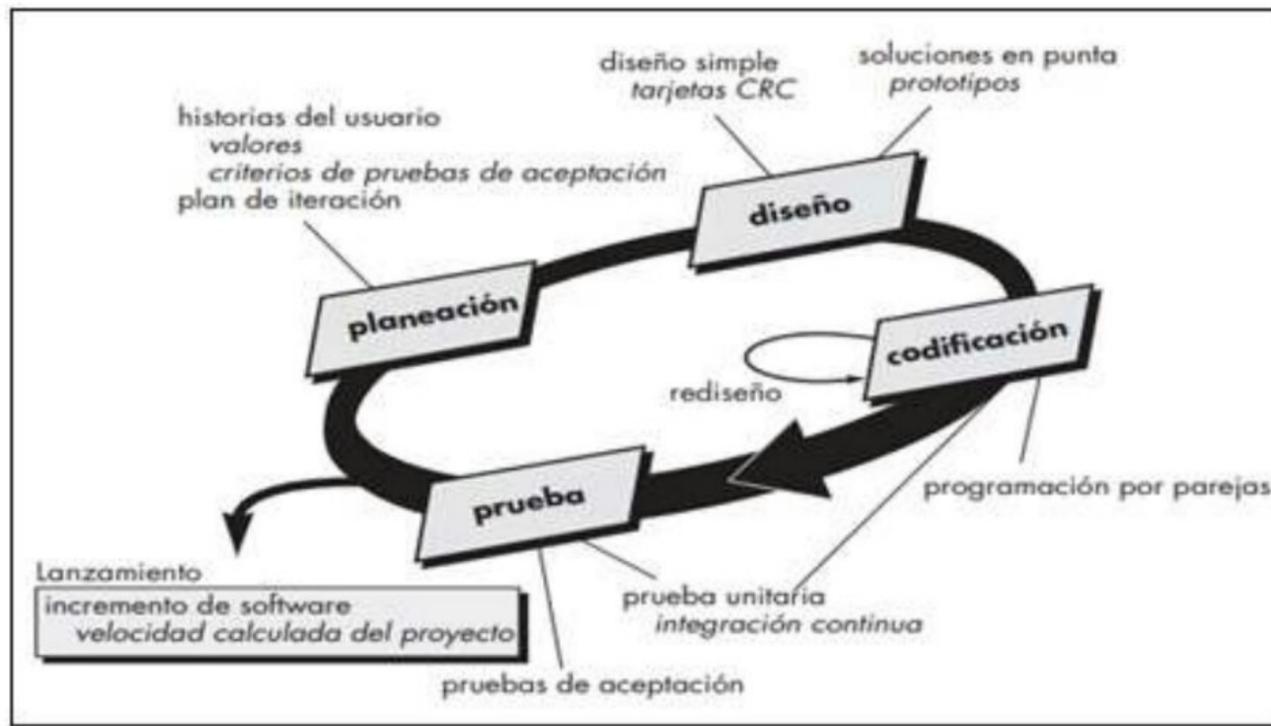
La agilidad se ha convertido en la palabra mágica de hoy para describir un proceso del software moderno. Todos son ágiles. Un equipo ágil es diestro y capaz de responder de manera apropiada a los cambios. El cambio es de lo que trata el software en gran medida. Hay cambios en el software que se construye, en los miembros del equipo, debidos a las nuevas tecnologías, de todas clases y que tienen un efecto en el producto que se elabora o en el proyecto que lo crea. Deben introducirse apoyos para el cambio en todo lo que se haga en el software; en ocasiones se hace porque es el alma y corazón de éste. Un equipo ágil reconoce que el software es desarrollado por individuos que trabajan en equipo, y que su capacidad, su habilidad para colaborar, es el fundamento para el éxito del proyecto.

Sin embargo por definición, las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

2.7.2.1 XP (Programación Extrema)

La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas.

Figura 4. El Proceso De La Programación Extrema



Fuente: Pressman, Ingeniería de Software (2010)

2.7.2.2 SCRUM (Proceso Unificado Relacional)

Scrum (nombre que proviene de cierta jugada que tiene lugar durante un partido de rugby), se caracteriza por ser la metodología del caos, que se basa en una estructura de desarrollo incremental, esto es, cualquier ciclo de desarrollo del producto o servicio se desgrana en pequeños proyectos divididos en distintas etapas: análisis, desarrollo y testing. En la etapa de desarrollo encontramos lo que se conoce como interacciones del proceso o Sprint, es decir, entregas regulares y parciales del producto final.

También podríamos mencionar otras metodologías de desarrollo ágil.

2.8 METODOLOGÍA UWE

La Ingeniería Web ha sido definida como una disciplina emergente que promueve el empleo de enfoques sistemáticos, disciplinados y cuantificables, para lograr el desarrollo eficiente de sistemas y aplicaciones Web con atributos de alta calidad. En particular, dicha disciplina científico-tecnológica se enfoca en la proposición, estudio sistemático, experimentación y mejora continua de metodologías, técnicas y herramientas que constituyan el soporte esencial del desarrollo de aplicaciones Web a lo largo de todo su ciclo de vida. De esta manera, abarca procesos de ingeniería de requisitos, diseño arquitectónico y detallado, construcción, evaluación y evolución,

que tienen en cuenta características y aspectos que diferencian a las aplicaciones Web de otros tipos de sistemas de información, software o aplicaciones tradicionales. (Rotta, Pallota, Klikailo, & Belloni, 2016, pág. 254)

Sin embargo, el desarrollo de una aplicación Web incluye elementos que no son comunes a una aplicación de escritorio. Esto requiere cambios importantes en la forma de realizar y controlar el proceso de desarrollo. Es decir, pasar de una Ingeniería de Software a una Ingeniería Web. Una de las primeras metodologías desarrolladas fue la Ingeniería Web basada en UML. UWE es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Language) para sus modelos y sus métodos, lo que facilita la transición. La metodología define claramente la construcción de cada uno de los elementos del modelo. (Nieves Guerrero, Ucán Pech, & Menéndes Domínguez, 2014, pág. 137)

El enfoque metodológico UML-Based Web Engineering (UWE) está destinado a proveer soporte en el desarrollo de aplicaciones y sistemas Web con especial foco en la sistematización, personalización y la generación semi automática de código. Es un enfoque Orientado a Objetos, iterativo e incremental basado tanto en el Unified Modeling Language (UML) como en el Unified Software Development Process (UP). UWE utiliza notación y diagramas UML para el análisis y diseño de aplicaciones Web. Para características Web específicas como nodos y links externos, el perfil de UWE incluye estereotipos, tags y restricciones definidas por los elementos del modelado. UWE cubre aspectos de navegación, presentación, procesos de negocios, así como también, aspectos de adaptación. (Rotta, Pallota, Klikailo, & Belloni, 2016, pág. 254)

2.8.1 Fases de la Metodología UWE

En su implementación se deben contemplar las siguientes etapas y modelos:

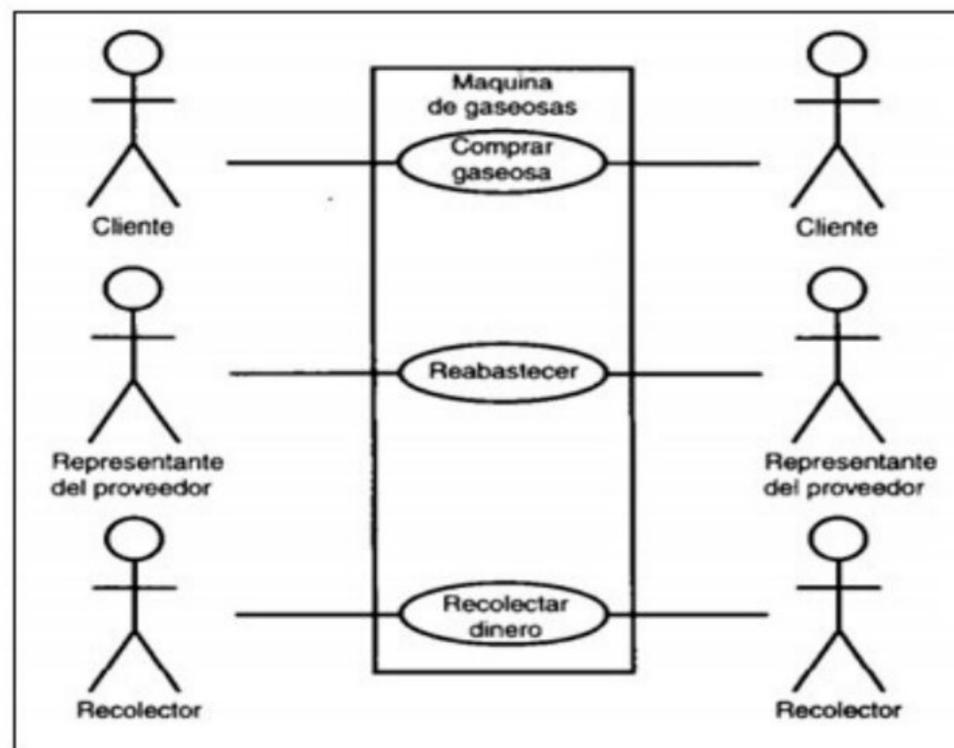
- **Análisis de requisitos.** Plasma los requisitos funcionales de la aplicación Web mediante un modelo de casos de uso.

- **Modelo de contenido.** Define, mediante un diagrama de clases, los conceptos a detalle involucrados en la aplicación.
- **Modelo de navegación.** Representa la navegación de los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas.
- **Modelo de presentación.** Representa las interfaces de usuario por medio de vistas abstractas.
- **Modelo de proceso.** Representa el aspecto que tienen las actividades que se conectan con cada clase de proceso.

2.8.1.1 Análisis de requisitos

Utiliza diagramas de casos de uso de UML. Los estereotipos principales propuestos por el profile de UWE son: <<navigation>> para tareas de navegación y <<webProcess>> para otras tareas. Además, se puede extender este modelo con la realización de Diagramas de Actividades para aportar mayor comprensión acerca del sistema a desarrollar. En la figura 5 siguiente se presenta un diagrama de casos de uso sobre el caso de estudio que pretende especificar el comportamiento del sistema en su interacción con los usuarios mostrando la relación entre los actores y los casos de uso del sistema, ambos caracterizados con estereotipos definidos por UWE.

Figura 5. Modelo de Caso de Uso



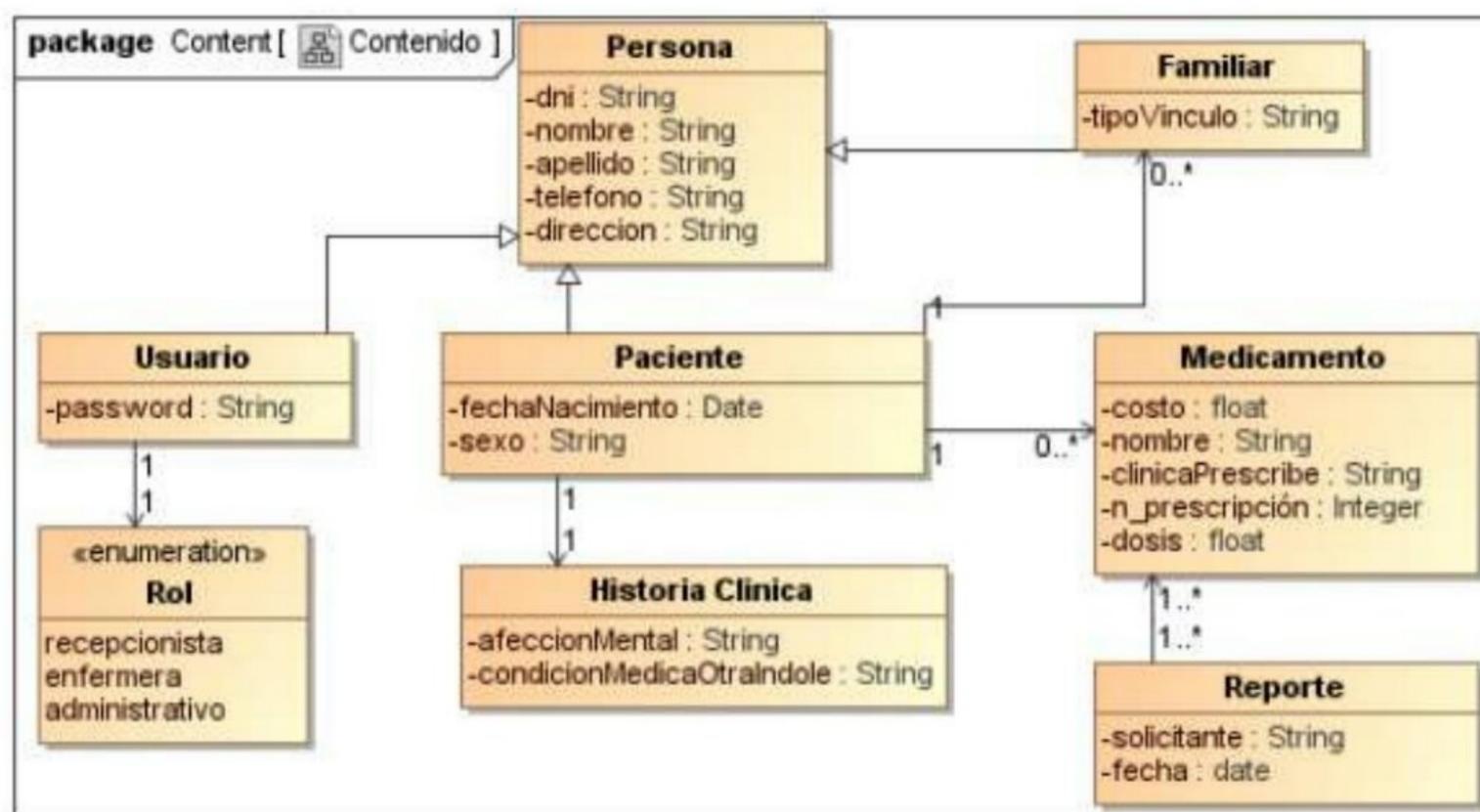
Fuente: Pressman (2010)

2.8.1.2 Modelo de Contenido

En él se representa la información del dominio, sus datos persistentes, mediante un diagrama de clases UML. En la figura 6 se presenta el modelo de contenido del MHC-PMS mediante un diagrama de clases de UML, donde se puede observar las distintas clases que forman parte del sistema con sus respectivos atributos y estereotipos definido por UWE.

El modelo de contenido es el modelo conceptual del dominio de aplicación tomando en cuenta los requerimientos especificados en los casos de uso y se representa con un diagrama de clases.

Figura 6. Modelo de Contenido



Fuente: Universidad Gastón Dachary (2016)

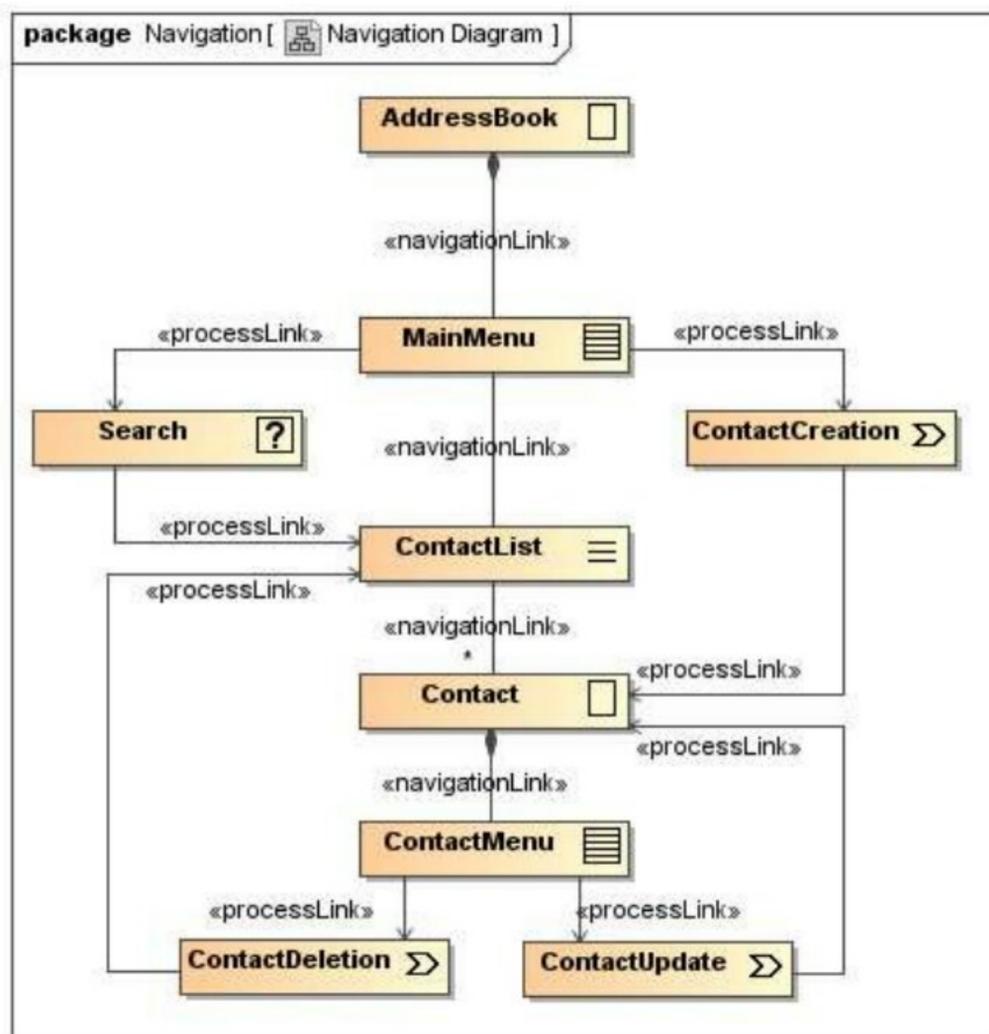
2.8.1.3 Modelo de Navegación

Tiene como objetivo la representación de nodos y enlaces de la estructura de hipertexto, y el diseño de las rutas de navegación mediante diagramas de clases UML. En la figura 7 se presenta el modelo de navegación del MHC-PMS, donde los distintos nodos, enlaces de la estructura de hipertexto, el diseño de las rutas de navegación y la

relación que existe entre los distintos nodos del sistema, se encuentran caracterizados por estereotipos definidos por UWE.

Dichos nodos se encuentran agrupados en paquetes para poder clasificarlos según al tipo de gestión del sistema al que pertenece, el cual puede ser de paciente o de usuarios.

Figura 7. Modelo de Navegación



Fuente: Universidad Gastón Dachary (2016)

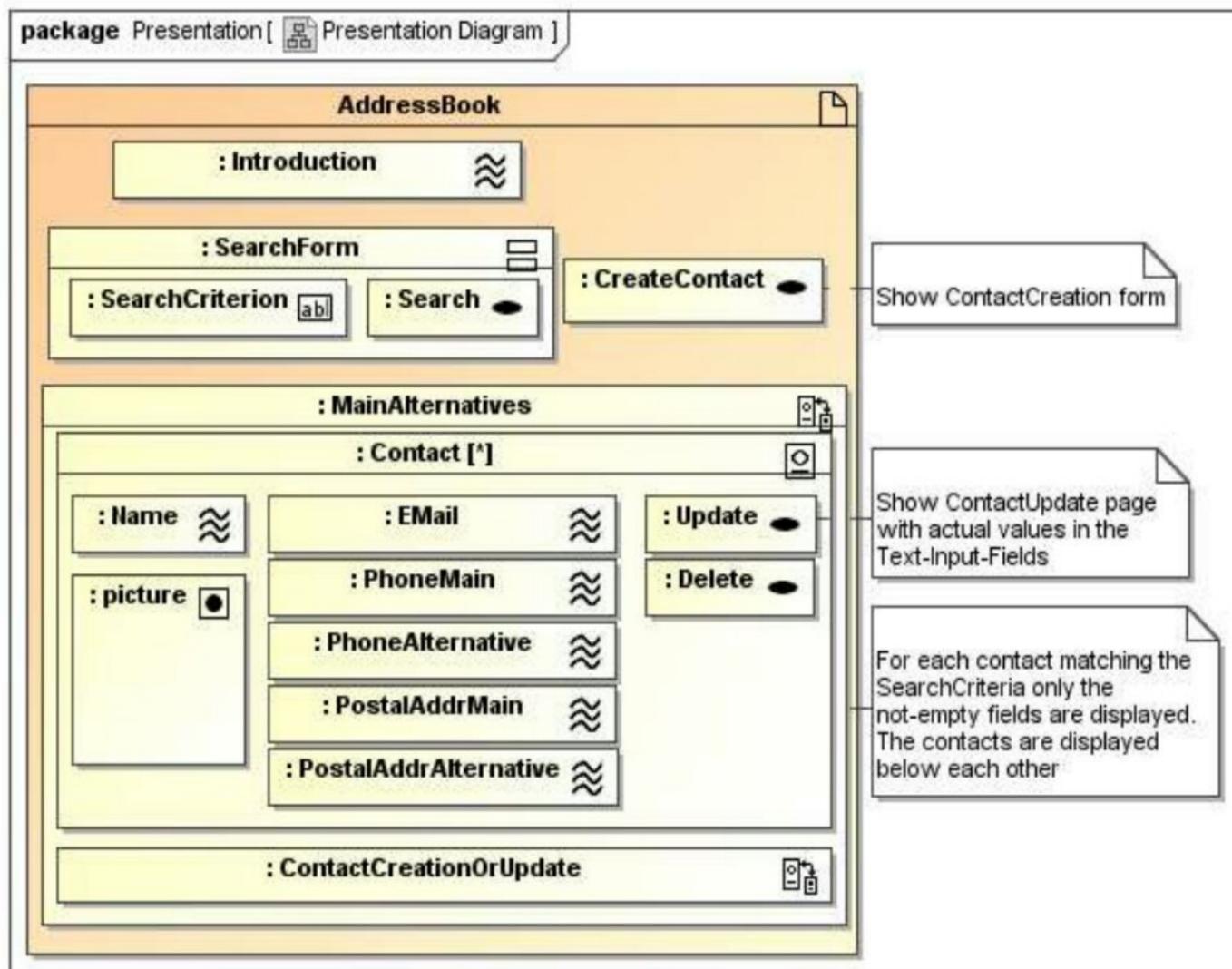
2.8.1.4 Modelo de Presentación.

Representa el layout subyacente a los modelos de navegación y procesos mediante una presentación abstracta, ya que una presentación concreta requeriría la especificación de propiedades físicas adicionales. Utiliza un diagrama de clases de UML para modelar la estructura.

En las figuras 8 y 9 se presentan los modelos de presentación para el caso de estudio tratado, donde se pretende mostrar las clases de navegación y de procesos a las que

pertenece a cada página web. Se puede observar, también, cómo se relacionan los elementos de distintas páginas, como llegar hasta ellas y los mensajes de error que pueden darse durante la interacción.

Figura 8. Modelo de Presentación



Fuente: Universidad Gastón Dachary (2016)

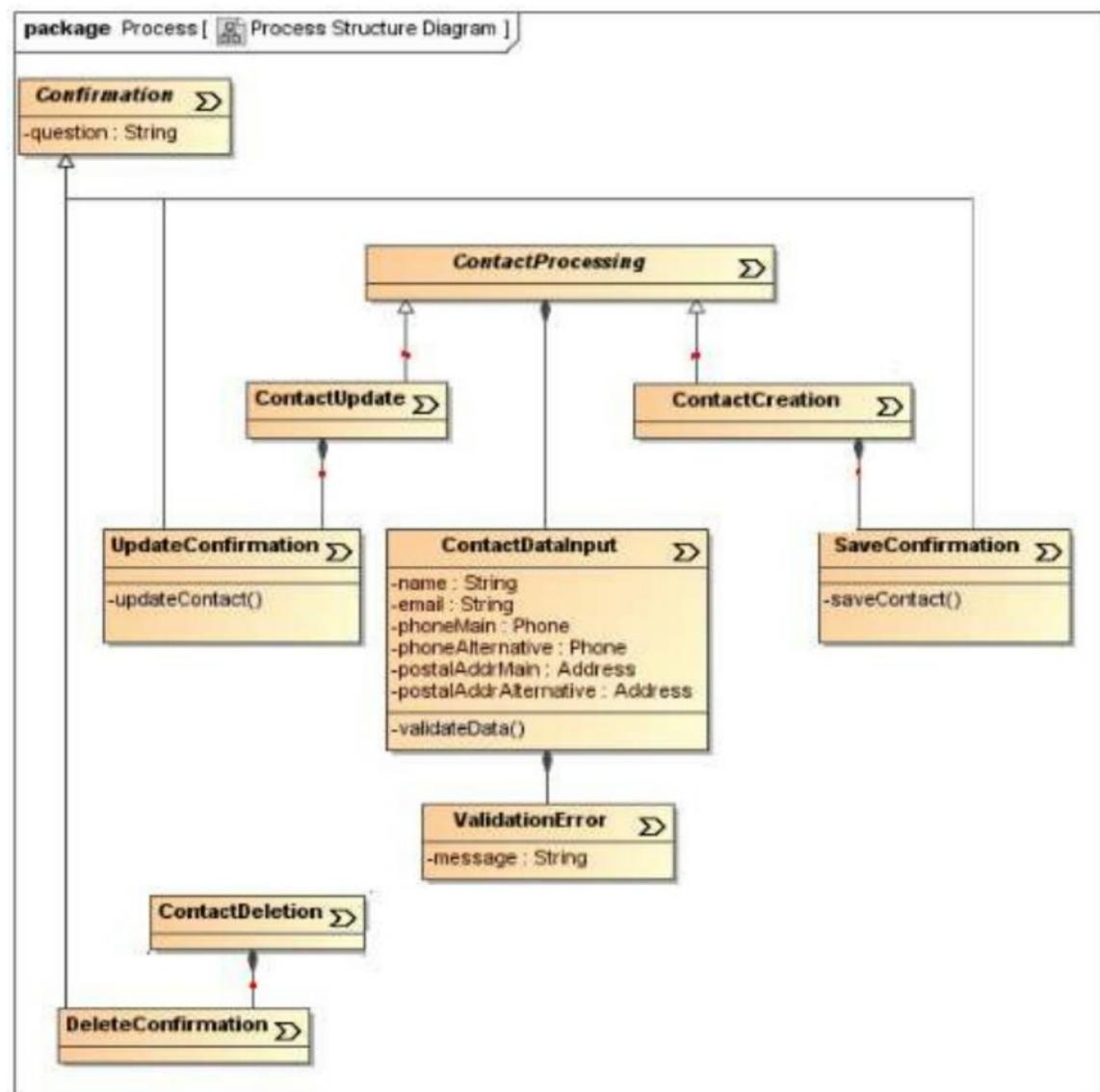
2.8.1.5 Modelo de Proceso

Representa los aspectos dinámicos de la aplicación Web y especifica funcionalidad cómo transacciones y workflows de actividades. Se modela mediante un diagrama de actividades de UML, y es resultado de refinar el diagrama de actividades modelado durante la especificación de requerimientos.

Muestra el flujo de la ejecución representado por nodos de actividad conectados, los nodos de control que proveen constructores de flujo de control como decisiones y sincronización y nodos de objetos que representan el flujo de datos. En la figura 10 se

presenta el modelo de estructura de procesos en donde mediante un diagrama de clases obtenido luego de haber usado la transformación de modelos de navegación a modelos de estructura de procesos, se pretende describir las relaciones entre las distintas «processClass».

Figura 9. Modelo del Proceso



Fuente: Universidad Gastón Dachary (2016)

2.9 ARQUITECTURA SOFTWARE

2.9.1 Patrón MVC

MVC es un patrón arquitectura, un modelo o guía que expresa cómo organizar y estructurar los componentes de un sistema software, sus responsabilidades y las relaciones existentes entre cada uno de ellos.

Su nombre, MVC, parte de las iniciales de Modelo-Vista-Controlador (Model-View-Controller, en inglés), que son las capas o grupos de componentes en los que

organizaremos nuestras aplicaciones bajo este paradigma.

➤ **Modelo**

En la capa Modelo encontraremos siempre una representación de los datos del dominio, es decir, aquellas entidades que nos servirán para almacenar información del sistema que estamos desarrollando. Por ejemplo, si estamos desarrollando una aplicación de facturación, en el modelo existirán las clases Factura, Cliente o Proveedor, entre otras.

➤ **La Vista**

Los componentes de la Vista son los responsables de generar la interfaz de nuestra aplicación, es decir, de componer las pantallas, páginas, o cualquier tipo de resultado utilizable por el usuario o cliente del sistema. De hecho, suele decirse que la Vista es una representación del estado del Modelo en un momento concreto y en el contexto de una acción determinada.

➤ **Controlador**

La misión principal de los componentes incluidos en el Controlador es actuar como intermediarios entre el usuario y el sistema. Serán capaces de capturar las acciones de éste sobre la Vista, como puede ser la pulsación de un botón o la selección de una opción de menú, interpretarlas y actuar en función de ellas. Por ejemplo, retornando al usuario una nueva vista que represente el estado actual del sistema, o invocando a acciones definidas en el Modelo para consultar o actualizar información.

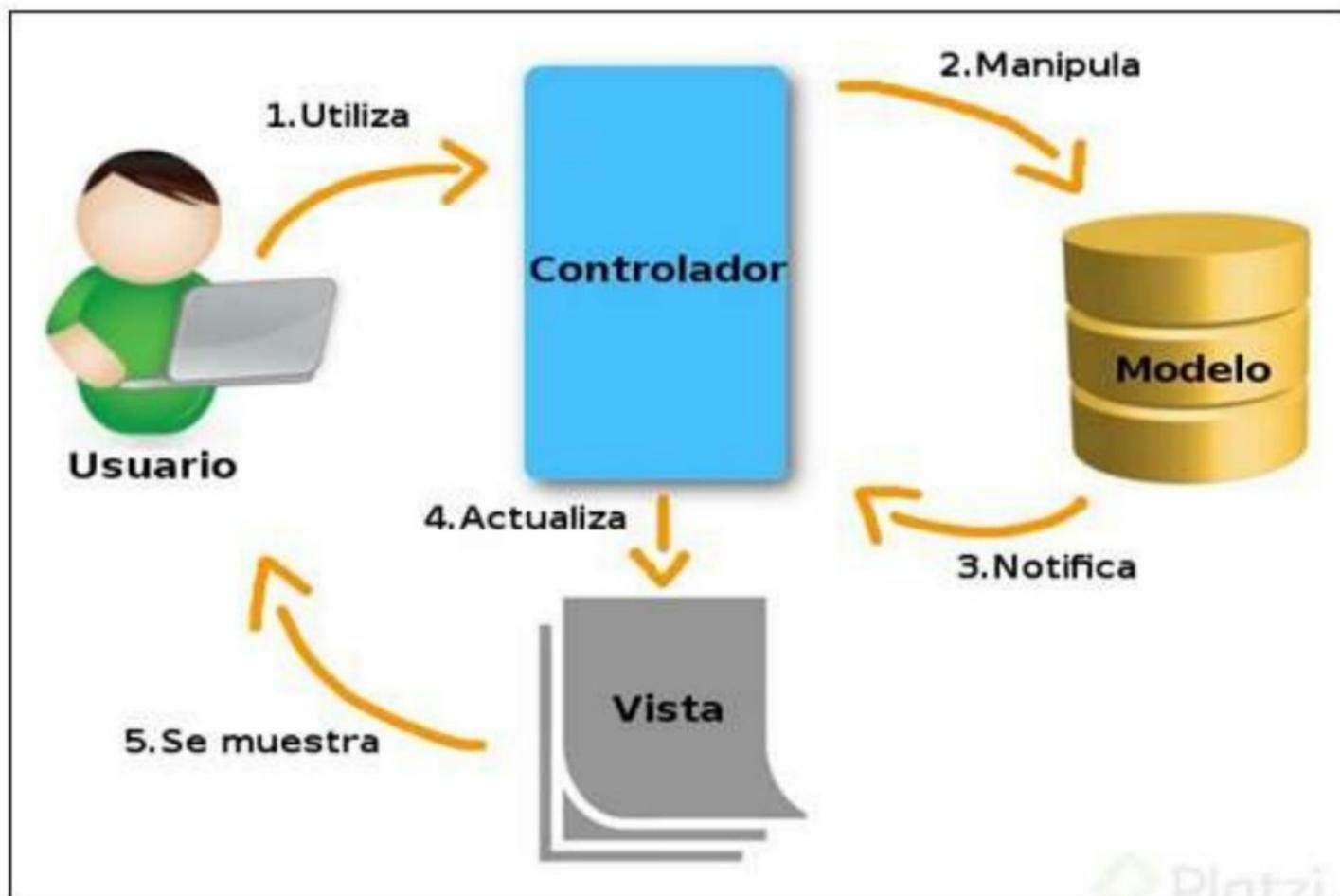
2.9.2 Ciclo de Vida del MVC

El siguiente diagrama refleja las relaciones existentes entre los componentes del Modelo, Vista y Controlador, y de éstos a su vez con el usuario, o cliente, del sistema.

Como se muestra en el diagrama, las acciones e información procedentes del usuario serán recogidas exclusivamente por los Controladores. Ningún componente de otra capa debe acceder a los datos generados desde el cliente, de la misma forma que sólo los componentes de la Vista estarán autorizados a generar interfaces

de usuario con las que enviar información de retorno.

Figura 10. Modelo Vista Controlador



Fuente: web

2.9.3 Ventajas y Desventajas

➤ Ventajas

Las principales ventajas de hacer uso del patrón MVC son:

- La separación del Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos.
- Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.

- Permite el escalamiento de la aplicación.

➤ **Desventajas**

Las desventajas de seguir el planteamiento de MVC son:

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje del patrón de diseño es más alta que usando otros modelos más sencillos.

Cabe mencionar que la comparación de ventajas y desventajas de MVC puede ser un tema muy subjetivo y se puede prestar como tema de debate, sin embargo se tomó la decisión usando principalmente los puntos mencionados anteriormente ya que en términos generales la balanza se inclina a favor del MVC en vez de en su contra. (Rivera López, 2008, pág. 8)

2.10 CALIDAD DE SOFTWARE

Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad. El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, ya que se requiere establecer métricas que permitan evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar. (Abud Figueroa, 2012, pág. 1)

2.10.1 Normas de Calidad ISO 9000

Un sistema de aseguramiento de la calidad se define como la estructura organizacional, responsabilidades, procedimientos, procesos y recursos necesarios para implementar la administración de la calidad. Los sistemas de aseguramiento de la calidad se crean para ayudar a las organizaciones a asegurar que sus productos y servicios satisfagan las expectativas del consumidor gracias a que cumplan con sus

especificaciones. Estos sistemas cubren una amplia variedad de actividades, que contemplan todo el ciclo de vida del producto, incluidos planeación, control, medición, pruebas e informes, así como la mejora de los niveles de calidad en todo el proceso de desarrollo y manufactura. La norma ISO 9000 describe en términos generales los elementos de aseguramiento de la calidad que se aplican a cualquier negocio, sin importar los productos o servicios ofrecidos. Para registrarse en alguno de los modelos del sistema de aseguramiento de la calidad contenidos en la ISO 9000, por medio de auditores externos se revisan en detalle el sistema y las operaciones de calidad de una compañía, respecto del cumplimiento del estándar y de la operación eficaz. Después de un registro exitoso, el grupo de registro representado por los auditores emite un certificado para la compañía. Auditorías semestrales de supervisión aseguran el cumplimiento continuo de la norma. (Pressman, 2010, pág. 379)

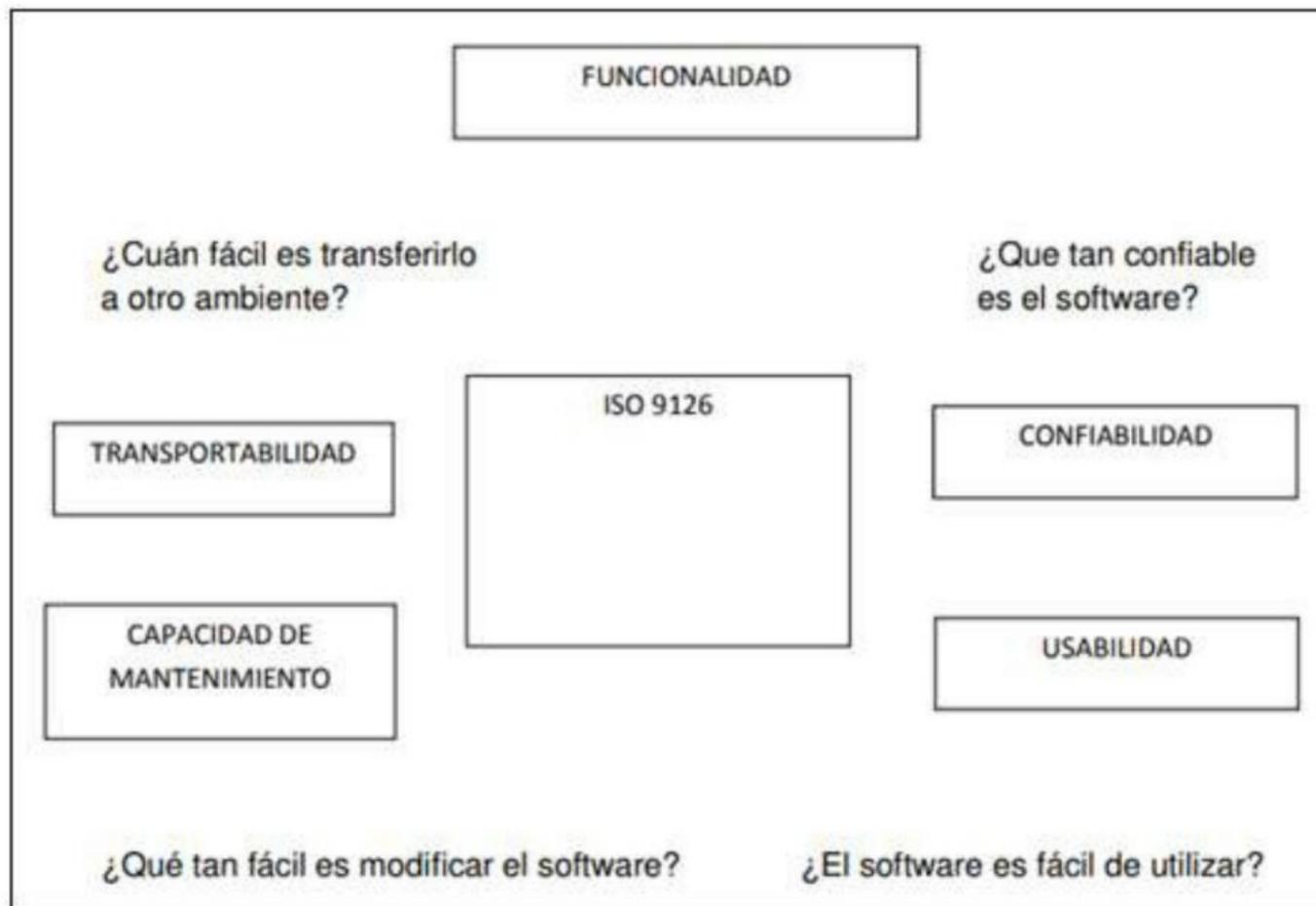
2.10.2 Factores de la Calidad ISO 9126

La norma ISO 9126 o ISO/IEC 9126 es un conjunto de características y subcaracterísticas definidas que permiten conocer el nivel de la calidad del software a través de un proceso de evaluación de acuerdo a las métricas o indicadores que presenta el modelo de calidad.

Según Modelo de Calidad de Software para la DGSC, <http://www.dgsc.go.cr/>, la norma ISO/IEC 9126 es un modelo de calidad estándar para productos de software, donde se describen las diferentes características y sub características que debe cumplir un sistema de software para que pueda ser considerado como un sistema de calidad. Además, este modelo también define una serie de métricas y se divide en dos partes:

- Calidad externa e interna del producto de software
- Calidad de uso del producto

Figura 11. Características ISO/IEC 9126



Fuente: ISO/IEC 9126(Traducción)

2.10.2.2 Funcionalidad

Se refiere a un conjunto de funciones y propiedades que tratan de satisfacer las necesidades. Sus atributos son: adecuación, exactitud, interoperabilidad, seguridad y cumplimiento funcional.

Los puntos de función se describen como medidas básicas desde donde se calculan métricas de productividad, estos se utilizan de las siguientes dos formas:

- Como una variable de estimación que se utiliza para dimensionar cada elemento del software.
- Como métricas de líneas base recopilada de proyectos anteriores y utilizados junto con variables de estimación para desarrollar proyecciones de costo y esfuerzo.

Para estimaciones de PF la descomposición funciona de la siguiente manera:

Tabla 3. Dominios de Información de Puntos Función

Dominio de información	Descripción
Número de entradas del usuario	Se encuentra cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.
Número de salidas del usuario	Se cuenta cada salida que proporciona al usuario información orientada a la aplicación, en este contexto la salida se refiere a informes, pantallas, mensajes de error y demás. Los elementos de datos particulares dentro de un informe no se encuentran de forma separada.
Número de peticiones al usuario	Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
Número de archivos	Se cuenta cada archivo maestro lógico (esto es un grupo lógico de datos que se puede ser una parte de una gran base de datos o un archivo independiente).
Número de interfaces externas	Se cuenta todas las interfaces legibles por la maquina (por ejemplo archivos de datos de disco), que se utilizan para transmitir información a otros sistemas.

Fuente: ISO/IEC 9126(Traducción)

Los puntos de función se calculan completando la siguiente tabla:

Tabla 4. Factores de Ponderación

Parámetros de medición	Cuenta	Simple	Medio	Complejo	Resultado
Número de entradas de usuario	N_1	3	4	6	$N_1 \cdot \text{factor}$
Número de salidas de usuario	N_2	4	5	7	$N_2 \cdot \text{factor}$
Número de peticiones de usuario	N_3	3	4	6	$N_3 \cdot \text{factor}$
Número de archivos	N_4	7	10	15	$N_4 \cdot \text{factor}$
Número de interfaces externas	N_5	5	7	10	$N_5 \cdot \text{factor}$
Cuenta total					$\sum(N_i \cdot \text{factor})$

Fuente: ISO/IEC 9126(Traducción)

Para calcular puntos de función (PF) se utiliza la siguiente relación:

$$PF = cuenta - total * [0,65 + 0,01 * \sum (F_i)]$$

En donde *cuenta - total* es la suma de todas las entradas de los factores de ponderación obtenidas en la tabla anterior.

F_i ($i = 1$ a 14), son valores de ajuste de complejidad según las respuestas a las siguientes preguntas:

Tabla 5. Valores de Ajuste de la Complejidad

Importancia	0	20	40	60	80	100
	%	%	%	%	%	%
Escala	No influencia	Incidental	Moderado	Medio	Significativo	Esencial
Factor	0	1	2	3	4	5
¿Requiere el sistema copias de seguridad y recuperación fiables?						
¿Se requiere comunicación de datos?						
¿Existen funciones de procesamiento distribuido?						
¿Es crítico el rendimiento?						
¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?						
¿Requiere el sistema entrada de datos interactiva?						
¿Requiere el sistema entrada de datos interactivos que las transacciones de entrada se lleven a cabo sobre múltiples entradas u operaciones?						
¿Se actualizan los archivos maestros de forma interactiva?						
¿Son complejas las entradas, las salidas, los						

archivos o peticiones?						
¿Es complejo el procesamiento interno?						
¿Se ha diseñado código para ser reutilizable?						
¿Están incluidas en el diseño la conversión y la instalación?						
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?						
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?						

Fuente: ISO/IEC 9126(Traducción)

También la métrica de adecuación según la siguiente tabla:

Tabla 6. Métrica de Adecuación

Nombre:	Completitud de implementación funcional
Propósito:	Cómo de completa es la implementación funcional.
Método de aplicación:	Contar las funciones faltantes detectadas en la evaluación y comparar con el número de funciones descritas en la especificación de requisitos.
Medición, fórmula:	$X = 1 - A/B$ <p>A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos</p>
Interpretación:	$0 \leq X \leq 1$ <p>Entre más cercano a 1, más completa.</p>
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ <p>A = count B = count</p>
Fuente de medición:	Especificación de requisitos Diseño

	Código fuente
	Informe de revisión

Fuente: ISO/IEC 9126(Traducción)

2.10.2.3 Fiabilidad

Se refiere a un conjunto de atributos que miden la capacidad que tiene el software para mantener un nivel de rendimiento óptimo, bajo determinadas condiciones y durante un periodo de tiempo determinado. Sus atributos son madurez, tolerancia a fallos y la capacidad de recuperación ante un fallo.

Para que un sistema sea fiable, se debe garantizar un nivel de seguridad. La seguridad se subdivide a su vez en confidencialidad, autenticación, control de acceso, integridad de los datos y responsabilidad de los usuarios. Para garantizarla se ofrecen distintos mecanismos como certificados digitales y sockets (SSL) y hace un tratamiento adecuado de la información personal y privada de los usuarios.

La confiabilidad de un sistema se calcula mediante la siguiente relación

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$
 Probabilidad de no hallar una falla: $P(T > t) = 1 - F(t)$
 Con: $F(t) = F_c * (e^{-\lambda/7 * 12})$

Dónde:

$F_c = 0,87$: funcionalidad del sistema

$\lambda = 1$: tasa de fallos dentro de un mes

También utilizando la métrica de madurez de la siguiente tabla:

Tabla 7. Métrica de Madurez

Nombre:	Suficiencia de las pruebas
Propósito:	Cuántos de los casos de prueba necesarios están cubiertos por el plan de pruebas.
Método de aplicación:	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.
Medición, fórmula:	$X = A/B$

	A = número de casos de prueba en el plan B = número de casos de prueba requeridos
Interpretación:	$0 \leq X$ Entre X sea mayor, mejor la suficiencia.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	A proviene del plan de pruebas B proviene de la especificación de requisitos
ISO/IEC 12207 SLCP:	Aseguramiento de Calidad Resolución de problemas Verificación
Audiencia:	Desarrolladores Mantenedores

Fuente: ISO/IEC 9126(Traducción)

2.10.2.4 Usabilidad

Se refiere a un conjunto de atributos que miden el esfuerzo cognitivo necesario que deben realizar los usuarios para utilizar el sistema de software. Sus atributos son comprensión, curva de aprendizaje y operatividad.

Utilizando la métrica de entendibilidad según la siguiente tabla:

Tabla 8. Métrica de Entendibilidad

Nombre:	Funciones evidentes
Propósito:	Qué proporción de las funciones del sistema son evidentes al usuario.
Método de aplicación:	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Medición, fórmula:	$X = A/B$ A = número de funciones (o tipos de funciones) evidentes al usuario B = total de funciones (o tipos de funciones)

Interpretación:	0 <= X <= 1 Entre más cercano a 1, mejor.
Tipo de escala:	Absoluta
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Especificación de requisitos Diseño Informe de revisión
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta

Fuente: ISO/IEC 9126(Traducción)

2.10.2.5 Eficiencia

Se refiere a un conjunto de atributos que miden la relación entre el rendimiento del software y la cantidad de recursos utilizados, dada una situación determinada. Sus atributos son tiempo de respuesta y recursos utilizados

La eficiencia se entiende como la capacidad del sistema para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados bajo condiciones determinadas.

Utilizando la métrica de comportamiento en el tiempo según la siguiente tabla:

Tabla 9. Métrica de Comportamiento en el Tiempo

Nombre:	Tiempo de respuesta
Propósito:	Cuál es el tiempo estimado para completar una tarea.
Método de aplicación:	Evaluar la eficiencia de las llamadas al SO y a la aplicación. Estimar el tiempo de respuesta basado en ello. Puede medirse: Todo o partes de las especificaciones de diseño. Probar la ruta completa de una transacción. Probar módulos o partes completas del producto. Producto completo durante la fase de pruebas.

Medición, fórmula:	X = tiempo (calculado o simulado)
Interpretación:	Entre más corto, mejor.
Tipo de escala:	Proporción
Tipo de medida:	X = time
Fuente de medición:	Sistema operativo conocido Tiempo estimado en llamadas al sistema
ISO/IEC 12207	Verificación
SLCP:	Revisión conjunta
Audiencia:	Desarrolladores Requeridores

Fuente: ISO/IEC 9126(Traducción)

2.10.2.6 Mantenibilidad

Se refiere a un conjunto de atributos relacionados con el esfuerzo necesario para realizar determinadas modificaciones en el producto. Sus atributos son la capacidad de ser analizado, capacidad para ser modificado, estabilidad y capacidad para ser probado.

El estándar IEEE 982.1 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad del producto de software, se determina con la siguiente relación

$$IMS = [M_T - (F_c + F_a + F_d)] / M_T$$

Dónde:

M_T = número de módulos en la versión actual.

F_c = número de módulos en la versión actual que se han cambiado.

F_a = número de módulos en la versión actual que se han añadido.

F_d = número de módulos en la versión anterior que se han borrado en la versión actual.

A medida que el IMS se aproxima a 1.0 se logra una madurez estable.

Utilizando la métrica de cambiabilidad según la siguiente tabla:

Tabla 10. Métrica de Cambiabilidad

Nombre:	Registro de cambios
Propósito:	¿Se registran adecuadamente los cambios a la especificación y a los módulos con comentarios en el código?
Método de aplicación:	Registrar la proporción de información sobre cambios a los módulos
Medición, fórmula:	$X = A/B$ A = número de cambios a funciones o módulos que tienen comentarios confirmados B = total de funciones o módulos modificados
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más registrable. 0 indica un control de cambios deficiente o pocos cambios y alta estabilidad.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Sistema de control de configuraciones Bitácora de versiones Especificaciones
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta
Audiencia:	Desarrolladores Mantenedores Requeridores

Fuente: ISO/IEC 9126(Traducción)

2.10.2.7 Portabilidad

Son atributos con la capacidad del software de ser transferido de un entorno a otro. Sus atributos son adaptabilidad, capacidad de instalación, coexistencia y capacidad de

reemplazamiento.

Utilizando la métrica de conformidad de transportabilidad según la siguiente tabla:

Tabla 11. Métrica de Conformidad de Transportabilidad

Nombre:	Conformidad de Transportabilidad
Propósito:	Cómo de transportable es el producto según las regulaciones, estándares y convenciones aplicables.
Método de aplicación:	Contar los artículos encontrados con conformidad y comparar con el número de artículos en la especificación que requieren conformidad.
Medición, fórmula:	$X = A/B$ A = número de artículos implementados de conformidad B = total de artículos que requieren conformidad
Interpretación:	$0 \leq X \leq 1$...// Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Especificación de conformidad y estándares, convenciones y regulaciones relacionados. Diseño Código fuente Informe de revisión
ISO/IEC 12207	Verificación
SLCP:	Revisión conjunta
Audiencia:	Requeridores Desarrolladores

Fuente: ISO/IEC 9126(Traducción)

2.11 ANÁLISIS DE COSTOS DE SOFTWARE COCOMO

Una de las tareas de mayor importancia en la administración de proyectos de software es la estimación de costos. Si bien es una de las primeras actividades, inmediatamente

posterior al establecimiento de los requerimientos, se ejecuta regularmente a medida que el proyecto progresa con el fin de ajustar la precisión en la estimación. Como sucede en cualquier modelo de estimación, no hay forma de compensar la falta o calidad de los datos de entrada y/o precisión de los valores de los factores de costo. El modelo COCOMO es un ejemplo de modelo algorítmico. (Gómez, López , Migani, & Otazú, 2010, pág. 5)

2.11.1 COCOMO II

Según Sommerville (2011, pág. 637), "El COCOMO II es un modelo empírico que se derivó al recopilar datos a partir de un gran número de proyectos de software. Dichos datos se analizaron para descubrir qué fórmulas se ajustaban mejor con las observaciones. Dichas fórmulas vinculan el tamaño del sistema y los factores del producto, proyecto y equipo, con el esfuerzo para desarrollar el sistema. COCOMO II es un modelo de estimación bien documentado y no registrado. COCOMO II se desarrolló a partir de los primeros modelos de estimación de costos COCOMO, que se basaron principalmente en el desarrollo de código original (Boehm, 1981; Boehm y Royce, 1989). El modelo COCOMO II toma en cuenta enfoques más modernos para el desarrollo de software, tales como el desarrollo rápido que usa lenguajes dinámicos, el desarrollo mediante composición de componentes y el uso de programación de base de datos. COCOMO II soporta el modelo en espiral de desarrollo, e incrusta submodelos que producen estimaciones cada vez más detalladas.

Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

COCOMO II tiene cuatro submodelos: modelo de composición de aplicación, modelo de diseño temprano, modelo de reutilización y modelo post-arquitectónico.

2.11.1.1 Modelo Post-Arquitectónico

El modelo post arquitectura se utiliza en la fase de construcción del producto software, fase en la cual se tiene a disposición varios parámetros necesarios para aplicar este

modelo de COCOMO II.

Según Sommerville (2011, pág. 642), “El modelo posarquitectónico es el más detallado de los modelos COCOMO II. Se usa una vez que está disponible un diseño arquitectónico inicial para el sistema, de manera que se conoce la estructura del subsistema. Entonces es posible hacer estimaciones para cada parte del sistema”.

“Como todos los modelos de estimación para software, los modelos COCOMO II requieren información sobre el dimensionamiento del software. Como parte de la jerarquía del modelo, están disponibles tres diferentes opciones de dimensionamiento: puntos objeto, puntos de función y líneas de código fuente”. (Pressman, 2010, pág. 609)

Por lo tanto, el modelo post arquitectura del modelo COCOMO II requiere información detallada de la estructura, del dimensionamiento del producto software.

Las ecuaciones que se utilizan son:

$$\text{➤ } E = a * (Kl)^b * m(X), \text{ en persona - mes}$$

$$\text{➤ } T_{dev} = c * (E)^d, \text{ en meses}$$

$$\text{➤ } P = \frac{E}{T_{dev}}, \text{ en personas}$$

Dónde:

E = es el esfuerzo requerido por el proyecto, en persona-mes.

T_{dev} = es el tiempo requerido por el proyecto, en meses.

P = es el número de personas requerido por el proyecto.

a , b , c y d son constantes con valores definidos en una tabla, según cada sub modelo.

Kl = es la cantidad de líneas de código, en miles.

$m(X)$ = Es un multiplicador que depende de 15 atributos.

2.11.1.2 Constantes de Complejidad

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

Tabla 12. Constantes de Complejidad

Modo	a	b	C	d
Orgánico	2.40	1.05	2.50	0.38
Semilibre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Fuente: Pressman (2010)

2.11.1.3 Factores Multiplicadores de Esfuerzo

Cada factor se cuantifica para un entorno de proyecto. La escala es muy baja - bajo – nominal – alto – muy alto – extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo *DATA* es calificado como *muy alto*, el resultado de la fórmula debe ser multiplicado por 1000).

El significado de los atributos es el siguiente, según su tipo:

➤ **De software**

- **RELY:** garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (*muy bajo*) hasta la posible pérdida de vidas humanas (*extremadamente alto*, software de alta criticidad).
- **DATA:** tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación: D/K donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
- **CPLX:** representa la complejidad del producto.

➤ **De hardware**

- **TIME:** limitaciones en el porcentaje del uso de la CPU.
- **STOR:** limitaciones en el porcentaje del uso de la memoria.
- **VIRT:** volatilidad de la máquina virtual.
- **TURN:** tiempo de respuesta requerido.

➤ **De Personal**

- **ACAP:** calificación de los analistas.
- **AEXP:** experiencia del personal en aplicaciones similares.
- **PCAP:** calificación de los programadores.
- **VEXP:** experiencia del personal en la máquina virtual.
- **LEXP:** experiencia en el lenguaje de programación a usar.

➤ **De Proyecto**

- **MODP:** uso de prácticas modernas de programación.
- **TOOL:** uso de herramientas de desarrollo de software.
- **SCED:** limitaciones en el cumplimiento de la planificación.

El valor de cada atributo, de acuerdo a su calificación:

Tabla 13. Variables Factor Ajuste del Esfuerzo

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						

Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

Fuente: IBM, Estimación del Costo del Software (2011)

2.12 SEGURIDAD DEL SISTEMA

2.12.1 Norma ISO 27001

La norma ISO/IEC 27001 que en siglas significa Technology Security Techniques viene a ser la evolución del estándar de buenas prácticas ISO creado en 1995, para lo cual su creación conlleva un progreso certificable llamado estándar 27001, este tipo de certificación facilitara a la Seguridad Informática al momento de establecer, implantar, operar, supervisar, mantener, mejorar un SGSI⁸. El funcionamiento de la norma ISO 27001 que sirve para brindar soporte de los datos que se encuentran registrados en la organización. Para ello se implementa un SGSI con estándar 27001 para el cuidado de la información brindando confidencialidad, disponibilidad e integridad de los datos para el buen uso de la información y no divulgación del mismo en Organizaciones ya sean grandes o pequeñas. (Cadme Ruiz & Duque Pozo, 2012, pág. 17)

2.12.1.1 Confidencialidad de datos

Es cuando un usuario o empleado de la empresa garantice seguridad al momento de ingresar a la información, divulgando dicha información a personas ajenas a la empresa; con ello se busca conseguir una seguridad donde los que puedan acceder a

⁸ SGSI Sistema de Gestión de Seguridad de la Información

los datos son los administradores del sistema o la misma gerencia.

2.12.1.2 Disponibilidad de datos

La integridad de datos es el acceder a la información de la empresa al tiempo o la hora que sea con el fin que los usuarios alteren, actualicen, respalden los datos útiles y no tener pérdidas financieras o de personal.

2.12.1.3 Integridad de datos

La integridad de los datos hace referencia a que los datos no pueden ser alterados por ningún tipo de personal, solo por la alta dirección, para ello deben de tener un tipo de seguridad que ayude al manejo debido de los datos para beneficio propio de la empresa.

2.13 PRUEBAS DE ACEPTACIÓN

El objetivo principal de las pruebas de funcionamiento es aportar calidad al producto que se está desarrollando.

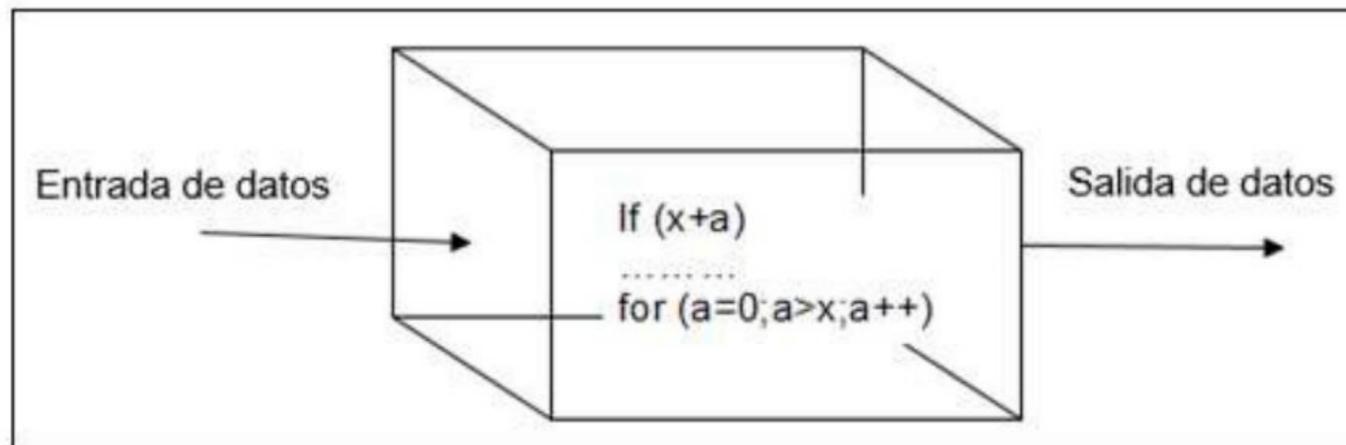
Para llevar a cabo las pruebas verificaremos el comportamiento del programa sobre un conjunto de casos de prueba. Estos casos de prueba se generarán mediante técnicas y estrategias específicas de pruebas que nos ayudarán a conseguir la búsqueda de los errores de un programa. Para encontrar los errores, dos de las técnicas más utilizadas en las pruebas son las técnicas de caja blanca y caja negra. Sin embargo a la hora de probar tenemos que establecer prioridades. Una de las prioridades más importantes que hay que tener en cuenta son los recursos de los que se va a disponer en el proyecto. Al realizar un análisis de los riesgos para asegurar que se dispone de recursos suficientes para poder llevar a cabo las pruebas. Estos recursos irán desde el personal, hasta las herramientas que se vayan a utilizar. (Sánchez Peño, 2015, pág. 23)

2.13.1 Pruebas de Caja Blanca

La técnica de caja blanca, a veces definida como prueba de “caja de cristal” o “caja transparente”, es una técnica de diseño de casos de prueba que usa la estructura de

control para obtener los casos de prueba. Dentro de esta estructura de control podemos encontrar la estructura de un componente de software como puede ser sentencias de decisiones, caminos distintos del código, la estructura de una página web, etc. (Sánchez Peño, 2015, pág. 31)

Figura 12. Caja Blanca



Fuente: Pruebas de Software. Fundamentos y Técnicas (2015)

Los métodos de prueba de caja blanca aportan los siguientes puntos:

- Garantizan que todas las rutas del código se revisan al menos una vez.
- Revisan las condiciones lógicas.
- Revisan estructuras de datos.

2.13.1.2 Pruebas de Ruta Básica

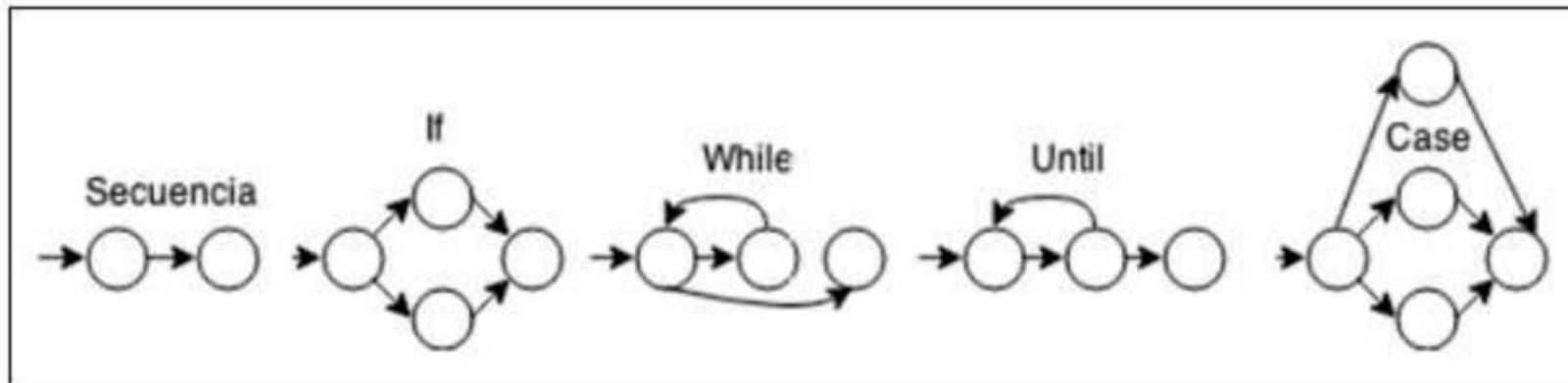
La prueba de ruta básica es un método de prueba de caja blanca, que inicialmente propuso Tom McCabe. Este tipo de pruebas se basa en diseñar un caso de prueba por cada camino independiente del programa, con esta técnica se intenta garantizar que se prueben todos los caminos de ejecución del programa al menos una vez.

Para aplicar las pruebas de ruta básica hay que conocer la representación que se conoce como grafo de flujo y las rutas de programa independiente.

➤ Notación de grafo de flujo

Esta representación se realiza a partir del PDL(Program Design Language) o pseudocódigo, y es una representación de los caminos que puede tomar un programa durante su ejecución.

Figura 13. Notación de Grafo de Flujo



Fuente: Pruebas de Software. Fundamentos y Técnicas (2015)

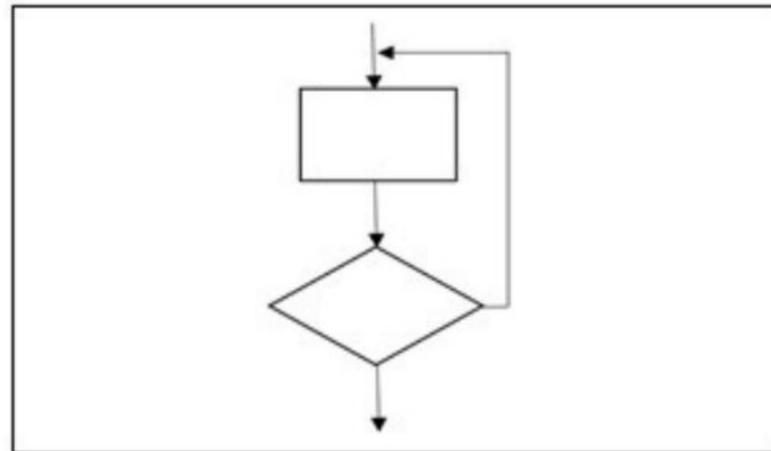
➤ **Pruebas de ciclos o bucles**

Este tipo de prueba se centra exclusivamente en la validación de las construcciones de bucles.

Para Beizer, citado por Pruebas de Software, Fundamentos y Técnicas (2015, p.36), define que pueden existir cuatro tipos diferentes de bucles:

- Simples, las pruebas pueden aplicarse como bucles simples donde “n” es el número máximo de iteraciones permitidas por el bucle.
 1. Saltar por completo el ciclo.
 2. Pasar una sola vez a través del ciclo.
 3. Dos pasadas a través del ciclo.
 4. Realizar “m” pasadas a través del ciclo con $m < n$.
 5. Hacer $n-1$, n y $n+1$ pasos por el bucle.

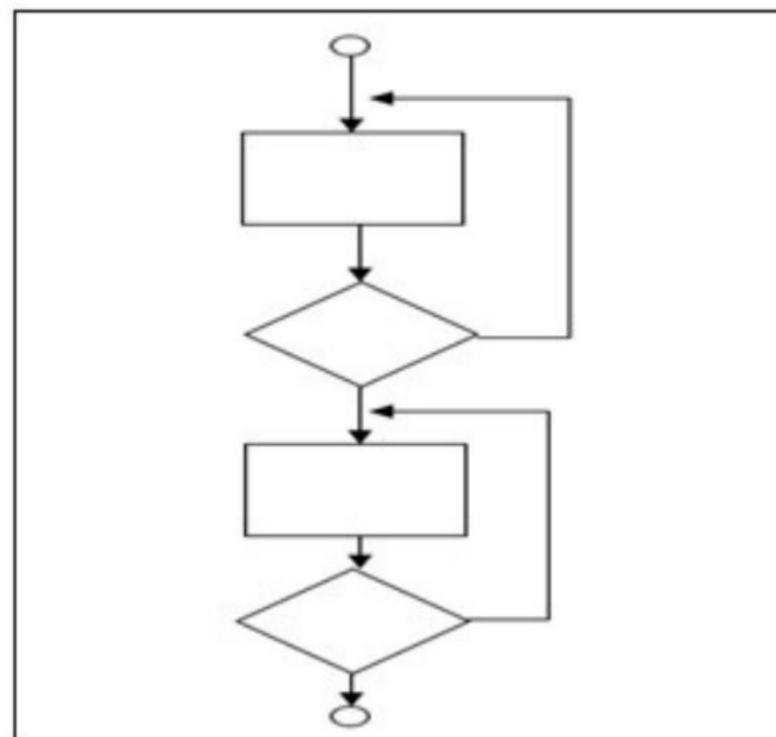
Figura 14. Bucle Simple



Fuente: Pruebas de Software. Fundamentos y Técnicas (2015)

- Concatenados, los ciclos concatenados se pueden enfocar como bucles simples, siempre que cada ciclo sea independiente de los otros, pero si dos o más ciclos que se concatenan no son independientes, es decir, se usa el valor del ciclo 1 como el valor inicial del ciclo 2, entonces se recomienda usar el enfoque aplicado a bucles anidados.

Figura 15. Bucle concatenado

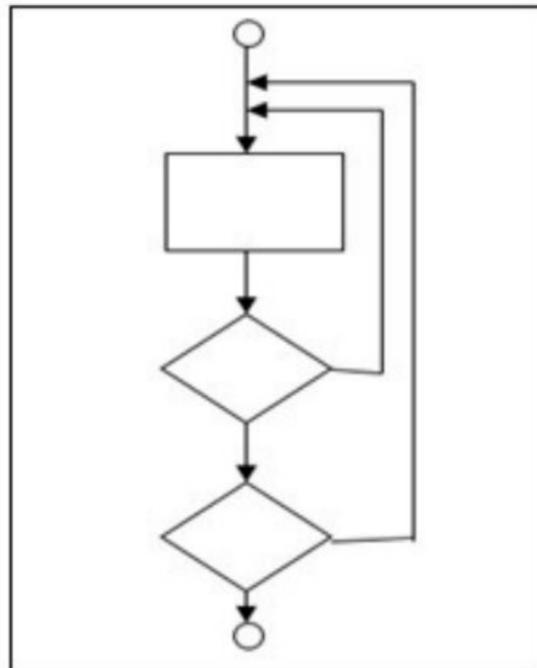


Fuente: Pruebas de Software. Fundamentos y Técnicas (2015)

- Anidados, el número de pruebas con este tipo de bucles crece exponencialmente conforme el nivel de bucle anidado aumenta. Para reducir el número de pruebas propone seguir los siguientes pasos:

1. Comenzar con el bucle más interno y establecer todos los otros bucles a valores mínimos.
2. Realizar pruebas de ciclo simple para el ciclo más interno mientras se mantienen los ciclos exteriores en sus valores mínimos de parámetro de iteración. Agregar otras pruebas para valores fuera de rango.
3. Trabajar hacia fuera y realizar pruebas para el siguiente ciclo pero mantener los otros ciclos exteriores en valores mínimos y los otros ciclos anidados en valores típicos.
4. Continuar hasta que todos los ciclos se hayan probado.

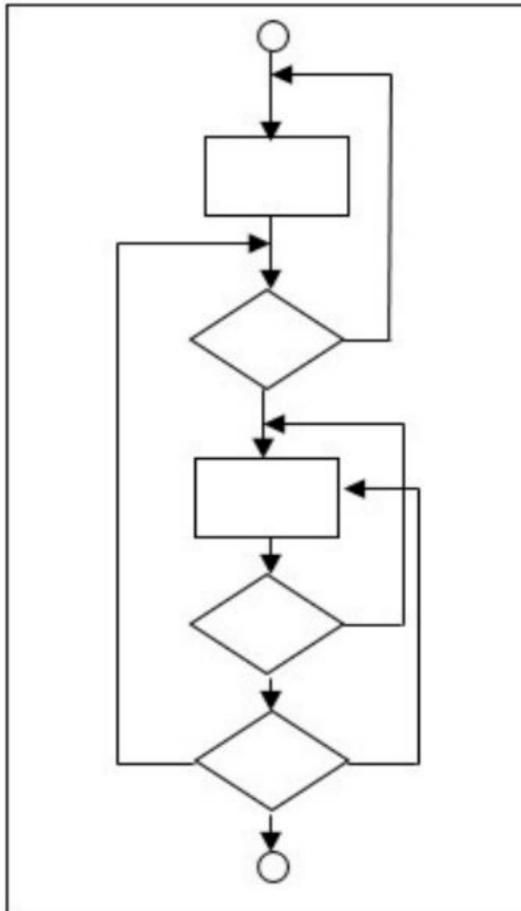
Figura 16. Bucle anidado



Fuente: Pruebas de Software. Fundamentos y Técnicas (2015)

- No estructurados, estos ciclos BelZER también los denomina como “horribles” y, siempre que se pueda, esta clase de bucles deben ser rediseñados en base a una programación estructurada.

Figura 17. Bucle no estructurado

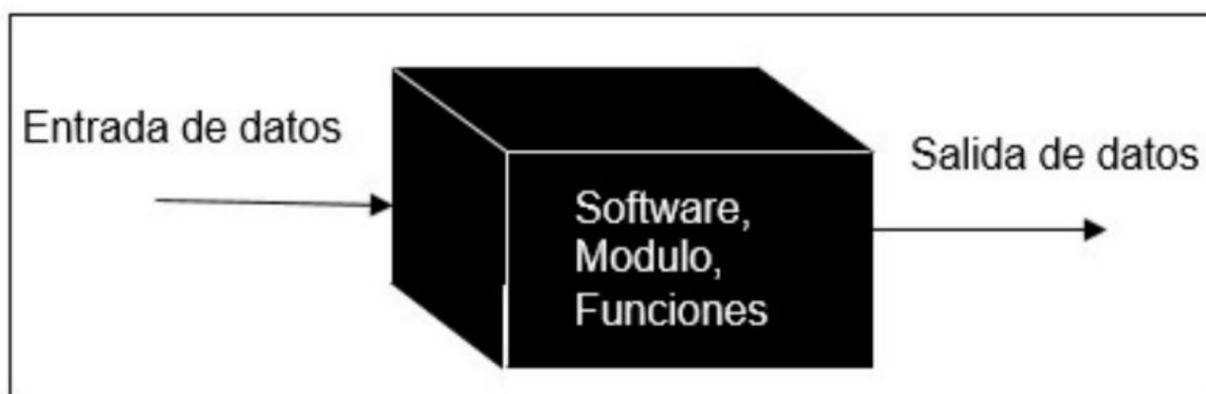


Fuente: Pruebas de Software. Fundamentos y Técnicas (2015)

2.13.2 Pruebas de Caja Negra

Las técnicas de diseño de caja negra, también llamadas pruebas de comportamiento, son las que utilizan el análisis de las especificación, tanto funcional como no funcional, sin tener en cuenta la estructura interna del programa para diseñar los casos de prueba y, a diferencia de pruebas de caja blanca, estas pruebas se suelen realizar durante las últimas etapas de la prueba.

Figura 18. Caja Negra



Fuente: Pruebas de Software. Fundamentos y Técnicas (2015)

Con los métodos de caja negra se intenta encontrar los errores:

- Funciones incorrectas o faltantes.
- Errores de inicialización y terminación.
- Errores de interfaz.
- Errores en las estructuras.

Hay varios métodos que se pueden aplicar a la hora de escoger la técnica de caja negra como modelo para las pruebas.

2.13.2.2 Partición de equivalencia

En las pruebas de partición de equivalencia, los valores de entrada del programa o del sistema se dividen en grupos que vayan a tener un comportamiento similar, de manera que puedan ser procesados de la misma forma. Las particiones de equivalencia o clases son aplicables a datos válidos y datos no válidos. También pueden aplicarse a los valores de salida, valores internos, valores relativos al tiempo o a los parámetros de interfaz.

Las clases de equivalencia se definen de acuerdo a las siguientes directrices:

- Si una condición de entrada especifica un rango, se definen una clase de equivalencia valida y dos invalidas.
- Si una condición de entrada requiere un valor especifico, se definen una clase de equivalencia valida y dos invalidas
- Si una condición de entrada especifica un miembro de un conjunto, se definen una clase de equivalencia valida y otra invalida.
- Si una condición de entrada es booleana, se definen una clase de equivalencia valida y otra invalida.

Al aplicar estas directrices se desarrollaran y ejecutaran los casos de prueba para cada objeto de los datos del dominio de entrada.

2.13.2.3 Análisis del valor limite

Es mayor el número de errores que se presenta en los límites del dominio de entrada que en el centro.

Esta técnica de diseño de casos de prueba complementa la técnica de partición de equivalencia u otras pruebas de caja negra.

Los valores máximos y mínimos de una partición son sus valores límites. Las pruebas pueden diseñarse para cubrir tanto los valores límites válidos, como no válidos.

Las directrices para el análisis del valor límite son similares a las de partición de equivalencia.

- Si una condición de entrada especifica un rango limitado por los valores a y b, los casos de prueba deben diseñarse con esos valores, además de los que se encuentran por encima y por debajo de ellos.
- Si una condición de entrada especifica diversos valores, deben desarrollarse casos de prueba que ejerciten los números máximo y mínimo. También se prueban los valores ubicados por encima y por debajo de estos máximos y mínimos.
- Si la estructura interna de datos del programa tiene límites prescritos debe diseñarse un caso de prueba para ejercitar los límites de la estructura de datos.

2.13.2.4 Pruebas de tabla de decisión

Estas pruebas se realizan cuando la lógica a probar está basada en decisiones o, dicho de otra manera, donde la lógica predominante es del tipo if, then y else.

Las tablas de decisión constituyen una buena manera de reflejar los requisitos del sistema que contienen condiciones lógicas y de documentar el diseño del sistema interno, Incluyen las condiciones de activación, a menudo combinaciones de verdadero y falso, para todas las condiciones de entrada y las acciones resultantes para cada combinación.

La ventaja de la tabla de decisión es que crea combinaciones que de otro modo no se habrían encontrado.

2.13.2.5 Pruebas de transición de estado

Un sistema puede mostrar respuestas diferentes en función de sus condiciones

actuales o su estado. Estos aspectos del sistema pueden mostrarse mediante el diagrama de transición de estados.

El objeto de estas pruebas es partir de un estado inicial y recorrer los diferentes estados mediante eventos que desencadenan cambios de estado.

Una tabla de estado muestra la relación entre los estados y las entradas, y eventualmente puede poner de manifiesto posibles transiciones no válidas.

Las pruebas de transición de estados deben ejecutar todas las funciones de un estado al menos una vez. Pueden diseñarse pruebas para cubrir una secuencia típica de estados, cubrir todos los estados, cubrir todos los estados, ejercitar secuencias específicas de transiciones o probar transiciones válidas.

2.13.2.6 Pruebas de caso de uso

Estas pruebas son las que se derivan de los casos de uso. Un caso de uso expresa todas las maneras de utilizar un sistema para alcanzar una meta particular para un usuario. En conjunto, los casos de uso le proporcionan todos los caminos útiles de usar el sistema e ilustran el valor que este provee.

Los casos de uso dejan claro lo que hará un sistema y, por omisión intencional, lo que no hará. Estos posibilitan una visión efectiva, el manejo del alcance y el desarrollo incremental de sistemas de cualquier tipo y cualquier tamaño.

Los casos de prueba derivados de los casos de uso resultan muy útiles a la hora de descubrir defectos en todos los caminos útiles durante el uso real del sistema.

Los casos de uso son de gran utilidad para diseñar las pruebas de aceptación con la participación del cliente/usuario.

CAPÍTULO III

MARCO METODOLÓGICO

3.1 INTRODUCCIÓN

Esta sección del documento describe la implementación del marco metodológica de desarrollo UWE aplicado en el presente proyecto “Sistema Web de Gestión De Historial Clínico Veterinario y Control de Adopciones”, teniendo en cuenta todo el marco aplicativo que incluye costos, seguridad calidad, etc.

3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL

La Clínica Veterinaria U.P.E.A. brinda diferentes servicios para las mascotas y otros en la urbe alteña, la Clínica se ubica en la zona Villa Esperanza, en un espacio reducido dentro de la Universidad Pública de El Alto.

La Clínica actualmente en sus servicios realiza un registro de las mascotas que visitan la clínica para su atención correspondiente de forma manual, se registra los datos del propietario y de la mascota, posteriormente una vez realizado la revisión médica se procede a llenar el diagnóstico y el tratamiento adecuado. Una vez terminado el registro se procede a archivar el registro, en carpetas físicas.

Cabe mencionar que dentro del registro del historial no se tiene un identificador único tanto para los propietarios como de las mascotas, por lo cual al momento de buscar un historial de una mascota que ya haya visitado la clínica, se hace una tarea difícil, por lo cual se hace el llenado de otra ficha de historial, incurriendo en duplicidad de datos.

Los diferentes servicios son promocionados a través de banners en instalaciones de la Clínica, y en redes sociales, también se pudo evidenciar que muchos de los clientes de la clínica buscan alternativas para informar y recibir información de mascotas en adopción.

3.3 ANÁLISIS DE REQUERIMIENTOS

3.3.1 Obtención de Requerimiento

Los requerimientos son una parte muy importante en este capítulo, ya que a partir de este se tiene el pilar principal para las siguientes fases de la metodología UWE, la obtención de requerimientos se obtuvo de acuerdo a la siguiente tabla:

Tabla 14. Obtención de Requerimientos

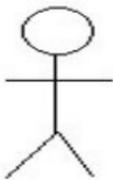
Tareas	Descripción
Entrevista	<ul style="list-style-type: none"> ➤ Se realizó la entrevista con la Clínica Veterinaria U.P.E.A. ➤ Director de La Clínica.
Observaciones	Dificultad en la búsqueda de historial clínico y los tratamientos de las mascotas que visitaron la Clínica.
Documentación	La Clínica proporcionó la información de los historiales clínicos, tanto el modelo del documento como los datos clínicos.

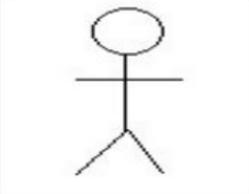
Fuente: Elaboración propia

3.3.2 Identificación de Actores

Los actores que intervienen en la clínica son parte muy importante del proyecto, ya que estos están en interacción del manejo constante del historial clínico, los actores y las funciones que cumplen se muestra a continuación:

Tabla 15. Identificaciones de actores

Actor	Descripción
Administrador 	Director Es el encargado de la administración general de la Clínica, además de ser el Médico de especialidad de la Clínica encargado de la supervisión de los historiales clínicos y tratamientos.
Usuario 	Pasante – Beca Trabajo Estudiante que realiza sus prácticas profesionales, se encarga de colaborar en las actividades dentro de la clínica de acuerdo a instrucciones del director de la Clínica. También encargado de los registros de los historiales clínicos y tratamientos.

Invitado 	Personas que acuden a la clínica, personas interesadas en los servicios y otros de la clínica.
---	--

Fuente: Elaboración propia

3.3.3 Identificación de Requerimientos

Es muy importante obtener los requerimientos correctos, sin embargo estos deben ser clasificados en requerimientos funcionales y no funcionales.

3.3.3.1 Requerimientos Funcionales

Son los requerimientos que cumplen las reglas del negocio y sus características

Tabla 16. Análisis de Requerimientos Funcionales

Rol	Descripción	Función
R1-1	Autenticación	<ul style="list-style-type: none"> ➤ Se accederá el sistema, verificando el acceso a usuarios permitidos. ➤ Se establecerá los permisos de acuerdo al rol de usuario.
R1-2	Administración de Usuarios	<ul style="list-style-type: none"> ➤ El sistema podrá crear, ver, editar, eliminar y buscar a los usuarios registrados.
R1-3	Administración de Propietarios	<ul style="list-style-type: none"> ➤ El sistema podrá crear, ver, editar, eliminar y buscar a los propietarios registrados. ➤ El sistema podrá verificar si el propietario cuenta con otras mascotas atendidas en la clínica.
R1-4	Administración de Mascotas	<ul style="list-style-type: none"> ➤ El sistema podrá crear, ver, editar, eliminar y buscar a las mascotas registradas.
R1-5	Administración de Diagnósticos	<ul style="list-style-type: none"> ➤ El sistema podrá crear, ver, editar y eliminar, el diagnóstico correspondiente a la mascota. ➤ El sistema podrá buscar el diagnóstico de acuerdo a la mascota

		a la cual le pertenezca.
R1-6	Historial Clínico	➤ El sistema mostrará toda la información del historial clínico de acuerdo a la mascota a la cual le pertenezca.
R1-7	Tratamiento Clínico	El sistema mostrará toda la información del tratamiento clínico de acuerdo a la mascota a la cual le pertenezca.
R1-8	Administración de Adopciones	➤ El sistema podrá crear, ver, editar, eliminar y buscar a las publicaciones de adopciones registrados.
R1-9	Reporte de Historial Clínico	➤ El sistema podrá generar un reporte del historial clínico de acuerdo a la mascota que lo requiera.
R1-10	Reporte de Tratamiento Clínico	➤ El sistema podrá generar un reporte del tratamiento clínico de acuerdo a la mascota que lo requiera.
R1-11	Datos Estadísticos	➤ El sistema podrá mostrar los datos estadísticos y gráficos de la clínica

Fuente: Elaboración propia

3.3.3.2 Requerimientos No Funcionales

Son los requerimientos del rendimiento de los procesos o del sistema.

Tabla 17. Análisis de Requerimientos No Funcionales

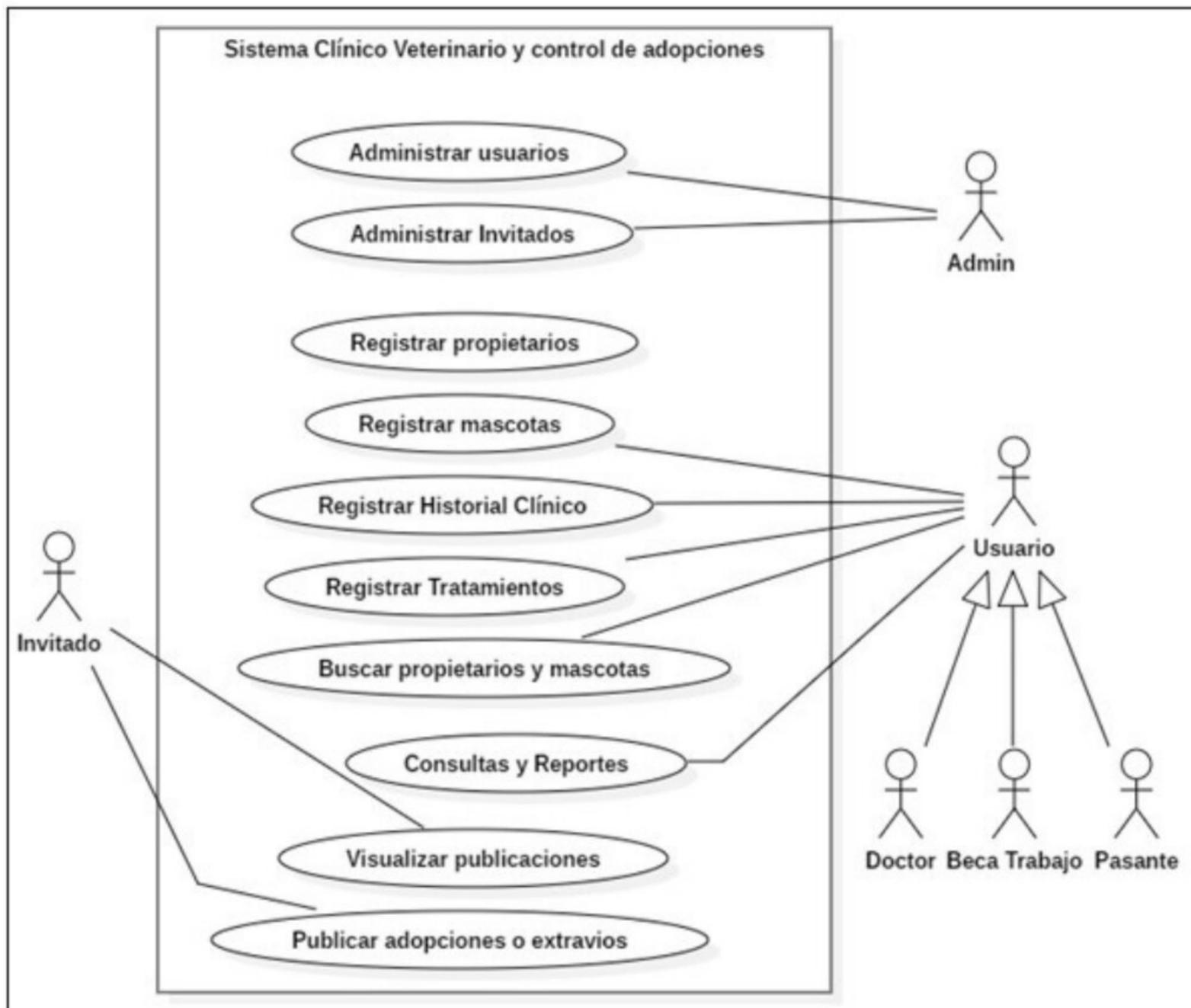
Rol	Función
R2-1	El sistema de visualizarse y funcionar correctamente en los diferentes navegadores.
R2-2	La interfaz gráfica debe ser fácil y entendible para el usuario.
R2-3	Toda transacción realizada debe responder al usuario en la brevedad que lo requiera.
R2-4	Soporte y mantenimiento, para brindar disponibilidad del sistema en la red.

Fuente: Elaboración propia

3.3.4 Diagrama de Caso de Uso en General

A continuación se muestra el modelado, de la interacción de los actores con el sistema de manera general.

Figura 19. Casos de Uso del Sistema

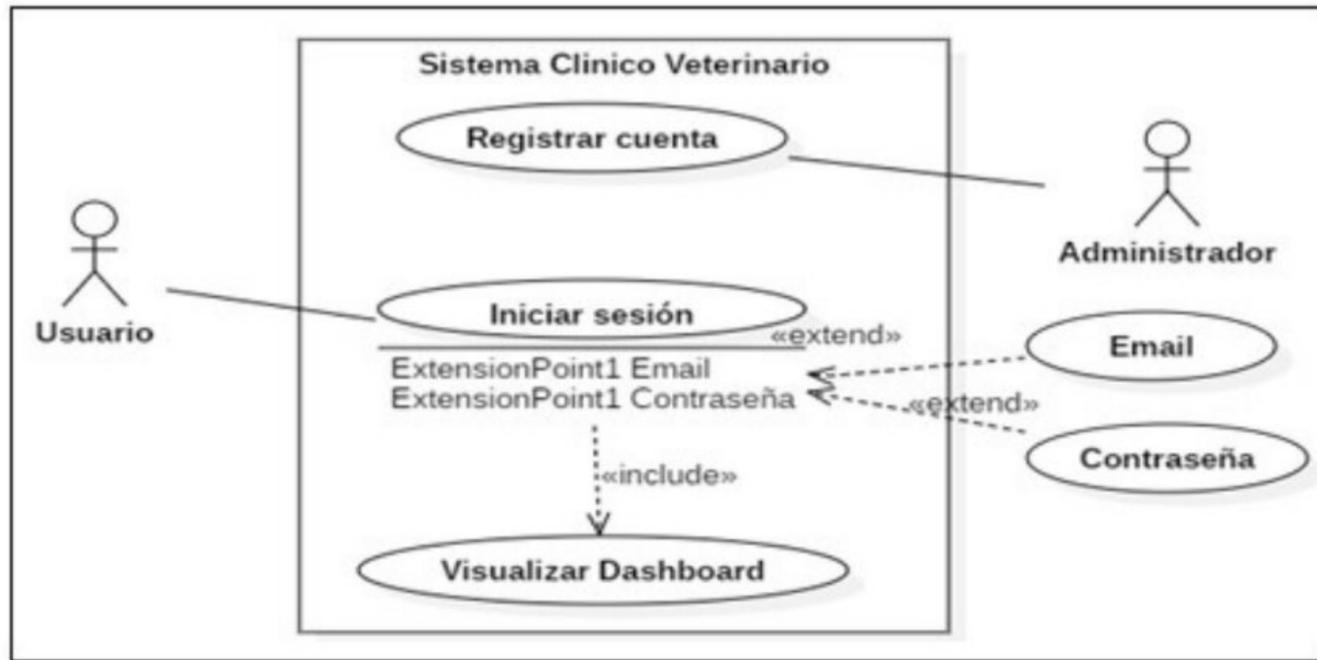


Fuente: Elaboración propia

➤ Diagrama de Caso de Uso Ingresar del Sistema Clínico

A continuación se muestra caso de uso para ingresar al sistema, en la siguiente figura.

Figura 20. Caso de Uso Ingresar al Sistema Clínico



Fuente: Elaboración propia

Tabla 18. Descripción de Caso de Uso: Ingresar al Sistema Clínico

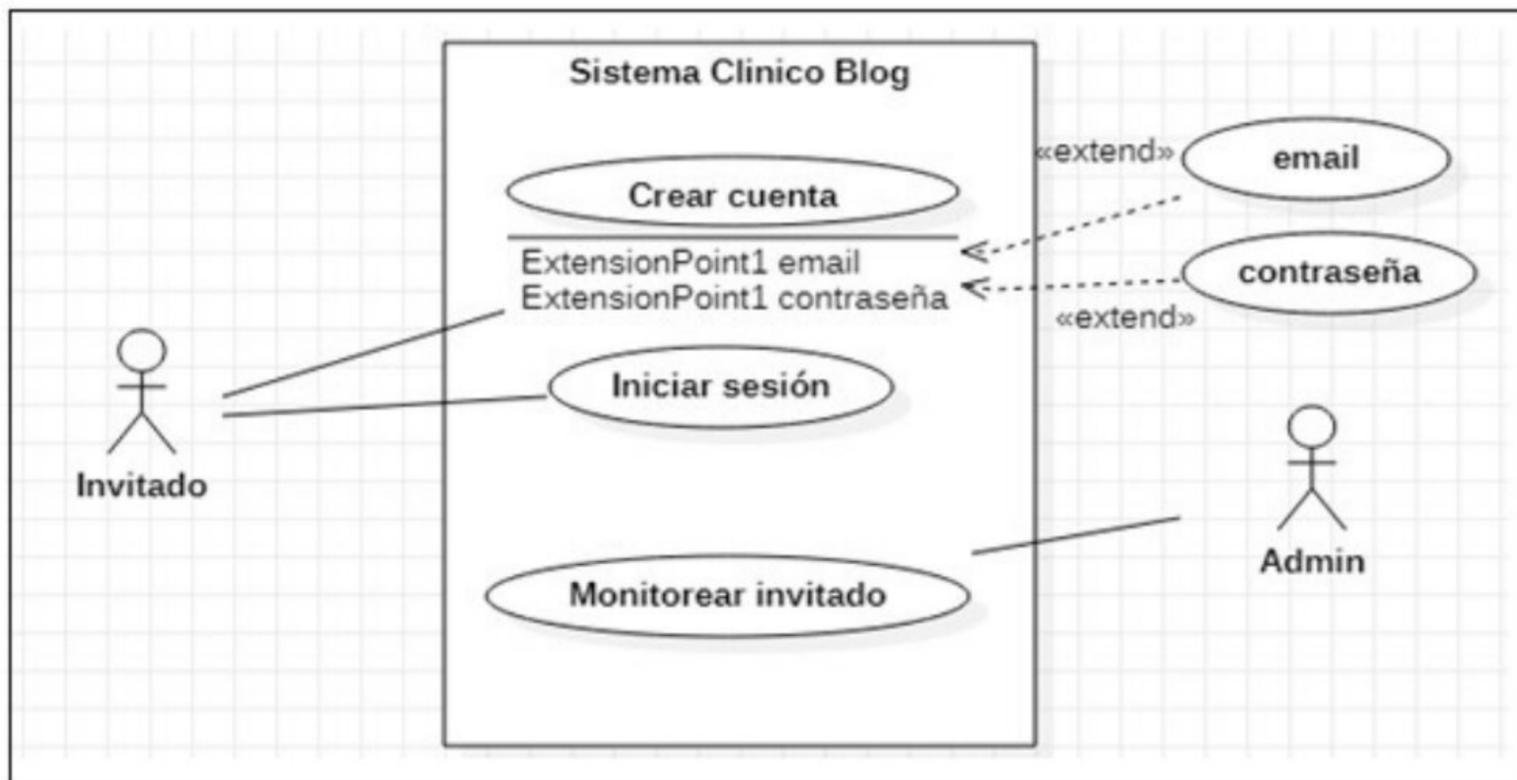
Caso De Uso:	Ingresar al Sistema Clínico
Requisito De Usabilidad	Administrador, Usuario
Precondiciones	El usuario debe contar con acceso a internet
Descripción	El administrador registra sus datos. El administrador monitorea las publicaciones de este.
Excepciones	No haya sistemas para el registro
Postcondiciones	El usuario ha ingresado al dashboard, habilitado para que pueda registrar datos del historial clínico.

Fuente: Elaboración propia

➤ **Diagrama de Caso de Uso Ingresar al Sistema Publicaciones**

A continuación se muestra el caso de uso para el ingreso de las publicaciones de las mascotas de acuerdo a la siguiente figura.

Figura 21. Caso de Uso: Ingresar al Sistema Publicaciones



Fuente: Elaboración propia

Tabla 19. Descripción Caso de Uso: Ingresar al Sistema Publicaciones

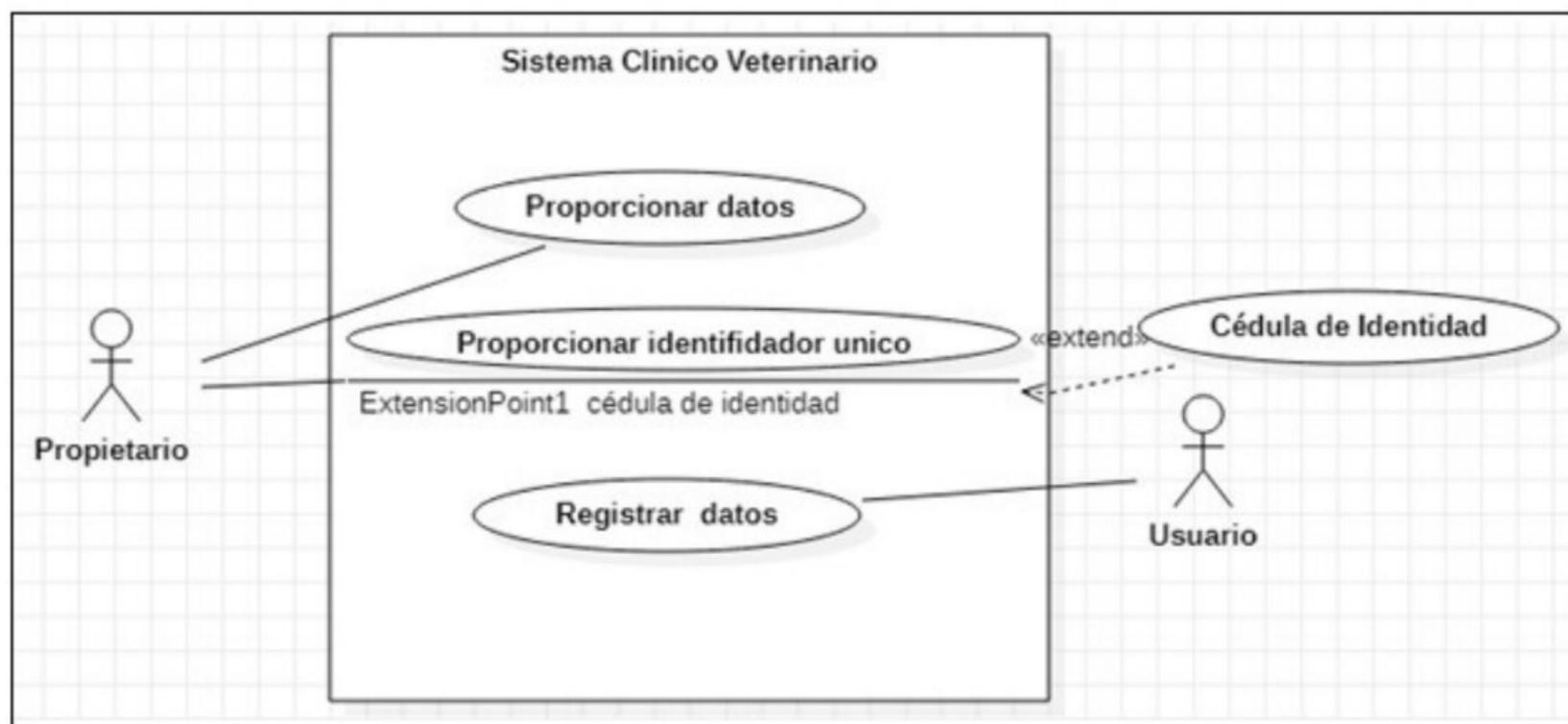
Caso De Uso:	Ingresar al Sistema Publicaciones
Requisito De Usabilidad	administrador, invitado
Precondiciones	El invitado debe contar con acceso a internet
Descripción	El invitado registra sus datos. El administrador monitorea las publicaciones de este.
Excepciones	no haya sistemas para el registro
Postcondiciones	El invitado ha ingresado al dashboard, habilitado para que pueda publicar.

Fuente: Elaboración propia

➤ **Diagrama de Caso de Uso Registrar Propietario**

A continuación se muestra el caso de uso para el registro de propietarios de acuerdo a la siguiente figura.

Figura 22. Caso de Uso: Registrar Propietario



Fuente: *Elaboración propia*

Tabla 20. Descripción Caso de Uso Registrar Propietario

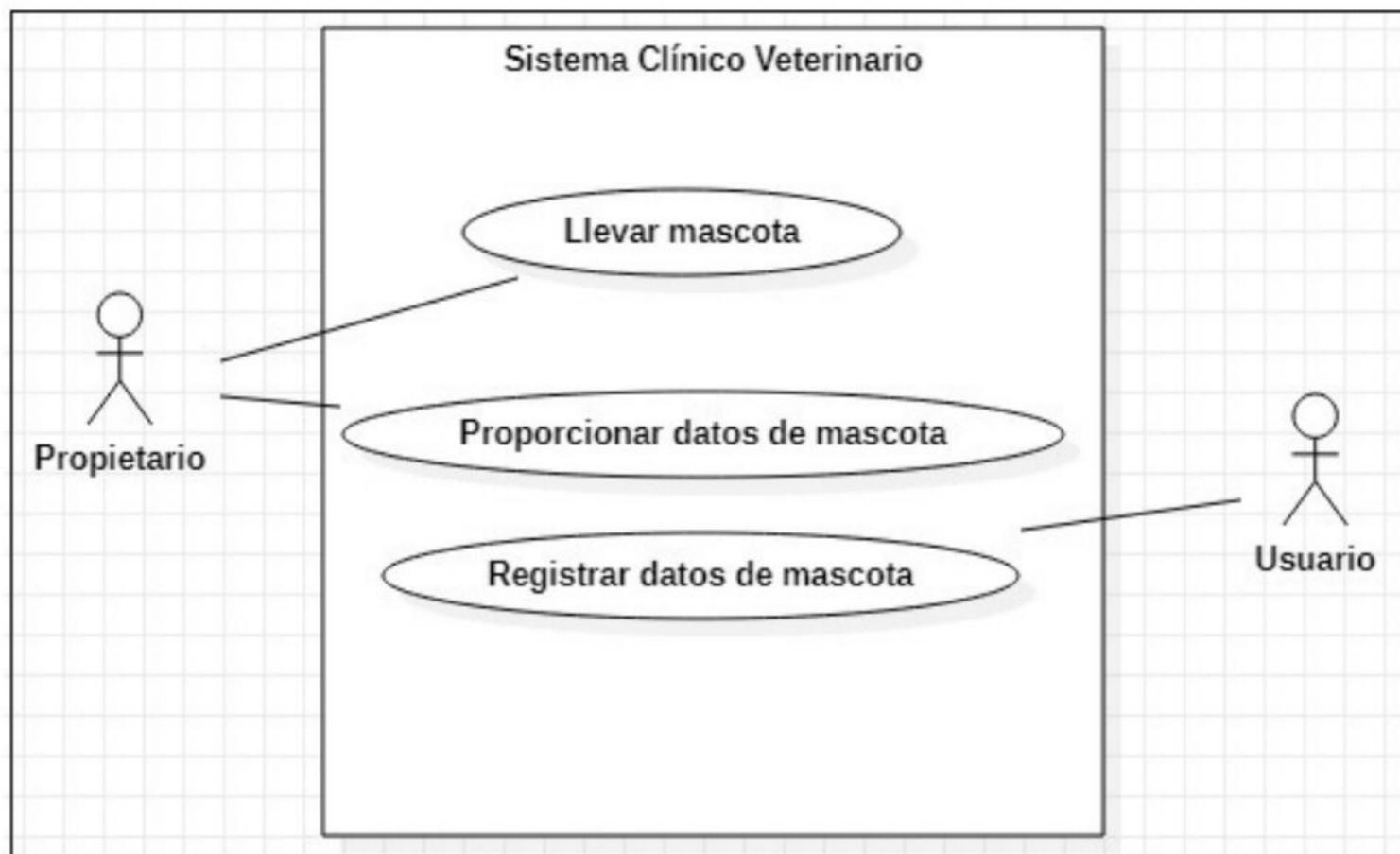
Caso De Uso:	Registrar Propietario
Actores	Usuario, propietario
Resumen	El administrador registra datos del propietario
Requisitos De Usabilidad	El administrador puede dar de baja a los usuarios que sean necesarios
Precondiciones	El propietario debe portar la cedula de identidad
Escenario Principal	El usuario pide los datos del propietario. El sistema verifica que no haya duplicidad de cedula de identidad
Excepciones	El sistema no está habilitado
Postcondiciones	El usuario está registrado

Fuente: *Elaboración propia*

➤ **Diagrama de Caso de Uso Registrar Mascota**

A continuación se muestra el caso de uso para el registro de las mascotas de acuerdo a la siguiente figura.

Figura 23. Caso de Uso: Registrar Mascota



Fuente: Elaboración propia

Tabla 21. Descripción Caso de Uso: Registrar Mascota

Caso De Uso:	Registrar mascota
Requisito De Usabilidad	Usuario, propietario
Precondiciones	El cliente debe aportar la documentación correspondiente para el registro.
Descripción	<p>El cliente solicita atención médica para su mascota</p> <p>El cliente proporciona los datos personales al usuario</p> <p>El usuario solicita el formulario de</p>

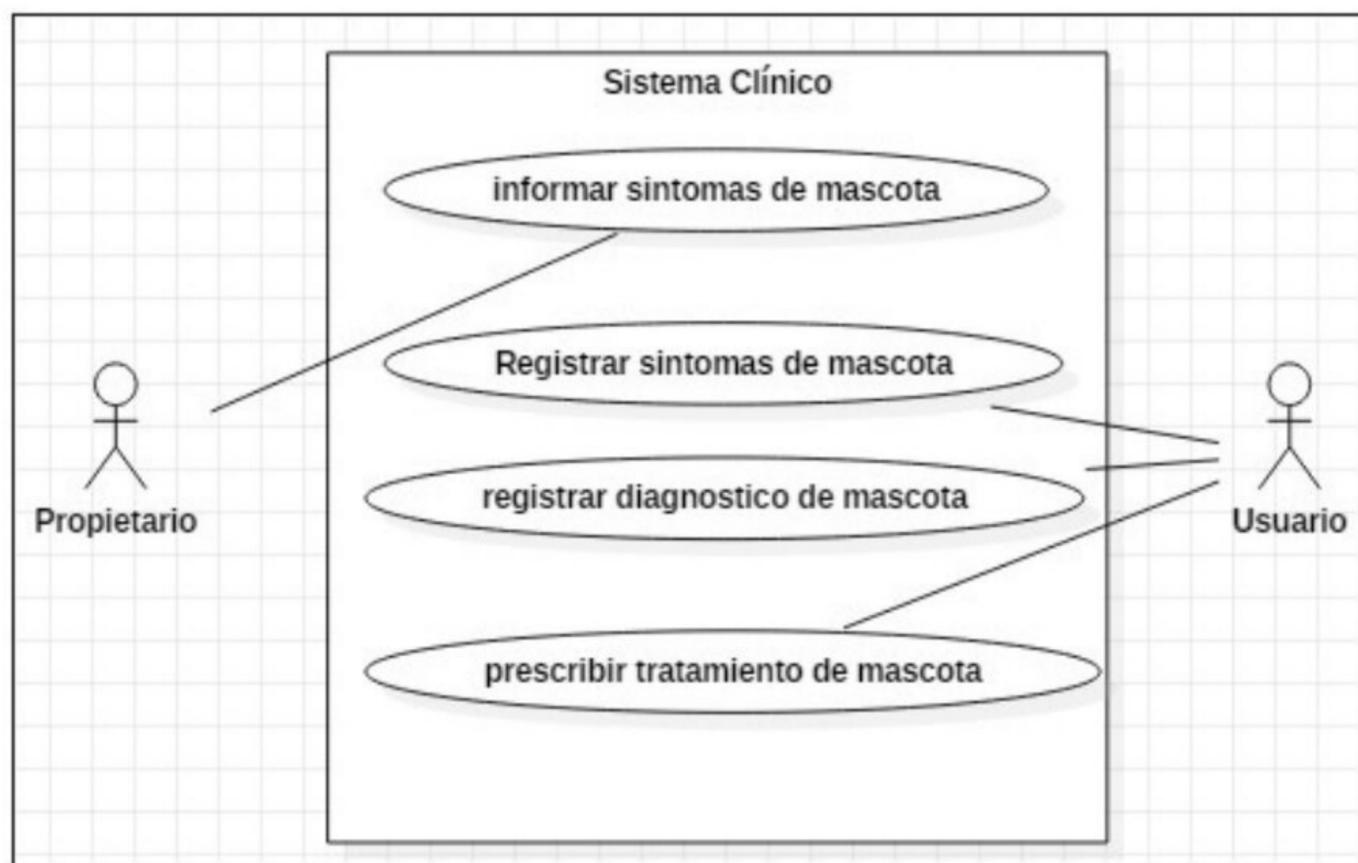
	registro El sistema redirige al usuario a un formulario con los datos necesarios a rellenar por el usuario El sistema almacena la información
Excepciones	no haya sistemas para el registro
Postcondiciones	el cliente ya ha efectuado su registro efectivamente

Fuente: Elaboración propia

➤ **Diagrama de Caso de Uso Registrar Diagnóstico**

Podemos observar el caso de uso para el registrar el diagnostico de las mascotas de acuerdo a la siguiente figura.

Figura 24. Caso de Uso: Registrar Diagnóstico



Fuente: Elaboración propia

Tabla 22. Descripción Caso de Uso: Registrar Diagnóstico

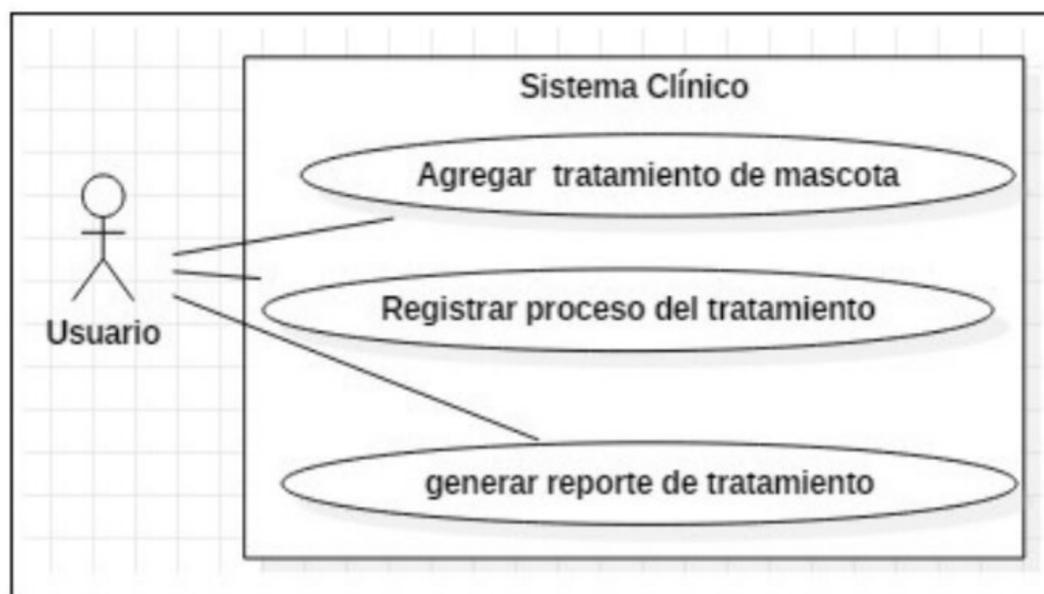
Caso De Uso:	Registrar Diagnóstico
Requisito De Usabilidad	Usuario, Propietario
Precondiciones	El propietario debe de llevar a la mascota
Descripción	El propietario debe de indicar todos los síntomas para registrarlo. El usuario debe revisar y registrar la revisión. De acuerdo a la revisión se tratara de dar un tratamiento.
Excepciones	no haya sistemas para el registro
Postcondiciones	El diagnóstico puede ser actualizado de acuerdo al tratamiento de la mascota.

Fuente: Elaboración propia

➤ **Diagrama de Caso de Uso Registrar Tratamiento**

A continuación se muestra el caso de uso para registrar los tratamientos de las mascotas de acuerdo a la siguiente figura.

Figura 25. Caso de Uso: Registrar Tratamiento



Fuente: Elaboración propia

Tabla 23. Descripción Caso de Uso: Registrar Tratamiento

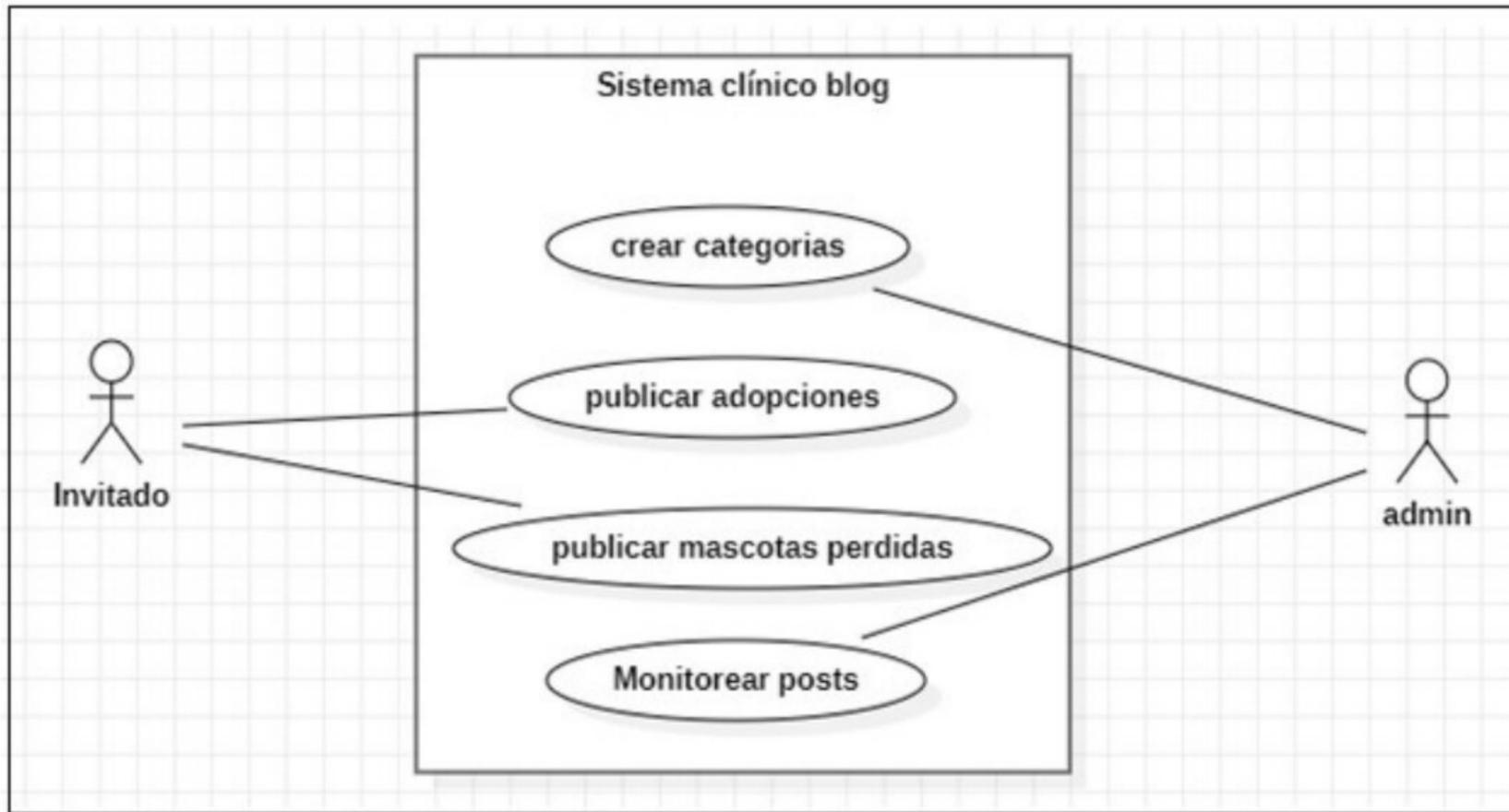
Caso De Uso:	Registrar Tratamiento
Requisito De Usabilidad	Usuario
Precondiciones	El propietario debe estar registrado en el sistema. La mascota debe contar con registro en el sistema.
Descripción	El usuario debe revisar y registrar el tratamiento. El tratamiento se lleva a cabo en varios días, debe actualizarse los datos. Se muestra la vista de los tratamientos de la mascota. Se genera el reporte de tratamiento.
Excepciones	No haya sistema para el registro.
Postcondiciones	El tratamiento puede ser actualizado de acuerdo a la evolución recuperatoria de la mascota.

Fuente: Elaboración propia

➤ **Diagrama de Caso de Uso Publicar Adopción**

A continuación se muestra el caso de para la publicación de adopciones de acuerdo a la siguiente figura:

Figura 26. Caso de Uso: Publicar Adopción



Fuente: Elaboración propia

Tabla 24. Descripción Caso de Uso: Publicar Adopción

Caso De Uso:	Publicar Adopción
Requisito De Usabilidad	Invitado, Administrador
Precondiciones	El invitado debe registrarse
Descripción	El Administrador puede crear categorías y etiquetas. El invitado puede publicar alguna adopción o alguna publicación de extravió.
Excepciones	no haya sistemas para el registro
Postcondiciones	El administrador monitorea las publicaciones.

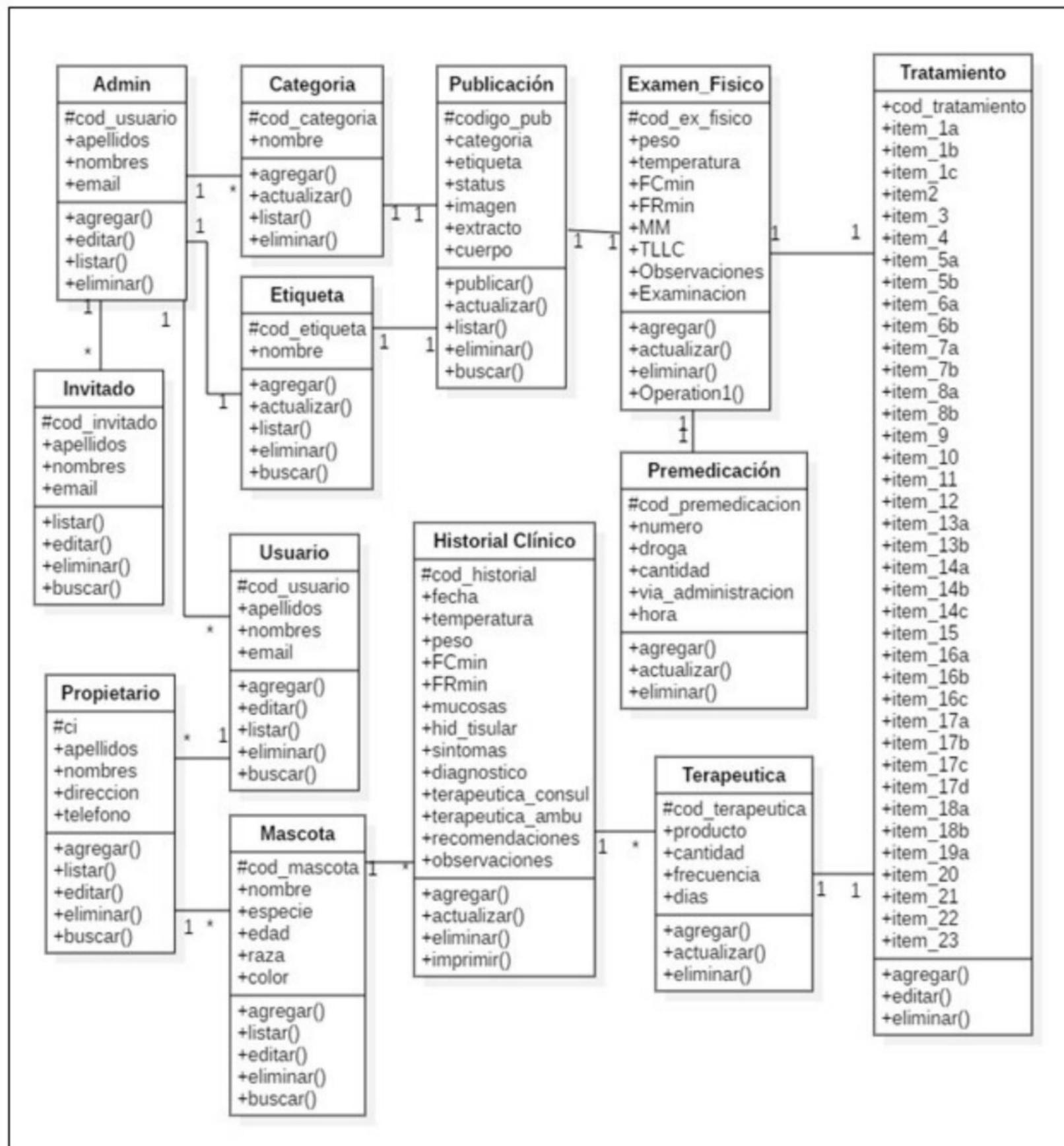
Fuente: Elaboración propia

3.4 MODELO DE CONTENIDO

3.4.1 Diagrama de Clases

A continuación se muestra el modelado, de la interacción de los actores con el sistema de manera general.

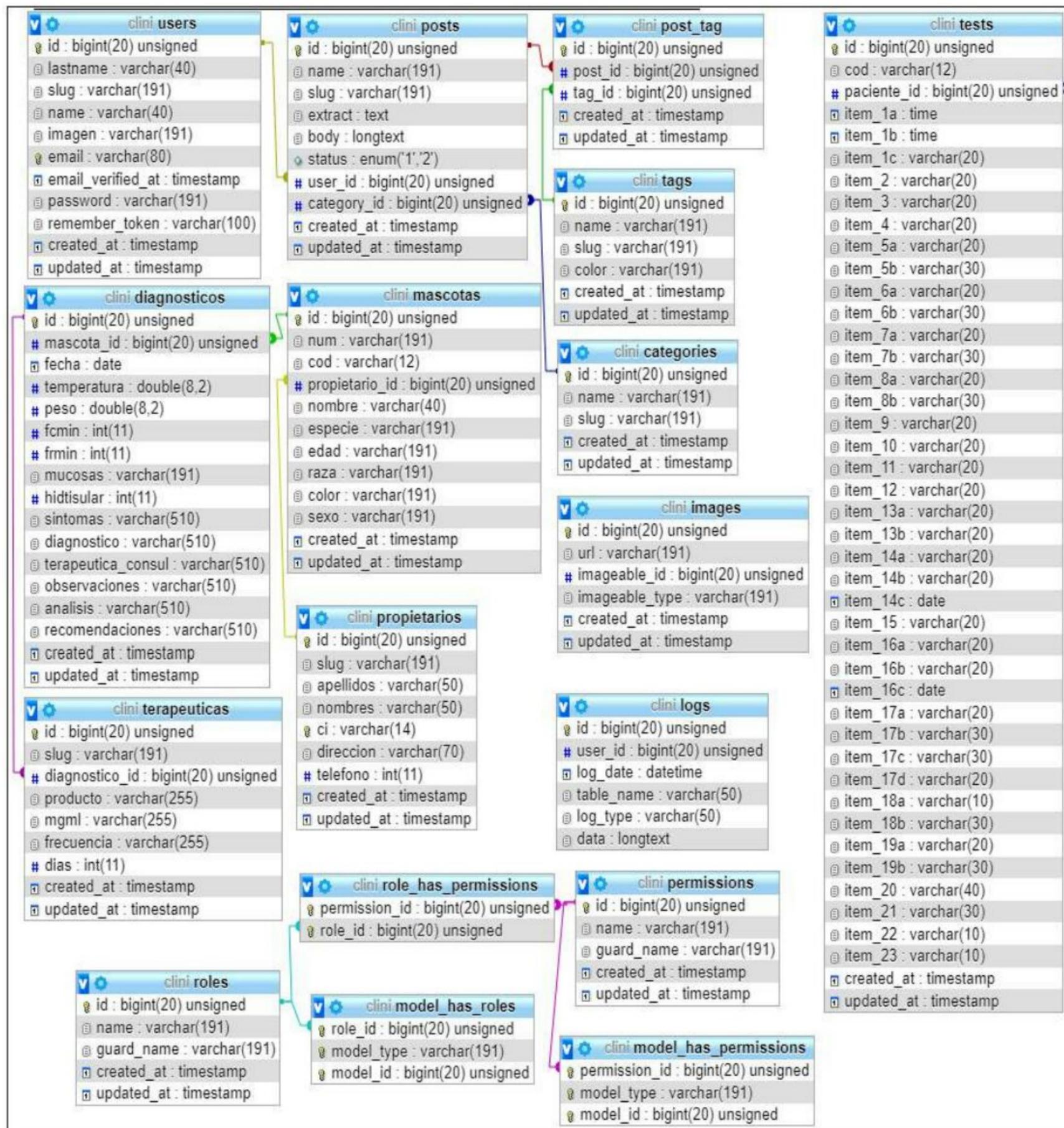
Figura 27. Diagrama de Clases



Fuente: Elaboración propia

A continuación se muestra el modelo entidad relación para el registro de las mascotas de acuerdo a la siguiente figura.

Figura 28. Modelo de Datos Relacional



Fuente: Elaboración propia

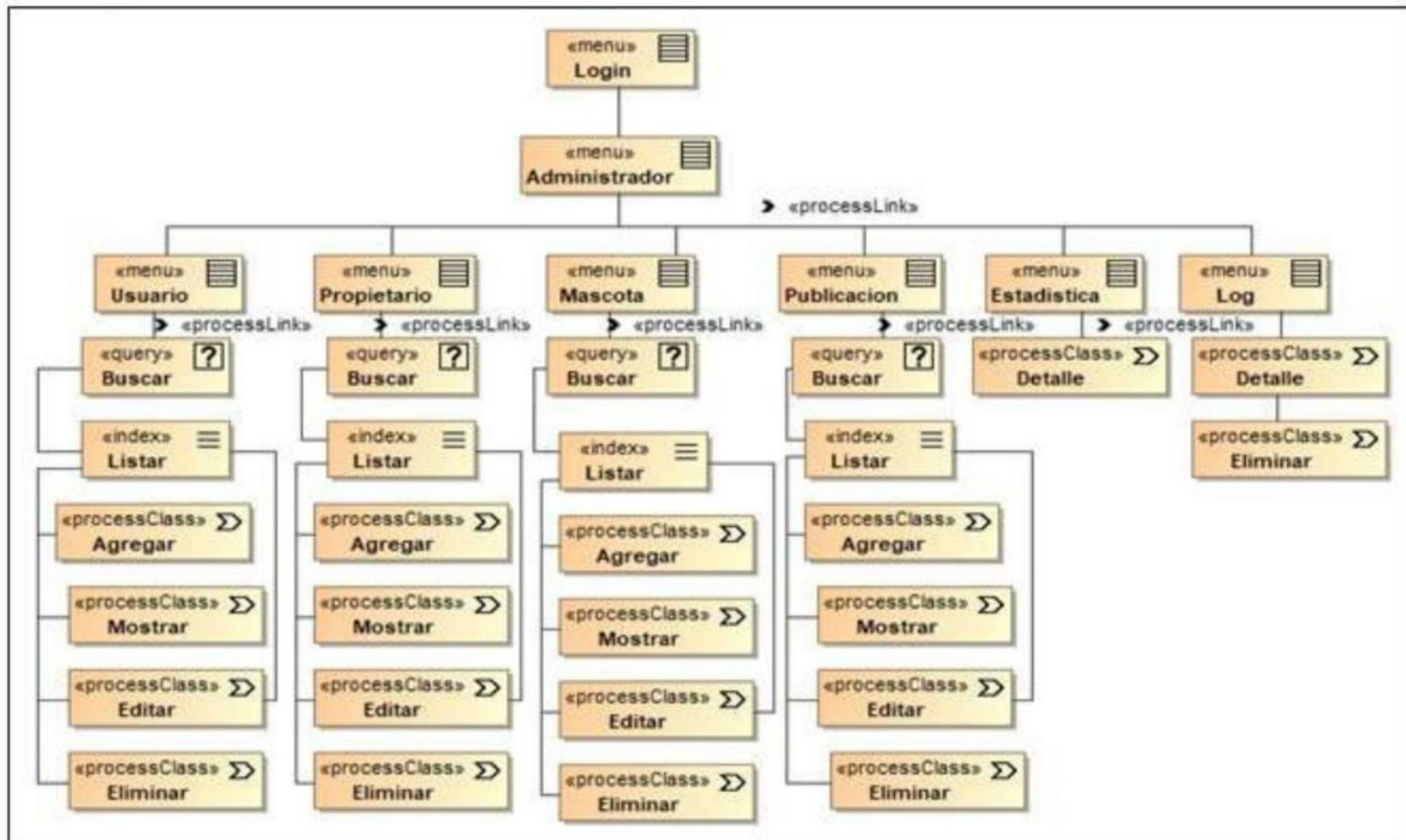
3.5 MODELO DE NAVEGACIÓN

Representa la navegación de los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas.

➤ Diagrama de Navegación: Administrador

A continuación se muestra el diagrama de navegación del administrador de acuerdo a la siguiente figura.

Figura 29. Diagrama de Navegación Administrador

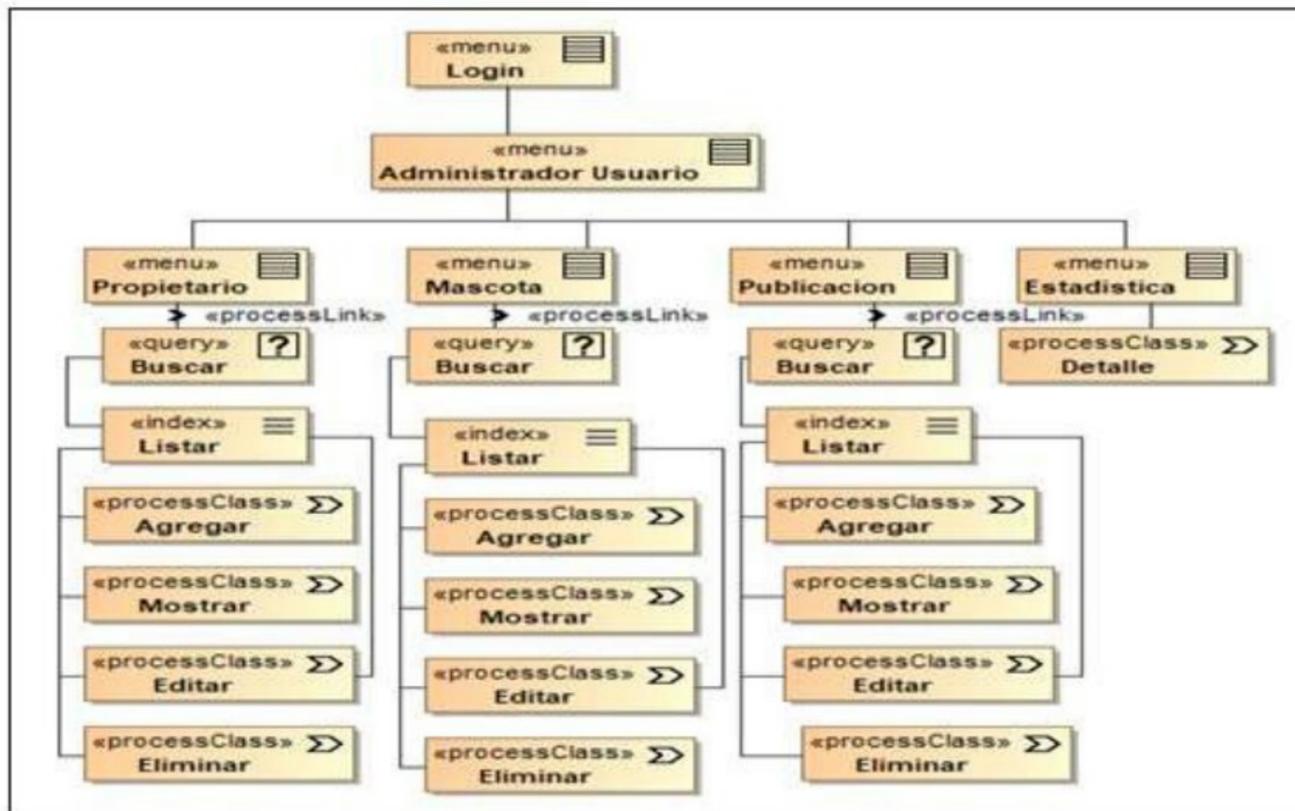


Fuente: Elaboración propia

➤ Diagrama de Navegación: Administrador Usuario

A continuación se muestra el diagrama de navegación para el usuario, de acuerdo a la siguiente figura.

Figura 30. Diagrama de Navegación: Usuario

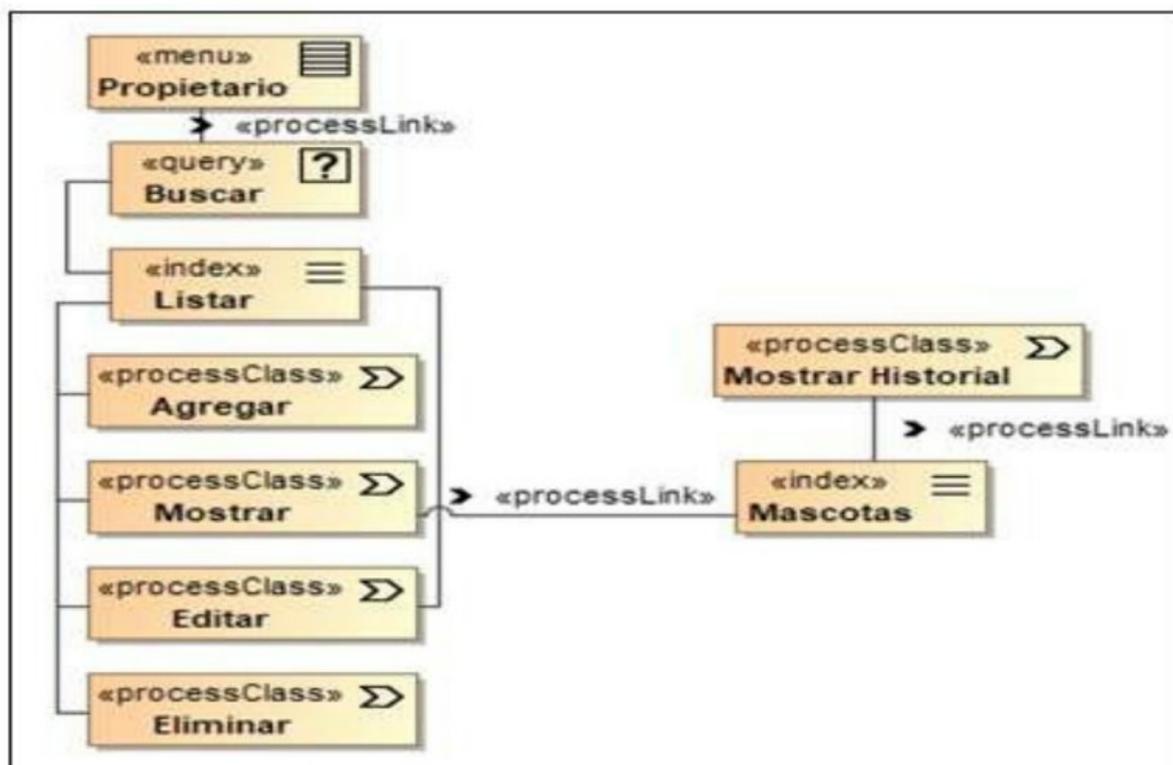


Fuente: Elaboración propia

➤ **Diagrama de Navegación: Mostrar Historial**

A continuación se muestra el diagrama de navegación para mostrar el historial clínico de acuerdo a la siguiente figura.

Figura 31. Diagrama de navegación: Mostrar Historial

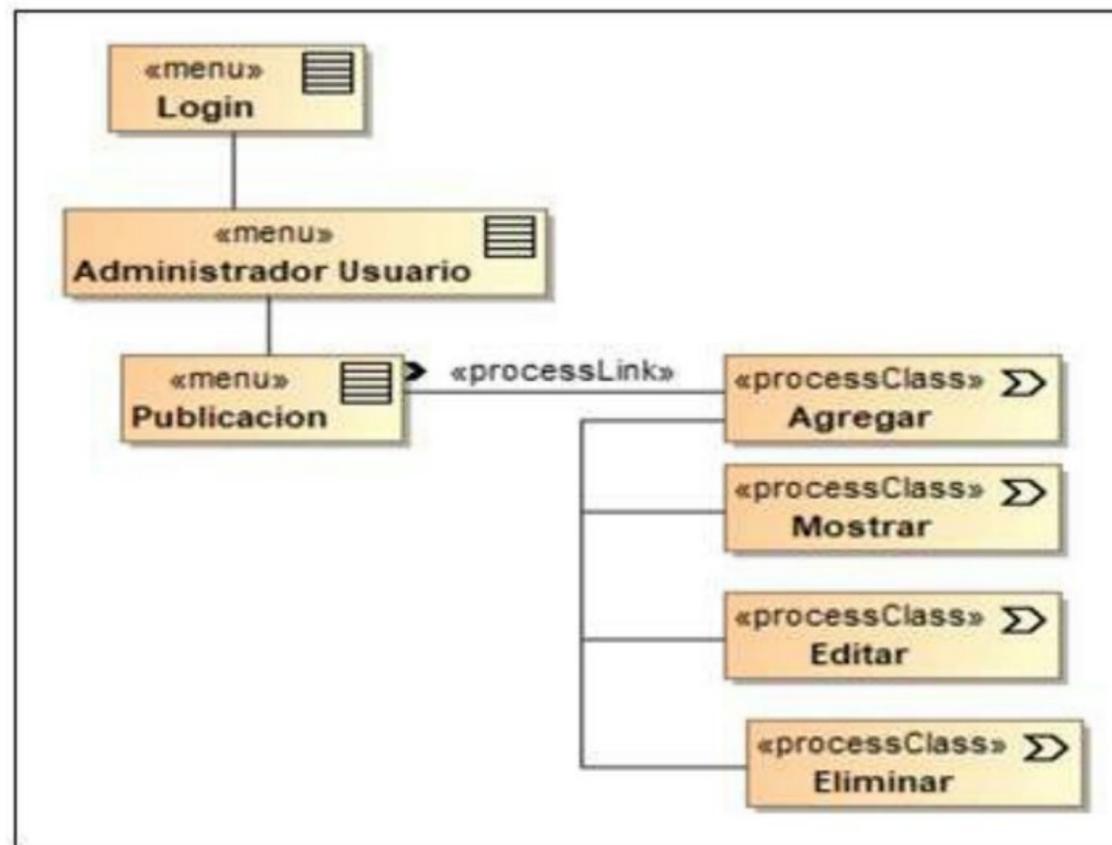


Fuente: Elaboración propia

➤ Diagrama de Navegación: Invitado

A continuación podemos observar el diagrama de navegación para el rol invitado de acuerdo a la siguiente figura.

Figura 32. Diagrama de navegación: Invitado



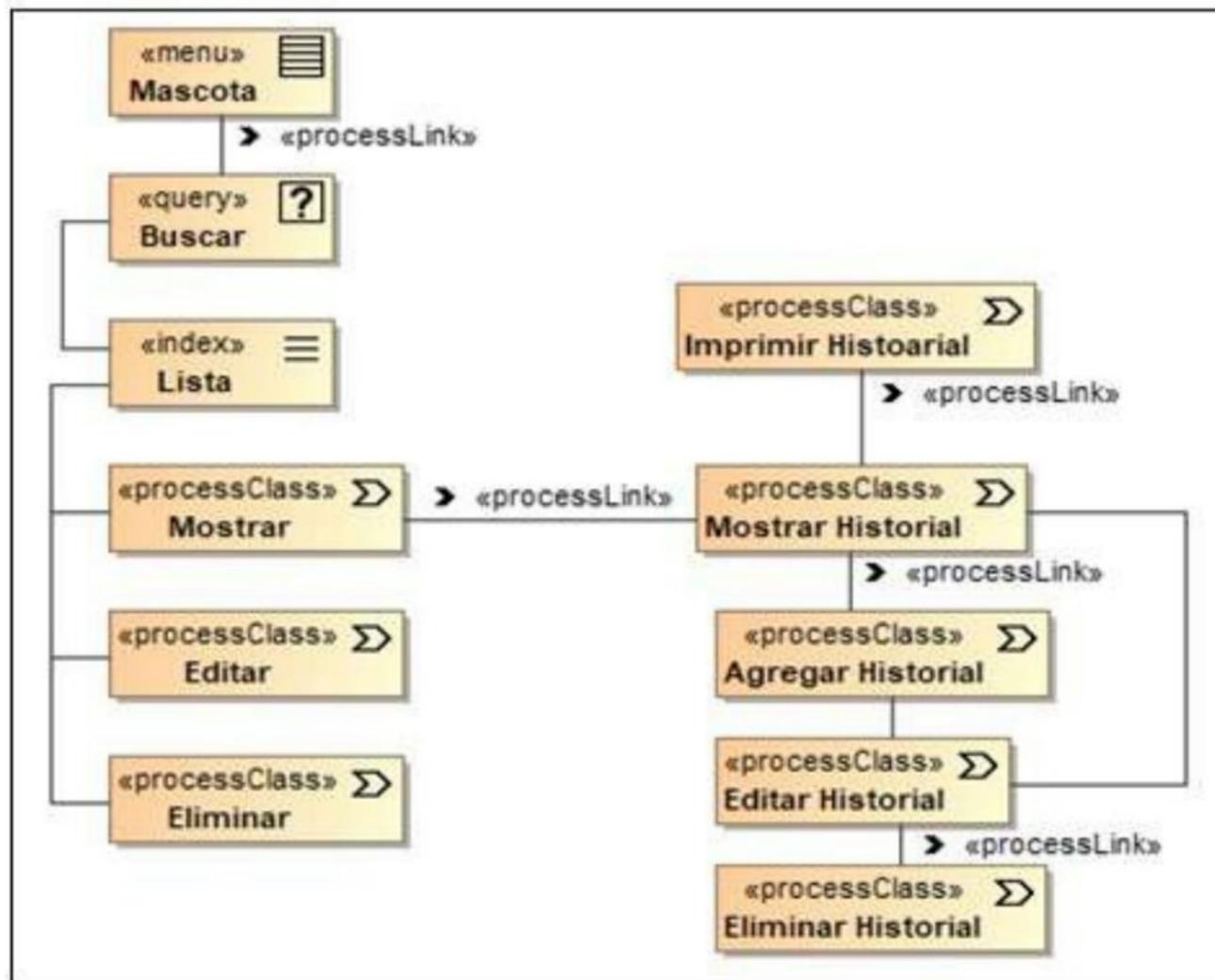
Fuente: Elaboración propia

➤ Diagrama de Navegación: Historial Clínico

En la siguiente figura podemos observar el diagrama de navegación para el caso de historial clínico, donde podemos observar el menú, el buscador, la lista de mascotas, donde cada registro puede mostrarse, editarse y eliminarse de acuerdo a la fila seleccionada.

De acuerdo a este diagrama también podemos navegar hacia el historial de una mascota seleccionada, de esta manera agregar, editar, eliminar e imprimir el historial clínico.

Figura 33. Diagrama de Navegación: Historial Clínico



Fuente: Elaboración propia

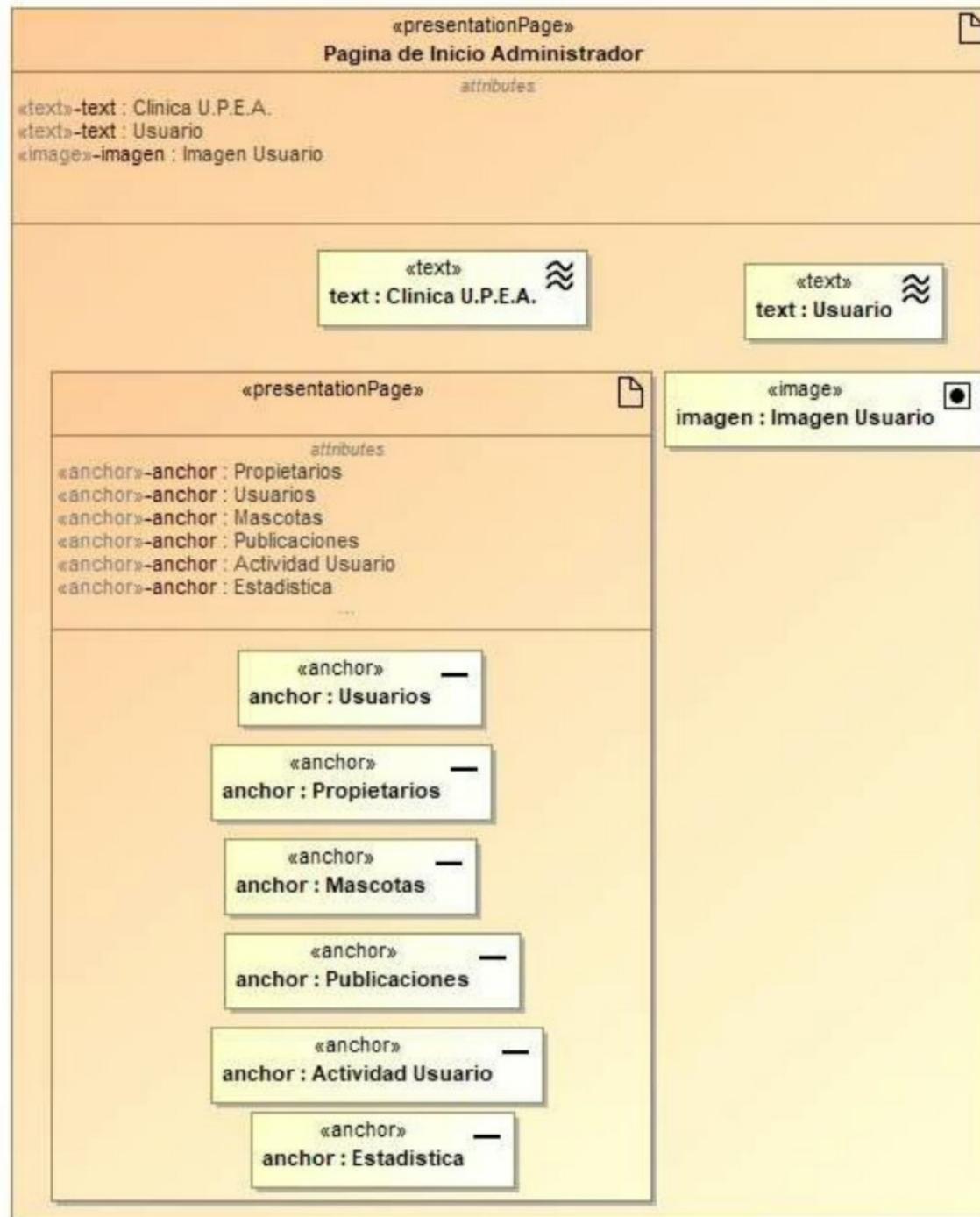
3.6 MODELO DE PRESENTACIÓN

Representa las interfaces de usuario por medio de vistas abstractas.

➤ Diagrama de presentación: Administrador

A continuación se muestra el diagrama de presentación del administrador de usuarios, en cual podemos observar los interfaces que tendrá esta ilustración de acuerdo a la siguiente figura.

Figura 34. Diagrama de Presentación: Administrador de usuarios

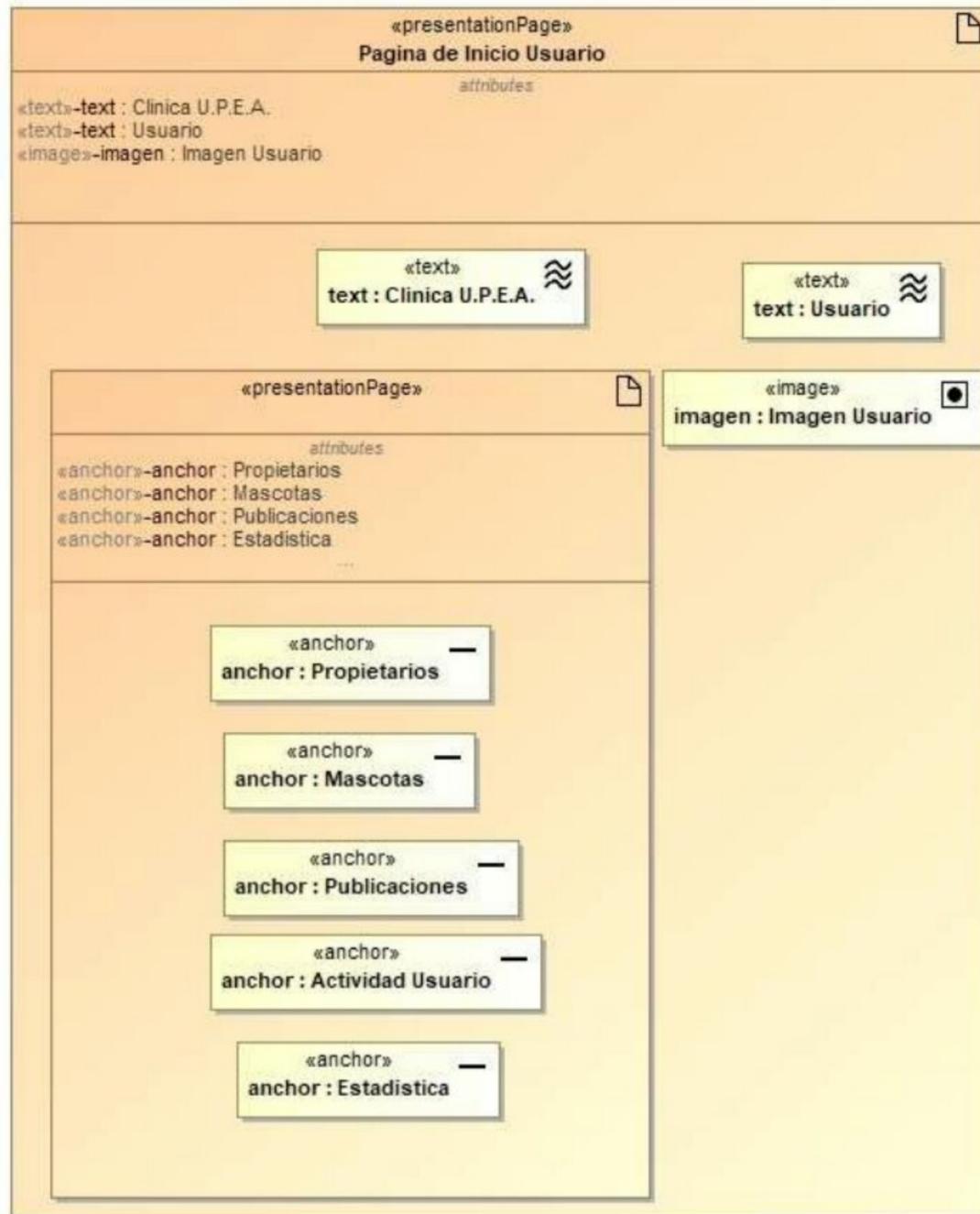


Fuente: Elaboración propia

➤ **Diagrama de Presentación: Usuario**

A continuación se muestra el diagrama de presentación para representar las interfaces de los usuarios de acuerdo a la siguiente figura.

Figura 35. Diagrama de Presentación: Usuario

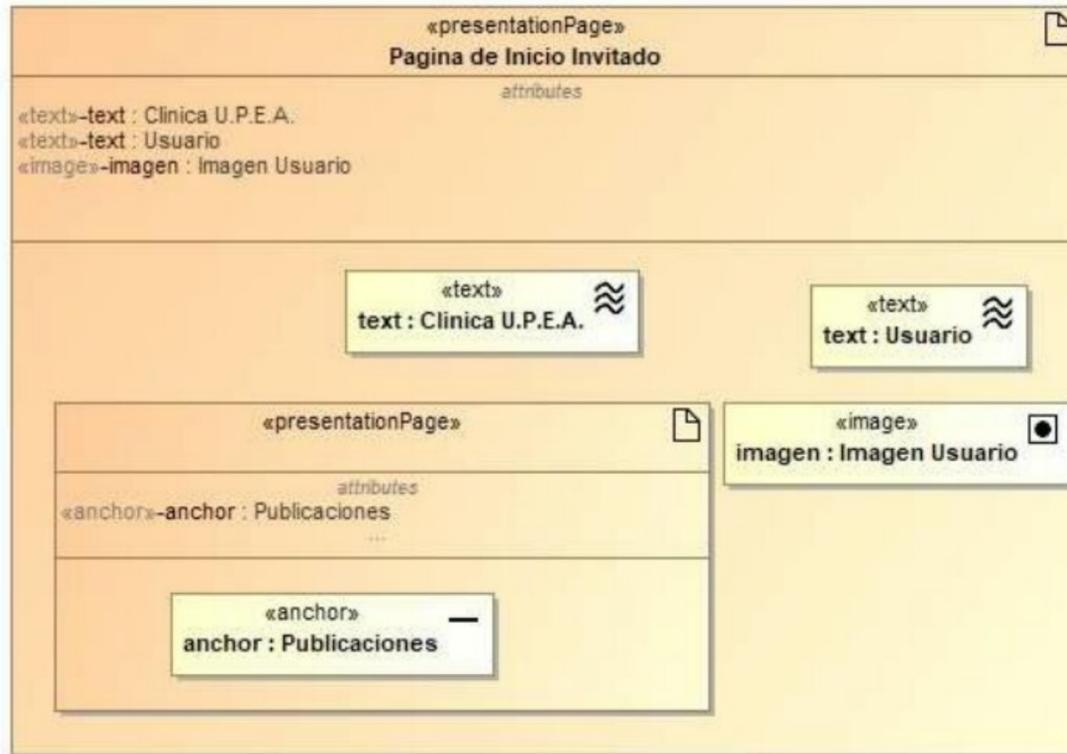


Fuente: Elaboración propia

➤ **Diagrama de Presentación: Invitado**

A continuación se muestra el diagrama de presentación para representar las interfaces de los invitados de acuerdo a la siguiente figura.

Figura 36. Diagrama de Presentación: Mostrar Invitado



Fuente: Elaboración propia

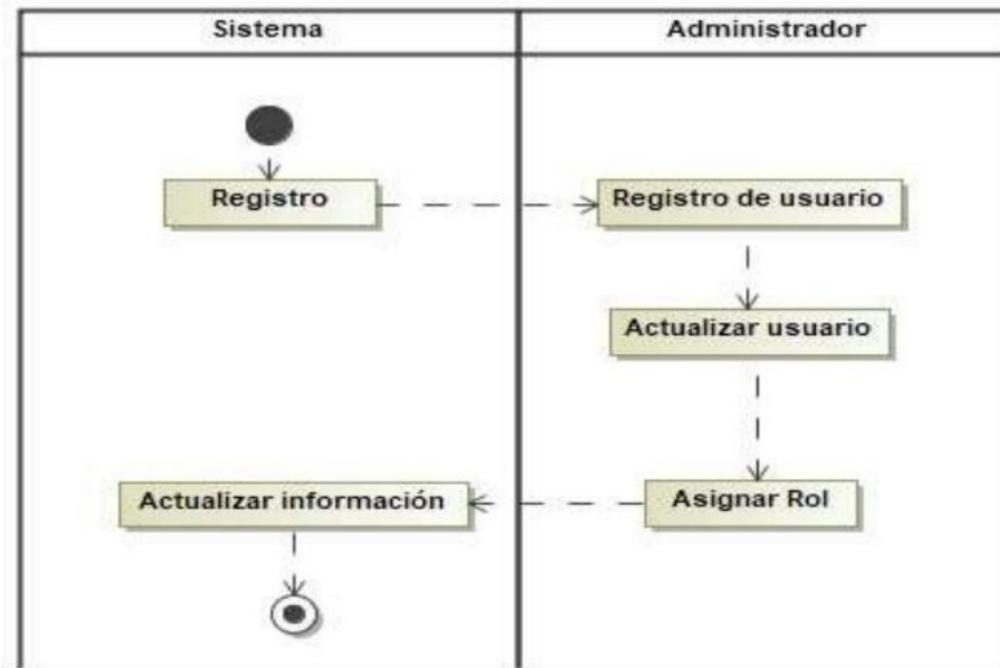
3.7 MODELO DE PROCESO

Representa los aspectos dinámicos de la aplicación Web y especifica funcionalidad cómo transacciones de actividades. Se modela mediante un diagrama de actividades de UML.

➤ **Diagrama de Actividades: Gestión de usuario**

A continuación se muestra el diagrama de actividades para representar las funcionalidades de los usuarios de acuerdo a la siguiente figura.

Figura 37. Diagrama de Actividades: Gestión de usuario

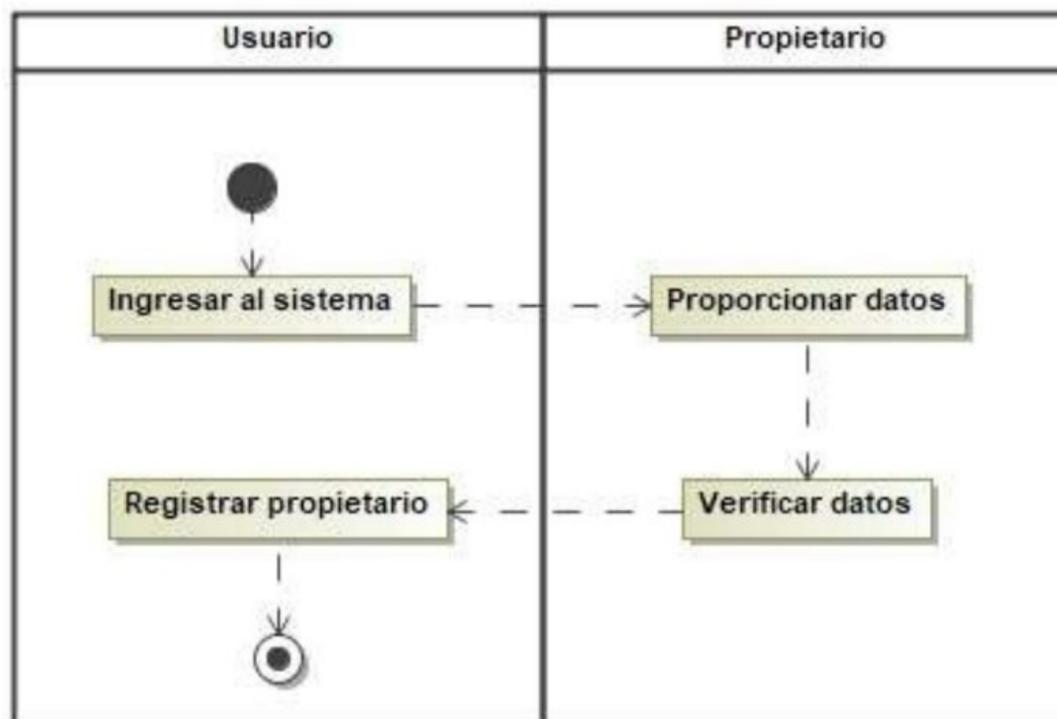


Fuente: Elaboración propia

➤ **Diagrama de Actividades: Registro propietario**

A continuación se muestra el diagrama de actividades para representar las funcionalidades del registro de propietarios, de acuerdo a la siguiente figura.

Figura 38. Diagrama de Actividades: Registro propietario

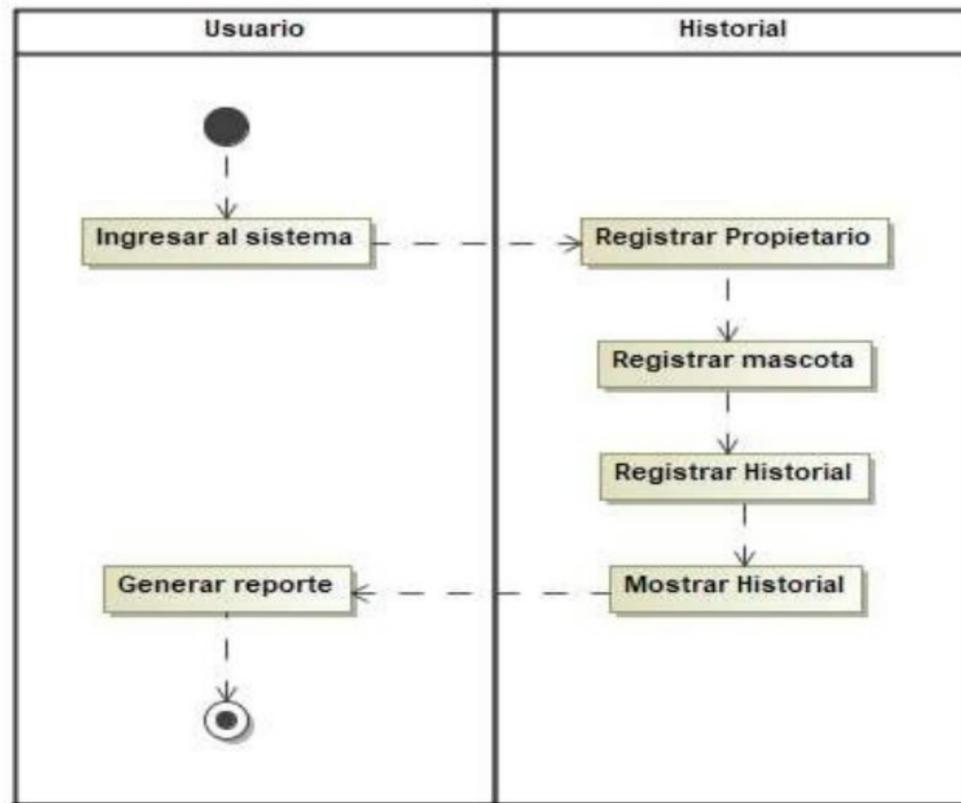


Fuente: Elaboración propia

➤ **Diagrama de Actividades: Gestión Historial**

A continuación se muestra el diagrama de actividades para representar las funcionalidades del historial clínico, de acuerdo a la siguiente figura.

Figura 39. Diagrama de Actividades: Gestión Historial

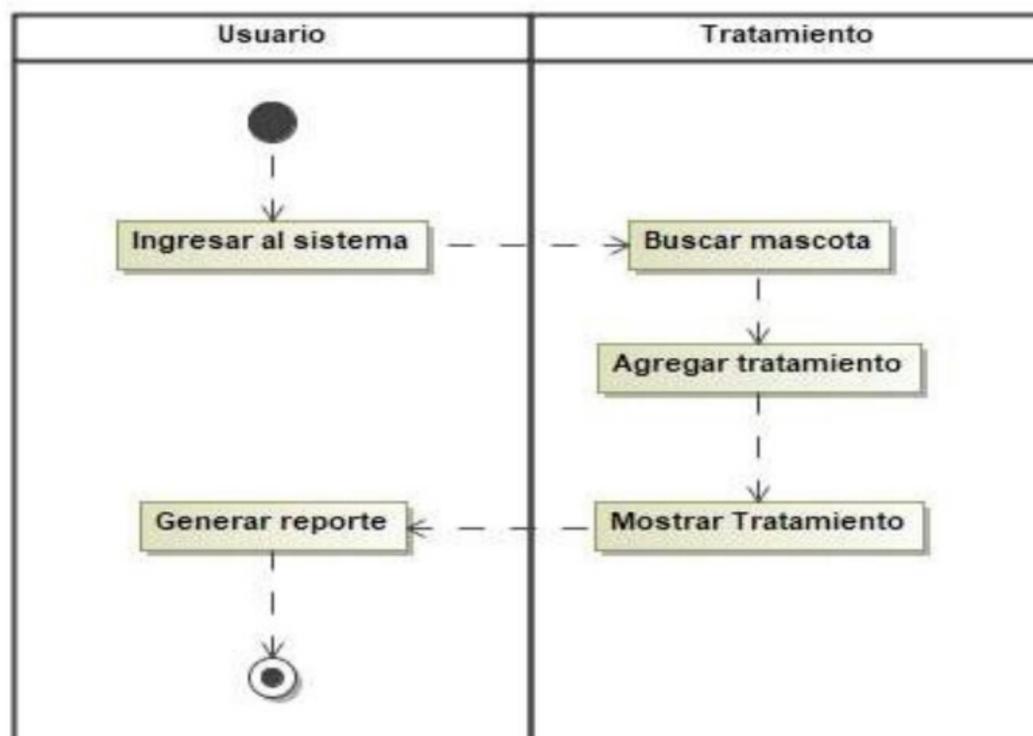


Fuente: Elaboración propia

➤ **Diagrama de Actividades: Gestión Tratamientos**

A continuación se muestra el diagrama de actividades para representar las funcionalidades del registro de propietarios, de acuerdo a la siguiente figura.

Figura 40. Diagrama de Actividades: Gestión de Tratamiento



Fuente: Elaboración propia

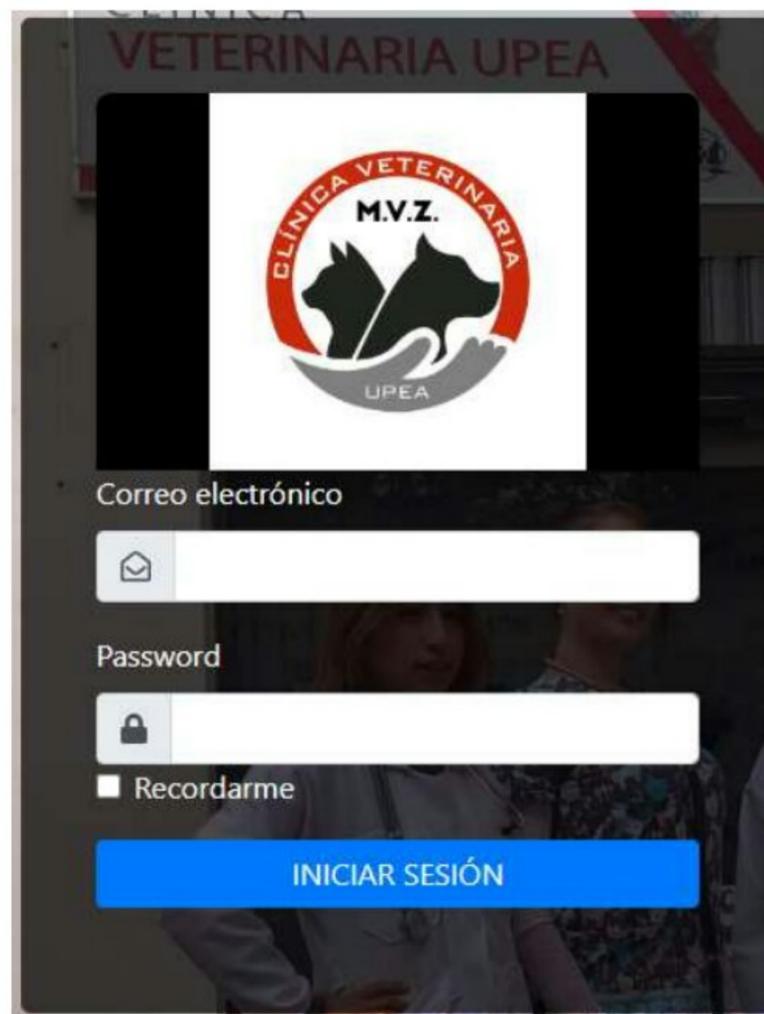
3.8 IMPLEMENTACIÓN DEL SISTEMA

Para el desarrollo de las aplicaciones web se requiere de ciertas metodologías acordes a las características de la plataforma donde será ejecutada. Para el presente proyecto de acuerdo al modelado de UWE, que es una metodología basada en UML.

3.8.1 Interfaz Inicio de Sesión

A continuación se muestra el interfaz de acceso al Sistema, de acuerdo a la siguiente figura.

Figura 41. Pantalla: Control de Acceso



Fuente: Elaboración propia

En la siguiente figura podemos observar, una vez que ya se tuvo el acceso a se muestra el control de acceso al sistema.

Figura 42. Pantalla de Inicio



Fuente: Elaboración propia

3.8.2 Módulos que integran el Sistema

3.8.2.1 Módulo Administración de Usuarios

A continuación se muestra el interfaz de administración de usuarios, donde se da a conocer los principales datos de los usuarios registrados en una lista donde podemos observar los botones de agregar, mostrar, editar y eliminar usuario, de acuerdo a la siguiente figura.

Figura 43. Pantalla: Lista de Usuarios

Lista de Usuarios

[+ Nuevo Usuario](#)

Mostrar registros por página

Buscar:

No	Usuario	Rol	Foto	Email	Acciones
1	Fabiola	usuarios		fabimkm@gmail.com	 eliminar
2	Juan Carlos	usuarios		jcarlo09@gmail.com	 eliminar
3	Pedro	administrador		pedromln@gmail.com	 eliminar

Fuente: Elaboración propia

A continuación se muestra el formulario de registro de los usuarios, donde podemos observar que se llenan los campos con los principales datos, la contraseña correspondiente contraseña, la foto de perfil, se le asigna el rol de usuario, de acuerdo a la siguiente figura.

Figura 44. Formulario: Registro de Usuarios

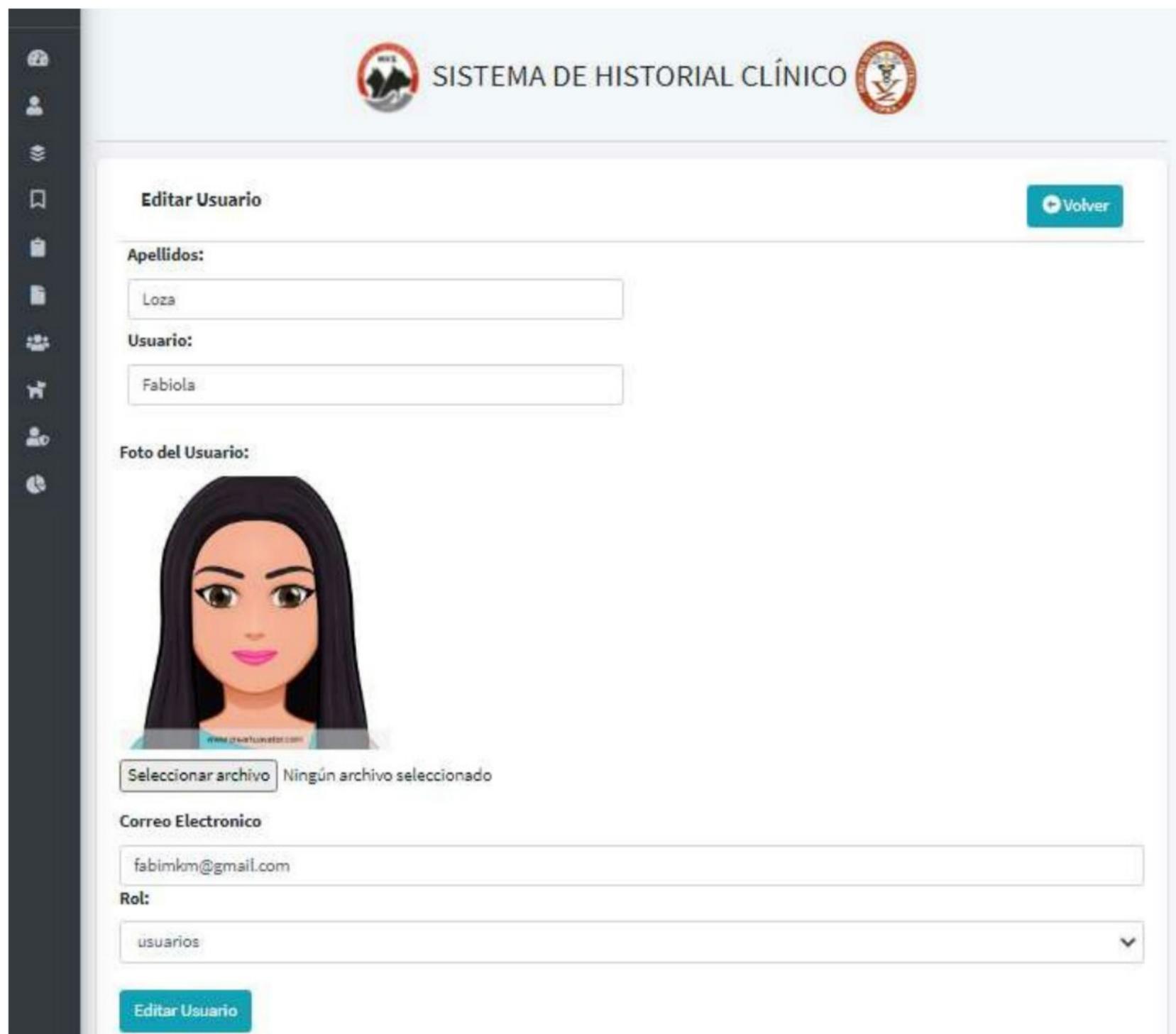
The image shows a web form for user registration. On the left, there is a dark vertical sidebar with several icons. The main form area is titled "Registro de Usuarios" and contains the following elements:

- A blue button labeled "Volver" with a circular arrow icon.
- Two input fields: "Apellidos:" and "Nombres:".
- A section for "Foto del Usuario:" containing a placeholder image with the text "Sorry, no photo available" and a "Seleccionar archivo" button. Below the button, it says "Ningún archivo seleccionado".
- An input field for "Correo Electronico".
- An input field for "Password".
- An input field for "Confirmar Password".
- A dropdown menu for "Rol:" with "super-admin" selected.
- A blue button labeled "Guardar Usuario" at the bottom.

Fuente: Elaboración propia

A continuación se muestra el formulario de modificación de los usuarios, donde podemos observar que los campos se encuentran con los datos del usuario seleccionado en la lista principal, y podemos actualizar los datos de este usuario, de acuerdo a la siguiente figura.

Figura 45. Formulario: Modificar Usuarios



The screenshot shows a web interface for a 'SISTEMA DE HISTORIAL CLÍNICO'. The page title is 'Editar Usuario'. On the right, there is a 'Volver' button. The form contains the following fields:

- Apellidos:** A text input field containing 'Loza'.
- Usuario:** A text input field containing 'Fabiola'.
- Foto del Usuario:** A section containing a cartoon avatar of a woman with long black hair. Below the avatar is a file selection button labeled 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'.
- Correo Electronico:** A text input field containing 'fabimkm@gmail.com'.
- Rol:** A dropdown menu with 'usuarios' selected.

At the bottom left of the form is an 'Editar Usuario' button.

Fuente: Elaboración propia

3.8.2.2 Módulo Administración de Propietarios

A continuación se muestra el interfaz de administración de propietarios, donde se da a conocer los principales datos de los propietarios registrados, de acuerdo a la siguiente figura.

Figura 46. Formulario: Registro Propietario

SISTEMA DE HISTORIAL CLÍNICO

PROPIETARIOS

[+ Agregar Propietario](#) [Imprimir](#)

Mostrar 10 registros por página

No	Apellidos	Nombres	C.I.	Dirección	Acciones
1	Cardenas Lazo	Valeria	7204104	Villa Alemania	Ver Editar eliminar
2	Quispe Huayta	Gabriel	7985692	Villa Bolivar D	Ver Editar eliminar
3	Flores Casas	Daniel	7044588	Obrajes	Ver Editar eliminar

Fuente: Elaboración propia

Figura 47. Formulario: Modificar Propietario

SISTEMA DE HISTORIAL CLÍNICO

Editar Propietario [Volver](#)

Apellidos

Nombres

C.I.

Dirección

Telefono

[Guardar](#)

Fuente: Elaboración propia

Figura 48. Pantalla: Información de Propietario

The screenshot displays the 'SISTEMA DE HISTORIAL CLÍNICO' interface. At the top, there are two logos: 'SERVICIO VETERINARIO MVZ' and 'MEDICINA RESPONSABLE Y POSICIONADA M.P.E.A.'. Below the logos, the page is divided into two main sections: 'Datos del propietarios' and 'Mascotas del propietarios'.

Datos del propietarios section includes a 'Volver' button and a table with the following data:

Datos del propietario					
Apellidos	Cardenas Lazo	Nombres	Valeria	C.I.	7204104
Telefono	67617937	Dirección	Villa Alemania		

Mascotas del propietarios section includes an 'Agregar mascota' button and a table with the following data:

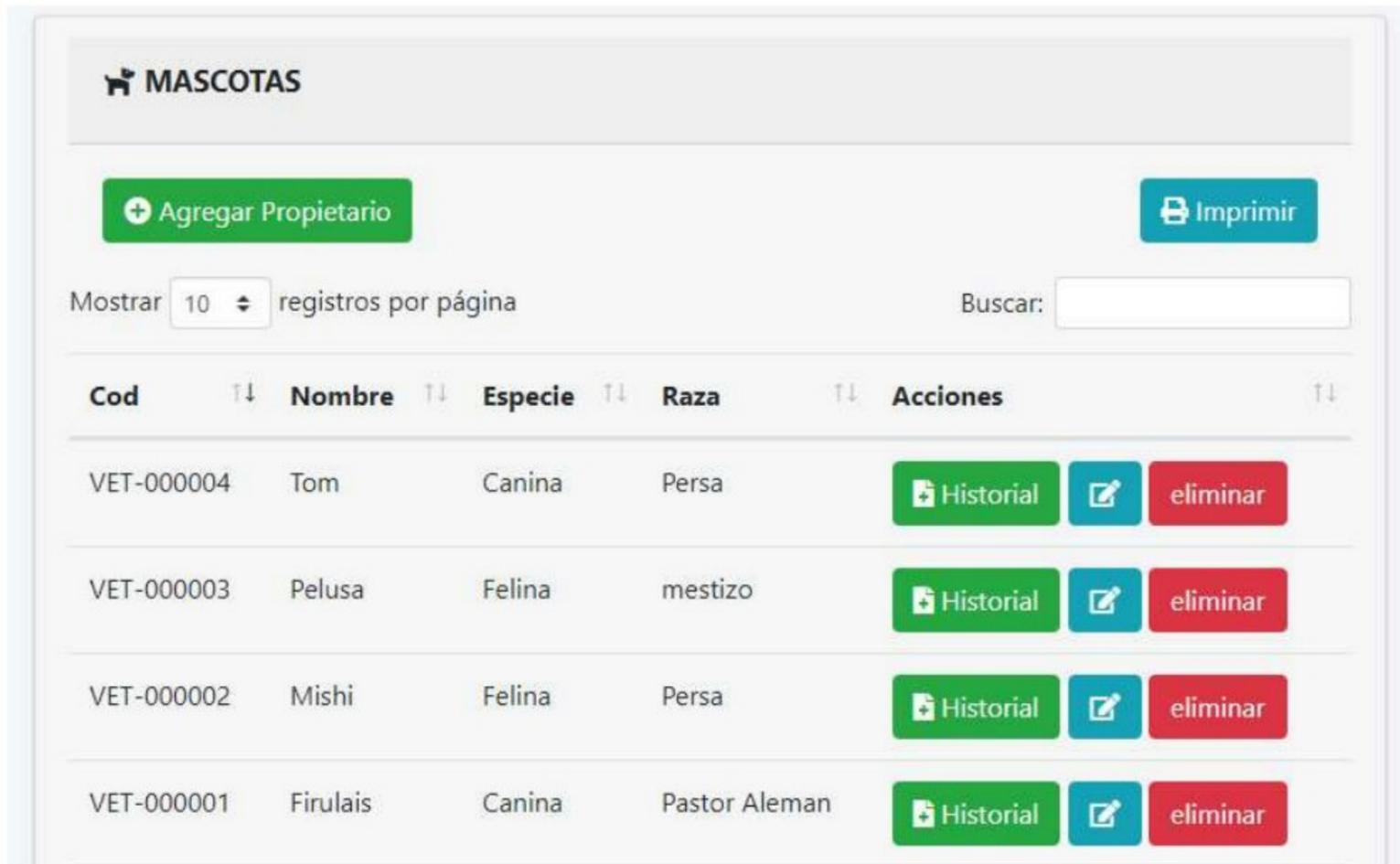
Cod	Nombre	Especie	Raza	Historial
VET-000001	Firulais	Canina	Pastor Aleman	Historial
VET-000004	Tom	Canina	Persa	Historial

Fuente: Elaboración propia

3.8.2.3 Módulo Administración de Mascota

A continuación se muestra el interfaz de administración de mascotas, donde se da a conocer los principales datos de los pacientes registrados, de acuerdo a la siguiente figura.

Figura 49. Pantalla: Lista de Mascotas



Cod	Nombre	Especie	Raza	Acciones
VET-000004	Tom	Canina	Persa	Historial ✎ eliminar
VET-000003	Pelusa	Felina	mestizo	Historial ✎ eliminar
VET-000002	Mishi	Felina	Persa	Historial ✎ eliminar
VET-000001	Firulais	Canina	Pastor Aleman	Historial ✎ eliminar

Fuente: Elaboración propia

Figura 50. Formulario: Modificar Mascota



Editar Mascota [Volver](#)

Nombre **Especie** **Edad**

Raza **Color** **sexo**

[Guardar](#)

Fuente: Elaboración propia

3.8.2.4 Módulo Administración de Historial

A continuación se muestra el interfaz de administración de historial, donde se muestra el formulario de registro de historial clínico, de acuerdo a la siguiente figura.

Figura 51. Formulario: Registro de Diagnostico

Diagnóstico

Fecha

dd/mm/aaaa

T° (°C) Peso (Kg) FC/min

FR/min Mucosas Hid. Tisular (seg)

SIGNOS Y SÍNTOMAS CLÍNICOS (ANAMNESIS)

DIAGNÓSTICO CLÍNICO (PRESUNTIVO)

TERAPÉUTICA EN CONSULTORIO/VACUNAS/DESP/OTROS

Observaciones...

Análisis o Exámenes Solicitados...

Recomendaciones...

Guardar Cancelar

STICO CLINICO: Disemper canino canónico crónico (tercera fase).

Fuente: Elaboración propia

Figura 52. Formulario: Registro Terapéutica

Terapéutica Ambulatoria

Producto

mg/ml Frecuencia # dias

Agregar Cancelar

Fuente: Elaboración propia

Figura 53. Pantalla: Historial de Mascota

Volver
Agregar Historial
Tratamiento
Imprimir



Universidad Pública de El Alto
CARRERA DE MEDICINA VETERINARIA Y ZOOTECNIA
CLÍNICA VETERINARIA U.P.E.A.



HISTORIA CLÍNICA

COD: VET-000004

***DATOS DEL PROPIETARIO**

Apellidos:	Cardenas Lazo	Nombre(s):	Valeria
Dirección:	Villa Alemania	Telefono:	67617937

***DATOS DE LA MASCOTA (Paciente)**

Nombre:	Tom	Especie:	Canina	Edad:	6 meses
Raza:	Persa	Color:	Blanco	Sexo:	Macho

Fecha: 2021-09-23
[Editar](#)
[Eliminar](#)

T° (°C):	12.6	Peso(Kg):	5	FC/min	20
FR/mín:	40	Mucosas:	rosadas	Hid.Tisular(Seg):	30

SIGNOS Y SÍNTOMAS CLÍNICOS (ANAMNESIS): Secreción nasal mucopurulento, lagañas, letargía, pérdida de apetito de hace dos días, movimientos involuntarios a nivel parpados superiores, nariz, y pulpejos hiperqueratinizados, vómitos frecuente desde hace días.

DIAGNÓSTICO CLINICO: Distemper canino canonico crónico (tercera fase).

TERAPÉUTICA EN CONSULTORIO VACUNA/DESP/OTROS: Complejo B = 2ml

TERAPÉUTICA AMBULATORIA: [+ Agregar](#)

Producto(N. comercial o principio activo)	mg/ml	Frecuencia	Número de días	Acciones
Glucosamin en sobre	5	C/12 horas	7	Editar Eliminar

OBSERVACIONES: No tiene vacunas.

ANÁLISIS O EXAMENES SOLICITADOS: Hemograma y Test rápido de Distemper canino.

RECOMENDACIONES: El tratamiento consiste en tres días.

Fuente: Elaboración propia

3.8.2.5 Módulo de Reportes

A continuación se muestra el reporte generado del Historial Clínico de los pacientes registrados, de acuerdo a la siguiente figura.

Figura 54. Reporte: Historial Clínico de Mascota

1 / 2 | - 50% + | [Iconos de navegación]

Cod: VET-000004



HISTORIA CLÍNICA

***DATOS DEL PROPIETARIO**

Apellidos: Cardenas Lazo Nombres: Cardenas Lazo
Dirección: Villa Alemania Telefono: 87617937

***DATOS DE LA MASCOTA**

Nombre: Tom Edad: 6 meses Especie: Canina
Raza: Persa Color: Blanco Sexo: Macho

Fecha: 2021-09-23

T (°C): 12.6 Peso (Kg): 5 FC/min: 20
FR/min: 40 Mucosas: rosadas Hid. Tis. (seg): 30

SIGNOS Y SÍNTOMAS CLINICOS (ANAMNESIS): Secreción nasal mucopurulenta, lagañas, letargia, pérdida de apetito de hace dos días, movimientos involuntarios a nivel parpados superiores, nariz, y pupilas hiperqueratinizadas, vómitos frecuente desde hace días.

DIAGNÓSTICO CLINICO (PRESUNTIVO): Distemper canino canónico crónico (tercera fase).

TERAPÉUTICA EN CONSULTORIO VACUNA/DESP/OTROS: Complejo B = 2ml

TERAPÉUTICA AMBULATORIA:

Producto(N. comercial o principio activo)	mg/ml	Frecuencia	Número de días
Glucosamin en sobre	5	C/12 horas	7

OBSERVACIONES: No tiene vacunas.

Fuente: Elaboración propia

3.8.2.6 Módulo de Administración de Adopciones

A continuación se muestra el formulario de registro de publicaciones, de acuerdo a las categorías y etiquetas a las que pertenezcan, de acuerdo a la siguiente figura.

Figura 55. Formulario de Registro de Publicaciones

Nombre

Slug

Categoría

Adopciones ▼

Etiquetas

Canino Felino

Estado

Borrador Publicado



Imagen que se mostrará en el post

Seleccionar archivo Ningún ar...leccionado

Subir una foto de la mascota, que se quiere dar en adopción, si no tuviese una se asignará una por defecto.

Extracto

Paragraph ▼ **B** *I* @ := ½= | ≡ ≡ | ▼ ▼ ↶ ↷

Cuerpo del post:

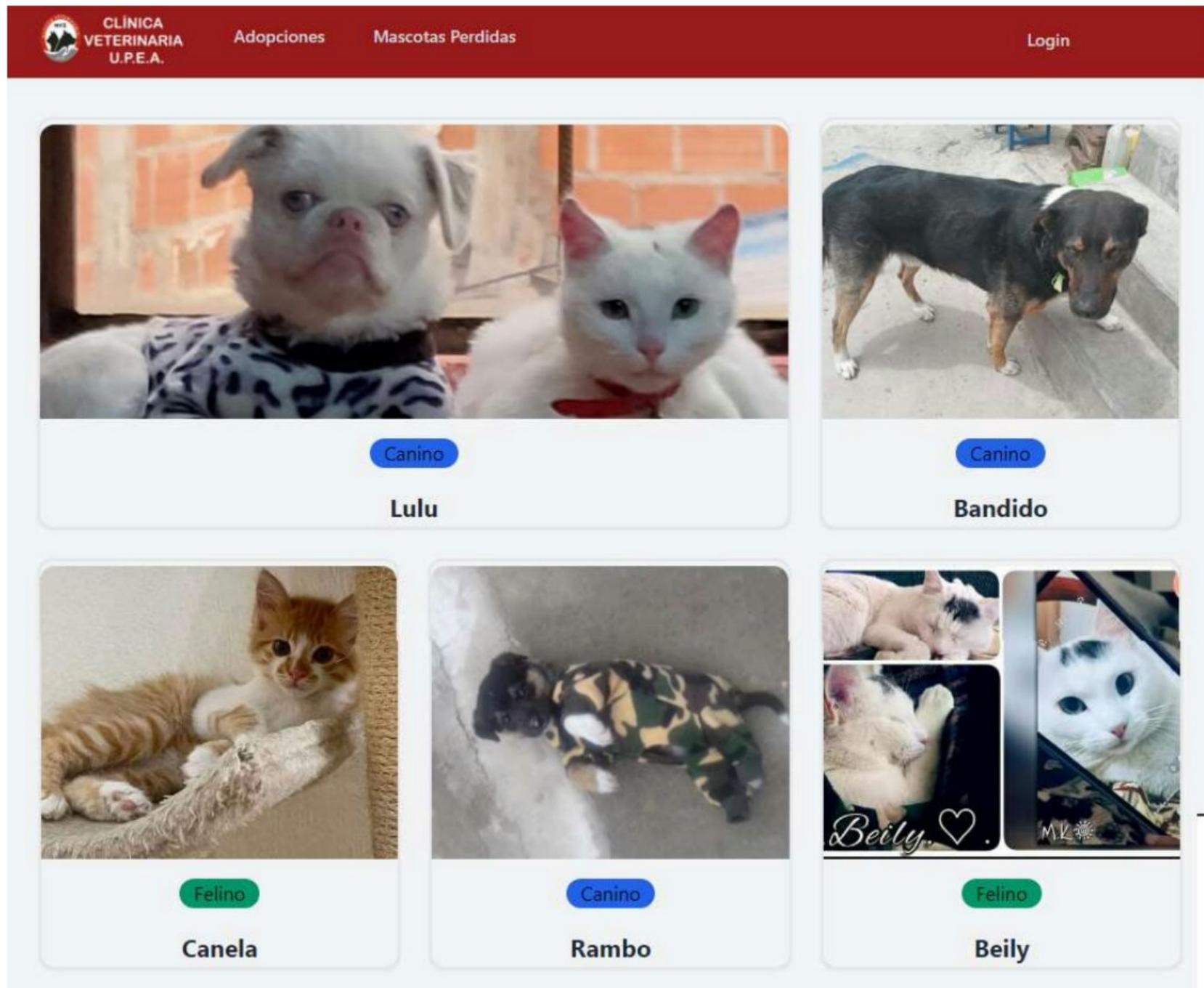
Paragraph ▼ **B** *I* @ := ½= | ≡ ≡ | ▼ ▼ ↶ ↷

Crear post

Fuente: Elaboración propia

A continuación se muestra el interfaz de publicaciones, de acuerdo a las categorías y etiquetas a las que pertenezcan, de acuerdo a la siguiente figura.

Figura 56. Interfaz de Publicaciones



Fuente: Elaboración propia

3.8.2.7 Reporte Estadístico

A continuación se muestra el reporte estadístico generado por la cantidad de mascotas de acuerdo a la especie de los pacientes registrados, de acuerdo a la siguiente figura.

Figura 57. Pantalla: Estadística de Mascotas



Fuente: Elaboración propia

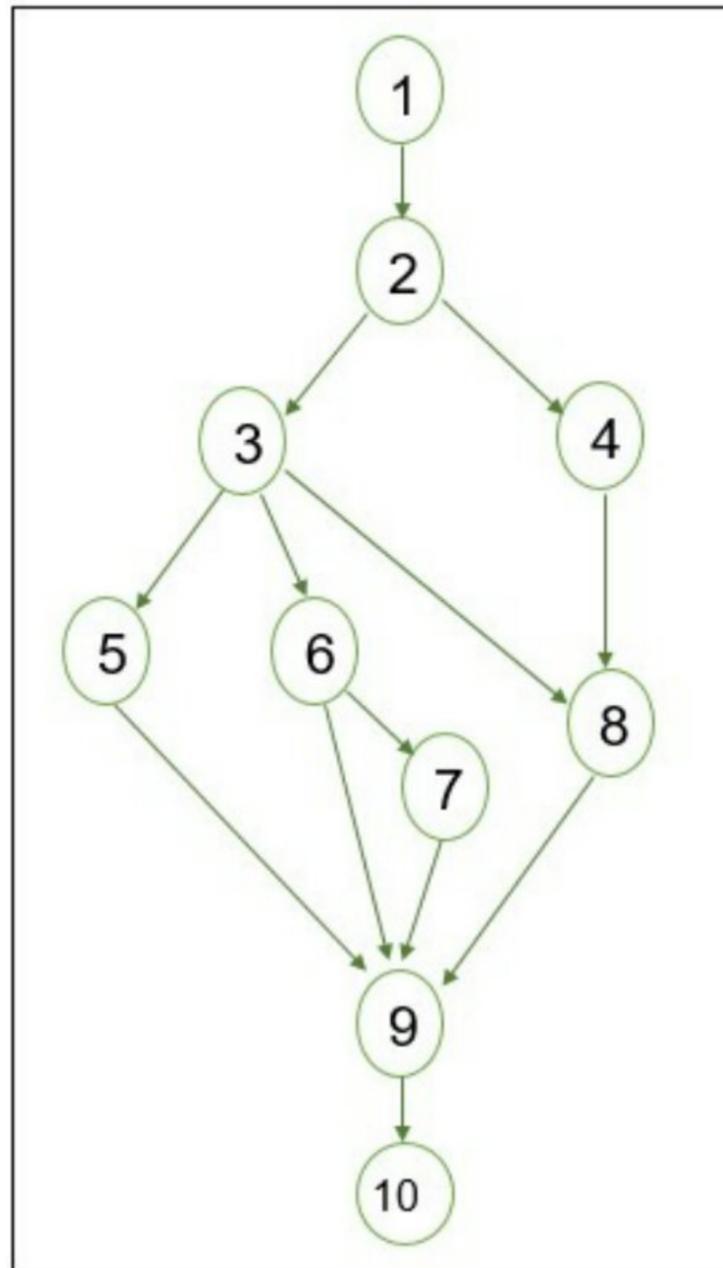
3.9 PRUEBAS DE SOFTWARE

Para las pruebas de software se utilizará las pruebas de caja blanca y las pruebas de caja negra, las cuales probaran que se obtiene una determinada salida para una entrada determinada y además de probar la funcionalidad en conjunto del sistema.

3.9.1 Pruebas de Caja Blanca

Esta prueba se basa en el cálculo de las regiones que deben ser consideradas como caminos independientes del sistema, y estableciendo cuáles con las entradas para que se ejecutan cada una de las regiones. A continuación de forma general se muestra los caminos independientes del sistema:

Figura 58. Caja Blanca



Fuente: Elaboración propia

Donde:

- Inicio del Sistema (1)
- Menú Principal (2)
- Administrador de Usuarios (3)
- Reportes (4)
- Propietarios (5)
- Mascotas(6)
- Historial Clínico (7)
- Tratamientos (8)
- Consultas de información (9)
- Fin del Sistema (10)

Ya realizado el grafo generado de acuerdo de las características del sistema, se procede a determinar la complejidad ciclométrica del grafo de acuerdo a la siguiente formula:

$$V(G) = A - N + 2$$

Donde:

$$A = 13 \text{ (Aristas)}$$

$$N = 10 \text{ (Nodos)}$$

Por tanto: $V(G) = 13 - 10 + 2 = 5$

Camino 1: 1-2-3-5-9-10

Camino 2: 1-2-3-6-9-10

Camino 3: 1-2-3-6-7-9-10

Camino 4: 1-2-3-8-9-10

Camino 5: 1-2-4-8-9-10

Estos caminos representan condiciones en las que al menos se ejecuta los nodos establecidos en el camino, la especificación de estos caminos son los siguientes:

Camino 1: El sistema se ejecuta al instante, una vez verificado los datos de acceso del administrador.

Camino 2: Se registra a los usuarios y se muestra ya en listado de los usuarios.

Camino 3: Se hace la consulta de los usuarios, propietarios y mascotas de acuerdo a los registros.

Camino 4: Se ejecuta los reportes de acuerdo a solicitud, estos pueden ser listas, o reporte de historial clínico.

Camino 5: Se registra al propietario y mascota en el historial clínico.

3.9.2 Pruebas de Caja Negra

Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

Para realizarla prueba de caja negra se tomará en cuenta los casos de ingreso al

sistema, registro de propietarios, registro de mascotas y registro de diagnóstico clínico.

Tabla 25. Caso de Prueba de Ingreso al Sistema

Caso de Prueba de Ingreso al Sistema	
Descripción	Acceso al sistema mediante email y contraseña. Se permite el acceso al sistema una vez verificado los datos de acceso.
Condición de Ejecución	El usuario debe estar en la página para ingresar al sistema una vez verificado email y contraseña.
Entradas	<p>El sistema muestra la página de autenticación email y contraseña.</p> <ul style="list-style-type: none"> ➤ Los usuarios del Sistema pueden realizar esta tarea, introducir los datos de acceso y presionar el botón de ingresar para ingresar al sistema. ➤ Los usuarios ingresan al sistema para realizar las tareas que corresponden dentro del sistema. <p>El Sistema despliega la pantalla de inicio para el usuario correspondiente de acuerdo al rol de este.</p>
Resultados Esperados	El sistema verifica al usuario de acuerdo a sus registros, para luego permitir el acceso correspondiente.
Evaluación de Prueba	Se realiza el ingreso al Sistema con la verificación previa, de manera satisfactoria.

Fuente: Elaboración propia

A continuación se muestra el caso de prueba de uso para el registro de usuario y mascota respectivamente, en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, de acuerdo a las siguientes tablas.

Tabla 26. Caso de Prueba Registro de Usuarios

Caso de Prueba Registro de Usuarios	
Descripción	Registrar a los Usuarios. Permite realizar el registro de los Usuarios con un rol asignado.
Condición de Ejecución	El Administrador debe de registrar a los usuarios con los permisos correspondientes.
Entradas	<ul style="list-style-type: none"> ➤ El sistema muestra el formulario de registro de usuarios. ➤ El administrador (Director), encargado de registrar a los usuarios y monitorearlos.
Resultados Esperados	Registrar a los usuarios con el rol en específico.
Evaluación de Prueba	Se realizó el registro de forma satisfactoria.

Fuente: Elaboración propia

Tabla 27. Caso de Prueba Registro de Mascota

Caso de Prueba Registro de Mascotas	
Descripción	Registrar a las mascotas. Permite realizar el registro de las mascotas con su respectivo propietario.
Condición de Ejecución	El Administrador y los Usuarios deben de registrar a los pacientes (Mascotas).
Entradas	<ul style="list-style-type: none"> ➤ El Administrador y los Usuarios deben de acceder al sistema. ➤ El sistema muestra el formulario de registro del propietario y posteriormente la del paciente (Mascota).
Resultados Esperados	Registrar los datos del propietario, y de su mascota en el formulario correspondiente.
Evaluación de Prueba	Se realizó el registro de forma satisfactoria.

Fuente: Elaboración propia

A continuación se muestra el caso de prueba de uso para el registro de diagnóstico clínico, en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, de acuerdo a la siguiente tabla.

Tabla 28. Caso de Prueba Registro de Diagnóstico Clínico

Caso de Prueba Registro de Diagnóstico Clínico	
Descripción	Registrar el Diagnóstico Clínico. Permite realizar el registro Diagnóstico de los pacientes (Mascotas).
Condición de Ejecución	El Administrador y Usuarios deben de registrar los diagnósticos de los pacientes (mascotas)
Entradas	<ul style="list-style-type: none"> ➤ El sistema muestra el formulario de registro de Diagnostico de Mascota. ➤ El administrador y Usuarios, encargados de registrar los Diagnósticos, después de haber registrado los propietarios y los datos de la mascota.
Resultados Esperados	Registrar el diagnostico correspondiente de acuerdo al paciente ya registrado.
Evaluación de Prueba	El registro se realizó de forma satisfactoria.

Fuente: Elaboración propia

Se realizó las pruebas de caja negra, en el registro de mascotas y registro de diagnóstico, de esta forma se realiza el proceso de registro del historial clínico.

3.10 CALIDAD

Las métricas de calidad es una de las partes más importantes de este proyecto, ya que definen de una y otra forma la calidad del software. Por lo cual se aplicará la norma ISO 9126 el cual es un estándar internacional para evaluar el software.

3.10.1 Normas ISO 9126

3.10.1.1 Funcionalidad

El presente proyecto satisface las necesidades de la clínica, como ser en la

administración dentro de la Clínica Veterinaria U.P.E.A. Una de las características de la norma ISO 9126 es medir la funcionalidad, la cual se realiza de acuerdo a la complejidad del sistema, para esta medición se debe determinar las siguientes características:

Tabla 29. Características de funcionalidad.

Características	Descripción
Número de entradas de Usuario	Cuando el usuario ingresa los datos de acceso
Número de salidas de Usuario	Se cuenta como salida la información que proporciona del usuario al sistema.
Número de peticiones de Usuario	Es una entrada en línea la cual lleva a la generación de una respuesta inmediata por parte del sistema.
Número de Archivos	Se toma en cuenta cada archivo lógico maestro, estos son un agrupamiento lógico de datos como ser parte de una base de datos o archivos.
Números de interfaces externas	Se cuentan todas las interfaces legibles por pantalla.

Fuente: Elaboración propia

De acuerdo a estos conceptos, aplicando al sistema desarrollado se tiene:

A continuación se muestra los conceptos aplicando al sistema desarrollado, se obtienen los siguientes datos, de acuerdo a la siguiente tabla.

Tabla 30. Parámetros de medición

Parámetros de Medición	Cuenta
Número de entradas de Usuario	6
Número de salidas de Usuario	4
Número de peticiones de Usuario	5
Número de Archivos	12
Números de interfaces externas	1

Fuente: Elaboración propia

➤ Factores de Ponderación

En la siguiente tabla podemos observar las cinco características con factores de ponderación para el cálculo de punto función.

Ya obtenidos los valores, y seleccionando la ponderación, realizamos el cálculo de punto función de acuerdo a la siguiente tabla:

Tabla 31. Cálculo de Punto de Función

Parámetros de Medición	Cuenta	Factor	Resultado
Número de entradas de Usuario	6	4	24
Número de salidas de Usuario	4	5	20
Número de peticiones de Usuario	5	4	20
Número de Archivos	12	10	120
Números de interfaces externas	1	7	7
Total			191

Fuente: Elaboración propia

Este resultado de punto de función se debe de ajustar en función a las características ambientales del sistema. De acuerdo a los valores de ajuste de complejidad F_i , donde i varía de 1 hasta 14, los valores de ajuste de complejidad basados en las respuestas a la preguntas formuladas de la siguiente tabla:

Tabla 32. Valores de Ajuste de Complejidad

	Importancia	0 %	20 %	40 %	60 %	80 %	100 %	F_i
	Escala	No	Incidencia	Moderado	Medio	Significativ	Esencial	
No	Factor	0	1	2	3	4	5	
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?					X		4

2	¿Se requiere comunicación de datos?				X			3
3	¿Existen funciones de procesamiento distribuido?				X			3
4	¿Es crítico el rendimiento?				X			3
5	¿Se ejecuta el sistema en un sistema operativo existente y fuertemente utilizado?					X		4
6	¿Requiere el sistema entrada de datos interactiva?	X						1
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?			X				2
8	¿Se actualizan los archivos maestros de forma interactiva?				X			3
9	¿Son complejos las entradas, las salidas, los archivos o peticiones?					X		4
10	¿Es complejo el procesamiento interno?						X	5
11	¿Se ha diseñado el código para ser reutilizable?			X				2
12	¿Están incluidas en el diseño la conversión y la instalación?			X				2
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?					X		4
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?				X			3
FACTOR DE COMPLEJIDAD TOTAL (FCT) ΣFi								43

Fuente: Elaboración propia

Calculando el punto de función mediante la siguiente ecuación:

$$PF = Cuenta\ Total * (0.65 + 0.01 * \sum Fi)$$

Donde

Cuenta total: es la suma del producto del factor de ponderación y valores de los parámetros.

$\sum Fi$: es la sumatoria de los valores de ajuste de la complejidad.

Calculando:

$$PF = 191 * (0.65 + 0.01 * 43)$$

$$PF = 206.28$$

Considerando el máximo ajuste de la complejidad $\sum Fi = 70$ calculamos al 100% el nivel de confianza de la siguiente manera:

$$PF_{max} = 191 * (0.65 + 0.01 * 70)$$

$$PF_{max} = 257.85$$

Una vez obtenido ambos valores, de acuerdo a la fórmula de la funcionalidad se obtiene el siguiente resultado:

$$funcionalidad = \frac{PF}{PF_{max}} * 100\%$$

$$funcionalidad = \frac{206.28}{257.85} * 100\%$$

$$funcionalidad = 80.4\%$$

De esta forma podemos concluir que la funcionalidad es de 80.4%, la interpretación nos dice que existe un 80.4% de funcionar sin riesgos, y un 19.6 % de probabilidad de que el sistema colapse.

3.10.1.2 Confiabilidad

La confiabilidad del presente proyecto, se medirá de acuerdo a la probabilidad de operación libre de fallos, los cuales son:

$$P(T \leq t) \text{ Probabilidad de fallas}$$

$$P(T \leq t) = 1 - F(t) \text{ Probabilidad de trabajo sin fallas}$$

Para calcular la confiabilidad del sistema se toma en cuenta en un periodo de tiempo en el que se ejecuta y se obtiene muestras.

$$F(t) = f * e^{-\mu * t}$$

$$\mu = \frac{\text{Número de fallas de acceso}}{\text{Números de acceso al sistema}}$$

Donde:

f: Funcionalidad del sistema

μ : Probabilidad de error que puede tener el sistema

t : Tiempo de duración de gestión del sistema (días)

Consideramos un periodo de 21 días, para el tiempo de prueba donde se obtuvieron que de cada 10 ejecuciones se presenta una falla, por lo que el cálculo será el siguiente:

$$F(t) = f * e^{-u*t}$$

$$F(t) = 0.804 * e^{-\frac{1}{10}*21}$$

$$F(t) = 0.0985 * 100 = 9.85 \%$$

Reemplazando en las fórmulas de probabilidades:

$$P(T \leq t) \Rightarrow P(T \leq t) = 0.0985 = 9.85 \%$$

$$P(T \leq t) = 1 - F(t) \Rightarrow P(T \leq t) = 1 - 0.0985 = 0.9015 = 90.15 \%$$

Por lo tanto, de acuerdo a los cálculos realizados se tiene una confiabilidad del 90.15%, en un periodo de prueba de 21 días.

3.10.1.3 Usabilidad

Se realiza una evaluación para conocer si el sistema satisface los requerimientos establecidos por el usuario, entonces para esto se medirá de acuerdo a encuestas planteadas a los usuarios del sistema, la calificación tuvo un rango de 0 a 100% y los resultados son los siguientes:

Tabla 33. Escala de Valoración Para La Evaluación de Uso

Escala	Valor
Muy Bueno	5
Bueno	4
Regular	3
Malo	2

Fuente: Elaboración propia

En la siguiente tabla nos muestra los resultados, de la encuesta que se realizó.

Tabla 34. Cuestionario de Evaluación de Uso

No	Preguntas	Si	No	Evaluación
1	¿Puede utilizar con facilidad el sistema?	5	0	1
2	¿Puede controlar operaciones que el sistema solicite?	3	2	0.6
3	¿El sistema permitió la retroalimentación de información?	3	2	0.6
4	¿El sistema cuenta con interfaz gráfica agradable a la vista?	5	0	1
5	¿La respuesta del sistema es satisfactoria?	4	1	0.8
6	¿Le parecen complicadas las funciones del sistema?	4	1	0.8
7	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	1
8	¿Durante el uso del sistema se produjo errores?	3	2	0.6
	TOTAL			6.4

Fuente: Elaboración propia

Para determinar la usabilidad del sistema se utiliza la siguiente ecuación:

$$U = \left(\sum \frac{X_i}{n} \right) * 100 \%$$

Realizando los cálculos con los datos obtenidos:

$$U = \frac{6.4}{8} * 100\%$$

$$U = 80\%$$

De acuerdo a estos cálculos, se tiene un 80% de usabilidad, eso quiere decir una comprensión o entendimiento de los usuarios en el porcentaje mencionado.

3.10.1.4 Mantenibilidad

El mantenimiento se da de acuerdo a las modificaciones del sistema, esto quiere decir a los nuevos requerimientos y actualizaciones según los usuarios del sistema. Por lo cual el índice de madurez del software se determina de acuerdo a la siguiente ecuación:

$$IMS = \frac{[Mt - (Fa + Fb + Fc)]}{Mt}$$

Donde:

Mt: Número de módulos en la versión actual

Fa: Número de módulos en la versión actual que se han añadido

Fb: Número de módulos en la versión actual que se han modificado

Fc: Número de módulos de la versión anterior que se han borrado en la versión actual

En este sistema se tienen los siguientes valores de acuerdo a las variables ya mencionadas, se muestra la información en la siguiente tabla:

Tabla 35. Datos Obtenidos del Sistema

Variable	Valor
Mt	6
Fa	0
Fb	1
Fc	0

Fuente: Elaboración propia

Realizamos los cálculos:

$$IMS = \frac{[6 - (0 + 1 + 0)]}{6} * 100\%$$

$$IMS = 0.83 * 100\%$$

$$IMS = 83\%$$

Por lo cual el sistema tiene un 83% de índice de madurez de software, que es la facilidad de mantenimiento, y el 17% corresponde al margen de error a los cambios y modificaciones.

3.10.1.5 Portabilidad

El presente proyecto de acuerdo a la metodología está diseñada en un entorno de acceso vía web, por lo cual la portabilidad por el lado del servidor y por el lado del cliente, entonces la portabilidad se medirá bajo los siguientes aspectos:

- Software del Servidor
- Hardware del Servidor
- Sistema Operativo del Servidor

En conclusión de acuerdo a estas características el sistema es portable tanto en hardware como en software.

3.11 SEGURIDAD

La seguridad se convierte en un pilar fundamental para el desarrollo de software, para este proyecto se tomaron en cuenta normas internacionales que permiten el aseguramiento del sistema.

3.11.1 ISO 27001

Es una norma internacional que describe cómo gestionar la seguridad de la información en una institución donde contempla la confidencialidad, disponibilidad e integridad de los datos. En el presente proyecto se dará a conocer de qué manera se hacen cumplir estos puntos muy importantes en la parte de seguridad.

3.11.1.1 Confidencialidad de datos

➤ Seguridad a Nivel de Base de datos

El presente proyecto continuamente interactúa con las base de datos, por lo cual la información es manejada constantemente, información que debe ser manejada con mucha responsabilidad, por lo que debe tener garantía de que la información no debe ser divulgada, ni expuesta a ciberataques. Por lo cual la forma de resguardar los datos se realizó con lo siguiente:

- Para el acceso al sistema se necesita el nombre de usuario y contraseña, la cual es encriptada al momento del registro.
- A través del framework usado en este proyecto se tiene seguridad a nivel de vistas, donde personas ajenas no pueden acceder a través de la url.
- Se usa token de seguridad “csrf”, para cada sesión de usuario activa manejada por la aplicación, esto evita ataques de inyección sql.

3.11.1.2 Disponibilidad de datos

La disponibilidad de datos es el acceder a la información al tiempo o la hora que sea con el fin que los usuarios alteren, actualicen, respalden los datos útiles del sistema. El servidor provee un servicio ininterrumpido de acceso, de esta forma se tiene una

disponibilidad para el usuario para que pueda acceder al Sistema Clínico en cualquier momento siempre y cuando se tenga acceso a la red.

3.11.1.3 Integridad de datos

La integridad de los datos hace referencia a que los datos no pueden ser alterados por ningún tipo de personal, el sistema ofrece al administrador, la capacidad de monitorear las actividades de los usuarios al realizar el registro de historial clínico, este monitoreo registra todas las actividades que realiza el usuario: la fecha y hora de acceso, registros, modificaciones, eliminaciones de datos. De esta forma el Sistema Clínico cuenta con la confiabilidad en la integridad de los datos.

3.12 COSTOS

El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el presente proyecto por lo cual se aplicará el modelo COCOMO (Constructive Cost Model), este análisis determina la calidad y cantidad de recursos necesarios.

3.12.1 COCOMO II

Para el desarrollo de la estimación de costos COCOMO II, la estimación del sistema ha sido desarrollado bajo las líneas de código "*LDC*" (líneas de código), con este dato obtenemos "*Kl*" (cantidad de líneas de código en miles). Por lo que la evaluación del sistema ha sido considerada en, 18.566 *Kl*, bajo el siguiente cálculo.

$$Kl = \frac{LDC}{1000}$$
$$Kl = \frac{18566}{1000}$$
$$Kl = 18.566$$

Para completar las variables de las fórmulas de costos, los coeficientes que se usarán serán los valores que se detallan en la siguiente tabla:

Tabla 36. Constantes de Complejidad

Modo	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semilibre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Fuente: Elaboración propia

Ecuaciones para calcular el costo de Software:

Tabla 37. Ecuaciones del modelo COCOMO II

Variable	Ecuación	Tipo/Unidad
Esfuerzo requerido por el proyecto	$E = a * (Kl)^b * m(X)$	Persona-mes
Tiempo requerido del proyecto	$T_{dev} = c * (E)^d$	Mes
Número de personas requeridas	$P = \frac{E}{T_{dev}}$	Persona
Costo Total	$CT = SueldoMes * NP * T_{dev}$	\$us

Fuente: Elaboración propia

Para obtener los valores de factor de ajustes del esfuerzo, utilizaremos la siguiente tabla.

Tabla 38. Variables factor ajuste del esfuerzo

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65

Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	
Total $m(X)$						0.395

Fuente: Elaboración Propia

Ya teniendo todas las variables, realizamos los cálculos correspondientes de acuerdo con las formulas de la tabla 37. Para los valores de las constantes a, b, c y d, tomamos en cuenta que la cantidad de líneas de código en miles no sobrepasa los 50, por lo tanto usamos los valores del tipo orgánico de acuerdo a la tabla 36.

Se tiene:

Calculando el Esfuerzo:

$$E = a * (Kl)^b * m(X)$$

$$E = 2.4 * (18.566)^{1.05} * 0.395$$

$$E = 20.37 \text{ [personas - mes]} \sim \mathbf{20} \text{ [personas - mes]}$$

Calculando el tiempo requerido del proyecto:

$$T_{dev} = 2.5 * (20.37)^{0.38}$$
$$T_{dev} = 7.86 \text{ [meses]} \sim \mathbf{8} \text{ [meses]}$$

Calculando el número de personas requeridas:

$$P = \frac{E}{T_{dev}}$$
$$P = \frac{20.37}{7.86}$$
$$P = 2.59 \sim \mathbf{3} \text{ [personas]}$$

Calculando el Costo Personas mes (Salario promedio 2450 Bs equivalente 350 \$)

$$\text{Costo Mes} = P * \text{Salario promedio}$$

$$\text{Costo Mes} = 2.59 * 350$$

$$\text{Costo Mes} = 906.5 \$$$

Calculando el Costo Total del Proyecto:

$$\text{Costo Total} = \text{Costo Mes} * T_{dev}$$

$$\text{Costo Total} = 906.5 * 8 = 7672 \$$$

En conclusión de acuerdo a los cálculos, se requiere un estimado de 3 personas trabajando por el lapso de 8 meses con un costo total de 7252 \$us que equivalen a 50162.08 Bs.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

En conclusión se obtuvo como producto final previo análisis y diseño, codificación, metodología de pruebas de software, pruebas e implementación, Sistema Web De Gestión De Historial Clínico Veterinario Y Control De Adopciones. Este sistema satisface las necesidades de la Clínica Veterinaria U.P.E.A., cumple con el objetivo general y los objetivos específicos que fueron planteados en el presente proyecto.

El sistema brinda una considerable mejoría en la atención a los usuarios de la clínica, optimiza el nivel de atención y servicios de los pacientes, mejora la organización y buen manejo de los tratamientos. Para la obtención final del proyecto se realizaron las siguientes actividades:

- El Sistema Web De Gestión De Historial Clínico Veterinario Y Control De Adopciones, resolvió los problemas de automatización en los diferentes procesos y registros de información de los usuarios y clientes.
- El módulo de registros de Historial Clínico, facilita el registro de la información de los propietarios, datos de la mascota, diagnóstico clínico, terapéutica, y tratamientos.
- El sistema cuenta con la selección de roles y permisos, para que los usuarios de acuerdo al nivel del rol que tengan, accedan a la información que se le asigne.
- La información esta almacenada en una base de datos de manera segura, encriptando las contraseñas, brindando la disponibilidad al usuario en tiempo real y cuando lo necesite.
- Los tiempos para los procesos de registro de propietarios, pacientes, diagnósticos, terapéutica y tratamientos, se han reducido considerablemente optimizando el nivel de servicio en la Clínica.
- La búsqueda de información se realiza de forma rápida, al igual que la impresión de los reportes de Historial Clínico de acuerdo a lo que requiera el usuario.

4.2 RECOMENDACIONES

Para el presente proyecto se cumplieron con los objetivos trazados, con el fin buscar el mejoramiento del sistema se propone las siguientes recomendaciones:

- El mantenimiento, mediante una revisión periódica por cierto periodo de tiempo es recomendable para un funcionamiento adecuado del sistema.
- Recomendar al administrador realizar las copias de seguridad, para resguardar la información del Sistema.
- Capacitar a los usuarios para que cambien las contraseñas en un periodo de tiempo para mejorar la seguridad de los accesos.
- Capacitar a nuevos administradores mediante manual de funciones, para que puedan realizar operaciones en el sistema, y de esta manera administrarlos correctamente.
- Para cuando se requiera aumentar módulos o realizar alguna actualización, se recomienda revisar la documentación de este proyecto, ya que el sistema contiene elementos que son reutilizables para otros módulos nuevos.

BIBLIOGRAFÍA

- Abud Figueroa, M. A. (2012). *Calidad en la Industria del Software. La Norma ISO-9126*.
- Cadme Ruiz, C. M., & Duque Pozo, D. F. (2012). *Auditoría de Seguridad Informática ISO 27001*. Cuenca.
- Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2005). *PHP y MySQL*.
- Desconocido. (2015). *Aprendiendo Laravel*.
- EUATM. (1992). *Introducción a la Web*.
- Figueroa, M. A. (2012). *Calidad en la Industria del Software. La Norma ISO-9126*. Mexico D.F., Mexico: revista upiicsa.
- Gallegos, J. (2003). *Desarrollo Web con PHP y MySQL*. Madrid, España: Grupo Anaya.
- Gómez, A., López, M., Migani, S., & Otazú, A. (2010). *Un Modelo de Estimación de Proyecto de Software*.
- Gómez, A., López, M., Migani, S., & Otazú, A. (2010). *COCOMO Un Modelo de Estimación de Costos*.
- Guerrero, Ucán, & Menéndes. (2014). *UWE en Sistema de Recomendación de Objetos de Aprendizaje*. Yucatán, Mexico : Revista Latinoamericana de Ingeniería de Software.
- Javier, G. A. (2018). *Curso Bootstrap*.
- Morales, I. (2019). *Enginner*. Quito, Ecuador: Rimorsoft.
- Nieves Guerrero, C. G., Ucán Pech, J. P., & Menéndes Dominguez, V. H. (2014). *UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio*.
- Pressman, R. (2010). *Ingeniería de Software: Un Enfoque Práctico*. México: Mc Graw.
- Regina Baena, G., Mendoza Mendez, R., & Dorantes, E. (2019). *IMPORTANCIA DE LA NORMA ISO/EIC 27000 EN LA IMPLEMENTACION DE UN SISTEMA DE GESTION DE LA SEGURIDAD DE LA INFORMACIÓN*. Mexico: Revista contribuciones a la Economía.
- Rivera López, A. (2008). *Sistema Asistente para la Generación de Horarios de Curso*. Puebla.
- Roja, D. (2018). *Ingeniería de Sistemas, Apuntes*.
- Rotta, D., Pallota, G., Klikailo, H., & Belloni, E. (2016). *Un caso de estudio sobre la aplicación UWE para la generación de Sistemas Web*.
- Sampedro, L., Marques Díaz, J., & Vargas, F. (2002). *Ingeniería y Desarrollo*.
- Sánchez Peño, J. M. (2015). *Pruebas de Software. Fundamentos y Técnicas*.
- Santillan Aldan, J. (2002). *Biblios*.
- Sommerville. (2011). *Ingeniería de Software*. Pearson.
- Vespito. (2011). *MySQL 5.0 Reference Manual*. Barcelona.

SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES



MANUAL DE USUARIO

AUTOR: DAN CRUZ MAMANI CASAS

**VERSIÓN 1.0
2021**

1. OBJETIVO

Brindar las instrucciones del software denominado Sistema Web De Gestión De Historial Clínico Veterinario Y Control De Adopciones, para los usuarios de este Sistema.

2. OPCIONES DEL SISTEMA

a) Ingreso al Sistema

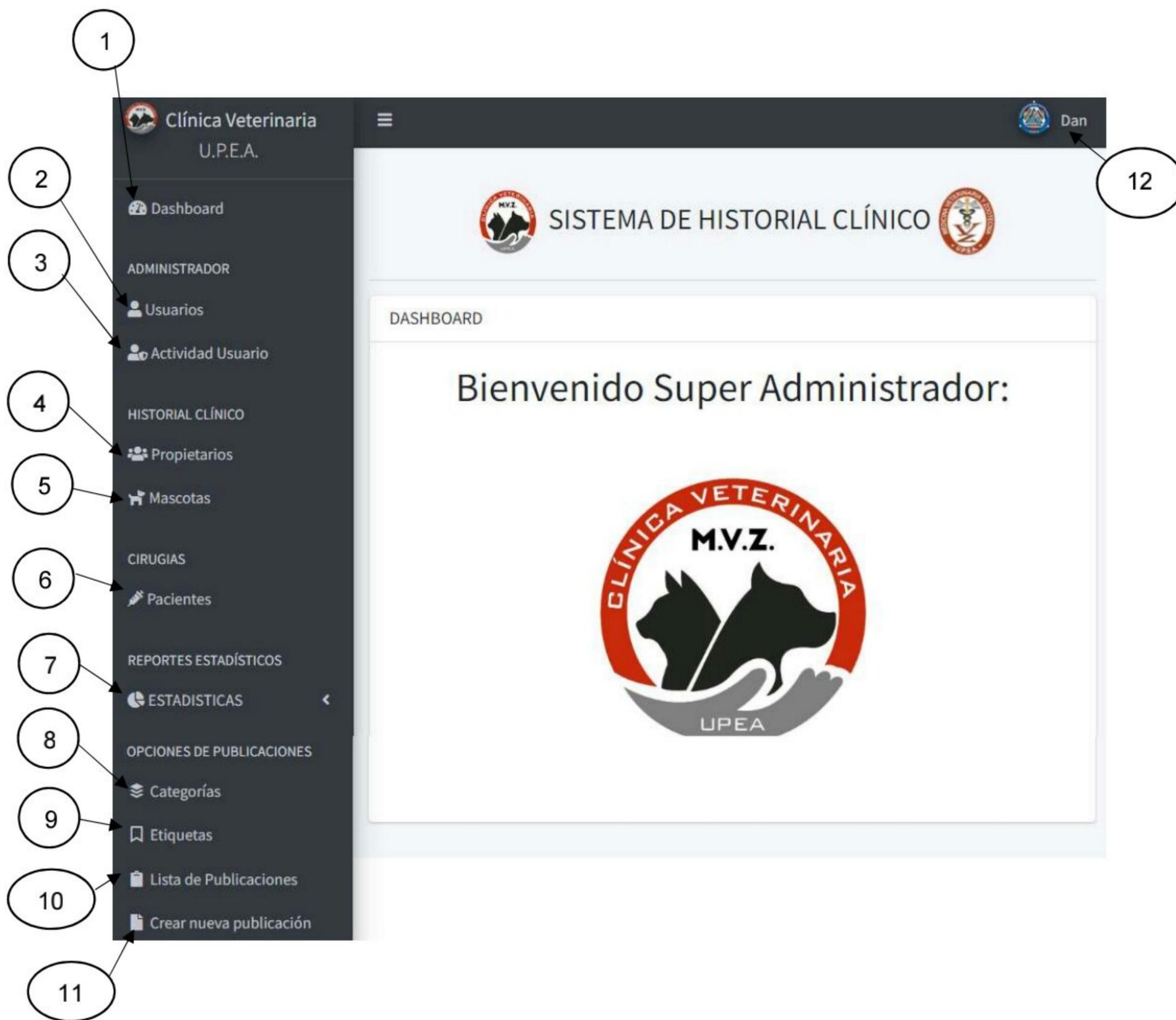
En esta pantalla el usuario puede loguearse de acuerdo al rol que se tenga, donde se requiere el email y la contraseña correspondiente.

The image shows a login interface for 'CLINICA VETERINARIA UPEA'. At the top, there is a logo featuring two black silhouettes of dogs facing each other, with the text 'CLINICA VETERINARIA M.V.Z. UPEA' around them. Below the logo, there are two input fields. The first is labeled 'Correo electrónico' and has a circled '1' next to it with an arrow pointing to the field. The second is labeled 'Password' and has a circled '2' next to it with an arrow pointing to the field. Below these fields is a checkbox labeled 'Recordarme' and a blue button labeled 'INICIAR SESIÓN'.

Pantalla de Ingreso al Sistema (Login)

1: Campo Email: Introducir el email del usuario registrado.

2: Campo Contraseña: Introducir la contraseña de usuario.



Pantalla de inicio (Home)

1: Dashboard: Pantalla Inicio

2: Usuarios: Opción que muestra la lista de usuarios.

3: Actividad Usuario: Opción que muestra las actividades que realizan los usuarios.

4: Propietarios: Opción que muestra la lista de propietarios.

5: Mascotas: Opción que muestra la lista de mascotas.

6: Pacientes: Opción que muestra la lista de mascotas que entraron a cirugía.

7: Estadísticas: Opción que muestra los reportes estadísticos.

- 8: Categorías: Opción que muestra la lista de categorías de las publicaciones.
- 9: Etiquetas: Opción que muestra la lista de etiquetas de las publicaciones.
- 10: Lista de Publicaciones: Opción que muestra la lista de publicaciones.
- 11: Nueva Publicación: Opción que muestra el formulario para un nuevo registro.
- 12: Nombre de usuario: foto del usuario, nombre del usuario y rol, además la opción para salir de la sesión.

b) Pestaña Usuarios

The screenshot shows the 'Lista de Usuarios' page. The sidebar on the left contains navigation options like 'Dashboard', 'ADMINISTRADOR', 'Usuarios', 'Actividad Usuario', 'HISTORIAL CLÍNICO', 'Propietarios', 'Mascotas', 'CIRUGIAS', 'Pacientes', 'REPORTES ESTADÍSTICOS', 'ESTADÍSTICAS', and 'OPCIONES DE PUBLICACIONES'. The main content area has a title 'Lista de Usuarios' and a green '+ Nuevo Usuario' button. Below this is a 'Mostrar' dropdown set to '10 registros por página' and a search bar labeled 'Buscar:'. A table lists three users with columns for 'No', 'Usuario', 'Rol', 'Foto', 'Email', and 'Acciones'. The 'Acciones' column contains three icons: a green 'ver datos' icon, a blue 'editar' icon, and a red 'eliminar' icon. Numbered callouts 1 through 6 are placed around the interface to identify these key elements.

Lista Principal De Usuarios

- 1. Botón de Nuevo Usuario: Opción que dirige al formulario de registro de usuario.
- 2. Selector de cantidad de datos: Opción para seleccionar el número de registros que desean ser visualizados.
- 3. Buscador: Buscador de usuarios, de acuerdo a los campos de la lista.
- 4. Botón de ver datos: opción que dirige a los datos completos del Usuario seleccionado.
- 5. Botón de Editar: opción que dirige al formulario de modificación los datos del Usuario seleccionado que se desea modificar.

6. Botón de Eliminar: opción que elimina los datos del Usuario que se desea seleccionado.

Registro de Usuario

The screenshot shows the 'Registro de Usuarios' form. At the top, there are two logos: the MVZ logo on the left and the UPEL logo on the right. The title 'SISTEMA DE HISTORIAL CLÍNICO' is centered between them. Below the logos, the text 'Registro de Usuarios' is displayed. A blue button labeled 'Volver' is positioned to the left of the form fields. The form contains several input fields: 'Apellidos:' and 'Nombres:' are text boxes; 'Foto del Usuario:' is a large grey box with a white silhouette of a person, and below it is a file selection area with a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'; 'Correo Electronico', 'Password', and 'Confirmar Password' are text boxes; and 'Rol:' is a dropdown menu currently showing 'super-admin'. A blue button labeled 'Guardar Usuario' is at the bottom of the form.

Información del Usuario

The screenshot shows the 'Información del Usuario' page. At the top, there are two logos: the MVZ logo on the left and the UPEL logo on the right. The title 'SISTEMA DE HISTORIAL CLÍNICO' is centered between them. Below the logos, the text 'Datos del Usuario' is displayed. A blue button labeled 'Volver' is positioned to the right of the title. Below the title, there is a black header bar with the text 'DATOS DEL USUARIO' in white. The user information is displayed in a table-like format:

Apellidos:	Mamani Casas	Usuario:	Dan
Rol	super-admin	Email:	dancruzito@gmail.com

At the bottom right of the page, there is a logo for 'INSTITUTO DE SISTEMAS UPEL'.

Editar Usuario

SISTEMA DE HISTORIAL CLÍNICO

Editar Usuario[Volver](#)

Apellidos:

Usuario:

Foto del Usuario:



Seleccionar archivo Ningún archivo seleccionado

Correo Electronico

Rol:

[Editar Usuario](#)

c) Opción Actividad Usuario

Lista de todas las actividades de todos los usuarios

Showing 1 USUARIO to 10 of 26 records

TIPO LOG: LISTA: FECHA DESDE: FECHA HASTA:

ID	FECHA Y HORA	TIPO LOG	USUARIO	ACCIÓN
26	2021-11-25T18:19:34.000000Z - hace 1 hora	<input type="button" value="edit"/> from mascotas	Guadalupe kmamani09@gmail.com	<input type="button" value="VER"/>
25	2021-11-25T18:18:33.000000Z - hace 1 hora	<input type="button" value="edit"/> from propietarios	Guadalupe kmamani09@gmail.com	<input type="button" value="VER"/>
24	2021-11-25T17:56:49.000000Z - hace 1 hora	<input type="button" value="edit"/> from diagnosticos	Guadalupe kmamani09@gmail.com	<input type="button" value="VER"/>
23	2021-11-25T15:13:10.000000Z - hace 4 horas	<input type="button" value="edit"/> from mascotas	Guadalupe kmamani09@gmail.com	<input type="button" value="VER"/>
22	2021-11-25T15:11:40.000000Z - hace 4 horas	<input type="button" value="edit"/> from propietarios	Guadalupe kmamani09@gmail.com	<input type="button" value="VER"/>
21	2021-11-25T15:00:06.000000Z - hace 4 horas	<input type="button" value="edit"/> from diagnosticos	Guadalupe kmamani09@gmail.com	<input type="button" value="VER"/>
20	2021-11-25T14:35:10.000000Z - hace 4 horas	<input type="button" value="login"/>	Dan dancruzito@gmail.com	<input type="button" value="VER"/>
19	2021-11-25T13:35:58.000000Z - hace 5 horas	<input type="button" value="login"/>	Dan dancruzito@gmail.com	<input type="button" value="VER"/>
18	2021-11-25T13:31:02.000000Z - hace 5 horas	<input type="button" value="edit"/> from propietarios	Guadalupe kmamani09@gmail.com	<input type="button" value="VER"/>

A continuación hacemos click en ver en cualquier opción para ver los detalles.

The modal window displays the following information:

INFO	
Tipo	<input type="button" value="edit"/>
Lista	propietarios
Fecha	hace 1 hora - 2021-11-25T18:18:33.000000Z
Usuario	Guadalupe - kmamani09@gmail.com

CAMPO	PREVIOUS	CURRENT
id	40	40
slug		
apellidos	Alberto Molina	Apoma
nombres	Jhaneth	Brigida
ci	040	040
direccion	Z/ Rosas Pampa	Z/ Mercedario
telefono	77783123	61148702

d) Opción Administración de Propietarios

The screenshot displays the 'SISTEMA DE HISTORIAL CLÍNICO' interface. The sidebar menu on the left includes options like 'Dashboard', 'ADMINISTRADOR', 'Usuarios', 'Actividad Usuario', 'HISTORIAL CLÍNICO', 'Propietarios' (highlighted), 'Mascotas', 'CIRUGIAS', 'Pacientes', 'REPORTES ESTADÍSTICOS', 'ESTADISTICAS', and 'OPCIONES DE PUBLICACIONES'. The main content area shows a table of owners with columns for 'No', 'Apellidos', 'Nombres', 'C.I.', and 'Acciones'. The table contains four rows of data. Callouts 1 through 7 identify key UI elements: 1 points to the 'Agregar Propietario' button; 2 points to the 'Mostrar 10 registros por página' dropdown; 3 points to the search input field; 4 points to the 'Mostrar datos' icon; 5 points to the 'Editar' icon; 6 points to the 'eliminar' button; and 7 points to the 'Imprimir' button.

No	Apellidos	Nombres	C.I.	Acciones
1	Aruquipa	Juan Carlos	046	[Mostrar datos] [Editar] [eliminar]
2	Apaza Astorillo	Olga	045	[Mostrar datos] [Editar] [eliminar]
3	Ayala	Yeaneth	044	[Mostrar datos] [Editar] [eliminar]
4	Hilaya Choque	Daniel	043	[Mostrar datos] [Editar] [eliminar]

Lista Principal de Propietarios

1. Botón de Nuevo Propietario: Opción que dirige al formulario de registro de propietario.
2. Selector de cantidad de datos: Opción para seleccionar el número de registros por página.
3. Buscador: Buscador en tiempo real, de acuerdo a los campos de la lista.
4. Botón de mostrar datos: opción que dirige a los datos del Propietario seleccionado.
5. Botón de Editar: opción que dirige al formulario de editar Propietario seleccionado.
6. Botón de Eliminar: opción para eliminar los datos del Propietario seleccionado.
7. Botón de Impresión: Imprime la lista de todos los Propietario registrados.

Registro de Propietario

The screenshot shows the 'Registro de Propietarios' form. The header includes the application name 'SISTEMA DE HISTORIAL CLÍNICO' and the user 'Dan'. The sidebar on the left lists navigation options: Dashboard, ADMINISTRADOR (Usuarios, Actividad Usuario), HISTORIAL CLÍNICO (Propietarios, Mascotas), and CIRUGIAS (Pacientes). The form fields are: Apellidos, Nombres, C.I., Dirección, and Telefono. A 'Siguiete' button is at the bottom left, and a 'Volver' button is at the top right.

Datos del Propietario

Donde nos muestra los datos del propietario, y las mascotas registradas pertenecientes a este.

The screenshot shows the 'Datos del propietario' section with a table of owner information and the 'Mascotas del propietarios' section with a table of registered pets. A 'Volver' button is at the top right, and an 'Agregar mascota' button is at the top right of the pet section.

Datos del propietario					
Apellidos	Aruquipa	Nombres	Juan Carlos	C.I.	046
Telefono	74944555	Dirección	Z/Villa Dolores C/2 N°50		

Mascotas del propietarios				
Cod	Nombre	Especie	Raza	Historial
VET-000046	Beca	Canina	Mestizo	Historial

Editar Datos Propietario



SISTEMA DE HISTORIAL CLÍNICO



Editar Propietario

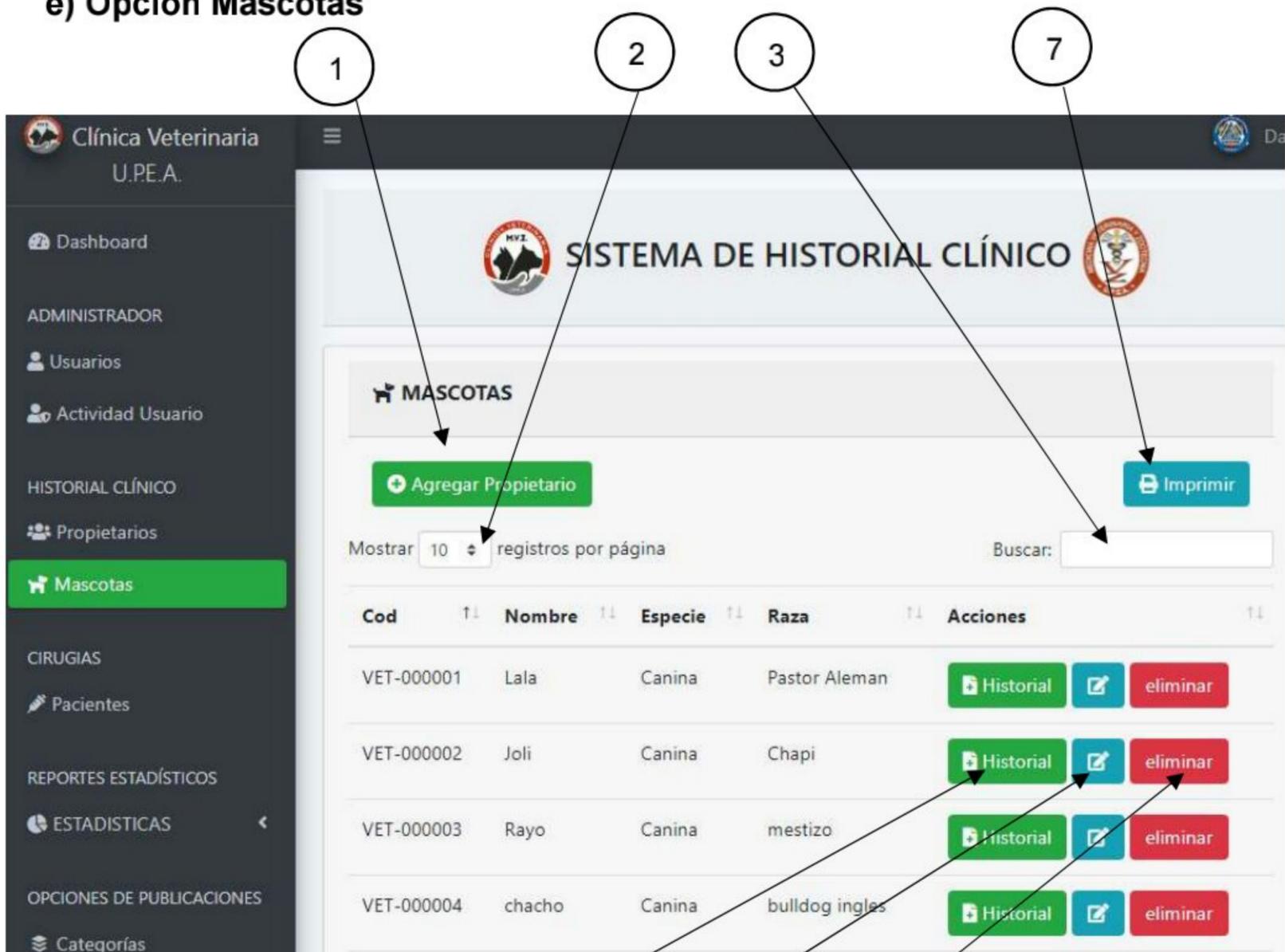
Volver

Apellidos **Nombres**

C.I. **Dirección** **Telefono**

Guardar

e) Opción Mascotas



Clínica Veterinaria U.P.E.A.

Dashboard

ADMINISTRADOR

Usuarios

Actividad Usuario

HISTORIAL CLÍNICO

Propietarios

Mascotas

CIRUGIAS

Pacientes

REPORTES ESTADÍSTICOS

ESTADÍSTICAS

OPCIONES DE PUBLICACIONES

Categorías



SISTEMA DE HISTORIAL CLÍNICO



MASCOTAS

Agregar Propietario Imprimir

Mostrar 10 registros por página

Buscar:

Cod	Nombre	Especie	Raza	Acciones
VET-000001	Lala	Canina	Pastor Aleman	Historial eliminar
VET-000002	Joli	Canina	Chapi	Historial eliminar
VET-000003	Rayo	Canina	mestizo	Historial eliminar
VET-000004	chacho	Canina	bulldog ingles	Historial eliminar

1

2

3

4

Lista principal de Mascotas

1. Botón de nuevo Propietario: Opción que dirige al formulario de registro de Propietarios.
2. Selector de cantidad de datos: Opción para seleccionar el número de registros a visualizar por página.
3. Buscador: Buscador en tiempo real, de acuerdo a los campos de la lista.
4. Botón de mostrar Historial: opción que dirige al Historial Clínico de la Mascota seleccionada.
5. Botón de Editar: opción que dirige a los datos de la mascota seleccionada que se desea modificar.
6. Botón de Eliminar: opción para eliminar los datos de la Mascota seleccionada
7. Botón de Impresión: Se genera el reporte de la lista de todas las mascotas registrada.

Registro de Mascota

Tomar en cuenta, que para acceder a este formulario de registro se debe de registrar primero al Propietario de este.

Editar Datos de la Mascota

SISTEMA DE HISTORIAL CLÍNICO

Editar Mascota

Volver

Nombre	Especie	Edad
Lala	Canina	10 años
Raza	Color	sexo
Pastor Aleman	Amarillo Beige	Hembra

Guardar

Mostrar Historial

[Volver](#) [Agregar Historial](#) [Imprimir](#)

 **Universidad Pública de El Alto**
CARRERA DE MEDICINA VETERINARIA Y ZOOTECNIA
CLÍNICA VETERINARIA U.P.E.A. 

HISTORIA CLÍNICA

COD: VET-000001

***DATOS DEL PROPIETARIO**

Apellidos: Aro Nombre(s): Wilfredo
Dirección: Z. Villa Ingenio, Calle Copacabana, #2064 Telefono: 79119375

***DATOS DE LA MASCOTA (Paciente)**

Nombre: Lala Especie: Canina Edad: 10 años
Raza: Pastor Aleman Color: Amarillo Beige Sexo: Hembra

Fecha: 2020-10-30 [Editar](#) [Eliminar](#)

T° (°C): 37.4 Peso(Kg): 28.65 FC/min: 120
FR/min: 60 Mucosas: Rosadas Hid.Tisular(Seg): 3

SIGNOS Y SÍNTOMAS CLÍNICOS (ANAMNESIS): Desde el lunes ya no tiene apetito, el miercoles se postro, presenta lagañas y neoplasia inguinal en izquierdo, de tamaño centrado y hiperqueratinizado en los pulpejos.

DIAGNÓSTICO CLINICO: NN

TERAPÉUTICA EN CONSULTORIO VACUNA/DESP/OTROS: Comp. B = 4ml Dexametazona 0.15 mg/Kg = 2.14 Gentamicina 20mg/Kg = 3.81 ml I.M.

TERAPÉUTICA AMBULATORIA: [Agregar](#)

Producto(N. comercial o principio activo)	mg/ml	Frecuencia	Número de días	Acciones
---	-------	------------	----------------	----------

OBSERVACIONES:

ANÁLISIS O EXAMENES SOLICITADOS:

RECOMENDACIONES:

Opciones principales de Historial Clínico

1. Botón de agregar historial: Opción que abre un formulario emergente de nuevo diagnóstico.
2. Reporte de Historial Clínico de la mascota: Opción que genera el reporte pdf del Historial Clínico.
3. Opción Editar Diagnóstico: Opción que abre una ventana emergente para editar los datos del diagnóstico.

4. Botón de eliminar Diagnóstico: Opción que dirige al Historial Clínico de la Mascota seleccionada.

5. Botón de Agregar: Opción que abre una ventana emergente para agregar terapéutica ambulatoria.

Formulario Agregar Diagnóstico

Diagnóstico

Fecha
dd/mm/aaaa

T° (°C)

Peso (Kg)

FC/min

FR/min

Mucosas

Hid. Tisular (seg)

SIGNOS Y SÍNTOMAS CLÍNICOS(ANAMNESIS)

DIAGNÓSTICO CLÍNICO (PRESUNTIVO)

TERAPÉUTICA EN CONSULTORIO/VACUNAS/DESP/OTROS

Observaciones...

Análisis o Exámenes Solicitados...

Recomendaciones...

Guardar Cancelar

Formulario Editar Diagnóstico

Terapéutica Ambulatoria ✕

Fecha
30/10/2020 📅

T° (°C) 37.4	Peso (Kg) 28.65	FC/min 120
FR/min 60	Mucosas Rosadas	Hid. Tisular (seg) 3

SIGNOS Y SÍNTOMAS CLÍNICOS (ANAMNESIS)

Desde el lunes ya no tiene apetito, el miércoles se postro, presenta lagañas y neoplasia inguinal en izquierdo, de tamaño centrado y hiperqueratinizado en los pulpejos.

DIAGNÓSTICO CLÍNICO (PRESUNTIVO)

NN

TERAPÉUTICA EN CONSULTORIO/VACUNAS/DESP/OTROS

Comp. B = 4ml Dexametazona 0.15 mg/Kg = 2.14 Gentamicina 20mg/Kg = 3.81 ml I.M.

Observaciones...

Análisis o Exámenes Solicitados...

Recomendaciones...

Guardar **Cancelar**

FR/min: 32 Mucosas: Palida Hid.Tisular(Seg):

Formulario Agregar Terapéutica Ambulatoria

Terapéutica Ambulatoria

Producto

mg/ml

Frecuencia

días

Agregar Cancelar

Wilfredo
79119375
Edad:
Sexo:
FC/min
Hid.Tisular(Seg):
Mucosas: Rosadas
Desde el lunes ya no tiene apetito, el miércoles se postro; present

f) Opción Publicación de Adopciones

Listado de publicaciones

+ Nueva Publicación

Ingrese el nombre de una publicación

ID	Nombre	Editar	Eliminar
5	Lulu	Editar	Eliminar
4	Bandido	Editar	Eliminar
3	Canela	Editar	Eliminar
2	Rambo	Editar	Eliminar
1	Beily	Editar	Eliminar

1

2

3

4

Opciones principales de Publicación

1. Botón de Búsqueda de Publicaciones: Opción que busca las publicaciones registradas en tiempo real.
2. Botón de nueva Publicación: Opción que abre un formulario para registrar una nueva publicación.
3. Opción Editar Publicación: Opción que abre una ventana emergente para editar los datos de la Publicación.
4. Botón de eliminar Publicación: Elimina una publicación seleccionada.

Formulario Publicación

Crear nueva publicación

Nombre

Slug

Categoría

Adopciones ▼

Etiquetas

Canino Felino

Estado

Borrador Publicado

Imagen que se mostrará en la publicación

Ningún archivo seleccionado

Subir una foto de la mascota, que se quiere dar en adopción, si no tuviese una se asignará una por defecto.



Introducción

Paragraph ▼ **B** *I*          

Cuerpo del publicación:

Paragraph ▼ **B** *I*          

Editar Publicación

Nombre

Slug

Categoría

Etiquetas

Canino Felino

Estado

Borrador Publicado

Imagen que se mostrará en la publicación

Ningún archivo seleccionado

Subir una foto de la mascota, que se quiere dar en adopción, si no tuviese una se asignará una por defecto.



Introducción

Paraphon

Extravió de Lulu

Cuerpo del publicación:

Paraphon

El pasado viernes 12 de junio, por la 16 de julio lulu se perdió, no conoce muy bien la calle. Por favor cualquier dato a la Clínica.

SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES



MANUAL TÉCNICO

AUTOR: DAN CRUZ MAMANI CASAS

VERSIÓN 1.0
2021

1. PRINCIPALES REQUERIMIENTOS

1.1 Requerimientos Funcionales

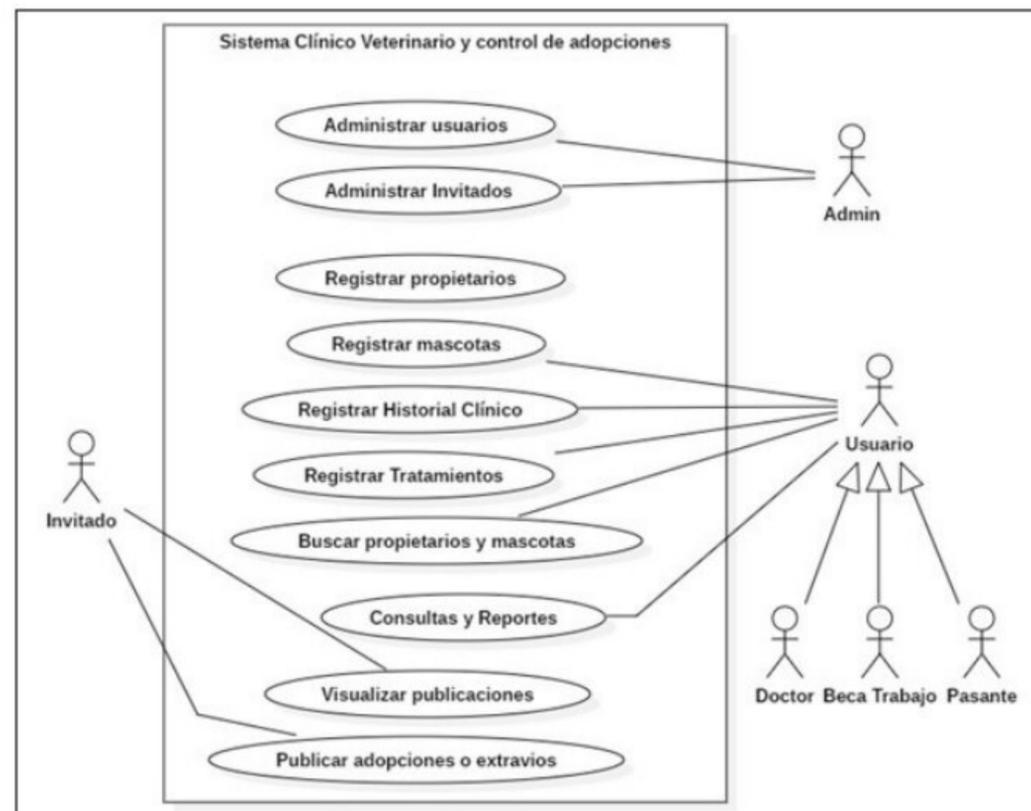
- Acceso al Sistema mediante verificación de datos de acceso.
- Administrar los roles y permisos de acuerdo a los usuarios registrados.
- Administración de Propietarios.
- Administración de Mascotas.
- Tratamiento Clínico.
- Reporte Estadístico.
- Reporte Historial Clínico.
- Administración de Adopciones.

1.2 Requerimientos No Funcionales

- Sistema Compatible con los diferentes navegadores.
- Interfaz gráfica debe responder al usuario cuando lo requiera
- Transacciones deben responder las necesidades de los usuarios.
- Soporte y Mantenimiento, para disponibilidad del sistema.

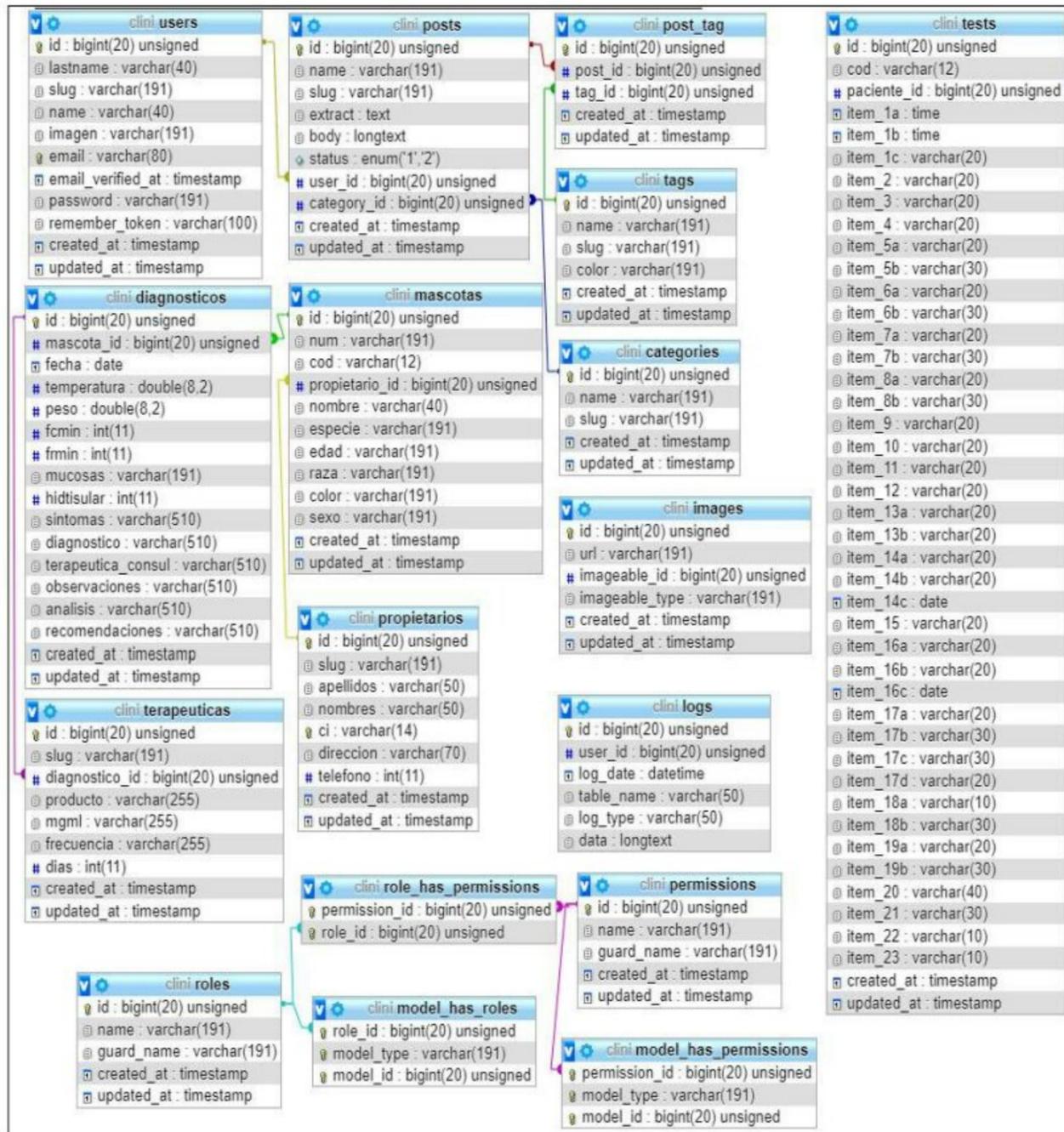
2 DISEÑO DEL SISTEMA

2.1 Caso de Uso General



3 BASE DE DATOS

3.1 Modelo de Base de Datos Relacional



4 HERRAMIENTA DE DESARROLLO

Las herramientas para el desarrollo del sistema serán los siguientes:

➤ Hypertext Pre.Procesor “PHP”

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

➤ MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo

licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation.

➤ **Laravel**

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP, basado en el modelo vista controlador, intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP

5 REQUERIMIENTO E INSTALACIÓN

5.1 Requerimientos

A continuación se lista los requerimientos del sistema para la implementación:

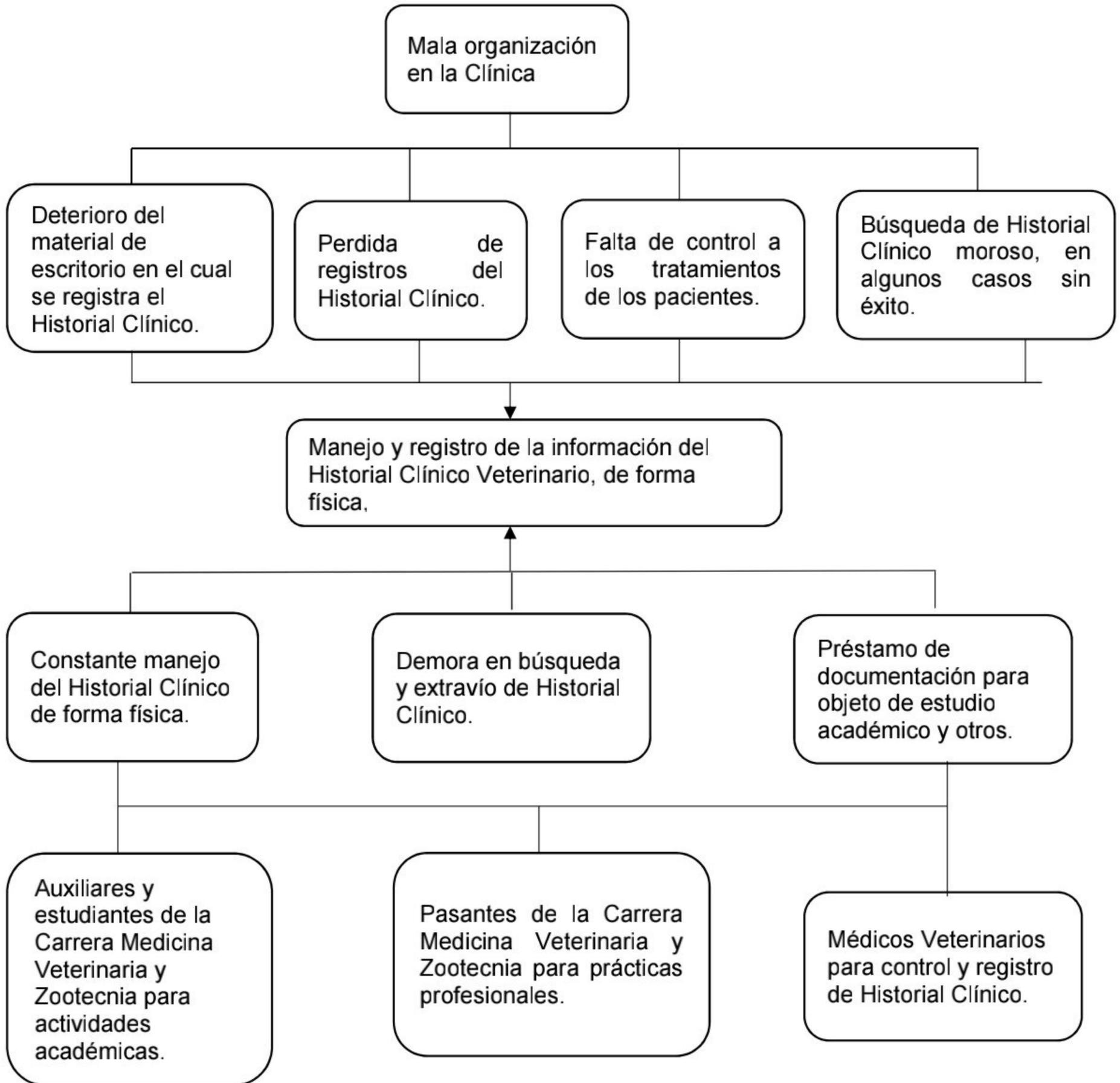
- PHP versión mínima 7.2 para arriba.
- Servidor Apache instalado.
- Acceso a Terminal para ejecución de comandos.
- Composer.
- Node JS
- Acceso a Terminal para ejecución de comandos.

El sistema cuenta con un flujo considerable por lo que se recomienda tener la capacidad de acuerdo a la siguiente especificación:

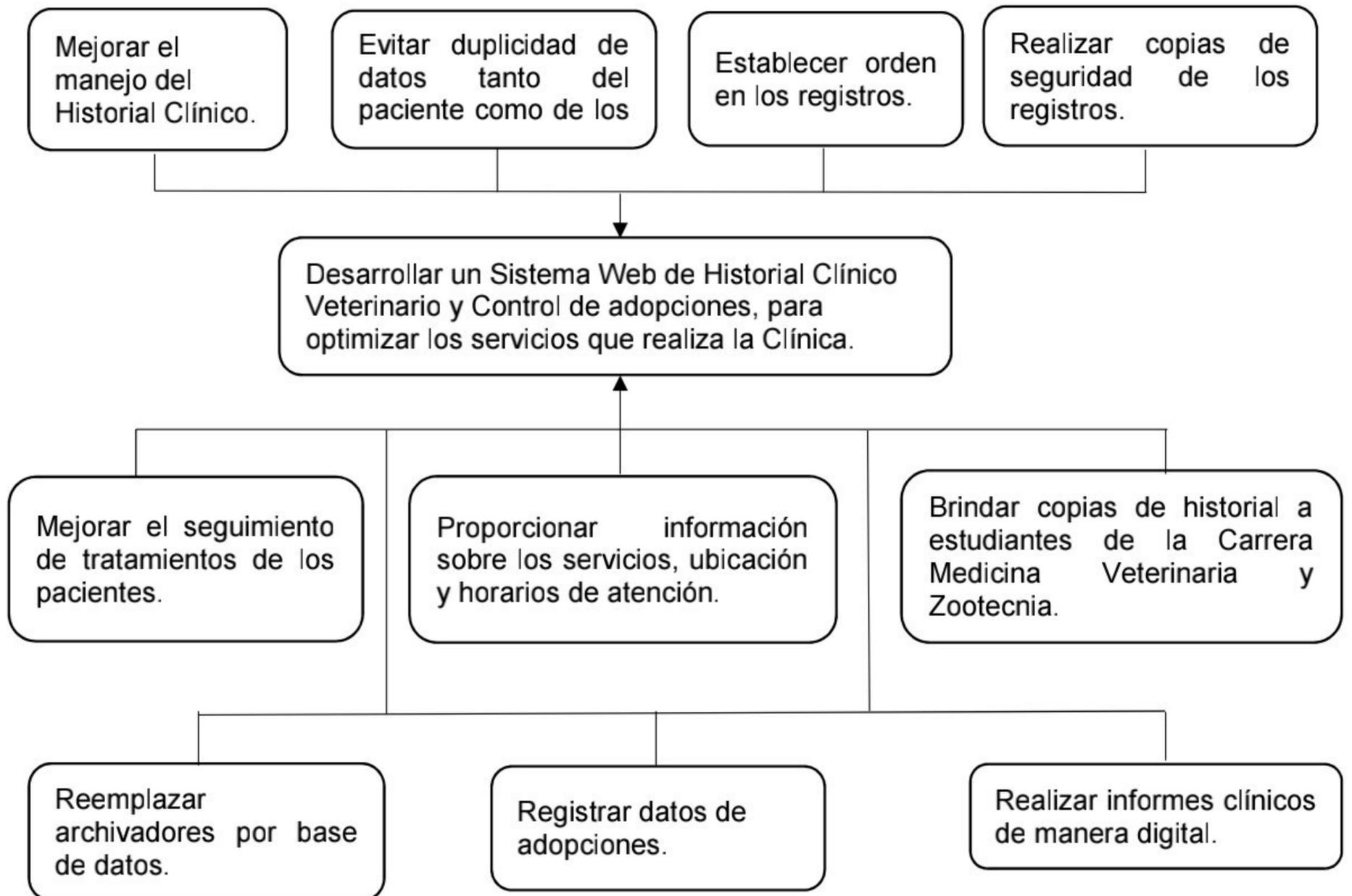
- 20 GB de almacenamiento como mínimo.

ANEXOS

Anexo A. Árbol de Problemas



Anexo B. Árbol de Objetivos



Anexo C. Herramienta De Recolección De Datos

Este cuestionario muestra las preguntas que se aplicaron en la entrevista que se realizó al M.V.Z. Lic. Cecilio Coaricona Nina, Director Clínica U.P.E.A. para obtener información del panorama de la clínica.

1. ¿Qué servicios brinda la Clínica Veterinaria U.P.E.A.?
2. ¿Quiénes son parte del personal y que cargos jerárquicos tienen dentro de la Clínica?
3. ¿De qué manera se realiza el manejo de archivos?
4. ¿Quiénes tienen acceso a los archivos de Historial Clínico?
5. ¿Quiénes acceden a los servicios de la Clínica?
6. Podría describir los campos del formulario de Historial del Clínico
7. ¿Cuáles son los servicios de mayor frecuencia y sus precios?
8. ¿Con que herramientas tecnológicas cuenta la Clínica?
9. ¿Qué otros servicios se brinda a parte de los servicios clínicos?
10. ¿De qué manera ayudan a las mascotas que se deseen dar en adopción?

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, **Dan Cruz Mamani Casas** estudiante con C.I. 6933062 L.P. mediante la presente declaro de manera pública que la propuesta del **TRABAJO DE GRADO** titulada “**SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES**” es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados,

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, Noviembre del 2021.



Dan Cruz Mamani Casas
C.I.: 6933062 L.P.
email: dancruzito@gmail.com

El Alto, noviembre de 2021

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR DE LA CARRERA – INGENIERÍA DE SISTEMAS

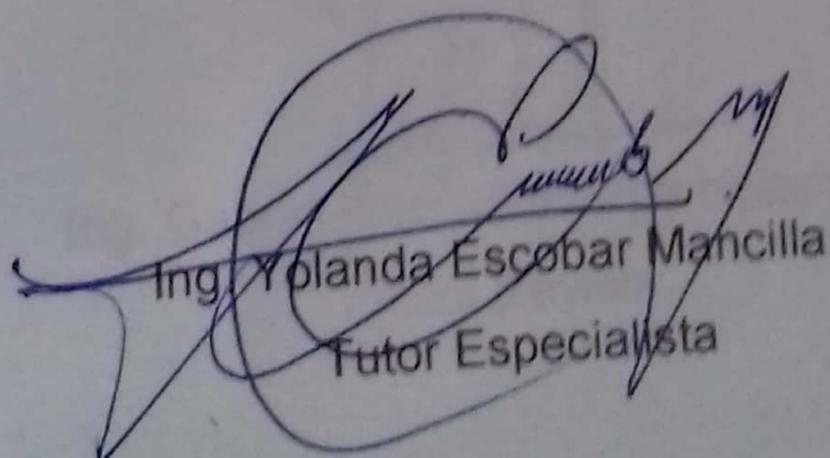
Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, mi conformidad con el Proyecto de Grado denominado: **"SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, CARRERA MEDICINA VETERINARIA Y ZOOTECNIA, CLINICA VETERINARIA U.P.E.A."**, que propone el postulante Univ.: **Dan Cruz Mamani Casas**, con C.I.: **6933062 L.P.** y R.U.: **15000109**, para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.


Ing. Yolanda Escobar Mancilla
Tutor Especialista

El Alto, noviembre de 2021

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR DE LA CARRERA – INGENIERÍA DE SISTEMAS

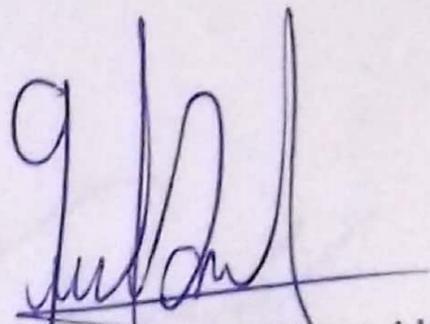
Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, mi conformidad con el Proyecto de Grado denominado: **"SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, CARRERA MEDICINA VETERINARIA Y ZOOTECNIA, CLINICA VETERINARIA U.P.E.A."**, que propone el postulante Univ.: **Dan Cruz Mamani Casas**, con C.i.: **6933062 L.P.** y R.U.: **15000109**, para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.



Ing. Gabriel Reynaldo Sirpa Huayhua
Tutor Revisor

El Alto, noviembre de 2021

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR DE LA CARRERA – INGENIERÍA DE SISTEMAS

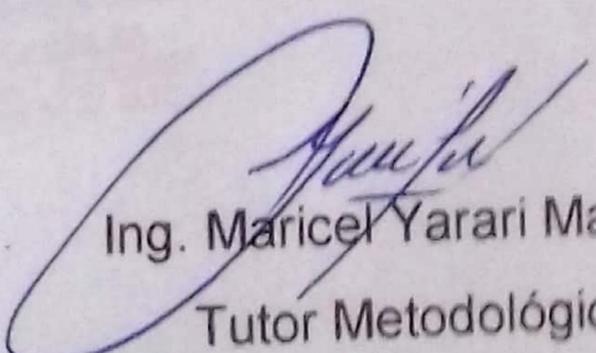
Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, mi conformidad con el Proyecto de Grado denominado: "**SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, CARRERA MEDICINA VETERINARIA Y ZOOTECNIA, CLINICA VETERINARIA U.P.E.A.**", que propone el postulante Univ.: **Dan Cruz Mamani Casas**, con C.i.: **6933062 L.P.** y R.U.:**15000109**, para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.



Ing. Maricel Yarari Mamani

Tutor Metodológico

Universidad Pública De El Alto

Creada por ley 2115 de 5 de septiembre del 2000 y Autónoma por ley 2556 de 12 de Noviembre del 2033

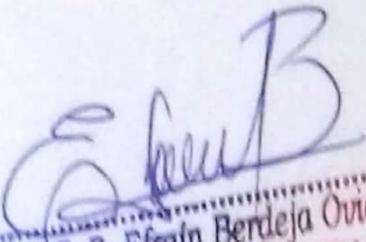
El Suscrito M.V.Z. Lic. R. Efraín Berdeja Ovidio, Director de la Carrera Medicina Veterinaria y Zootecnia de la Universidad Pública de El Alto, a petición del interesado:

CERTIFICA

Que el Sr. **DAN CRUZ MAMANI CASAS** con Cédula de Identidad N° **6933062** expedido en la ciudad de La Paz y Registro Universitario N° 15000109, estudiante de la Carrera de **INGENIERÍA DE SISTEMAS**, formalizó la implementación del Sistema a la Clínica Veterinaria U.P.E.A.

El proyecto de Grado titulado: **"SISTEMA WEB DE GESTIÓN DE HISTORIAL CLÍNICO VETERINARIO Y CONTROL DE ADOPCIONES, CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, CARRERA MEDICINA VETERINARIA Y ZOOTECNIA, CLINICA VETERINARIA U.P.E.A."**

Es cuanto tengo a bien certificar en honor a la verdad y para fines que convengan al interesado.


Lic. M.V.Z. R. Efraín Berdeja Ovidio
DIRECTOR DE CARRERA
MEDICINA VETERINARIA Y ZOOTECNIA
UNIVERSIDAD PÚBLICA DE EL ALTO

El Alto, noviembre de 2021