



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SISTEMAS

**DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE
LA VETERINARIA “ANIMAL’S VET”**

TRABAJO DE TITULACIÓN
TIPO: PROYECTO TÉCNICO

Presentado para optar al grado académico de:
INGENIERO EN SISTEMAS INFORMÁTICOS

AUTORES: CARLOS JOHNNY GREFA AJON
BERNABE SAMUEL ARAGON VARGAS
DIRECTORA: DRA. NARCISA DE JESUS SALAZAR

Riobamba-Ecuador
2019

@2019, Carlos Johnny Grefa Ajon, Bernabé Samuel Aragón Vargas

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal de Tesis certifica que: El trabajo de Investigación: “DESARROLLO DE UN SISTEMA WEB PARA LA GESTION DE PROCESOS DE LA VETERINARIA ANIMAL’S VET”, de responsabilidad de los señores Carlos Johnny Grefa Ajon y Bernabé Samuel Aragón Vargas, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

FIRMA

FECHA

Ing. Washington Luna E. PhD
**DECANO DE LA FACULTAD
DE INFORMÁTICA Y
ELECTRÓNICA**

Ing. Patricio Moreno C.
**DIRECTOR DE LA ESCUELA DE
INGENIERÍA EN SISTEMAS**

Dra. Narcisa de Jesús Salazar
**DIRECTORA DEL TRABAJO
DE TITULACIÓN**

Ing. Gloria Arcos Medina Mgs.
MIEMBRO DEL TRIBUNAL

Nosotros Carlos Johnny Grefa Ajon con el número de cedula 1803898970 y Bernabé Samuel Aragón Vargas con el número de cedula 1600488678, somos responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Como autores reconozco la responsabilidad legal y académica de los contenidos del trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Carlos Johnny Grefa Ajon
Bernabé Samuel Aragón Vargas

AGRADECIMIENTO

Agradezco primeramente a Dios por darme salud y vida, a mi familia por tener la paciencia y el esfuerzo que hicieron para poder darme el estudio, y el estar siempre junto a mí en los momentos difíciles que se llegó a pasar al estar lejos de casa. El sendero no fue fácil seguir pero gracias a los buenos amigos, consejos y vivencias de mi familia se pudo llegar a concluir una meta más propuesta.

Agradecer a la Dra. Narcisa Salazar y la Ing. Gloria Arcos directoras de tesis, por su tiempo y paciencia que nos guiaron a culminar de mejor manera este trabajo de titulación.

Carlos

Agradezco a Dios por darme salud y vida para llegar a este momento. Al apoyo y amor que me brindaron mis padres. A mis amigos que me acompañaron y apoyaron en toda la carrera. En especial a Evelyn Sánchez, quien estuvo conmigo en cada paso que di durante este esforzado camino. A los docentes que supieron impartir sus conocimientos.

Bernabé

DEDICATORIA

El presente trabajo va dedicado en primer lugar a Dios, que me dio salud, fuerza y mucha sabiduría para terminar esta etapa de mi vida, gracias a su infinita gracia, pude llegar hasta donde estoy. También agradecer a mis padres, que supieron enseñarme el valor del esfuerzo y perseverancia, a nunca rendirme a pesar todo, además de ser mi apoyo, motivándome siempre a salir adelante.

Bernabé

Quiero dedicar este logro a mis padres Carlos y Judith por apoyarme en todo momento por los valores que me han inculcado por sus consejos y su paciencia todo lo que he conseguido es gracias a ellos. El sendero no fue fácil viví momentos difíciles que se llegó al estar lejos de casa pero ignoré todo tipo de obstáculo y cumplí uno de mis más anhelados objetivos. Finalmente agradezco a los docentes, compañeros y amigos (Jasón Gómez y J.A) por todo el apoyo brindado a lo largo de la carrera, por su tiempo, amistad y por los conocimientos que me transmitieron.

Carlos

TABLA DE CONTENIDO

ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS	xii
RESUMEN	xvii
ABSTRACT	xviii
INTRODUCCIÓN	1
CAPÍTULO I	6
1. MARCO TEÓRICO REFERENCIAL	6
1.1. Procesos en una veterinaria	6
1.2. Trabajos relacionados	6
1.3. SCRUM	7
1.4. Arquitectura MVC	9
1.5. NetBeans IDE 8.2	11
1.6. Java	11
1.7. JavaScript	12
1.8. Payara Server	13
1.9. Base de Datos PostgreSQL	14
1.10. Sistema web	15
1.11. Diseño de un sistema web	15
1.12. Hojas de Estilo	16
1.13. FRAMEWORK	17
1.14. Css Grid	18
1.15. Calidad de software	19
1.16. ISO 25010-Mantenibilidad	20
CAPÍTULO II	24
2. MARCO METODOLÓGICO	24
2.1. Métodos y técnicas	24
2.2. Herramientas para medir la mantenibilidad	26
2.3. Indicadores Mantenibilidad	27

2.3.1. <i>Analizabilidad</i>	27
2.3.2. <i>Capacidad para ser modificado</i>	28
2.3.3. <i>Reusabilidad-Estabilidad</i>	28
2.3.4. <i>Capacidad para ser probado</i>	28
2.4. Escenarios de Prueba	29
2.5. Población y muestra	29
2.6. Desarrollo del sistema web para la gestión de los procesos de la veterinaria "Animal's Vet"	30
2.6.1. <i>Información de la veterinaria</i>	30
2.6.2. <i>Personas y roles involucrados en el proyecto</i>	30
2.7. Descripción del proceso de la veterinaria.....	31
2.7.1. <i>Especificación de requerimientos</i>	33
2.7.2. <i>Análisis y Gestión de riesgos</i>	36
2.8. Fase de planificación	40
2.8.1. <i>Historias de usuario</i>	41
2.8.2. <i>Product backlog</i>	42
2.8.3. <i>Sprint Product backlog</i>	44
2.8.4. <i>Sprint 1</i>	47
2.8.5. <i>Sprint 2</i>	48
2.8.6. <i>Sprint 3</i>	49
2.8.7. <i>Sprint 4</i>	51
2.8.8. <i>Sprint 5</i>	51
2.9. Fase de desarrollo.....	53
2.9.1. <i>Diseño de la arquitectura</i>	53
2.9.2. <i>Estándar de programación</i>	53
2.9.3. <i>Diseño de interfaces</i>	53
2.9.4. <i>Diseño de la base de datos</i>	54
2.10. Fase de cierre	57
2.10.1. <i>Burndown chart</i>	57

CAPÍTULO III	59
3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS	59
3.1. Analizabilidad	59
3.1.1. <i>Complejidad Ciclomática</i>	59
3.1.2. <i>Densidad de comentarios</i>	60
3.2. Capacidad para ser modificado	61
3.2.1. <i>Escenario 1</i>	61
3.2.2. <i>Escenario 2</i>	62
3.3. Reusabilidad-Estabilidad	64
3.3.1. <i>Escenario 1</i>	64
3.3.2. <i>Escenario 2</i>	65
3.4. Capacidad de ser probado	67
3.4.1. <i>Escenario 1</i>	67
3.4.2. <i>Escenario 2</i>	67
3.5. ANÁLISIS DE RESULTADOS	69
3.5.1. <i>Analizabilidad</i>	69
3.5.2. <i>Capacidad para ser cambiado</i>	71
3.5.3. <i>Reusabilidad-Estabilidad</i>	72
3.5.4. <i>Capacidad de ser probado</i>	73
3.6. Respuestas a preguntas de investigación	76
3.6.1. <i>¿Cómo se realizan actualmente los procesos relacionados la gestión de la veterinaria “Animal’s Vet”?</i>	76
3.6.2. <i>¿Qué tecnologías se va a utilizar para el desarrollo del sistema web para la veterinaria “Animal’s Vet”?</i>	76
3.6.3. <i>¿Cómo son los parámetros para demostrar la mantenibilidad del sistema?</i>	76
CONCLUSIONES	
RECOMENDACIONES	
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1:	Características de PostgreSQL	14
Tabla 2-1:	Tamaño de un proyecto realizado con CSS GRID	19
Tabla 3-1:	Indicadores Mantenibilidad	23
Tabla 1-2:	Herramientas utilizadas para medir la mantenibilidad	26
Tabla 2-2:	Clasificación Complejidad Ciclomática	27
Tabla 3-2:	Clasificación densidad de comentarios.....	27
Tabla 4-2:	Clasificación de la capacidad para ser modificado.....	28
Tabla 5-2:	Clasificación de la reusabilidad-estabilidad	28
Tabla 6-2:	Clasificación de la capacidad de ser probado	29
Tabla 7-2:	Personas y Roles.....	30
Tabla 8-2:	Riesgos Proyecto	36
Tabla 9-2:	Riesgos Proyecto	37
Tabla 10-2:	Riesgos Negocio.....	37
Tabla 11-2:	Rango Probabilidades.....	38
Tabla 12-2:	Impacto Riesgo.....	38
Tabla 13-2:	Probabilidades	38
Tabla 14-2:	Impacto Riesgos	39
Tabla 15-2:	Exposición Riesgo	39
Tabla 16-2:	Exposición Riesgo	40
Tabla 17-2:	Método de estimación T-shirt.....	41
Tabla 18-2:	Prioridad Alta	42
Tabla 19-2:	Prioridad Media.....	43
Tabla 20-2:	Prioridad Baja.....	44
Tabla 21-2:	Sprint Backlog.....	45
Tabla 22-2:	Detalle Sprint 1.....	48
Tabla 23-2:	Detalle Sprint 2.....	49
Tabla 24-2:	Detalle Sprint 3.....	50
Tabla 25-2:	Detalle Sprint 4.....	51
Tabla 26-2:	Detalle Sprint 5.....	52
Tabla 1-3:	Resultados Complejidad Ciclomática.....	69
Tabla 2-3:	Resultados Densidad de comentarios	70
Tabla 3-3:	Resultados Cambiabilidad	71
Tabla 4-3:	Resultados Reusabilidad-Estabilidad	72
Tabla 5-3:	Capacidad para ser probado.....	73

Tabla 6-3: Tabla de Resultados Escenario 1	74
Tabla 7-3: Indicador de evaluación de la mantenibilidad	74
Tabla 8-3: Resumen de resultados Escenario 2.....	75
Tabla 9-3: Indicador de evaluación de la mantenibilidad	75

ÍNDICE DE FIGURAS

Figura 1-1:	Eventos de Scrum	8
Figura 2-1:	Elementos del patrón MVC	10
Figura 3-1:	Partes de una regla Css.	17
Figura 4-1:	CSS GRID LAYOUT	19
Figura 5-1:	ISO/IEC 25010.....	22
Figura 1-2:	Diagrama de procesos historia clínica y cirugía de la veterinaria Animal's Vet.	31
Figura 2-2:	Diagrama de procesos de adopciones de la veterinaria Animal's Vet	32
Figura 3-2:	Diagrama de procesos proveedores, insumos de la veterinaria Animal's Vet.	32
Figura 4-2:	Diseño de la arquitectura	53
Figura 5-2:	Módulos del sistema de la veterinaria Animal's Vet.	54
Figura 6-2:	Diagrama de base de datos de la veterinaria Animal's Vet 1.	55
Figura 7-2:	Diagrama de base de datos de la veterinaria Animal's Vet	55
Figura 8-2:	Diagrama de base de datos de la veterinaria Animal's Vet 3.	56
Figura 9-2:	Diagrama de base de datos de la veterinaria Animal's Vet 4.	56
Figura 10-2:	Diagrama de Despliegue	57
Figura 11-2:	Diagrama de componentes.	57
Figura 1-3:	Complejidad Ciclomática LN.	59
Figura 2-3:	Complejidad Ciclomática AD.....	60
Figura 3-3:	Complejidad Ciclomática IU.	60
Figura 4-3:	Densidad de comentarios Escenario1.	60
Figura 5-3:	Densidad de comentarios Escenario2.	61
Figura 6-3:	Cambiabilidad Acceso a Datos Escenario1	61
Figura 7-3:	Cambiabilidad Lógica de Negocio Escenario1.....	62
Figura 8-3:	Cambiabilidad Interfaz de usuario Escenario1.	62
Figura 9-3:	Cambiabilidad Acceso a Datos Escenario2.	63
Figura 10-3:	Cambiabilidad Lógica de negocios Escenario2.	63
Figura 11-3:	Cambiabilidad Interfaz de Usuario Escenario2.....	64
Figura 12-3:	Reusabilidad-Estabilidad Acceso a Datos Escenario1.....	64
Figura 13-3:	Reusabilidad-Estabilidad Lógica de Negocio Escenario1.	65
Figura 14-3:	Reusabilidad-Estabilidad Interfaz de Usuario Escenario1.....	65
Figura 15-3:	Reusabilidad-Estabilidad Acceso a Datos Escenario2.....	66
Figura 16-3:	Reusabilidad-Estabilidad Lógica de Negocio Escenario2.	66
Figura 17-3:	Reusabilidad-Estabilidad Interfaz de Usuario Escenario2.....	66

Figura 18-3: Capacidad para ser probado Escenario1.	67
Figura 19-3: Capacidad para ser probado Escenario2.	68

ÍNDICE DE GRÁFICOS

Gráfico 1-2: Burndown chart del proyecto 1	58
Gráfico 1-3: Complejidad Ciclomática Resultados.....	69
Gráfico 2-3: Densidad de Comentarios Resultados.....	70
Gráfico 3-3: Cambiabilidad Resultados.....	71
Gráfico 4-3: Reusabilidad-Estabilidad Resultados.....	72
Gráfico 5-3: Capacidad para ser probado Resultados.....	73

ÍNDICE DE ANEXOS

Anexo A: Historias de Usuario

Anexo B: Hoja de Gestión de Riesgo

Anexo C: Diccionario de Datos

Anexo D: Manual de Usuario

ÍNDICE DE ABREVIATURAS

GNU Licencia Pública General

HTML Lenguaje de Marcado de Hipertexto

HTTP HyperText Transfer Protocol

MVC Modelo, Vista, Controlador

REST Representational State Transfer

BD Base de datos

XML Lenguaje de marcado similar

ISO International Organization for Standardization

RESUMEN

En el presente trabajo de titulación el objetivo fue Desarrollar un Sistema Web para automatizar los procesos de la Clínica Veterinaria “Animal’s Vet”, para el cual se aplicó la metodología ágil SCRUM en la que se obtuvo 58 historias de usuario y 12 historias técnicas, a partir de esto se realizó la planificación en la que se determinó la fecha de entrega del sistema funcional al cliente. El sistema cuenta con tres roles de Usuario que son: Administrador, Secretaria y Veterinario para lo cual se desarrolló los siguiente módulos de usuarios: administrador, secretaria, veterinario y proveedor, para desarrollarlos se utilizaron las siguientes herramientas: Lenguaje de programación Java, gestor de base de datos PostgreSQL y el Framework CSS GRID, el sistema se albergó en el servidor de aplicaciones Payara. La estructura de sistema que se utilizó fue el Modelo-Vista-Controlado (MVC), esta estructura permitió tener una fácil estructuración del código. Finalmente se evaluó la mantenibilidad del sistema, con el cual, a partir de varios análisis iniciales sobre la Analizabilidad, Cambiabilidad, Reusabilidad-Estabilidad y Capacidad para ser probado, se obtuvo un porcentaje de Mantenibilidad del 59.75%, a partir de esto se realizaron correcciones al sistema para posteriormente volver a analizar el sistema. Se obtuvo como resultado final un porcentaje del 88.5%, lo cual representó un aumento significativo, por el análisis presentado se recomienda hacer uso de las métricas de calidad basadas en la norma ISO/IEC 25010 para futuros proyectos de software.

Palabras Claves: < TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, <METODOLOGÍA ÁGIL SCRUM>, <NORMA ISO/IEC 25010>, <TECNOLOGÍAS DE DESARROLLO DE SOFTWARE>, <APLICACIÓN WEB>, <MANTENIBILIDAD>, <MÉTRICAS DE CALIDAD>.

ABSTRACT

In the present degree work the objective was to Develop a Web System to automate the processes of the Veterinary Clinic "Animal's Vet", for which the agile SCRUM methodology was applied in which 58 user histories and 12 technical stories were obtained, to from this, the planning was carried out in which the delivery date of the functional system to the customers was determined. The system has three User roles that are: Administrator, Secretary and Veterinarian, for which the following user modules were developed: administrator, secretary, veterinarian and provider, the following were used to develop those tools: Java programming language, PostgreSQL database manager and the CSS GRID Framework, the system was hosted on the Payara application server. The system structure used was the Model - View - Controller (MVC), this structure allowed for easy code structuring. Finally, the maintainability of the system was evaluated, based on several initial analyzes on the Analyzability, Changeability, Reusability - Stability and Capacity to be tested, a 59.17% maintainability percentage was obtained, from which corrections were made to the system for subsequently re-analyze its maintainability. As a final result, a percentage of 88.5% maintainability was obtained, which represented a significant increase, for the analysis presented it is recommended to use the quality metrics based on the ISO / IEC 25010 standard for future software projects.

KEY WORDS: <TECHNOLOGY AND ENGINEERING SCIENCES>, <SOFTWARE ENGINEERING>, <AGIL SCRUM METHODOLOGY>, <ISO/IEC 25010 STANDARD>, <SOFTWARE DEVELOPMENT TECHNOLOGY>, <WEB APPLICATION>, <MAINTAINABILITY>, <QUALITY METRICS>.

INTRODUCCIÓN

Actualmente, contar con un sistema web permite a las empresas tener un mayor alcance y un impacto en los clientes, ya que la información sobre lo que se realiza siempre va a estar disponible para todas las personas, mejorando así la comunicación entre el cliente y la empresa.

La veterinaria “Animal’s Vet” es una empresa que desde hace 4 años brinda a la ciudadanía sus servicios de atención a mascotas. Esta empresa está buscando renovar su proceso de atención a los clientes, mediante la implementación de un sistema que le permita automatizar el mismo. En reuniones mantenidas con la Dra. Diana Paucar, gerente propietaria de la veterinaria, manifestó que actualmente todos los procesos que se realizan en la veterinaria, como: registro de los dueños de los animales, las historias clínicas de las mascotas, ingreso para cirugías, controles de insumos de la veterinaria, y datos de las mascotas que están disponibles para la adopción, se hacen de forma manual, tomando así mucho tiempo en la toma de datos para la posterior atención de los mismos.

Llevar el proceso de esta manera, ha traído inconvenientes a la veterinaria, tales como: pérdidas de datos de los dueños de las mascotas, historias clínicas incompletas, mal manejo del inventario de los insumos, etc. Causando así, en muchas ocasiones, una ineficiencia al momento de poder atender a los animales, generando molestias e incomodidades a la clientela.

Por lo tanto, se propone desarrollar un sistema web que permita llevar todo esto de una forma ordenada, correcta y sin problemas de pérdidas de datos. De esta manera, siempre se tendrá disponible la información de cada uno de los procesos que se realizan.

Se plantea realizar el sistema web con las herramientas de desarrollo de software como: CSS grid, el cual se utilizará para el diseño de la interfaz del sistema, se guardarán los datos en PostgreSQL, y el sistema estará albergado en el servidor Payara.

Antecedentes

La veterinaria “Animal’s Vet”, durante 4 años ha atendido a la clientela en su oficina situada en las calles Av. Lizarzaburu manzana 12 (Junto a Nissan). La veterinaria dispone de algunos servicios que son ofrecidos al público, cómo: Cirugías para las mascotas, adopciones, peluquería, insumos de mascotas. El control y registro de los procesos (historia clínica, personal, mascotas, adopciones, alarmas, facturación, citas) se han venido realizando de manera manual, ralentizando así el servicio hacía las personas y dificultando llevar un control preciso de los animales internados, el trámite para dar en adopción a alguna mascota, etc. Además, de tener

inconvenientes con los permisos de funcionamiento otorgados por el Municipio de Riobamba al no tener la información de los animales organizada y accesible. Actualmente cuentan con 2 médicos veterinarios y 2 asistentes que ayudan en la atención de la oficina. El local está separado en 4 sub-departamentos que son: hospitalización, consultorio 1 y 2, y peluquería.

La veterinaria “Animal’s Vet” al no tener los registros y control al día de las mascotas puede tener inconvenientes con el municipio al no tener la información de todas las macotas que han sido atendidos por la veterinaria. Esto puede llevar a que haya multas, la suspensión del permiso de operación que puede dar al cierre de la veterinaria.

La veterinaria actualmente cuenta con un sistema que sólo realiza el registro de la información de los animales, esto retrasa y dificulta la gestión de los demás procesos, por lo que ha tenido inconvenientes al momento de llevar registros de las historias clínicas, cirugías, insumos, proveedores, dueño de la mascota, adopciones, esto ocasionaba pérdida de información importante sobre las mascotas, pérdida de clientes y pérdidas económicas.

El sistema con el que cuenta la veterinaria, no disponía de ningún tipo documentación, como el manual de usuario o manual técnico, además de que fue imposible acceder al código fuente de la misma, por tales motivos, realizar un aumento de funcionalidades al sistema fue un trabajo que no se podía realizar, y el esto llevó a realizar desde cero un sistema completo que completo que contenga todos los procesos automatizados.

La veterinaria actualmente no cuenta con un sistema que facilite la gestión de sus procesos, por lo que ha tenido inconvenientes al momento de llevar registros de los animales que ingresan al lugar, así como el control de las cirugías planificadas a las mascotas y los insumos veterinarios que se necesitan, además no tienen un control de proveedores de productos.

El desarrollo del proyecto, que gestiona la veterinaria “Animal’s Vet”, se realizará con la finalidad de registrar los datos de los animales ingresados en el lugar, los cuales son internados para proceder con el proceso de curación necesario, además, la veterinaria dispone de mascotas las cuales puede dar en adopción a las personas que se interesen en alguna de ellas.

Formulación del Problema

¿Cómo lograr una adecuada mantenibilidad en la implementación de la aplicación web para automatizar los procesos de la Veterinaria “Animal’s Vet”?

Sistematización del problema

¿Cómo se realizan actualmente los procesos relacionados la gestión de la veterinaria “Animal’s Vet”?

¿Qué tecnologías se va a utilizar para el desarrollo del sistema web para la veterinaria “Animal’s Vet”?

¿Cómo son los parámetros para demostrar la mantenibilidad del sistema (Modularidad, Analizabilidad, Capacidad para ser modificado, Capacidad para ser probado)?

Justificación teórica

Para el desarrollo del sistema para la veterinaria “Animal’s Vet” se utilizarán las herramientas:

Framework CSS-GRID es una característica nativa en CSS con la que puedes crear retículas fácilmente y evitar tanto uso de floats. Actualmente este framework es el mejor para el desarrollo de aplicaciones web, ya que corrige algunas de los errores que presentan frameworks tan populares como Bootstrap.

CSS Grid layout contiene funciones de diseño dirigidas a los desarrolladores de aplicaciones web. El CSS grid se puede utilizar para lograr muchos diseños diferentes. También se destaca por permitir dividir una página en áreas o regiones principales, por definir la relación en términos de tamaño, posición y capas entre partes de un control construido a partir de primitivas HTML.

Al igual que las tablas, el grid layout permite a un autor alinear elementos en columnas y filas. Sin embargo, con CSS grid son posibles muchos más diseños y de forma más sencilla que con las tablas. Por ejemplo, los elementos secundarios de un contenedor de cuadrícula podrían posicionarse para que se solapen y se superpongan, de forma similar a los elementos posicionados en CSS GRID.

CSS Grid

- Puede hacer que nuestra cuadrícula tenga exactamente la cantidad de columnas que deseamos ya sean 5 columnas, 7 columnas, 9 columnas o una cantidad explícita.
- Más flexibilidad al realizar cambios en el diseño.
- Los proyectos realizados presentan un peso en disco mucho más ligero. CSS.

BOOTSTRAP

- Como la cuadrícula Bootstrap está dividida en doce columnas, llegarás a tener muchos problemas si quieres un diseño de 5 columnas, 7 columnas, 11 columnas. O cualquier cosa que no sume 12.
- Presenta más complicaciones para realizar cambios en el diseño.
- Los proyectos realizados presentan un peso en disco muy superior.

Se ocupará el IDE de desarrollo de Netbeans, la versión 8.2, este IDE permite desarrollar en lenguajes de programación como C++, C, JAVA, etc. Permittiéndonos realizar el código para el

desarrollo del sistema en JSP, el cual es similar a PHP pero usa el lenguaje de programación JAVA.

Razones por las cuales se ha visto necesario el implementar un sistema automatizado para realizar los procesos de la veterinaria “Animal’s Vet”, basándose en los requerimientos que se solicitan.

Justificación aplicativa

La veterinaria “Animal’s Vet”, actualmente maneja los registros de manera manual dando como resultado un retraso en el manejo y atención de las mascotas ingresadas al lugar. Esto evita que la veterinaria pueda tomar una acción rápida cuando tenga que responder a las situaciones que se presenten con los animales ingresados. La veterinaria “Animal’s Vet”.

El sistema consiste en llenar el formulario de ingreso del animal, trabajo que lo realiza la secretaria, posteriormente, los veterinarios registrarán el proceso de revisiones clínicas y cirugías de los animales.

En el sistema se podrán generar distintos reportes que puedan facilitar ver de manera resumida y estadística, los procesos que se realizan dentro de la veterinaria. Los reportes generados por el sistema serán archivados de manera física y contarán como respaldos para justificar las decisiones tomadas dentro de la veterinaria (Agricultura).

Con el Sistema Web propuesto, la veterinaria logrará gestionar la información de los diferentes animales de la veterinaria, al automatizar los procesos (cirugía, personas, registro de proveedores, registro de insumos, registro de razas etc.), se accederá a una información centralizada y persistente que puede ser administrada en tiempo real por medio de una conexión a red. Esto brindará al personal administrativo, información sólida y eficaz de dichos procesos, ayudándolos en la toma de decisiones en base a datos relevantes.

El desarrollo del sistema web permitirá que la información de la veterinaria se encuentre alojada en un servidor web (por medio de una base de datos) haciéndola persistente, y que esté disponible para el personal administrativo del lugar. Así poder evitar las sanciones que se imponen de acuerdo a la Resolución de AGROCALIDAD 12 (“06NOR2016-RESOLUCION04.pdf”, s/f), en la cual se detalla que se debe llevar un registro del personal que labora en el lugar, y los datos actualizados y precisos de los animales que ingresan a la veterinaria para de esta manera seguir actualizando el permiso de funcionamiento.

Objetivos

Objetivo general

Desarrollar un sistema web para gestión de los procesos de la veterinaria “Animal’s Vet” de la ciudad del Riobamba.

Objetivos específicos

- Determinar cómo se realizan actualmente la gestión de la veterinaria “Animal’s Vet” para su posterior automatización.
- Analizar el framework CSS-GRID para el desarrollo de la aplicación web.
- Evaluar la mantenibilidad con el estándar ISO 25010.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

Con respecto al trabajo que aborda la problemática

Una clínica veterinaria es un lugar en donde practica la medicina veterinaria, la cual es la ciencia médica que está encargada de prevenir, prescribir y aliviar las enfermedades que afecten a los animales domésticos, salvajes o de producción. Dentro de una clínica veterinaria se realizan varias funciones, entre las cuales tenemos: Cirugías, Venta de insumos comestibles o de utilidad, recepción y entrega de animales para la adopción.

1.1. Procesos en una veterinaria

Dentro de una clínica veterinaria se realizan varios procesos.

- **Registro de los dueños de las mascotas:** Este proceso consiste en registrar al dueño de cada mascota en el sistema, para poder relacionarlos con la ficha médica que se crear posterior a esto.
- **Creación de ficha médica:** Este proceso se da cuando un paciente llega a la Clínica por primera vez lo cual da lugar a que se registren sus datos principales y los de su dueño.
- **Registro de admisión:** Este proceso se lleva a cabo cuando el paciente llega a la Clínica para recibir atención. Suponiendo que el paciente ya tenga ficha se procede a registrar su ingreso tomado en cuenta la persona con la que llega a la Clínica.
- **Evaluación médica:** Consiste en recoger los datos relativos al paciente en cuanto a signos y síntomas que este experimenta, seguidamente se hace un examen físico para dar un tratamiento adecuado y el correspondiente diagnóstico, pronóstico y receta.
- **Facturación:** Tras la evaluación del paciente el propietario se acerca a pagar por los servicios prestados por la Clínica. Luego de la cancelación en efectivo se entrega la correspondiente factura (David *et al.*, 2012).

Todos los procesos detallados son los que se desarrollan en una clínica veterinaria en general, el sistema permitirá automatizar cada uno de ellos, los cuales se dividirán en módulos, y de esta manera agilizar los tiempos de atención para cada cliente.

1.2. Trabajos relacionados

Existen varios trabajos relacionados con la gestión de veterinarias como las siguientes:

Sistema de gestión para la Clínica Veterinaria de la Universidad Central del Ecuador la cual se creó con el propósito de automatizar los procesos que se realizan dentro de la clínica, tales como: creación de ficha médica, registro de admisión, evaluación médica y facturación. El sistema se realizó con Java, la base de datos del sistema se realizó en PostgreSQL, y está almacenado en un

servidor de aplicaciones JBoss 5.1, además se utilizó la metodología Desarrollo Iterativo Incremental (David *et al.*, 2012).

Un ejemplo de ellos, en un estudio realizado donde se determinó que el tiempo que un programador invierte en mantenimiento es alrededor del 61% frente al 39% dedicado al desarrollo(Singer and Janice, 2003).

Desarrollo e implementación de un sistema web para la administración de la clínica veterinaria de la fundación protección animal Ecuador. Este sistema se desarrolló para suplir la necesidad de tener automatizado información de la clínica veterinaria, como: administración de historias clínicas de las mascotas que se llegaban a atender durante el día y el proceso de adopción y rescate de mascotas. Para este tema de tesis se utilizó la metodología ICONIX, las herramientas OpenSource y la arquitectura Java Enterprise Edition (Wladimir and Siguencia, 2012).

Implementación de un sistema en la nube para controlar y gestionar procesos clínicos en una veterinaria de mascotas que permita controlar y gestionar los procesos clínicos administrativos que se realizan en la Veterinaria Mascotas, mediante el uso de herramientas web como ZK Framework, Apache Tomcat y MySQL server. Se aplicó investigación de campo y bibliográfica como parte de la fase de análisis utilizando las técnicas de encuestas y entrevistas que fueron dirigidas a los involucrados en los procesos administrativos de la clínica; y a expertos en diseño y desarrollo de software, con la intención de validar los requerimientos funcionales, no-funcionales, de accesibilidad y criterios de usabilidad definidos para el proyecto. El diseño e implementación del proyecto permitió disminuir los tiempos de respuesta y controlar el flujo de información que se maneja en la clínica, mediante la optimización de los procesos principales del negocio, mejorando la disponibilidad, integridad y seguridad de la información relacionadas a datos administrativos y clínicos como los registros contables y fichas médicas de los pacientes(Haz Lopez *et al.*, 2017).

1.3. SCRUM

Es un marco de trabajo ligero y de fácil entendimiento en donde se emplean varias técnicas y procesos que sirven para un propósito en específico y fundamentalmente para el éxito de SCRUM y su uso(Schwaber and Sutherland, 2014, p. 5).

Los roles principales que se desempeña en la metodología ágil SCRUM son: el product Owner, equipo de desarrollo y scrum master.

El primero el product owner es el encargado de identificar, priorizar e ir listando las funcionalidades que van al principio de cada uno de los Sprints, además interactúa directamente con el equipo y va revisando los resultados que se van dando en cada uno de los sprints. Cabe

recalcar que en SCRUM existe una única persona que sirve como dueño del producto (Deemer *et al.*, 2015, p. 4).

El equipo de desarrollo es el encargado de ir construyendo lo que el dueño del producto dispone, todos los miembros del equipo trabajan conjuntamente deciden qué y cuantas funcionalidades son las que se van a ir desarrollando en cada uno de los sprints y así ir englobando conocimientos e ideas para lograr cumplir de la mejor manera los objetivos planteados (Deemer *et al.*, 2015, p. 5).

El scrum master es el que debe garantizar que la metodología sea entendida y adoptada, además de que el equipo se acople a cada una de las reglas, prácticas y teorías de scrum (Schwaber and Sutherland, 2014).

Artefactos de Scrum

Pila del producto

Es una lista ordenada y numerada de requerimientos, funcionalidades, cambios, correcciones que pueden ser utilizadas sobre el producto en el desarrollo para entregas futuras. Cada uno de los elementos de la pila del producto van ordenados de mayor a menor según el rango de importancia, y nunca dejan de cambiar por lo que no se le puede dar como completada ya que puede ser sometida a cambios y está en continuo crecimiento y evolución(Scrum Manager, 2016, p. 22).

Pila de sprint

Es la lista de tareas necesarias para construir las historias de usuario que se tomaran en cuenta en cada uno de los sprints. La pila del sprint la realiza el equipo de desarrollo en donde mediante las reuniones van indicando las prioridades que se les dará a cada una de las tareas aplicando técnicas como es la de T-shirt.(Scrum Manager, 2016, p. 25).

Eventos de Scrum



Figura 1-1: Eventos de Scrum

Fuente: (Scrum Manager, 2016).

Los eventos de SCRUM que presenta (Scrum Manager, 2016), define como trabaja en la metodología cada uno de ellos, los mismos que se muestran a continuación:

Planificación del Sprint

El dueño del producto define y explica las prioridades con las que contara, además que el equipo de desarrollo estima el esfuerzo de cada uno de los requerimientos y se realiza el sprint backlog.

Sprint

En este ciclo se define el tiempo que se demora, se recomienda que los sprints no se los realice de más de 6 semanas sino más bien en un tiempo inferior a un mes.

Scrum diario

En este evento cada miembro del equipo de desarrollo tiene 15 minutos como máximo para exponer las tareas realizadas, las tareas por hacer y si es que presenta inconvenientes al momento de desarrollarlas.

Revisión del sprint

Se realiza la presentación del avance del sprint, se plantean sugerencias en el equipo de desarrollo y se presenta el próximo sprint que se desarrollara.

Retrospectiva

El equipo de desarrollo conjuntamente analiza las fortalezas y debilidades con el fin de presentar mejoras en los siguientes sprints.

1.4. Arquitectura MVC

Es una arquitectura de diseño que se lo utiliza en sistemas que manipulan grandes cantidades de datos ya que permite separar el código de una manera más ordenada y estructurada, obteniendo así una programación en distintas capas(Rivera, 2008, p. 11).

Definición de las partes de la arquitectura MVC

Modelo: Es el que representa los datos que maneja el sistema. El sistema es el encargado de mantener los enlaces con la vista y el controlador y así mismo notificar de las modificaciones que se realizan en el modelo ya que el modelo no presenta ningún conocimiento algunos con la vista y el controlador (Yenisleidy Fernández Romero, 2012).

Vista: Es una representación visual de los datos enviados por el modelo. La vista es la encargada de representar visualmente el modelo y presenta al usuario. Tiene mayor preferencia de interacción con el controlador, pero puede trabajar también directamente con el modelo a través de una referencia al mismo (Yenisleidy Fernández Romero, 2012).

Controlador: Es una de las partes que le da sentido a las órdenes del usuario, brindado respuesta a los datos representados por el modelo, es el encargado de toda la interacción entre el modelo y

la vista. Al momento que se realicen modificaciones ya se tanto en la vista como en el modelo el controlador entra en acción en cualquiera de estos casos (Yenisleidy Fernández Romero, 2012).

Interrelación de los elementos de la arquitectura MVC

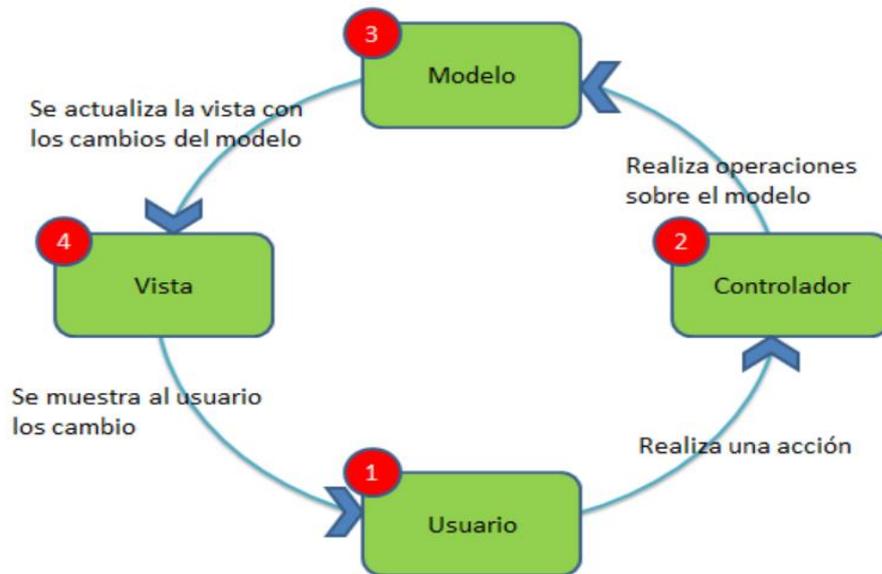


Figura 2-1: Elementos del patrón MVC
Fuente: (OSCAR BLANCARTE, 2014)

Mediante la siguiente figura a continuación se muestra la interrelación de los elementos del patrón MVC presentada por(OSCAR BLANCARTE, 2014).

- Primero el usuario interactúa y envía una acción hacia el controlador.
- El controlador lo recibe y realiza la petición y las debidas operaciones a realizar en el Modelo
- El modelo llama a la vista y es actualizada con los cambios que se realizaron en el modelo.
- La información por parte del modelo es recogida y es mostrada por la vista al usuario como una representación gráfica entendible para el usuario.

Ventajas

Entre las principales ventajas que presenta MVC según(Rivera, 2008), Tenemos las siguientes:

- Permite la separación entre los datos y la representación visual.
- Permite una independencia de funcionamiento
- Facilidad al momento de realizar el mantenimiento de ciertos errores que presente el sistema.
- Permite incorporar nuevos datos que requiera el sistema de una manera sencilla ya que cada una de las capas trabajan independientemente.

Desventajas

- Como cualquier patrón de diseño, MVC también presenta ciertas desventajas que se deben tomar en cuenta como presenta (Rivera, 2008).
- Al momento de separar el código en capas presenta mayor dificultad al realizar el sistema.
- Se incrementa considerablemente la cantidad de archivos en el sistema.
- En comparación a otros patrones la curva de aprendizaje de MVC es mucho más alta.

1.5. NetBeans IDE 8.2

Definición

Es una herramienta utilizada para el desarrollo de aplicaciones que se la puede programar en distintos lenguajes, pero principalmente trabaja con Java(Sdk and Netbeans, no date).

Es una herramienta que ayuda a integrar los lenguajes de programación con distintas plataformas, además de ayudar en la productividad a los desarrolladores al momento de crear, actualizar o implementar sistemas informáticos(González, 2015).

Ventajas

Según (González, 2015), netbeans presenta ventajas importantes a considerar las cuales se muestran a continuación:

- Es multiplataforma por lo que se lo puede ejecutar en cualquier sistema operativo.
- Cuenta con plugins para agregar funciones extras al sistema.
- Es una herramienta que no implica gastos ya que es de código abierto y gratuito.
- Es de fácil uso y presenta una comunidad muy amplia que brindan soporte al mismo.

Desventajas

Netbeans de acuerdo a (González, 2015), presenta ciertas desventajas que pueden ser consideradas al momento de escoger el IDE a utilizar como se muestran a continuación:

- Trabaja con mucha memoria por lo que si no se cuenta con un buen computador se tornaría lento.
- Lee todo el código por lo que si carga muchos proyectos trabajara lento.

1.6. Java

Definición

Es un lenguaje de programación de propósito general y orientado a objetos creado con el fin de ser utilizado para todo tipo de aplicaciones, ejecutándose en cualquier tipo de equipo con el sistema operativo con el que cuente.

Diferencia de java con otros lenguajes

Según (Vidal, 2001), presenta diferentes características que los hacen distinto a otros lenguajes, a continuación se muestran algunas de ellas:

- Es un lenguaje público y se puede conseguir kits de desarrollo de manera gratuita.
- Presenta una buena estructura y es de fácil aprendizaje
- Las aplicaciones que se desarrollan en este lenguaje son fiables, permitiendo controlar la seguridad siendo capaz de gestionar permisos y criptografía.

Ventajas

Muchos desarrolladores de software han migrado al lenguaje de Java valorando las ventajas con las que cuenta, a continuación, algunas presentadas por (Vidal, 2001).

- Presenta una gran funcionalidad por las librerías con las que cuenta incorporadas.
- La memoria es gestionada por el mismo lenguaje y no por el programador
- Las aplicaciones desarrolladas en este lenguaje presenta pocos errores posibles.

Desventajas

Según (NORMAS ISO 25000, 2019), java presenta ciertas desventajas como las que se muestra a continuación:

- Al momento de que el código es compilado se tomara cierto tiempo ya que java debe de interpretar el código para luego ejecutar el proceso.
- En comparación con el código nativo la máquina de java presenta cierta deficiencia de lentitud.

1.7. JavaScript

Este lenguaje de programación orientado a objetos actualmente es considerado la mejor opción para el desarrollo de aplicaciones web. Javascript tiene un parecido a C, C++ y Java, cómo por ejemplo el uso de las sentencias if, operadores, etc. (Ayoze, 2017).

Javascript se utiliza regularmente en navegadores web, lo cual permite interactuar con el usuario y controlar los navegadores modificando el contenido que aparece en pantalla.

En el libro de “Curso de Programación Web: JavaScript, Ajax y jQuery. 2ª Edición”, el autor especifica alguna de las características que tiene JavaScript, a continuación le lista alguna de ellas (Ayoze, 2017).

- Tiene muchas formas de probar el tipo de objeto.
- Las propiedades de los objetos se pueden añadir, modificar o eliminar.
- Es compatible con la mayoría de los navegadores.

- Soporta los elementos de sintaxis de programación estructurada del lenguaje C.
- Puede hacer diferenciación entre expresiones y comandos.
- Los tipos se asocian con valores y no con variables.
- En su mayoría está basado en objetos.
- Si un usuario tiene alguna deficiencia física o visual, se puede utilizar JS para optimizar su experiencia.
- Integra varios frameworks como: jQuery, Mootools, JSON, AJAX(Ayoze, 2017).

Ventajas

- La seguridad que brinda desarrollar con JS es casi total.
- Se ejecuta en la parte del cliente, de esta manera el servidor no se requerido más de lo que se necesita.
- Es fácil de poder aprender e implementar.
- Puede encajar correctamente con varios lenguajes y se puede usar en una gran variedad de aplicaciones(Ayoze, 2017).

Desventajas

- Los códigos de JS son visibles, por lo tanto, cualquier usuario puede leerlos.
- Desde el navegador, los usuarios pueden deshabilitar JS.
- Al ser código cliente-side, puede ser interpretado de manera diferente, dependiendo del navegador que se esté utilizando(Ayoze, 2017).

1.8. Payara Server

Es un servidor de aplicaciones de código abierto que permite ejecutar aplicaciones Java EE. Constantemente recibe actualizaciones, y cuenta con una de las mayores comunidades, lo cual permite tener un amplio soporte (Payara Services, 2019).

Es una plataforma de middleware

Según las características son:

- Licencia de código abierto.
- Soporta IDEs cómo: Eclipse, Netbeans, IntelliJ IDEA.
- Recibes actualizaciones cada trimestre.
- Los arreglos de seguridad son brindadas inmediatamente gracias a la gran comunidad.
- Arquitectura nativa de nube (Payara Services, 2019).

1.9. Base de Datos PostgreSQL

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto con más de 30 años de desarrollo activo que le ha ganado una sólida reputación de confiabilidad, solidez de funciones y rendimiento (Gibert and Pérez, 2002).

Es un gestor de base de datos relacionales que está orientada a objetos y es muy usado en entornos de software libre porque cumple con los estándares SQL92 y SQL99, además posee muchas funcionalidades avanzadas, lo cual lo coloca, en mucho de los casos, mejor que algunos SGBD comerciales.

El origen de PostgreSQL se remota a un gestor desarrollado en la Universidad de Berkely llamado POSTGRES en 1994. Desde ese momento ya contaba con varias prestaciones únicas en el mercado y que la mayoría de gestores comerciales lo han añadido con el pasar del tiempo.

PostgreSQL funciona en múltiples plataformas (la mayoría basadas en Unix) y, después de la versión 8.0, también funciona en Windows, además cuenta con un amplio conjunto de enlaces con varios lenguajes de programación como: C, C++, Java y Python.

¿Qué es una base de datos relacional?

Es un sistema que permite el manejo de acuerdo con las reglas del álgebra relacional. También se podría decir que es una colección de tablas interrelacionadas que posibilita el poder almacenar información para que posteriormente pueda utilizarse, para poder realizar las relaciones, se basa en la teoría de conjuntos.

Tabla 1-1: Características de PostgreSQL

CARACTERÍSTICAS	DETALLES
Plataforma	Microsoft Windows, MAC OS, Linux, Unix.
Lenguaje de programación	C, C++, Java, Java Web, etc.
Herramientas de Case	Power Designer
Transacciones	✓
Licencia	Gratuita
Administración	Se basa en usuarios y privilegios
Confiabilidad	Altamente confiable en cuanto a estabilidad se refiere
Soporta	Vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas.

Fuente: (Angelica Dominguez, 2013)

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

Ventajas

- Es un sistema estable, de alto rendimiento y gran flexibilidad.
- Permite migrar aplicaciones desde C/C++, Visual studio, etc.
- Puede manejar múltiples conexiones concurrentes de los clientes.

Desventajas:

- Puede ser lento en inserciones de bases de datos me menor tamaño.
- La sintaxis de varias de las sentencias no pueden ser intuitivas, lo cual requiere tener un nivel medio en lenguajes de SQL.

1.10. Sistema web

Una aplicación web en los últimos años se ha convertido en un sistema complejo con una interfaz de usuario que cada vez se parecen más a las aplicaciones de escritorio, una aplicación web brinda servicios a procesos de negocios de tamaño considerable, además consta de requisitos para la accesibilidad, respuesta y calidad.

La expansión de internet ha supuesto a que las necesidades de información de una organización tengan cambios, tales como:

- Que se pueda acceder a la información desde cualquier parte, ya sea dentro de la organización o fuera de ella.
- Que la información esté sea compartida en todo momento entre todas las partes que estén interesadas, de esta manera se puede acceder a los datos que les corresponda según la función.

Estas necesidades han supuesto una transición de aplicaciones de escritorio hacia las aplicaciones web. Además de que ha provocado el aumento en la complejidad de los sistemas, por ende, ha habido necesidad de encontrar opciones nuevas para diseñarlos y de aumentar su calidad de producción (Castejón Garrido, 2004).

Para el funcionamiento de una aplicación web se debe tener un servidor principal en donde esté alojado el sistema funcional, de esta manera los usuarios acceden al sistema a través de un navegador web instalado en sus estaciones de trabajo conectadas a la red (La *et al.*, 2013).

1.11. Diseño de un sistema web

Todos los sistemas para que puedan ser usados correctamente necesitan mostrar al usuario sus posibilidades de uso y tener claro las acciones que son posibles, esto lo hacen mediante su interfaz, la cual debe ser clara y entendible, que permita acceder sin problemas y sin frustración a los servicios que ofrece el sistema.

Cuando se inició con el desarrollo de sistemas web, existía un concepto erróneo del diseño de aplicaciones web, el cual decía que el diseño de interfaces trata sólo sobre poner gráficos, agregar decoración y dibujos independientes de la funcionalidad que tendrá el sistema (Martínez Gómez, Higuera Marín and Aguilar Díaz, 2013).

Para Jesse James Garet el diseño de un sitio web estudia, no como funciona el producto, sino como funciona cuando un usuario entra en contacto con él.

La primera generación de desarrollo de páginas web estaba basada en lenguaje de hipertexto HTML, en la cual se podía crear diseños sencillos, únicamente con textos, tablas y vínculos con pocos gráficos, la segunda generación dio origen a HTML2, la cual permitía utilizar íconos en vez de palabras, además de poder poner imágenes de fondo, en la tercera generación, HTML3, se añadieron más etiquetas HTML, y dispone de más opciones como tablas y hojas de estilo CSS. Esto ayudó a tener fondos con un patrón gráfico, y la capacidad de añadir animación al diseño web (GIF), además llega Flash aportando mucha capacidad de innovación gráfica.

En la actualidad el diseño de sistemas web tuvo la necesidad de evolucionar al desarrollo de sitios web responsive, esto quiere decir que el sistema se adapte a las características de la pantalla en la cual se visualiza, ya sea de un ordenador o un dispositivo móvil, debido a las nuevas características de demanda surgieron nuevas herramientas de apoyo al desarrollo de aplicaciones, denominados frameworks, además de variedad de lenguajes, posibilidades y varios caminos a elegir como por ejemplo JavaScript, PHP, XML, CSS, etc (Forero Palma, 2017).

1.12. Hojas de Estilo

Son líneas de código HTML o XML, que se orientan a dar un estilo específico a los elementos de un documento, se ocupan para poder personalizar el color, fuentes, en sí, todo el apartado visual de un sitio web. De esta manera se mantiene más limpio el código fuente de una página.

Hay 3 tipos de Hojas de estilo:

Externas: Son archivos independientes. Tienen la extensión .css y se llaman en la cabecera del código del sitio web en desarrollo, de esta manera se cargan los estilos definidos.

Internas: Va añadida dentro de la etiqueta HEAD del código de la página web.

InLine: Se añaden directamente en la etiqueta, esto quiere decir que el estilo de la etiqueta va escrito dentro de la misma.

Los CSS han cambiado por completo la manera en la que se puede desarrollar los sitios web, se ha reemplazado el uso de las tablas HTML para organizar elementos en el sitio web. Además, permiten manejar de manera fácil todas las zonas de la página, y ofrece una comodidad total cuando se realiza la diagramación del sitio web. Esto ayuda a depurar el código fuente y permite crear un estilo visual que se ajuste a las necesidades presentadas por el desarrollador del sistema y del usuario final (*Qué es una hoja de estilo CSS, no date*).

En la **Figura 3-1**, se detalla como esta compuesta una regla de CSS con su respectiva manera de establecer en el código de nuestro sistema.

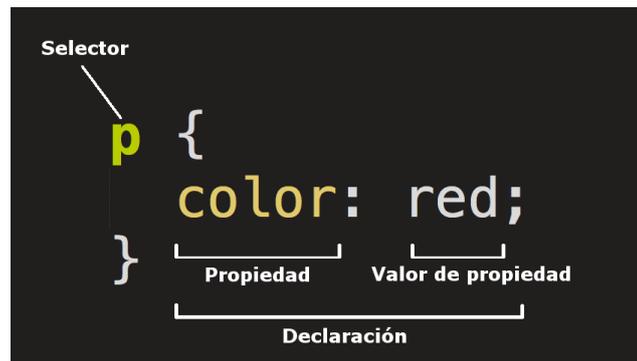


Figura 3-1: Partes de una regla Css.

Fuente:(Mozilla, 2019)

Selector: El elemento HTML en el que comienza la regla esta selecciona los elementos a dar estilo.

Declaración: Una sola regla como color: red; específica a cuál de las propiedades del elemento quieres dar estilo.

Propiedades: Maneras en las cuales puedes dar estilo a un elemento HTML.

Valor de la propiedad: A la derecha de la propiedad, después de los dos puntos (:), tenemos el valor de la propiedad (Mozilla, 2019).

1.13. FRAMEWORK

Un framework CSS es una estructura de software que consta de varios componentes (herramientas, hojas de estilo) que pueden ser personalizables e intercambiables para poder facilitar y agilizar el desarrollo de un sistema web. Las características principales que tiene un framework son: apresurar el proceso de desarrollo y reutilizar código, esto ayuda a promover las buenas prácticas de desarrollo (Gutiérrez, 2010).

Una ventaja importante que nos brinda utilizar un framework de desarrollo es que se puede utilizar patrones de diseño para desarrollar una aplicación. Uno de los más utilizados es el Modelo-Vista-Controlador (MVC). Además, el código que es parte de un framework se ha probado totalmente, esto quiere decir que se garantiza que funcione correctamente.

TIPOS DE FRAMEWORK

Actualmente, existen muchos tipos de framework, entre los cuales tenemos:

- Infraestructura de sistema: permite el desarrollo de sistemas de explotación, herramientas gráficas y plataformas web.
- Comunicativos: o también llamados software.
- De Empresa: Sirve para desarrollos específicos.
- Gestión de contenido: son los de tipo Content Management System (AcensTechnologies, 2018).

Los frameworks CSS que son completos pueden incluir utilidades para que el diseñador de la aplicación no deba esforzarse en los aspectos genéricos del diseño web, por tal motivo, los frameworks más avanzados incluyen herramientas para:

- Neutralizar los estilos que vienen por defecto en los navegadores, vendría a ser la hoja de estilo reset.css que los diseñadores utilizan.
- Manejar de manera correcta el texto, esto quiere decir que los contenidos del sistema web se vean exactamente igual en cualquier navegador, además de que sea adaptable para tener una mejor accesibilidad y acceso en cualquier medio o dispositivo.
- Crear cualquier Layout de forma sencilla, con la confianza de que todo funciona de manera adecuada en cualquier versión de un navegador (Uniwebsidad, 2006).

1.14. Css Grid

Actualmente Css Grid es el framework más poderoso para realizar diseños, en cuanto a CSS se refiere. Es un sistema bidimensional, esto quiere decir que se maneja con filas y columnas, esta es la principal diferencia con Flexbox, ya que este trabaja con un sistema unidimensional, esto quiere decir que usar CSS grid amplía nuestras posibilidades de desarrollo, ocupando características para crear páginas con diseños que hacer un tiempo atrás parecían casi imposibles(House, 2017).

Características de CSS Grid

- No tiene limitaciones a doce columnas.
- A diferencia de otro framework conocido como lo es Bootstrap, el cual está limitado a trabajar con doce columnas, a la larga esto causa muchos problemas si queremos un diseño con 5, 7 u otra cantidad de columnas.
- Con CSS grid, este problema queda resuelto, ya que podemos hacer que la cuadrícula tenga la cantidad exacta de columnas que deseamos.
- Soportes del Navegador.

- Otra de las ventajas importantes de manejar CSS grid es el soporte con el navegador (Jonathan Araujo, 2018).

En la **Figura 4-1**, se detalla el soporte con cada navegador que actualmente soporta el manejo de CSS grid.

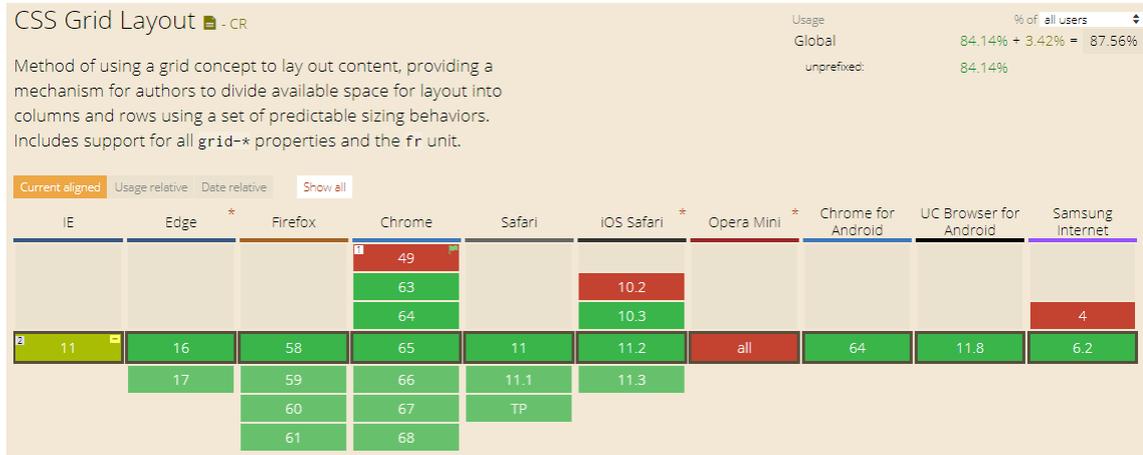


Figura 4-1: CSS GRID LAYOUT
Fuente:(Jonathan Araujo, 2018).

Más ligero

Los tamaños en disco de los proyectos realizados con CSS GRID son excesivamente más ligeros a comparación de proyectos realizados con Bootstrap.

Tabla 2-1: Tamaño de un proyecto realizado con CSS GRID

Tabla Tamaño	148KB (152.075 bytes)
Tamaño de disco	164KB (167.936 bytes)

Fuente:(Jonathan Araujo, 2018)
Realizado por: Carlos Grefa, Bernabé Aragón, 2019

1.15. Calidad de software

En los últimos años ha habido un aumento en la preocupación de la industria del software por el desarrollo sistemático de productos de calidad que satisfagan las necesidades y expectativas del cliente, del esta manera se disminuyen gastos posteriores y se optimizan los recursos (Castejón Garrido, 2004).

Para el autor Edwards Deming, la calidad tiene que ver los procesos que realiza una empresa internamente, además comenta en su libro “Out of the Crisis” que: “El control de calidad no significa alcanzar la perfección. Significa conseguir una eficiente producción con la calidad que espera obtener en el mercado.” (López Echeverry, Valencia Ayala and Cabrera, 2008, p. 2).

Por otro lado, para el autor Joseph Juran, la calidad tiene mucho más trasfondo que el producto y el nivel de satisfacción del cliente, sino que también se debe a mejorar la calidad de los procesos de la producción (López Echeverry, Valencia Ayala and Cabrera, 2008, p. 2).

Se podría decir que la calidad persigue los siguientes objetivos:

- Satisfacción de los clientes.
- Eficiencia en el uso de los recursos humanos.
- Disminución en el costo de las operaciones (López Echeverry, Valencia Ayala and Cabrera, 2008).

Una norma para evaluar los productos software fue el estándar ISO/IEC 9126, la cual plantea un estándar de calidad que tiene 3 aproximaciones: interna (calidad del código), externa (calidad en la ejecución) y en uso. Este estándar tiene 6 características para poder evaluar la calidad de los atributos del software entre ellos: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad. (Garzas, 2012).

En el año 2005 ISO/IEC publica la norma 25000 en remplazo al estándar ISO 9126. En este nuevo estándar se agregaron 2 nuevas características con respecto a las anteriores las cuales son: seguridad y compatibilidad (Calabrese *et al.*, 2017).

La norma ISO 25000 está compuesta por cinco divisiones que son:

- ISO/IEC 2501n es la división para el modelo de la calidad.
- ISO/IEC 2502n es la división para la medición de la calidad.
- ISO/IEC 2503n es la división para los requisitos de la calidad.
- ISO/IEC 2504n es la división para la evaluación de la calidad.
- ISO/IEC 2500n es la división para la gestión de la calidad.

La ISO/IEC 2501n presenta modelos de calidad, los cuales incluyen calidad interna, externa y en uso. Esta división está formada por:

- Estándar ISO/IEC 25010, el mismo que se utilizará para medir la mantenibilidad del sistema.
- Estándar ISO/IEC 25012 el cual ayuda a definir el modelo general para la calidad de datos (ISO, 2017).

1.16. ISO 25010-Mantenibilidad

El estándar ISO/IEC 25010 fue desarrollado para reemplazar el estándar ISO/IEC 9126, el cual fue uno de los estándares más utilizados para medir la calidad de un producto software. Esta norma forma parte de la familia de normas ISO 25000. Este estándar ayudará a determinar las

características de calidad a tener en cuenta para poder realizar la evaluación de un producto software ya finalizado (Yusof, Kane and Yusop, 2013).

Este estándar define un modelo de calidad del producto el cual se compone de ocho características, algunas de ellas se subdividen en 31 subcaracterísticas. Las características y subcaracterísticas del estándar ayudan a brindar congruencia terminológica para poder especificar, medir y evaluar la calidad de un producto software (Valenciano López *et al.*, 2015).

“Históricamente la mantenibilidad ha sido reconocida como una de las características más relevantes del software debido a su impacto directo sobre el costo de desarrollo y el propio mantenimiento. De hecho, estudios previos señalan la fase de mantenimiento como la fase donde más recursos se invierten en el ciclo de vida del software”(Irrazábal, 2015).

La dimensión de la mantenibilidad se puede dar debido a distintos tipos de necesidades, tales como:

- Perfectivas y Evolutivas: esta trata de la adición de nuevas prestaciones y/o funcionalidades al sistema, también trata sobre la mejora en el rendimiento de las que ya existen.
- Correctivas: se trata sobre corregir los fallos y defectos que puede tener el sistema, de igual manera los efectos que se derivan de estos. Si el defecto es muy leve, la corrección será fácil y sin demora.
- Adaptativas: trata de los cambios que se deben realizar a partir de cambios en el sistema operativo, hardware, arquitectura del sistema, etc. Suceden en un porcentaje muy bajo.

Actualmente la mantenibilidad es pieza fundamental a la hora de llevar a cabo un proceso de desarrollo de software, porque es el atributo de calidad de software que influye de manera directa en los costes y necesidades del mantenimiento, mientras mayor sea el porcentaje de mantenibilidad, menores serán los costes para mantener el sistema.

Existen factores que afectan de manera directa a la mantenibilidad, los más relevantes son:

- Proceso de desarrollo: La mantenibilidad siempre debe formar parte de manera directa en el proceso de desarrollo de un producto software, ya que será mayor la ganancia cuando la mantenibilidad se incorpore de manera propia en los sistemas.
- Documentación: cuando la documentación no está disponible los costes de mantenimiento del sistema se aumentan significativamente debido al tiempo que se requerirá para que se entienda como fue diseñado el software antes de poder empezar a modificarlo.
- Comprensión de programas: uno de los problemas básicos que causan altos costes en el mantenimiento, es la falta de comprensión humana en los programas existentes.

La mantenibilidad de un sistema podría ser considerada como la unión de dos propiedades, Reparabilidad y Flexibilidad.

Un sistema software se podría considerar como reparable si este permite la corrección de los defectos que tiene, con una cantidad limitada de trabajo, además, es flexible si permite realizar cambios que satisfagan nuevos requerimientos (Valenciano López *et al.*, 2015).

La mantenibilidad es la facilidad con la que se puede modificar un sistema o componente software para corregir errores, mejorar el rendimiento u otros atributos, o adaptarse a un entorno modificado (Coleman *et al.*, 1994).

La calidad se puede definir como el nivel de satisfacción en el cumplimiento de los requisitos de sus usuarios.

La Mantenibilidad del sistema se refiere a medir en el sistema: la modularidad, reusabilidad, analizabilidad, capacidad de ser modificado, capacidad para ser probado (Mera Paz, Miranda Gómez and Cuaran Rosas, 2017).

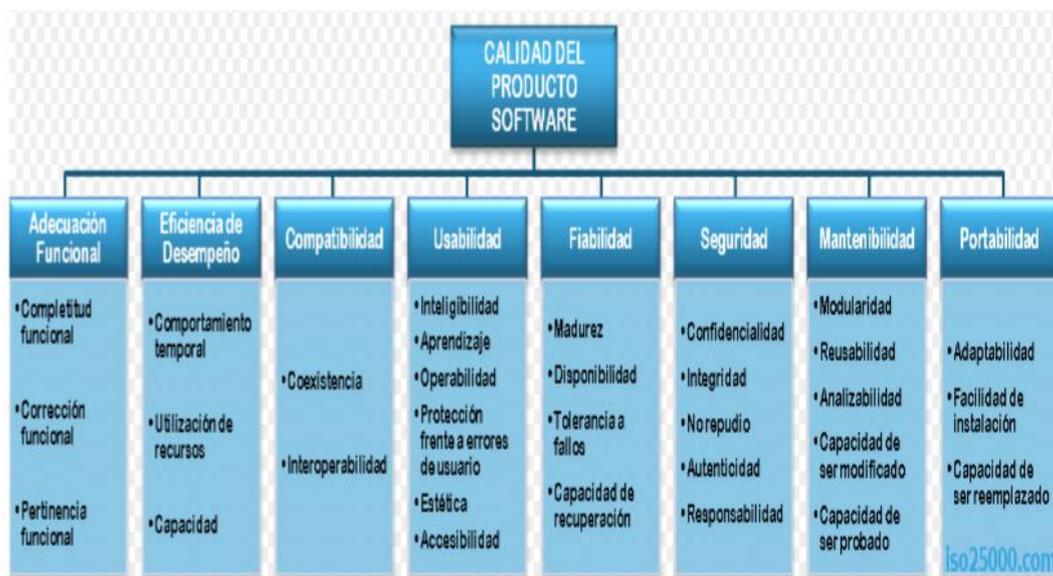


Figura 5-1: ISO/IEC 25010

Fuente: (Iso25000.com, 2019)

Tabla 3-1: Indicadores Mantenibilidad

Mantenibilidad	
Indicador	Definición
Analizabilidad.	Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
Reusabilidad- Estabilidad	Capacidad de un software que permite que un cambio sea utilizado en la reconstrucción de un nuevo sistema o del mismo.
Capacidad para ser modificado	Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
Capacidad para ser probado	Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 3-1** se presenta los indicadores de la mantenibilidad según la norma ISO/IEC 25010.

CAPÍTULO II

2. MARCO METODOLÓGICO

En este capítulo se utilizó la metodología ágil SCRUM lo que hace preciso seguir con el ciclo de vida del software que tienen 3 fases: de planificación, de desarrollo y de cierre, se detalla a continuación la gestión de los procesos de la veterinaria “Animal’s Vet”.

Tipo de Investigación

El tipo de investigación que contemplará este trabajo de titulación es de tipo aplicativo, ya que mediante los conocimientos adquiridos se dará solución al problema de la gestión de los procesos de la veterinaria “Animal’s Vet” a través de un sistema que permita agilizar los mismos, ya que en la actualidad no cuentan con toda información digitalizada, lo que les genera inconvenientes con ciertas instituciones públicas como la Municipalidad de Riobamba.

2.1. Métodos y técnicas

Para el desarrollo del sistema se fundamentan ciertos métodos y técnicas que se aplicarán en cada uno de los objetivos planteados y que se presentan a continuación:

2.1.1. *Objetivo 1: Determinar cómo se realizan actualmente la gestión de la veterinaria “Animal’s Vet” para su posterior automatización.*

Método Deductivo

Por medio de este método se definieron las técnicas que se vienen realizando en la veterinaria “Animal’s Vet”, el cual dará una idea clara y precisa de los procesos que son utilizadas para poder desarrollar el sistema web.

Técnicas

Observación

Mediante esta técnica se propuso realizar un análisis de manera general, como se lleva a cabo los procesos en la veterinaria “Animal’s Vet”, así se pudo tener una idea más clara del proceso que realiza.

Entrevista.

Por medio de esta técnica se pudo obtener de manera más precisa la información que será utilizada en el sistema web.

Se realizó una entrevista al personal que se encuentra laborando en la veterinaria, el cual proporcionó información acerca de los procesos que se vienen realizando, esto sirvió para entender la forma de cómo se realizan las técnicas para poder realizar el sistema informático.

2.1.1. Objetivo 2: Analizar el framework CSS-GRID para el desarrollo de la aplicación web.

Para realizar el sistema web de la gestión de los procesos de la veterinaria “Animal’s Vet” se manejó el framework CSS-GRID, el cual brinda un mejor el diseño de las interfaces.

Las herramientas a utilizar en el sistema son:

- **NetBeans IDE8.2:** Se utilizó el NetBeans como IDE de desarrollo para realizar la aplicación.
- **JavaScript:** Se utilizó este lenguaje en la parte del cliente por ser un lenguaje flexible y permite que la aplicación web sea más moderna e interactiva.
- **Payara Server:** Se utilizó este servidor para almacenar la aplicación.
- **Base de datos PostgreSQL:** Se utilizó como servidor BD multiplataforma para almacenar la información de manera segura.
- **Css-Grid:** Es un framework que se utilizó para el desarrollo de la interfaz del sistema.

Método Analítico

Para una investigación de manera más profunda acerca el framework CSS-GRID, se analizó la respectiva información, la cual facilitará entender de una mejor manera como utilizar esta herramienta y ayudará a comprender a cómo realizar el sistema web de manera precisa.

Técnicas

Técnica de investigación documental.

Esta técnica ayudará a recolectar de manera satisfactoria la información de diferentes sitios web, artículos científicos y libros, la cual facilitará conocer más acerca de dicha herramienta.

2.1.2. Objetivo 3: Evaluar la mantenibilidad con el estándar ISO 25010.

Método Analítico

Mediante esta técnica se determinó un correcto análisis de cada uno de los atributos que corresponden al estándar ISO 25010, y poder aplicar de manera acertada en la evaluación del sistema.

Técnicas.

Técnica documental

Mediante esta técnica se determinó un correcto análisis de cada uno de los atributos que corresponden al estándar ISO 25010, y poder aplicar de manera acertada en la evaluación del sistema.

Para el avance del sistema web de la gestión de los procesos se colocó como objetivos la mantenibilidad de dicho sistema, por esto se llegará a evaluar la mantenibilidad, especificando en la **Figura 5-1** los indicadores que se van a tomar en cuenta

2.2. Herramientas para medir la mantenibilidad

Tabla 1-2: Herramientas utilizadas para medir la mantenibilidad

Herramientas	Descripción	Métricas
SimpleCodeMetrics	Es un complemento que es proporcionado por Netbeans que es utilizado para medir, líneas de código, número de clases y métodos, complejidad ciclomática, gracias a esta herramienta se logró medir la anazabilidad.	Complejidad Ciclomática
Findbugs	Es un complemento que es proporcionado por Netbeans se lo maneja a la hora de encontrar ciertos errores en la codificación y en las variables de cuál es la mejor manera de establecer el código, gracias a esta herramienta se logró medir la Reusabilidad-Estabilidad.	Reusabilidad-Estabilidad
EasyPMD	Es un complemento que es proporcionado por Netbeans el cual indica el código que no era utilizado en el sistema y los comentarios que se encontraban fuera de lugar, gracias a esta herramienta se logró medir la Capacidad para ser modificado.	Capacidad para ser modificado
Cloc	Es una herramienta que permite encontrar el total de líneas de código que llega a tener nuestro sistema el cual servirá para tener una noción de cuanto se ha llegado a programar en todo el tiempo que se ha realizado el sistema, gracias a esta herramienta se logró medir la anazabilidad.	Densidad de comentarios
Junit	Es un complemento que es proporcionado por Netbeans el cual permite observar el total de pruebas unitarias hemos realizado por cada módulo, gracias a esta herramienta se logró medir, la capacidad para ser probado.	Capacidad para ser probado

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 1-2** se muestran las herramientas que son utilizadas para poder medir las diferentes condiciones de la mantenibilidad, y de esta manera medir de manera eficaz el estado mantenible del sistema.

2.3. Indicadores Mantenibilidad

2.3.1. Analizabilidad

Densidad de complejidad ciclomática

La complejidad ciclomática se basa en el cálculo de las trayectorias que tiene nuestro código se realizó un análisis de la complejidad ciclomática de cada uno de los componentes (MVC) que son los que cuenta el sistema (Lógica de Negocio, Acceso a datos y la Interfaz de Usuario), de esta manera se puede conseguir un indicador que es independiente al proyecto (Valenciano López *et al.*, 2015).

Tabla 2-2: Clasificación Complejidad Ciclométrica

Complejidad Ciclométrica	Evaluación Rango
1-10	Programa Simple, sin mucho riesgo
11-20	Más complejo, riesgo moderado
21-50	Complejo, programa de alto riesgo
>=50	Programa no testeable, muy alto riesgo

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

Fuente: (Valenciano López *et al.*, 2015)

En la **Tabla 2-2** se muestra la clasificación de la complejidad ciclométrica en sus diferentes rangos de evaluación para posterior evaluación.

Densidad de Comentarios

La densidad de comentarios calcula la cantidad de comentarios del sistema y el tamaño del mismo, es necesario realizar un análisis del código fuente, es decir líneas de código que contengan sentencias que no sean de comienzo o fin (llaves), de esta manera se puede conseguir un indicador que es independiente al proyecto (Valenciano López *et al.*, 2015).

Tabla 3-2: Clasificación densidad de comentarios

No	Defectos de la capacidad de la densidad de comentarios	Resultado defectos de la densidad de comentarios
1	71-100%	Muy Bueno
2	36-70%	Bueno
3	1-35%	Malo

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 3-2** se muestra la clasificación de la densidad de comentarios en sus diferentes rangos de evaluación para posterior evaluación.

2.3.2. Capacidad para ser modificado

La capacidad de ser modificado del sistema se calcula con la relación entre el número de defectos de modificabilidad del sistema y el tamaño (Valenciano López *et al.*, 2015).

Tabla 4-2: Clasificación de la capacidad para ser modificado

No	Defectos de la capacidad para ser modificado	Resultado defectos de la capacidad para ser modificado
1	76%-100%	Muy malo
2	51-75%	Malo
3	26-50%	Bueno
4	1-25%	Muy Buena

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

Fuente: (Valenciano López *et al.*, 2015)

En la **Tabla 4-2** se muestra la clasificación de la capacidad para ser modificado en sus diferentes rangos de evaluación para posterior evaluación.

2.3.3. Reusabilidad-Estabilidad

Mide la relación entre el número de defectos de reusabilidad y el tamaño del sistema. Este indicador demuestra lo complejo que es el software a la hora que va ser reutilizado (Valenciano López *et al.*, 2015).

Densidad de defectos de reusabilidad

Tabla 5-2: Clasificación de la reusabilidad-estabilidad

No	Defectos de la reusabilidad-estabilidad	Resultados defectos de la reusabilidad-estabilidad
1	1-25%	Muy Buena
2	26-50%	Bueno
3	51-75%	Malo
4	76%-100%	Muy malo

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

Fuente: (Valenciano López *et al.*, 2015)

En la **Tabla 5-2** se muestra la clasificación de la reusabilidad-estabilidad en sus diferentes rangos de evaluación para posterior evaluación.

2.3.4. Capacidad para ser probado

Las pruebas unitarias se deben realizar las respectivas trayectorias del código, las pruebas miden la relación del número de pruebas realizadas y la complejidad ciclomática, es un excelente indicador para realización de las pruebas.

- UTD: Densidad de Pruebas Unitarias.
- UN: Número de Pruebas Unitarias.

- CC: Complejidad Ciclomática (Valenciano López *et al.*, 2015).

Tabla 6-2: Clasificación de la capacidad de ser probado

No	Módulos	Resultados defectos de la capacidad para ser probado	Defectos de la capacidad para ser probado
1	13-17	Muy Bueno	75%-100%
2	7-12	Bueno	25%-75%
3	1-6	Malo	1%-25%

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 6-2** se muestra la clasificación de la capacidad para ser probado en sus diferentes rangos de evaluación para posterior evaluación.

2.4. Escenarios de Prueba

2.4.1. Mantenibilidad al culminar el desarrollo de la aplicación.

Escenario 1.

Después de culminar con el desarrollo del sistema web se aplicaron los indicadores de la mantenibilidad como son: Analizabilidad, Reusabilidad-Estabilidad, Capacidad para ser modificado, Capacidad para ser probado. Se tendrá dos escenarios con el porcentaje de mantenibilidad cada uno, en el primer escenario se analizará el sistema recién desarrollado sin aplicar los cambios.

2.4.2. Mantenibilidad después de aplicar las acciones de mejora.

Escenario 2.

Al aplicar la mantenibilidad después de las acciones de mejora en el sistema web, se analizará con los indicadores de mantenibilidad que son: Analizabilidad, Reusabilidad-Estabilidad, capacidad para ser modificado, Capacidad para ser probado. En el segundo escenario se evaluará el sistema para comprobar si efectivamente hubo cambios en el porcentaje de mantenibilidad, para poder determinar si se efectuaron o no correctamente los indicadores mencionados.

2.5. Población y muestra

Para poder medir la mantenibilidad se consideró todas las líneas de código utilizadas en el sistema, Líneas de código (LOC): $LOC = NCLOC(\text{líneas comentadas}) + CLOC(\text{Líneas en Blanco}) = 157775$. Con estos valores se obtendrá los resultados de la mantenibilidad del sistema

2.6. Desarrollo del sistema web para la gestión de los procesos de la veterinaria “Animal’s Vet”.

Estudio Preliminar

La veterinaria “Animal’s Vet”, durante 4 años ha atendido a la clientela en su oficina situada en las calles Av. Lizarzaburu manzana 12 (Junto a Nissan). El local está separado en 4 sub-departamentos que son: hospitalización, consultorio 1 y 2, y peluquería.

2.6.1. Información de la veterinaria

La veterinaria “Animal’s Vet” se encuentra ubicada en las calles Av. Lizarzaburu manzana 12, dispone de servicios como: cirugías para las mascotas, adopciones, peluquería, insumos de mascotas, actualmente la veterinaria cuenta con 2 médicos veterinarios y 2 asistentes que ayudan en la atención de la oficina, el local está separado en 4 sub-departamentos los cuales son hospitalización, consultorio 1 y 2, y peluquería.

2.6.2. Personas y roles involucrados en el proyecto

Tabla 7-2: Personas y Roles

PERSONA	ROL	INSTITUCIÓN
Ing. Gloria Arcos	SCRUM Master	ESPOCH-FIE
Dra. Diana Paucar	Product Owner	VETERINARIA ANIMAL’S VET
Carlos Grefa	Development Team	ESPOCH-EIS
Bernabé Aragón	Development Team	ESPOCH-EIS

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

En la **Tabla 7-2** se muestran las personas, roles y la institución de los que se encuentra involucrados en el proyecto

2.7. Descripción del proceso de la veterinaria.

En la veterinaria Animal's Vet se realizan los procesos de manera manual lo que les con lleva bastante tiempo en realizar los diferentes cambios en los métodos. En la **Figura 1-2** se establecieron los actores y procesos que realizan en la veterinaria (Historia Clínica, Cirugía y Adopciones), A continuación, se muestra las diferentes actividades que realizan los actores del sistema.

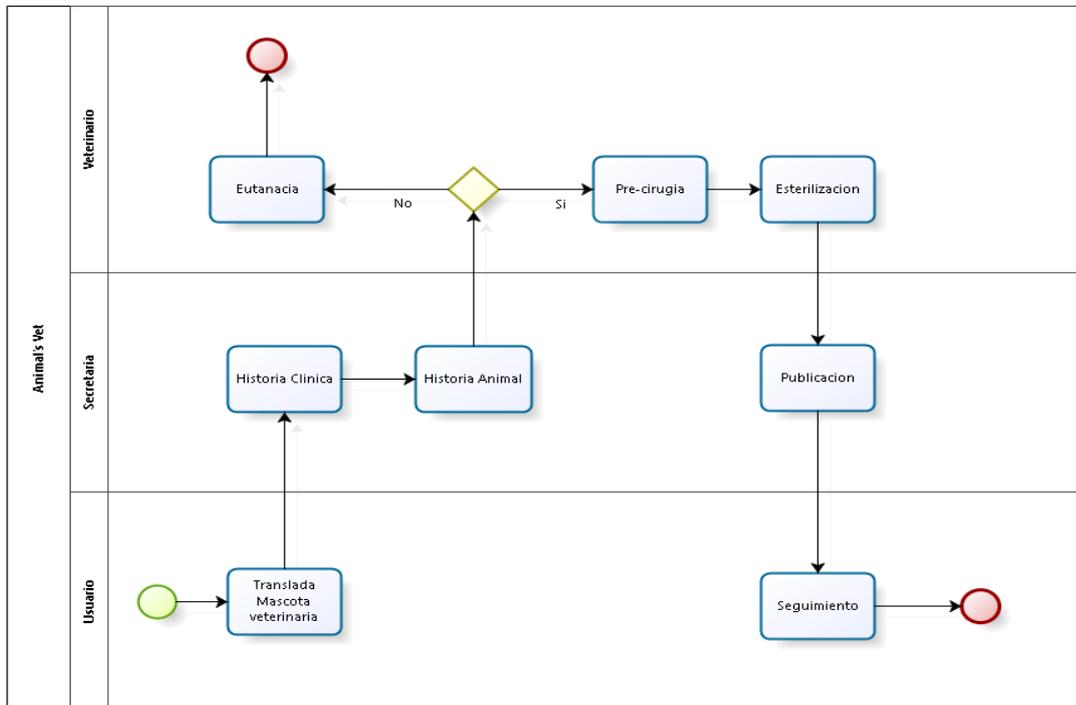


Figura 1-2: Diagrama de procesos historia clínica y cirugía de la veterinaria Animal's Vet.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

En la **Figura 2-2** se establecieron los actores y procesos que realizan en la veterinaria en este caso adopciones que hace la veterinaria, A continuación, se muestra las diferentes actividades que realizan los actores del sistema.

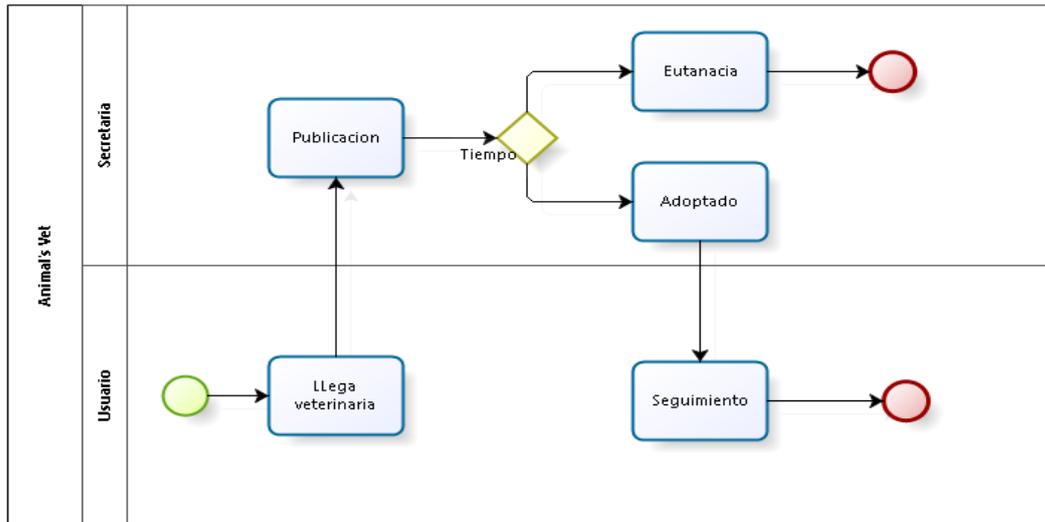


Figura 2-2: Diagrama de procesos de adopciones de la veterinaria Animal's Vet
 Realizado por: Carlos Grefa y Bernabé Aragón, 2019

En la **Figura 3-2** se establecieron los actores y procesos que realizan en la veterinaria en este caso los proveedores e insumos de la veterinaria, A continuación, se muestra las diferentes actividades que realizan los actores del sistema.

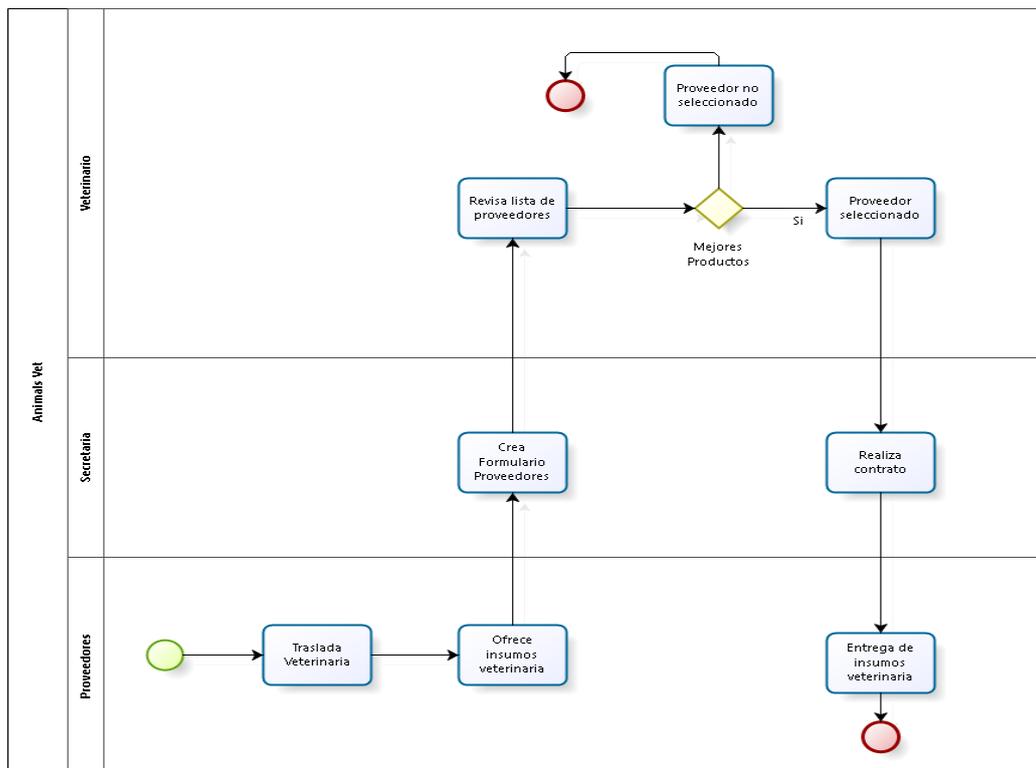


Figura 3-2: Diagrama de procesos proveedores, insumos de la veterinaria Animal's Vet.
 Realizado por: Carlos Grefa y Bernabé Aragón, 2019

2.7.1. Especificación de requerimientos

Un requerimiento es una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal característica que el sistema debe tener o es una restricción que el sistema debe satisfacer para ser aceptada por el cliente. (Camacho Antonio, 2005, p. 23)

Los requerimientos para el Sistema Web fueron establecidos y priorizados conjuntamente con la Dr. Diana Paucar. Mediante varias reuniones en las que se estableció las funcionalidades del sistema con un total de 70 tareas a realizar las cuales están divididas en 58 historias de usuario y 12 historias técnicas.

Requerimientos Funcionales

Historia clínica

1. Ingresar la revisión clínica del animal.
2. Ingresar las razas de caninos y felinos.
3. Modificar la revisión clínica del animal.
4. Buscar y listar la revisión clínica del animal.
5. Eliminar la revisión clínica del animal.
6. Visualizar la revisión clínica.

Usuario

1. Modificar personas que traerán y adoptarán a los animales
2. Buscar y listar personas que traerán y adoptarán a los animales
3. Visualizar la información de personas que traerán y adoptarán a los animales
4. Visualizar la información de razas de caninos y felinos
5. Buscar y listar los usuarios
6. Modificar al usuario
7. Modificar razas de caninos y felinos
8. Buscar y listar razas de caninos y felinos
9. Eliminar razas de caninos y felinos
10. Eliminar personas que traerán y adoptarán a los animales al centro
11. Eliminar al usuario
12. Visualizar los datos del usuario
13. Desarrollar la parte de autenticación de usuario para el sistema.

Alarma

1. Ingresar la alarma
2. Modificar la alarma

3. Buscar y listar las alarmas
4. Eliminar la alarma

Rol de usuario

1. Ingresar al usuario que utilizará el sistema informático
2. Ingresar el rol de usuario
3. Modificar el rol de usuario
4. Buscar y listar los roles de usuario
5. Eliminar el rol de usuario

Tipo domicilio

1. Ingresar el tipo de domicilio
2. Modificar el tipo de domicilio
3. Buscar y listar los tipos de domicilio
4. Eliminar el tipo de domicilio

Tipo cirugía

1. Ingresar las cirugías del animal
2. Modificar cirugías del animal y registrar cuantos insumos se utilizaron.
3. Ingresar el tipo de cirugía
4. Buscar y listar los insumos de cirugía
5. Ingresar insumos que usaran en las cirugías
6. Eliminar la cirugía del animal y los insumos que se utilizaron
7. Visualizar la información de cirugía del animal y los insumos que se utilizaron
8. Ingresar personas que traerán y adoptarán a los animales
9. Modificar el tipo de cirugía
10. Buscar y listar los tipos de cirugía
11. Modificar insumos de cirugía
12. Buscar y listar cirugías del animal
13. Eliminar insumos de cirugía
14. Visualizar la información de los insumos de cirugía
15. Eliminar el tipo de cirugía

Proveedores

1. Ingresar proveedores de insumos
2. Modificar proveedores de insumos
3. Buscar y listar proveedores de insumos
4. Eliminar proveedores de insumos
5. Visualizar la información de los proveedores de insumos

Diferentes categorías de tipos

Enfermedad

- Ingresar Enfermedad.
- Modificar Enfermedad.
- Visualizar Enfermedad.
- Buscar Enfermedad.
- Listar Enfermedad.

Síntomas

- Ingresar Síntomas.
- Modificar Síntomas.
- Visualizar Síntomas.
- Buscar Síntomas.
- Listar Síntomas.

Cirugía

- Ingresar Cirugía.
- Modificar Cirugía.
- Visualizar Cirugía.
- Buscar Cirugía.
- Listar Cirugía.

Domicilio

- Ingresar Domicilio.
- Modificar Domicilio.
- Visualizar Domicilio.
- Buscar Domicilio.
- Listar Domicilio.

Alarma

- Ingresar Alarma
- Modificar Alarma.
- Visualizar Alarma
- Buscar Alarma
- Listar Alarma.

Rol de Usuarios

- Ingresar Rol de Usuarios
- Modificar Rol de Usuarios
- Visualizar Rol de Usuarios
- Buscar Rol de Usuarios.
- Listar Rol de Usuarios.

Animal.

1. Ingresar Animal.
2. Modificar Animal.

3. Visualizar Animal
4. Buscar Animal.
5. Listar Animal.

Reportes

1. Realizar el reporte de mascotas adoptadas
2. Realizar el reporte de mascotas a adoptar
3. Realizar el reporte de la revisión clínica del animal
4. Realizar el reporte de los usuarios registrados
5. Realizar el reporte de alarmas.
6. Realizar el reporte de tipo de domicilios.
7. Realizar el reporte de roles de usuario.
8. Realizar el reporte de tipos de cirugía.

Requerimientos no funcionales

- Modularidad
- Reusabilidad
- Analizabilidad
- Capacidad para ser modificado
- Capacidad para ser probado

2.7.2. Análisis y Gestión de riesgos

El análisis de riesgos es un evento cuya probabilidad es incierta, pero en el caso de aparecer, tiene un efecto sobre los objetivos a realizarse en el sistema web, este tiene como propósito determinar los componentes del sistema que están expuestos a amenazas que estos pueden sufrir. El análisis de riesgos permitirá tomar medidas anticipadas las cuales certifiquen que el sistema web se desarrolle de manera satisfactoria.

Tabla 8-2: Riesgos Proyecto

RIESGOS DEL PROYECTO			
ID	Descripción	Tipo	Consecuencia
R1	Falta de compromiso del usuario con el cronograma de trabajo establecido	Proyecto	<ul style="list-style-type: none"> • Retraso en la entrega del proyecto. • No cumplen con las funcionalidades del sistema
R2	Ausencia temporal por parte de algún miembro del equipo de trabajo.	Proyecto	<ul style="list-style-type: none"> • Sobrecarga laboral al resto del equipo. • Retraso en la planificación.
R3	Mal ambiente de trabajo entre los integrantes del equipo de trabajo.	Proyecto	<ul style="list-style-type: none"> • La falta de comunicación provoca retrasos a la hora del desarrollo del sistema web.

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 8-2** nos indica los riesgos del proyecto tipo proyecto que pueda llegar a tener el sistema.

Tabla 9-2: Riesgos Proyecto

RIESGOS DEL TÉCNICOS			
ID	Descripción	Tipo	Consecuencia
R4	Interfaz poco amigable para el usuario	Técnico	<ul style="list-style-type: none"> Mal uso del sistema web por parte del usuario No se llega a utilizar todas las funcionalidades realizadas del sistema
R5	Falta de acceso a internet.	Técnico	<ul style="list-style-type: none"> Pérdida de Tiempo Retraso del Sistema.
R6	Diseño de la base de datos mal estructurada.	Técnico	<ul style="list-style-type: none"> Retraso en el Sistema Rediseño de la base de datos.

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 9-2** nos indica los riesgos del proyecto tipo técnico que pueda llegar a tener el sistema.

Tabla 10-2: Riesgos Negocio

RIESGOS DEL NEGOCIO			
ID	Descripción	Tipo	Consecuencia
R7	Cancelación de actividades de la veterinaria.	Negocio	<ul style="list-style-type: none"> Fracaso del proyecto
R8	Falta de compromiso por parte del personal.	Negocio	<ul style="list-style-type: none"> Información inconsistente para el desarrollo. Mal diseño del sistema.

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 10-2** nos indica los riesgos del proyecto tipo negocio que pueda llegar a tener el sistema.

Priorización de los riesgos

Se realiza la priorización de los riesgos con el objetivo de determinar el respectivo orden a los cuales deben ser considerados cada uno de los riesgos, así tener un mejor control durante la ejecución del sistema.

Se realizó un análisis de los riesgos de acuerdo al rango (baja, media o alta), así como el valor para precisar su prioridad, dicho estudio se basó en los posibles riesgos que se podrían ocurrir en las diferentes etapas del sistema.

Luego de realizar su respectivo análisis de los riesgos que puedan ocurrir durante el desarrollo del sistema se procedió a determinar la prioridad de acuerdo a los valores que se tienen en la exposición al riesgo.

Tabla 11-2: Rango Probabilidades

RANGO DE PROBABILIDADES	0%	1%-33%	34%-67%	68%-99%
DESCRIPCIÓN	Insignificante	Baja	Media	Alta
VALOR	1	2	3	4

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En la **Tabla 11-2** nos indica el rango de probabilidades que pueda llegar a tener el sistema.

Tabla 12-2: Impacto Riesgo

IMPACTO	Bajo	Medio	Alto
RETRASO	1 semana	2 semanas	1 mes
IMPACTO TÉCNICO	Ligero efecto en el desarrollo del proyecto.	Moderado efecto en el desarrollo del proyecto.	Severo efecto en el desarrollo del proyecto.
VALOR	2	3	4

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En este apartado se ha analizado la probabilidad de cada uno de los riesgos otorgando cierto porcentaje, probabilidad y valor en cuanto pueda ocurrir cada uno de los riesgos que se describen en la **Tabla 13-2**. Obteniendo así valores que sirven para ir clasificando de mayor a menor los riesgos que pueden presentarse en el sistema.

Tabla 13-2: Probabilidades

ID	DESCRIPCIÓN	PROBABILIDAD		
		Porcentaje	Probabilidad	Valor
R1	Falta de compromiso del usuario con el cronograma de trabajo establecido	32%	Baja	2
R2	Ausencia temporal por parte de algún miembro del equipo de trabajo.	70%	Alta	4
R3	Mal ambiente de trabajo entre los integrantes del equipo de trabajo.	39%	Medio	3
R4	Interfaz poco amigable para el usuario	50%	Medio	3
R5	Falta de acceso a internet.	10%	Baja	2
R6	Diseño de la base de datos mal estructurada.	45%	Medio	3
R7	Cancelación de actividades de la veterinaria.	25%	Baja	2
R8	Falta de compromiso por parte del personal.	10%	Baja	2

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En este apartado se ha analizado el impacto de cada uno de los riesgos otorgando cierto, impacto y valor en cuanto pueda ocurrir cada uno de los riesgos que se describen en la **Tabla 14-2**. Obteniendo así valores que sirven para ir clasificando de mayor a menor los riesgos que pueden presentarse en el sistema.

Tabla 14-2: Impacto Riesgos

ID	DESCRIPCIÓN	IMPACTO	
		Impacto	Valor
R1	Falta de compromiso del usuario con el cronograma de trabajo establecido	Alto	4
R2	Ausencia temporal por parte de algún miembro del equipo de trabajo.	Alto	4
R3	Mal ambiente de trabajo entre los integrantes del equipo de trabajo.	Media	3
R4	Interfaz poco amigable para el usuario	Media	3
R5	Falta de acceso a internet.	Alto	4
R6	Diseño de la base de datos mal estructurada.	Media	3
R7	Cancelación de actividades de la veterinaria.	Alta	4
R8	Falta de compromiso por parte del personal.	Alta	4

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

En este apartado se ha analizado la exposición de cada uno de los riesgos otorgando cierta, exposición y valor en cuanto pueda ocurrir cada uno de los riesgos que se describen en la **Tabla 15-2**. Obteniendo así valores que sirven para ir clasificando de mayor a menor los riesgos que pueden presentarse en el sistema.

Tabla 15-2: Exposición Riesgo

ID	DESCRIPCIÓN	EXPOSICIÓN	
		Exposición	Valor
R1	Falta de compromiso del usuario con el cronograma de trabajo establecido	Medio	3
R2	Ausencia temporal por parte de algún miembro del equipo de trabajo.	Alto	9
R3	Mal ambiente de trabajo entre los integrantes del equipo de trabajo.	Bajo	2
R4	Interfaz poco amigable para el usuario	Media	4
R5	Falta de acceso a internet.	Medio	4
R6	Diseño de la base de datos mal estructurada.	Media	4
R7	Cancelación de actividades de la veterinaria.	Alto	6
R8	Falta de compromiso por parte del personal.	Medio	3

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

Es un punto muy importante en el desarrollo del sistema ya que mediante la exposición de los riesgos que se muestra en la **Tabla 16-2** podemos tener en cuenta que van hacer los primeros riesgos en ser gestionados para evitar tener un impacto negativo dentro del sistema para eso hemos tomado valores que van de 1 a 2 identificado con un color verde que representa un exposición baja, de 3 a 5 representado por un color naranja y representa una exposición media y mientras que

los valores que van de 6 a 9 con un color rojo representa que van a tener una exposición alta en el desarrollo de nuestro sistema.

Tabla 16-2: Exposición Riesgo

ID	DESCRIPCIÓN	PRIORIDAD	EXPOSICIÓN AL RIESGO	
			Valor	Exposición
R2	Ausencia temporal por parte de algún miembro del equipo de trabajo.	4	9	ALTA
R7	Cancelación de actividades de la veterinaria.	4	6	ALTA
R1	Falta de compromiso del usuario con el cronograma de trabajo establecido.	3	3	MEDIA
R4	Interfaz poco amigable para el usuario	3	4	MEDIA
R5	Falta de acceso a internet.	3	4	MEDIA
R6	Diseño de la base de datos mal estructurada.	3	4	MEDIA
R8	Falta de compromiso por parte del personal.	3	3	MEDIA
R3	Mal ambiente de trabajo entre los integrantes del equipo de trabajo.	2	2	BAJA

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8. Fase de planificación

Para el cumplimiento de las tareas establecidas se utilizó la metodología Scrum el cual ayudó de manera satisfactoria la planificación del mismo. Se detallará las iteraciones las mismas que tienen un tiempo de duración y una fecha de inicio que es el 06 de marzo del 2019.

La planificación del proyecto a desarrollarse permite proporcionar un marco de trabajo en el cual se puede hacer estimaciones razonables de recursos, costos durante la ejecución del sistema.

Para el desarrollo del sistema, se establecen 5 sprints haciendo referencia cada uno de ellos a los módulos contemplados en el sistema.

Para realizar la estimación de cada tarea se utiliza el método de la talla de la camiseta o T-shirt. Las tallas o estimaciones del método son XS, S, M Y L como se lo presenta en la **Tabla 17-2**, a continuación:

Un punto estimado es igual a una hora de trabajo; un día de trabajo es de 4 horas realizado por una sola persona, por consecuente una semana de trabajo (5 días) cada sprint será de 4 semanas.

Tabla 17-2: Método de estimación T-shirt

Talla	Puntos estimados	Horas de trabajo
XS	6	5
S	18	10
XM	30	20
M	60	40
L	90	80

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8.1. *Historias de usuario*

Las historias de usuario representan los distintos requerimientos que se logran identificar en las diferentes reuniones donde se estableció los diferentes procesos que vaya a contener el sistema web. Las historias de usuario deben ser claras y concisas para que los programadores puedan agregar al sistema web en el tiempo que dure el sprint.

Las historias de usuario se encuentran separados en diferentes campos como son:

- ID.
- Nombre.
- Descripción.
- Responsable.
- Esfuerzo.
- Historias Técnicas.
- Pruebas de aceptación.
- Tareas de Ingeniería.

2.8.2. Product backlog

Product Backlog es una lista de los requerimientos que se van a desarrollarse en el sistema web. En las siguientes tablas se visualizan las diferentes tareas que se desarrollaran a lo largo del sistema.

Tabla 18-2: Prioridad Alta

Prioridad Alta		
ID	TAREAS	ESTIMACIÓN
HT-01	Cómo desarrollador como está estructurada la base de datos dentro de veterinaria.	5
HT-02	Cómo desarrollador necesito diseñar la base de datos en PostgreSQL	5
HT-03	Cómo desarrollador necesito conectarme a la base de datos en PostgreSQL.	5
HT-04	Cómo desarrollador necesito elaborar el diccionario de datos	5
HT-05	Cómo desarrollador requiero diseñar y desarrollar la interfaz de usuario.	5
HT-06	Cómo desarrollador necesito desarrollar la página principal del sistema.	5
HT-07	Cómo desarrollador requiero desarrollar la parte de autenticación de usuario.	5
HT-08	Cómo desarrollador requiero la Documentación del sistema requerido	5
HT-09	Cómo desarrollador requiero verificar el funcionamiento del sistema mediante pruebas reales.	20
HT-10	Cómo desarrollador requiero realizar el despliegue de la aplicación.	20
HU-01	Cómo desarrollador necesito ingresar al usuario que utilizará el sistema informático	20
HU-02	Cómo desarrollador necesito ingresar la alarma	20
HU-03	Cómo desarrollador necesito ingresar el tipo de domicilio	20
HU-04	Cómo desarrollador necesito ingresar el rol de usuario	20
HU-05	Cómo desarrollador necesito ingresar el tipo de cirugía	20
HU-06	Cómo desarrollador necesito ingresar la cirugías del animal	20
HU-07	Cómo desarrollador se necesita modificar cirugías del animal y registrar cuantos insumos se utilizaron.	5
HU-08	Cómo desarrollador se necesita realizar el reporte de mascotas adoptadas.	5

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

Tabla 19-2: Prioridad Media

PRIORIDAD MEDIA		
ID	TAREAS	ESTIMACIÓN
HU-09	Cómo desarrollador necesito modificar la revisión clínica del animal	5
HU-10	Cómo desarrollador necesito modificar al usuario	5
HU-11	Cómo desarrollador necesito modificar la alarma	5
HU-12	Cómo desarrollador necesito modificar el tipo de domicilio	5
HU-13	Cómo desarrollador necesito modificar el rol de usuario	5
HU-14	Cómo desarrollador necesito modificar el tipo de cirugía	5
HU-15	Cómo desarrollador necesito buscar y listar la revisión clínica del animal	10
HU-16	Cómo desarrollador necesito buscar y listar los usuarios	10
HU-17	Cómo desarrollador necesito buscar y listar las alarmas	40
HU-18	Cómo desarrollador necesito buscar y listar los tipos de domicilio	40
HU-19	Cómo desarrollador necesito buscar y listar los roles de usuario	40
HU-20	Cómo desarrollador necesito buscar y listar los tipos de cirugía	40
HU-21	Cómo desarrollador necesito modificar personas que traerán y adoptarán a los animales.	5
HU-22	Cómo desarrollador necesito buscar y listar los insumos de cirugía	5
HU-23	Cómo desarrollador necesito ingresar insumos que usaran en las cirugías	5
HU-24	Cómo desarrollador necesito buscar y listar personas que traerán y adoptarán a los animales.	5
HU-25	Cómo desarrollador necesito eliminar la cirugía del animal y los insumos que se utilizaron	20
HU-26	Cómo desarrollador necesito visualizar la información de cirugía del animal y los insumos que se utilizaron	10
HU-27	Cómo desarrollador necesito visualizar la información de personas que traerán y adoptarán a los animales.	10
HU-28	Cómo desarrollador necesito realizar el reporte de mascotas a adoptar	20
HU-29	Cómo desarrollador necesito ingresar proveedores de insumos	5
HU-30	Cómo desarrollador necesito ingresar personas que traerán y adoptarán a los animales.	5
HU-31	Cómo desarrollador necesito necesita ingresar las razas de caninos y felinos	5

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

Tabla 20-2: Prioridad Baja

PRIORIDAD BAJA		
ID	TAREAS	ESTIMACIÓN
HU-32	Cómo desarrollador necesito eliminar la revisión clínica del animal	5
HU-33	Cómo desarrollador necesito eliminar al usuario	10
HU-34	Cómo desarrollador necesito eliminar la alarma	10
HU-35	Cómo desarrollador necesito eliminar el tipo de domicilio	5
HU-36	Cómo desarrollador necesito eliminar el rol de usuario	5
HU-37	Cómo desarrollador necesito eliminar el tipo de cirugía	5
HU-38	Cómo desarrollador necesito visualizar la revisión clínica	5
HU-39	Cómo desarrollador necesito visualizar los datos del usuario	20
HU-40	Cómo desarrollador necesito realizar el reporte de la revisión clínica del animal	20
HU-41	Cómo desarrollador necesito realizar el reporte de los usuarios registrados	5
HU-42	Cómo desarrollador necesito realizar el reporte de alarmas	5
HU-43	Cómo desarrollador necesito realizar el reporte de tipo de domicilios	5
HU-44	Cómo desarrollador necesito realizar el reporte de roles de usuario	5
HU-45	Cómo desarrollador necesito realizar el reporte de tipos de cirugía	40
HU-46	Cómo desarrollador necesito modificar insumos de cirugía	20
HU-47	Cómo desarrollador necesito modificar proveedores de insumos	10
HU-48	Cómo desarrollador necesito modificar razas de caninos y felinos	10
HU-49	Cómo desarrollador necesito buscar y listar cirugías del animal	5
HU-50	Cómo desarrollador necesito buscar y listar proveedores de insumos	5
HU-51	Cómo desarrollador necesito buscar y listar razas de caninos y felinos	5
HU-52	Cómo desarrollador necesito eliminar insumos de cirugía	40
HU-53	Cómo desarrollador necesito eliminar proveedores de insumos	20
HU-54	Cómo desarrollador necesito eliminar razas de caninos y felinos	20
HU-55	Cómo desarrollador necesito eliminar personas que traerán y adoptarán a los animales.	5
HU-56	Cómo desarrollador necesito visualizar la información de los insumos de cirugía	5
HU-57	Cómo desarrollador necesito visualizar la información de los proveedores de insumos	5
HU-58	Cómo desarrollador necesito visualizar la información de razas de caninos y felinos	5
HT-11	Se requiere capacitar a los usuarios sobre el manejo del sistema y obtener recomendaciones.	5
HT-12	Se requiere el Manual técnico de los módulos del sistema	5
HT-13	Se requiere el Manual del Usuario para una guía de manejo del sistema.	10

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8.3. *Sprint Product backlog*

En la **Tabla 21-2** se detalla el plan de entrega se detalla cada iteración, se presenta el listado de tareas para el desarrollo del sistema web donde se presenta el esfuerzo realizado en horas.

Cada Sprint está compuesto por diferentes historias técnicas o de usuario, además se especifica que tiene una duración de 4 semanas, donde cada día equivale a 4 horas por persona, una semana 20 horas, finalmente dando un total de 160 horas por sprint.

Tabla 21-2: Sprint Backlog

N° Sprint	ID	N° Horas	Fecha Inicio	Fecha Fin	N° horas
1	HT-01	5	06/03/2019	03/04/2019	160
	HT-02	5			
	HT-03	5			
	HT-04	5			
	HT-05	5			
	HT-06	5			
	HT-07	5			
	HT-08	5			
	HT-09	20			
	HT-10	20			
	HU-01	20			
	HU-02	20			
	HU-03	20			
	HU-04	20			
2	HU-05	20	04/04/2019	03/05/2019	160
	HU-06	20			
	HU-07	5			
	HU-08	5			
	HU-09	5			
	HU-10	5			
	HU-11	5			
	HU-12	5			
	HU-13	5			

	HU-14	5			
	HU-15	10			
	HU-16	10			
	HU-17	40			
	HU-18	20			
3	HU-19	20	05/05/2019	03/06/2019	160
	HU-20	40			
	HU-21	5			
	HU-22	5			
	HU-23	5			
	HU-24	5			
	HU-25	20			
	HU-26	10			
	HU-27	10			
	HU-28	20			
	HU-29	5			
	HU-30	5			
	HU-31	5			
HU-32	5				
4	HU-33	10	04/06/2019	03/07/2019	160
	HU-34	10			
	HU-35	5			
	HU-36	5			
	HU-37	5			
	HU-38	5			
	HU-39	20			
	HU-40	20			
	HU-41	5			
	HU-42	5			

	HU-43	5			
	HU-44	5			
	HU-45	40			
	HU-46	20			
5	HU-47	10	04/07/2019	23/07/2019	160
	HU-48	10			
	HU-49	5			
	HU-50	5			
	HU-51	5			
	HU-52	40			
	HU-53	20			
	HU-54	20			
	HU-55	5			
	HU-56	5			
	HU-57	5			
	HU-58	5			
	HT-11	5			
HT-12	5				
TOTAL HORAS					800

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8.4. *Sprint 1.*

Para el primer sprint se muestra se realizó la recopilación de información referente a las necesidades determinando los requerimientos para el proyecto, basándose en esta información se definió la arquitectura del sistema como se describe en la siguiente **Tabla 22-2**.

Tabla 22-2: Detalle Sprint 1

Sprint 1				
Inicio: 03/06/2019	Fin: 03/04/2019	Esfuerzo Total: 160		
Pila del Sprint				
Backlog ID	Descripción	Esfuerzo	Tipo	Responsable
HT-01	Requiero una recopilación de requerimientos.	5	Análisis	Carlos Grefa
HT-02	Requiero realizar una planificación de trabajo para el sistema informático	5	Análisis	Bernabé Aragón
HT-03	Requiero definir las historias de usuario	5	Análisis	Carlos Grefa
HT-04	Requiero conocer como está estructurada la base de datos dentro de veterinaria.	5	Análisis	Bernabé Aragón
HT-05	Necesito diseñar la base de datos en PostgreSQL	5	Análisis	Carlos Grefa
HT-06	Necesito conectarme a la base de datos en PostgreSQL.	5	Análisis	Bernabé Aragón
HT-07	Necesito elaborar el diccionario de datos	5	Análisis	Carlos Grefa
HT-08	Requiero diseñar y desarrollar la interfaz de usuario.	5	Análisis	Bernabé Aragón
HT-09	Necesito desarrollar la página principal del sistema.	20	Análisis	Carlos Grefa
HT-10	Requiero desarrollar la parte de autenticación de usuario.	20	Análisis	Bernabé Aragón
HU-01	Necesito ingresar al usuario que utilizará el sistema informático	20	Desarrollo	Carlos Grefa
HU-02	Se necesita ingresar la alarma	20	Desarrollo	Bernabé Aragón
HU-03	Se necesita ingresar el tipo de domicilio	20	Desarrollo	Carlos Grefa
HU-04	Se necesita ingresar el rol de usuario	20	Desarrollo	Bernabé Aragón

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8.5. *Sprint 2*

Corresponde al desarrollo de los proceso de ingreso y modificación para el sistema, de buscar y listar para los módulos ya descritos. Los requisitos que se desarrollaron de describen en la siguiente **Tabla 23-2**.

Tabla 23-2: Detalle Sprint 2

Sprint 2				
Inicio: 04/04/2019	Fin: 03/05/2019	Esfuerzo Total: 160		
Pila del Sprint				
Backlog ID	Descripción	Esfuerzo	Tipo	Responsable
HU-05	Se necesita ingresar el tipo de cirugía	20	Desarrollo	Bernabé Aragón
HU-06	Se necesita ingresar la cirugías del animal	20	Desarrollo	Carlos Grefa
HU-07	Se necesita modificar cirugías del animal y registrar cuantos insumos se utilizaron.	5	Desarrollo	Bernabé Aragón
HU-08	Se necesita realizar el reporte de mascotas adoptadas.	5	Desarrollo	Carlos Grefa
HU-09	Se necesita modificar la revisión clínica del animal	5	Desarrollo	Bernabé Aragón
HU-10	Se necesita modificar al usuario	5	Desarrollo	Carlos Grefa
HU-11	Se necesita modificar la alarma	5	Desarrollo	Bernabé Aragón
HU-12	Se necesita modificar el tipo de domicilio	5	Desarrollo	Carlos Grefa
HU-13	Se necesita modificar el rol de usuario	5	Desarrollo	Bernabé Aragón
HU-14	Se necesita modificar el tipo de cirugía	5	Desarrollo	Carlos Grefa
HU-15	Se necesita buscar y listar la revisión clínica del animal	10	Desarrollo	Bernabé Aragón
HU-16	Se necesita buscar y listar los usuarios	10	Desarrollo	Carlos Grefa
HU-17	Se necesita buscar y listar las alarmas	40	Desarrollo	Bernabé Aragón
HU-18	Se necesita buscar y listar los tipos de domicilio	20	Desarrollo	Carlos Grefa

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8.6. *Sprint 3*

Corresponde al desarrollo de los proceso de buscar, listar y modificar para el sistema, para los módulos ya descritos. Los requisitos que se desarrollaron de describen en la siguiente **Tabla 24-2**.

Tabla 24-2: Detalle Sprint 3

Sprint 3				
Inicio: 05/05/2019	Fin: 03/06/2019	Esfuerzo Total: 160		
Pila del Sprint				
Backlog ID	Descripción	Esfuerzo	Tipo	Responsable
HU-19	Se necesita buscar y listar los roles de usuario	20	Desarrollo	Carlos Grefa
HU-20	Se necesita buscar y listar los tipos de cirugía	20	Desarrollo	Bernabé Aragón
HU-21	Se necesita modificar personas que traerán y adoptaran a los animales.	5	Desarrollo	Carlos Grefa
HU-22	Se necesita buscar y listar los insumos de cirugía	5	Desarrollo	Bernabé Aragón
HU-23	Se necesita ingresar insumos que usaran en las cirugías	5	Desarrollo	Carlos Grefa
HU-24	Se necesita buscar y listar personas que traerán y adoptaran a los animales.	5	Desarrollo	Bernabé Aragón
HU-25	Se necesita eliminar la cirugía del animal y los insumos que se utilizaron	5	Desarrollo	Carlos Grefa
HU-26	Se necesita visualizar la información de cirugía del animal y los insumos que se utilizaron	5	Desarrollo	Bernabé Aragón
HU-27	Se necesita visualizar la información de personas que traerán y adoptaran a los animales.	5	Desarrollo	Carlos Grefa
HU-28	Se necesita realizar el reporte de mascotas a adoptar	5	Desarrollo	Bernabé Aragón
HU-29	Se necesita ingresar proveedores de insumos	10	Desarrollo	Carlos Grefa
HU-30	Se necesita ingresar personas que traerán y adoptaran a los animales.	10	Desarrollo	Bernabé Aragón
HU-31	Se necesita ingresar las razas de caninos y felinos	40	Desarrollo	Carlos Grefa
HU-32	Se necesita eliminar la revisión clínica del animal	20	Desarrollo	Bernabé Aragón

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8.7. *Sprint 4*

Corresponde al desarrollo de los procesos de eliminar y visualizar para el sistema, de buscar y los módulos ya descritos. Los requisitos que se desarrollaron de describen en la siguiente **Tabla 25-2**.

Tabla 25-2: Detalle Sprint 4

Sprint 4				
Inicio: 04/06/2019	Fin: 03/07/2019	Esfuerzo Total: 160		
Pila del Sprint				
Backlog ID	Descripción	Esfuerzo	Tipo	Responsable
HU-33	Se necesita eliminar al usuario	10	Desarrollo	Bernabé Aragón
HU-34	Se necesita eliminar la alarma	10	Desarrollo	Carlos Grefa
HU-35	Se necesita eliminar el tipo de domicilio	5	Desarrollo	Bernabé Aragón
HU-36	Se necesita eliminar el rol de usuario	5	Desarrollo	Carlos Grefa
HU-37	Se necesita eliminar el tipo de cirugía	5	Desarrollo	Bernabé Aragón
HU-38	Se necesita visualizar la revisión clínica	5	Desarrollo	Carlos Grefa
HU-39	Se necesita visualizar los datos del usuario	20	Desarrollo	Bernabé Aragón
HU-40	Se necesita realizar el reporte de la revisión clínica del animal	20	Desarrollo	Carlos Grefa
HU-41	Se necesita realizar el reporte de los usuarios registrados	5	Desarrollo	Bernabé Aragón
HU-42	Se necesita realizar el reporte de alarmas	5	Desarrollo	Carlos Grefa
HU-43	Se necesita realizar el reporte de tipo de domicilios	5	Desarrollo	Bernabé Aragón
HU-44	Se necesita realizar el reporte de roles de usuario	5	Desarrollo	Carlos Grefa
HU-45	Se necesita realizar el reporte de tipos de cirugía	40	Desarrollo	Bernabé Aragón
HU-46	Se necesita modificar insumos de cirugía	20	Desarrollo	Carlos Grefa

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.8.8. *Sprint 5*

Corresponde a los requerimientos de eliminar, visualizar y los primeros reportes, reportes del sistema. Los requisitos que se desarrollaron de describen en la siguiente **Tabla 26-2**.

Tabla 26-2: Detalle Sprint 5

Sprint 5				
Inicio: 04/06/2019	Fin: 03/07/2019	Esfuerzo Total: 160		
Pila del Sprint				
Backlog ID	Descripción	Esfuerzo	Tipo	Responsable
HU-47	Se necesita modificar proveedores de insumos	10	Desarrollo	Carlos Grefa
HU-48	Se necesita modificar razas de caninos y felinos	10	Desarrollo	Bernabé Aragón
HU-49	Se necesita buscar y listar cirugías del animal	5	Desarrollo	Carlos Grefa
HU-50	Se necesita buscar y listar proveedores de insumos	5	Desarrollo	Bernabé Aragón
HU-51	Se necesita buscar y listar razas de caninos y felinos	5	Desarrollo	Carlos Grefa
HU-52	Se necesita eliminar insumos de cirugía	40	Desarrollo	Bernabé Aragón
HU-53	Se necesita eliminar proveedores de insumos	20	Desarrollo	Carlos Grefa
HU-54	Se necesita eliminar razas de caninos y felinos	20	Desarrollo	Bernabé Aragón
HU-55	Se necesita eliminar personas que traerán y adoptaran a los animales.	5	Desarrollo	Carlos Grefa
HU-56	Se necesita visualizar la información de los insumos de cirugía	5	Desarrollo	Bernabé Aragón
HU-57	Se necesita visualizar la información de los proveedores de insumos	5	Desarrollo	Carlos Grefa
HU-58	Se necesita visualizar la información de razas de caninos y felinos	5	Desarrollo	Bernabé Aragón
HT-11	Se requiere capacitar a los usuarios sobre el manejo del sistema.	5	Desarrollo	Carlos Grefa
HT-12	Se requiere el Manual técnico de los módulos del sistema	5	Desarrollo	Bernabé Aragón

Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.9. Fase de desarrollo

2.9.1. Diseño de la arquitectura

Se establece la manera como va estar organizado el sistema Desarrollo de un sistema web para la gestión de la veterinaria “Animal’s Vet”. Se muestra **Figura 4-2** la arquitectura MVC (Modelo-Vista-Controlador) ya que permite tener grandes cantidades de datos el código se tiene de una manera estructurada y organizada en diferentes capas, el modelo es donde llega la información de los datos de las mascotas, la vista es la que se encarga de recibir los datos del modelo y el controlador es el que se encarga de interactuar el modelo y la vista, el sistema gestor de base de datos que se utilizó fue PostgreSQL para almacenar la información.

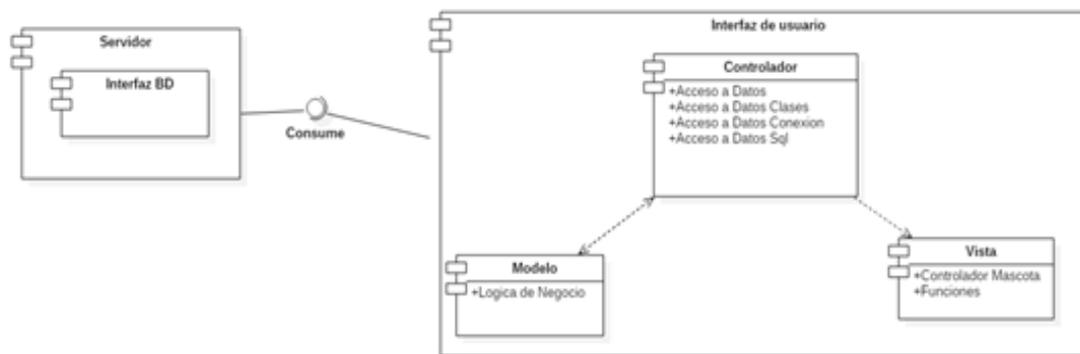


Figura 4-2: Diseño de la arquitectura
Realizado por: Carlos Grefa, Bernabé Aragón, 2019

2.9.2. Estándar de programación

Para llevar un mejor registro al momento de la programación se escogió el estándar de programación denominado Upper CamelCase el cual dará una mejor visión al momento de realizar el sistema “Desarrolla de un sistema web para la gestión de los procesos de la veterinaria Animal’s Vet”.

2.9.3. Diseño de interfaces

Módulos del sistema

Administrador: Para el sistema se requiere este usuario quien se encarga del ingreso, modificación y eliminación de algunas entidades del sistema

Veterinario: El veterinario es el usuario quien registra los animales, las revisiones clínicas y cirugías que se encuentra a su cargo y genera su reporte.

Secretaria: La secretaria es el usuario que puede realizar el registro de animales y realizar el proceso de adopción.

Proveedor: Los proveedores pueden realizar una competencia de entre los diferentes que concursan y puedan facturar los insumos a la veterinaria.

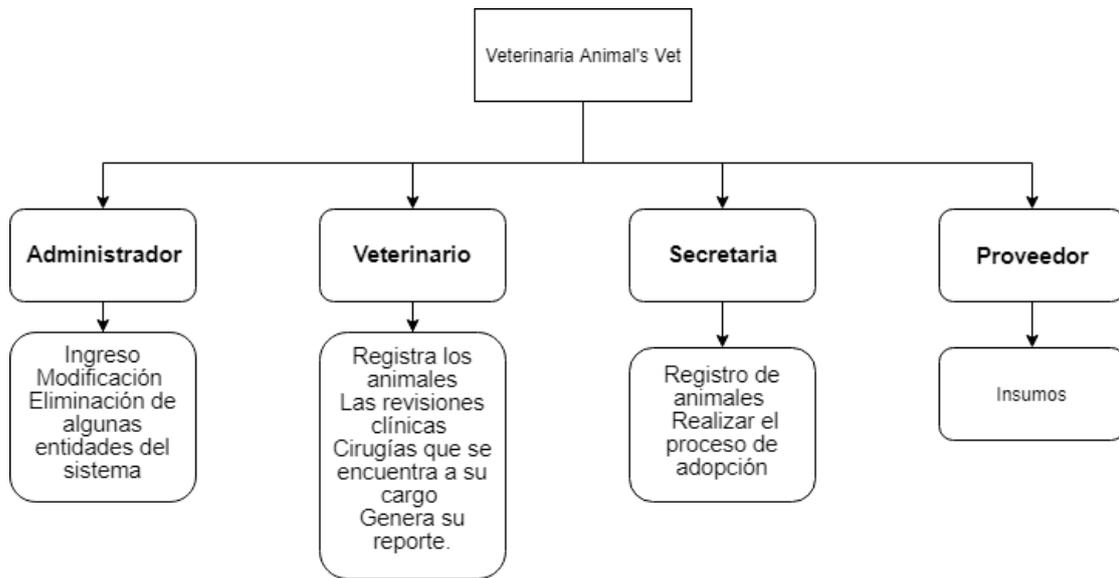


Figura 5-2: Módulos del sistema de la veterinaria Animal's Vet.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

En la **Figura 5-2** se muestra los módulos que tiene el sistema web.

2.9.4. Diseño de la base de datos

La Base de Datos para el Sistema Web está elaborado en PostgreSQL y cuenta con 17 tablas y 124 funciones para realizar las acciones de ingreso, modificación, eliminación y búsquedas.

En las **Figura 6-2, Figura 7-2, Figura 8-2, Figura 9-2** el esquema de la base de datos, se observa la relación entre tablas de la base de datos, y los campos que cada tabla contiene con sus respectivos tipos de dato (modelo lógico). DbVisualizer es la herramienta que se utilizó para diseñar la base de datos.

La base de datos del sistema está conformada por la tabla animal, la cual se relación con las tablas historia clínica y cirugía; la tabla historia clínica se relaciona con las tablas síntoma y enfermedad, debido a que una revisión clínica se puede o no registrar síntomas o enfermedades. La tabla cirugía se relaciona con las tablas tipo cirugía y con la tabla insumos, esta tabla posee los instrumentos que se utilizan para realizar una cirugía.

Otra de las relaciones entre la tabla animal es con la tabla persona, por la razón de quien adopta a un animal se registra a la persona, al igual quien deja al animal en el albergue. La tabla raza también se relaciona con la tabla animal, de la misma manera se relaciona la tabla usuario con la tabla animal para conocer quien registro al animal al sistema.

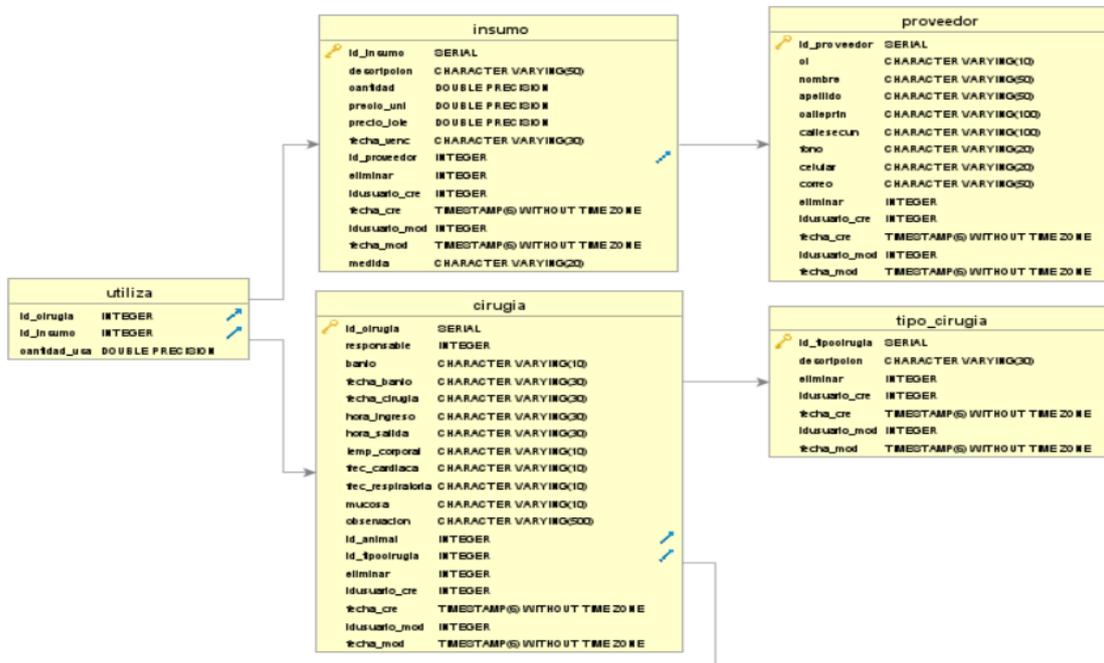
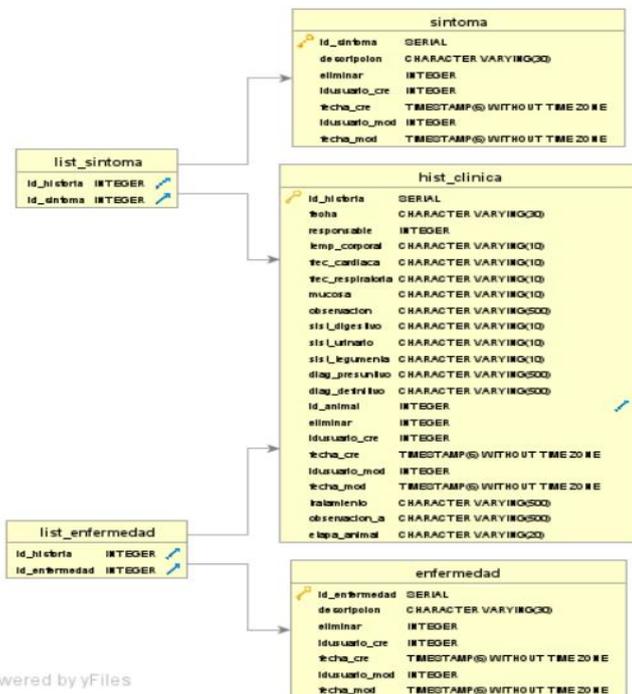


Figura 6-2: Diagrama de base de datos de la veterinaria Animal's Vet 1.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019



Powered by yFiles

Figura 7-2: Diagrama de base de datos de la veterinaria Animal's Vet
Realizado por: Carlos Grefa y Bernabé Aragón, 2019



Figura 8-2: Diagrama de base de datos de la veterinaria Animal's Vet 3.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

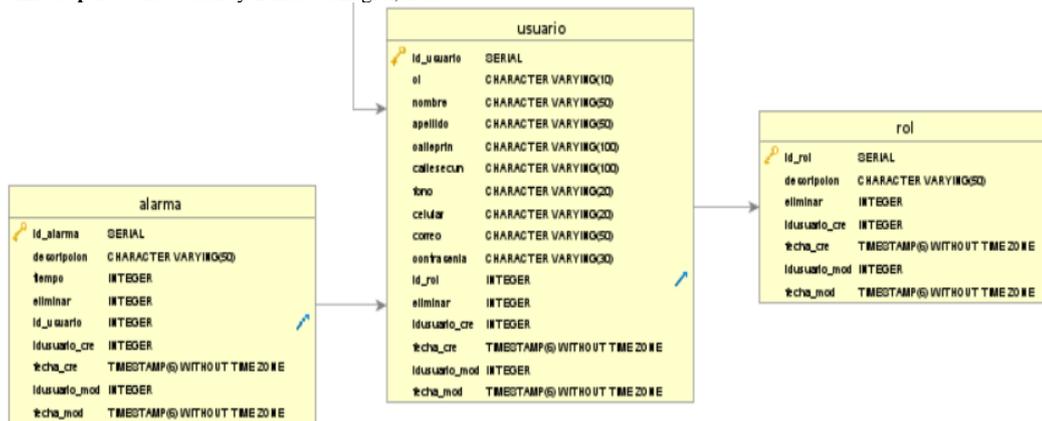


Figura 9-2: Diagrama de base de datos de la veterinaria Animal's Vet 4.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

2.9.5. Diagrama de despliegue

El diagrama de despliegue procede a modelar el funcionamiento del sistema en un período de elaboración del sistema web con la configuración del hardware que estará en cada nodo se realizó el diagrama de despliegue del sistema web “Animals Vet” **Figura 10-2**.

Se detalla en la **Figura 10-2**. Los nodos que se utilizó a la hora de realizar el sistema contando con diferentes actores y procesos que llega a tener el sistema: Consta de aplicación web el consta del nodo sistema en los que consta (Interfaz de Usuario, Lógica de negocio y Acceso a datos), el cual se almacena en el servidor de Base de datos.

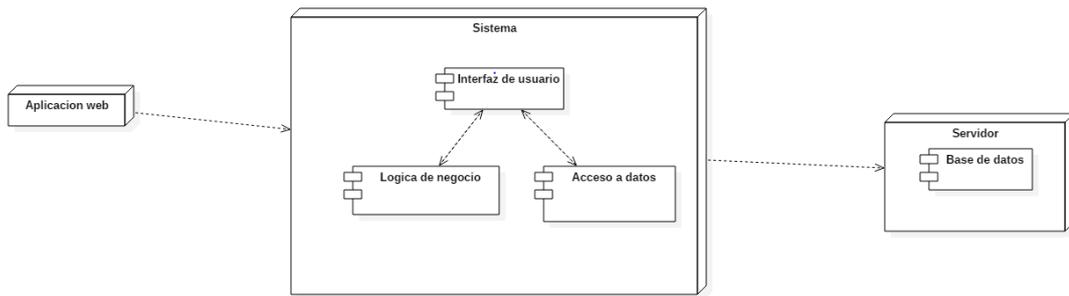


Figura 10-2: Diagrama de Despliegue
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

2.9.6. Diagrama de componentes

El diagrama de componentes representa como mostrar el sistema web para la gestión de la veterinaria “Animals Vet” se divide en sus diferentes componentes y la dependencia que existe entre cada uno.

En la **Figura 11-2**, se muestra como veterinaria tiene sus diferentes componentes entre los cuales consta con el servidor el cual conecta con la interfaz de usuario que contiene (Modelo, Vista y controlador). Del sistema de la veterinaria Animal’s Vet.

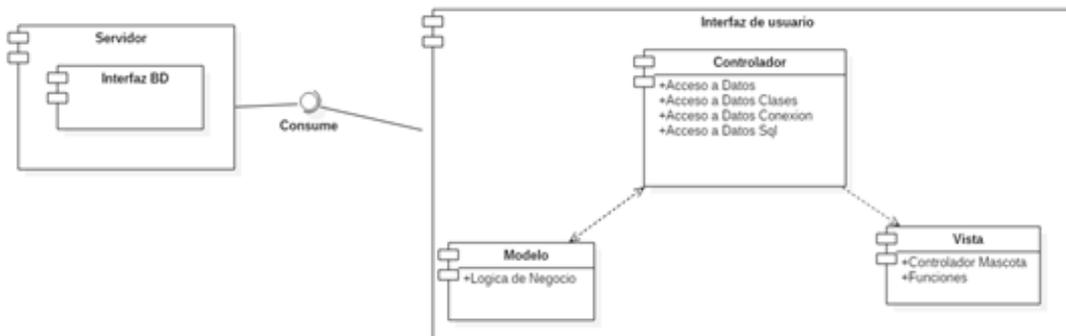


Figura 11-2: Diagrama de componentes.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

2.10. Fase de cierre

2.10.1. Burndown chart

Mediante la herramienta del Burn Down Chart se hace referencia al seguimiento y cumplimiento de las tareas realizadas durante el desarrollo del proyecto. Donde las iteraciones se representan en el eje X mientras que el esfuerzo se representa en el eje Y a 4 horas por persona, una semana 20 horas, dando un total de 160 horas por sprint. El gráfico está representado por dos líneas; la línea de color verde muestra el desarrollo real del proyecto y la línea de color azul plasma el desarrollo ideal.

Como se mencionó anteriormente durante el desarrollo fue necesario agregar algunas funcionalidades al sistema, luego de la capacitación de usuario por lo que fue relevante el incremento de tiempo que tomo realizar estos cambios; debido a esto se desarrolló un total de 70 tareas, distribuidas en 12 historias técnicas y 58 historias de usuario.

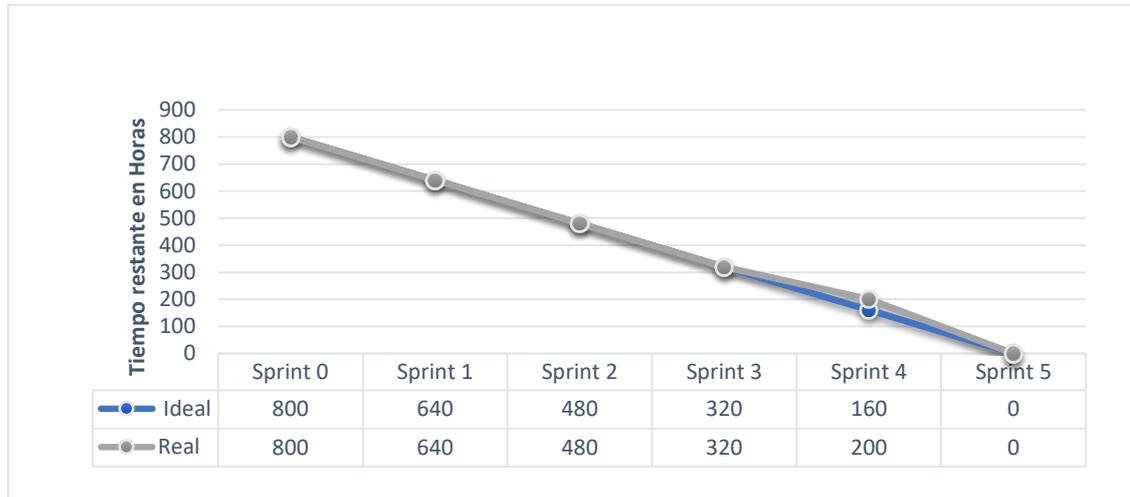


Gráfico 1-2: Burndown chart del proyecto 1

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

2.10.2. Gestión de Riesgos

Con la gestión de riesgos determinamos los posibles inconvenientes que se pueden llegar a tener durante el desarrollo del sistema, y los cuales pudimos llegar a evitar, así los objetivos no sufrieron ningún inconveniente a la hora de que se presenten alguno de los riesgos mencionados. Los riesgos que se encontraron fueron Ausencia temporal por parte de algún miembro del equipo de trabajo, el cual se solucionó de manera satisfactoria ya que durante un tiempo no se trabajó de manera correcta. El otro riesgo fue la falta de acceso al internet el cual se solucionó satisfactoriamente buscando otro diferente proveedor de internet, con el cual ya no se tuvo problemas. Los riesgos R2 Y R5 se convirtieron en problemas y las acciones planificadas llevarán a más tiempo de en realizar el sistema. Los riesgos R2 Y R5 se convirtieron en problemas y las acciones planificadas llevaron más tiempo en realizar el sistema.

Toda la información se encontrara detalladamente en el **ANEXO B**

CAPÍTULO III

3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS

En este capítulo se detallará los resultados que se obtuvieron al medir la mantenibilidad del sistema, tomando en cuenta el estándar ISO/IEC 25010.

Los conjuntos de datos tomados para medir la mantenibilidad del sistema fueron el total de líneas de código de la aplicación, los indicadores de mantenibilidad escogidos para el análisis son: Analizabilidad, Cambiabilidad, Reusabilidad-Estabilidad y Capacidad para ser probado, estos datos se tomaron a partir de cómo se encontraba el sistema antes de realizar los cambios para mejorar la mantenibilidad del mismo, es decir cuando se lo terminó de desarrollar.

3.1. Analizabilidad

3.1.1. Complejidad Ciclomática

Lógica de Negocio.

En la **Figura 1-3** se muestra un promedio de los métodos más altos de la complejidad ciclomática la cual fue medida en la lógica de negocio del sistema “Animal’s Vet”, utilizando la herramienta SimpleCodecMetrics se obtuvo estos valores.

```
Cyclomatic complexity
-----
Average cyclomatic complexity: 1.1451612903225807

Methods with the highest cyclomatic complexity:

LN::buscarProovedorNOMAPELN: 2
LN::buscarProovedorCILN: 2
LN::buscarUsuarioAPELLIDO_LN: 2
LN::buscarCirugiaIDLN: 2
LN::buscarUsuarioNOMBRE_LN: 2
```

Figura 1-3: Complejidad Ciclomática LN.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Acceso a Datos

En la **Figura 1-3** se muestra un promedio de los métodos más altos de la complejidad ciclomática la cual fue medida en el acceso a datos del sistema “Animal’s Vet”, utilizando la herramienta SimpleCodecMetrics se obtuvo estos valores.

```
Cyclomatic complexity
-----
Average cyclomatic complexity: 1.260806916426513

Methods with the highest cyclomatic complexity:

ConjuntoResultado::Fill: 4
ConjuntoResultado::Find: 4
AccesoDatos::ejecutaQuery: 3
AccesoDatos::ejecutaQuery: 3
f_insumo::insertar_insumo: 3
```

Figura 2-3: Complejidad Ciclomática AD.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Interfaz de usuario.

En la **Figura 3-3**, se muestra un promedio de los métodos más altos de la complejidad ciclomática la cual fue medida en la interfaz de usuario del sistema “Animal’s Vet”, utilizando la herramienta SimpleCodeMetrics se obtuvo estos valores.

```
-----
Cyclomatic complexity
-----
Average cyclomatic complexity: 1.5217391304347827

Methods with the highest cyclomatic complexity:

controladorAnimal::doPost: 6
controladorAnimalM::doPost: 6
razaEspecieControlador1::doGet: 2
razaEspecieControlador::doGet: 2
calcularTiempoDias::CalcularDias: 1
```

Figura 3-3: Complejidad Ciclomática IU.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

3.1.2. Densidad de comentarios

Escenario 1

En la **Figura 4-3** se muestra el total de líneas de código comentadas que existen en el sistema “Animal’s Vet” en el Escenario 1, el cual se obtuvo mediante la utilización de la herramienta Oclock.

```
github.com/AlDanial/cloc v 1.82 T=6.00 s (207.0 files/s, 56101.5 lines/s)
```

Figura 4-3: Densidad de comentarios Escenario1.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

CLOC líneas de comentarios (56101)/LOC (157775)=35.5%

Escenario 2

Densidad de comentarios:

En la **Figura 5-3** se muestra el total de líneas de código comentadas que existen en el sistema “Animal’s Vet” en el Escenario 1, el cual se obtuvo mediante la utilización de la herramienta Oclock.

```
github.com/AlDanial/cloc v 1.82 T=5.00 s (248.4 files/s, 67321.8 lines/s)
```

Figura 5-3: Densidad de comentarios Escenario2.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

CLOC líneas de comentarios (67321)/LOC (157775)=42.6%

3.2. Capacidad para ser modificado

3.2.1. Escenario 1

Acceso a datos

En la **Figura 6-3** se muestra el valor de 20 que son las líneas de código comentadas que se encuentran fuera de lugar y no son utilizadas dentro del sistema, este valor se obtuvo del acceso a datos mediante herramienta EasyPmd.

Search Results	Analyzer	Inspector	Output	Action Items ×
Description				
3 - Avoid returning from a finally block				
3 - Avoid returning from a finally block				
3 - Avoid returning from a finally block				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid unused private fields such as 'error'.				

EasyPmd - Medium: 20 TODO: 100

Figura 6-3: Cambiabilidad Acceso a Datos Escenario1

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Lógica de Negocio

En la **Figura 7-3**, se muestra el valor de 100 que son las líneas de código comentadas que se encuentran fuera de lugar y no son utilizadas dentro del sistema, este valor se obtuvo de la lógica de negocio mediante herramienta EasyPmd.

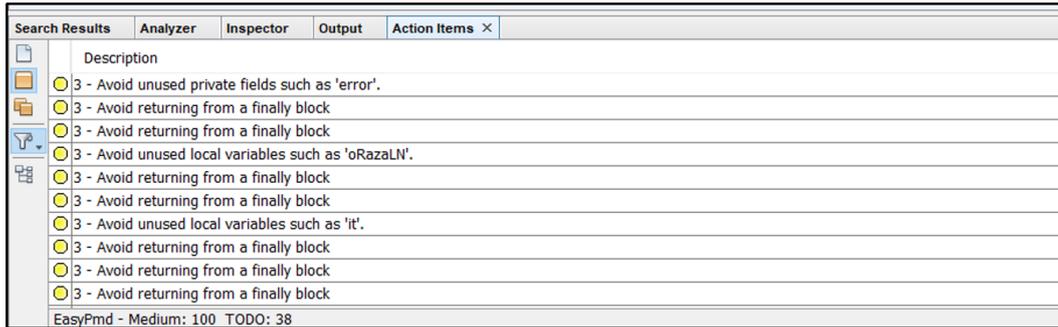


Figura 7-3: Cambiabilidad Lógica de Negocio Escenario1.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Interfaz de Usuario

En la **Figura 8-3**, se muestra el valor de 100 que son las líneas de código comentadas que se encuentran fuera de lugar y no son utilizadas dentro del sistema, este valor se obtuvo de la interfaz de usuario mediante herramienta EasyPmd.

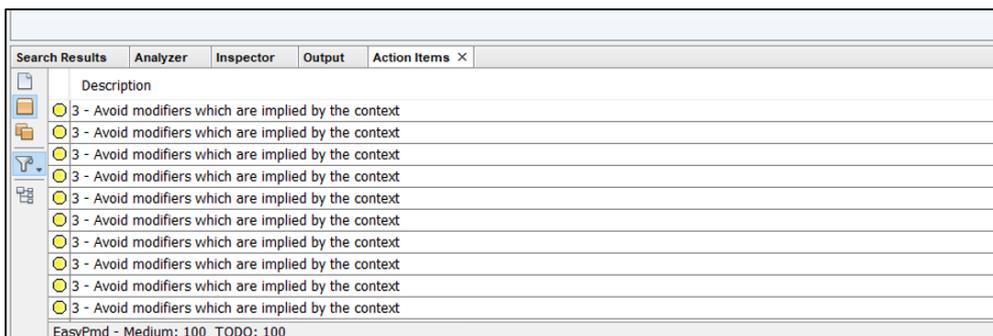


Figura 8-3: Cambiabilidad Interfaz de usuario Escenario1.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Número de defectos (220)/módulos (17)=12.94%

Es la densidad de los errores que pueden ser cambiados en el código fuente

3.2.2. Escenario 2.

Acceso a Datos.

En la **Figura 9-3**, se muestra el valor de 12 que son las líneas de código comentadas que se encuentran fuera de lugar y no son utilizadas dentro del sistema, este valor se obtuvo del acceso a datos mediante herramienta EasyPmd.

Search Results	Inspector	Output	Action Items ×
Description			File
3 - Avoid returning from a finally block			ConexionDatos.java
3 - Avoid returning from a finally block			ConexionDatos.java
3 - Avoid returning from a finally block			ConexionDatos.java
3 - Avoid unused private fields such as 'error'.			f_enfermedad.java
3 - Avoid unused private fields such as 'error'.			f_hist_clinica.java
3 - Avoid unused private fields such as 'error'.			f_insumo.java
3 - Avoid unused private fields such as 'error'.			f_list_enfermedades.java
3 - Avoid unused private fields such as 'error'.			f_list_sintomas.java
3 - Avoid unused private fields such as 'error'.			f_persona.java
3 - Avoid unused private fields such as 'error'.			f_proveedor.java

EasyPmd - Medium: 12 TODO: 100

Figura 9-3: Cambiabilidad Acceso a Datos Escenario2.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Lógica de Negocio.

En la **Figura 10-3**, se muestra el valor de 100 que son las líneas de código comentadas que se encuentran fuera de lugar y no son utilizadas dentro del sistema, este valor se obtuvo de la lógica de negocio mediante herramienta EasyPmd.

Search Results	Analyzer	Inspector	Output	Action Items ×
Description				
3 - Avoid unused private fields such as 'error'.				
3 - Avoid returning from a finally block				
3 - Avoid returning from a finally block				
3 - Avoid unused local variables such as 'oRazaLN'.				
3 - Avoid returning from a finally block				
3 - Avoid returning from a finally block				
3 - Avoid unused local variables such as 'it'.				
3 - Avoid returning from a finally block				
3 - Avoid returning from a finally block				
3 - Avoid returning from a finally block				

EasyPmd - Medium: 100 TODO: 38

Figura 10-3: Cambiabilidad Lógica de negocios Escenario2.
Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Interfaz de Usuario.

En la **Figura 11-3**, se muestra el valor de 100 que son las líneas de código comentadas que se encuentran fuera de lugar y no son utilizadas dentro del sistema, este valor se obtuvo de la interfaz de usuario mediante herramienta EasyPmd.

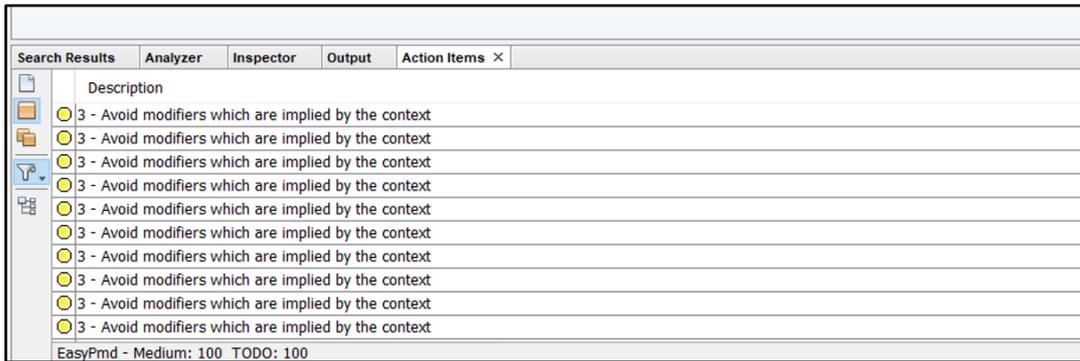


Figura 11-3: Cambiabilidad Interfaz de Usuario Escenario2.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Número de defectos (212)/módulos (17)=12.47%

Es la densidad de los errores que pueden ser cambiados en el código fuente.

3.3. Reusabilidad-Estabilidad

3.3.1. Escenario 1

Acceso a Datos

En la **Figura 12-3**, se muestran un total de 236 errores que se encontraron en los diferentes paquetes que conforma el acceso a datos, estos valores se encontraron a través de la herramienta FINDSBUGS.



Figura 12-3: Reusabilidad-Estabilidad Acceso a Datos Escenario1.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Lógica de Negocio

En la **Figura 13-3**, se muestran un total de 11 errores que se encontraron en los diferentes paquetes que conforma la lógica de negocio, estos valores se encontraron a través de la herramienta FINDSBUGS.



Figura 13-3: Reusabilidad-Estabilidad Lógica de Negocio Escenario1.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Interfaz de Usuario

En la **Figura 14-3**, se muestran un total de 22 errores que se encontraron en los diferentes paquetes que conforma la interfaz de usuario, estos valores se encontraron a través de la herramienta FINDSBUGS.

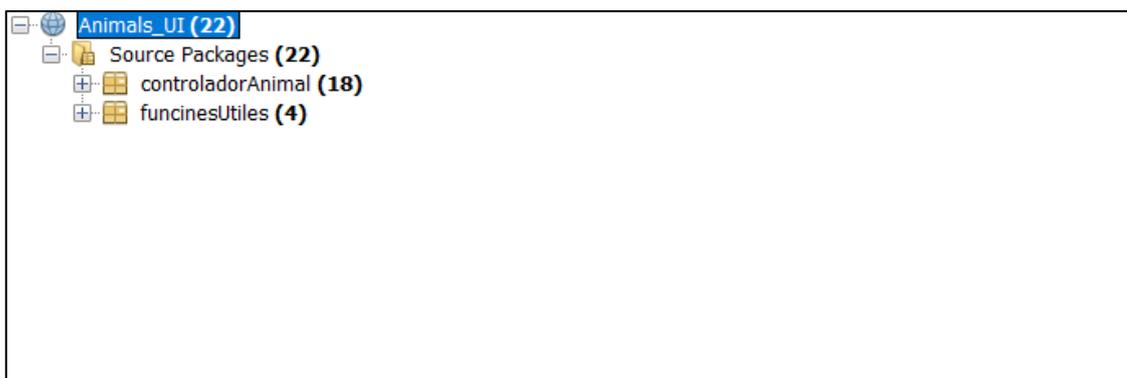


Figura 14-3: Reusabilidad-Estabilidad Interfaz de Usuario Escenario1.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Número de defectos (269)/módulos (17)=15.82%

3.3.2. Escenario 2

Acceso a Datos

En la **Figura 15-3**, se muestran un total de 161 errores que se encontraron en los diferentes paquetes que conforma el acceso a datos, estos valores se encontraron a través de la herramienta FINDSBUGS.

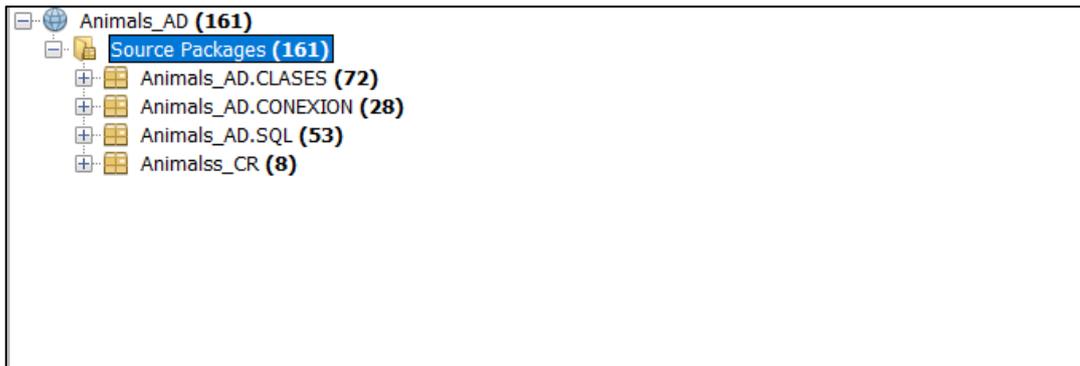


Figura 15-3: Reusabilidad-Estabilidad Acceso a Datos Escenario2.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Lógica de Negocio

En la **Figura 16-3**, se muestran un total de 6 errores que se encontraron en los diferentes paquetes que conforma el acceso a datos, estos valores se encontraron a través de la herramienta FINDSBUGS.



Figura 16-3: Reusabilidad-Estabilidad Lógica de Negocio Escenario2.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Interfaz de Usuario

En la **Figura 17-3**, se muestran un total de 4 errores que se encontraron en los diferentes paquetes que conforma la interfaz de usuario, estos valores se encontraron a través de la herramienta FINDSBUGS.



Figura 17-3: Reusabilidad-Estabilidad Interfaz de Usuario Escenario2.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Número de defectos (171)/módulos (17)=10.05%

3.4. Capacidad de ser probado

3.4.1. Escenario 1

En la **Figura 18-3**, se muestra el análisis de la prueba unitaria que se realizó en el acceso a datos para comprobar que la información ingresada sea íntegra, para lo cual se utilizó la herramienta Junit.

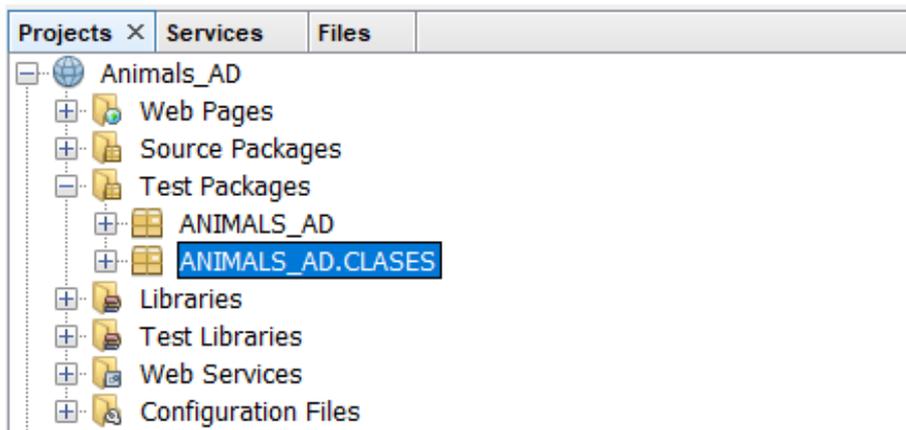


Figura 18-3: Capacidad para ser probado Escenario 1.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Función de Medición $UTD = UN/CC$ $1/1.12=1.01$

3.4.2. Escenario 2

En la **Figura 19-3**, se muestra el análisis de las pruebas unitarias que se realizó en todo el sistema para comprobar que la información ingresada sea íntegra, para lo cual se utilizó la herramienta Junit.

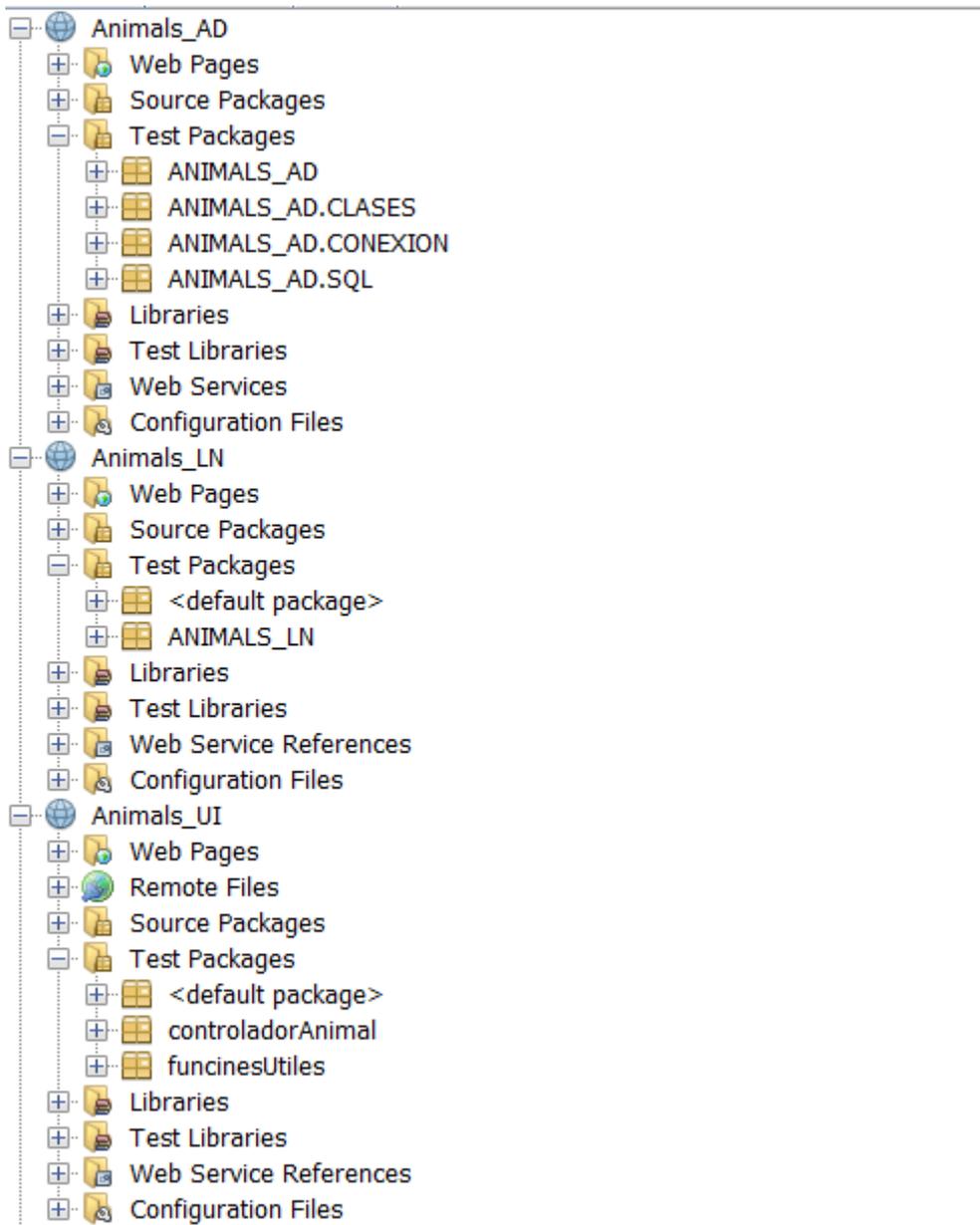


Figura 19-3: Capacidad para ser probado Escenario2.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Función de Medición $UTD = UN/CC$ 17/1.12=15.17

3.5. ANÁLISIS DE RESULTADOS

3.5.1. Analizabilidad

Complejidad Ciclomática

En la **Tabla 1-3**, se muestra una recopilación de todos los datos obtenidos en el análisis de cada uno de los componentes que se analizaron para la complejidad ciclomática.

Tabla 1-3: Resultados Complejidad Ciclomática

COMPLEJIDAD CICLOMÁTICA	
COMPONENTES	VALORES
Lógica de Negocio	1.145161
Acceso a Datos	1.260806
Interfaz de Usuario	1.521739
Promedio	1.309235

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Para realizar un mejor análisis de la complejidad ciclomática de todo el sistema, se realizó un promedio entre los 3 componentes analizados previamente para tener un valor de todo el sistema, el cual se detalla en el siguiente **Gráfico 1-3**.

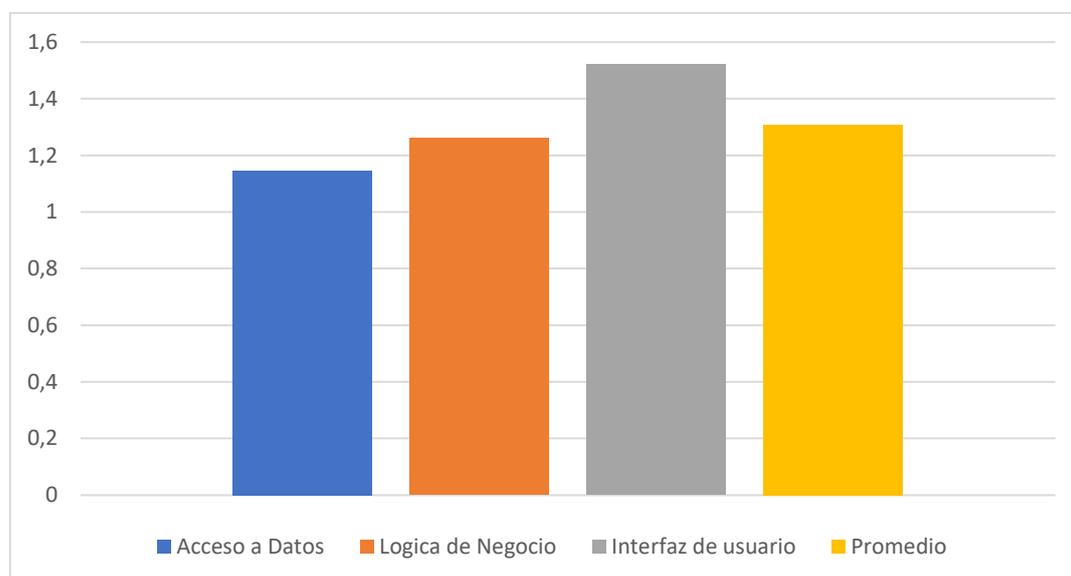


Gráfico 1-3: Complejidad Ciclomática Resultados.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Tomando en cuenta los valores obtenidos del análisis, se puede observar que la complejidad ciclomática del sistema web se encuentra en el rango de 1-10 lo que quiere decir que el software está categorizado como simple y sin mucho riesgo para entenderlo, lo cual se especifica de la **Tabla 2-2**.

DENSIDAD DE COMENTARIOS

En la **Tabla 2-3**, se muestran una recopilación de los datos obtenidos del escenario 1 que se realizó al culminar el software y del escenario 2 después de haber realizado las respectivas modificaciones en el sistema.

Tabla 2-3: Resultados Densidad de comentarios

DENSIDAD DE COMENTARIOS		
DESCRIPCIÓN	Escenario 1	Escenario 2
CLOC LÍNEA DE COMENTARIOS	56101	67321
LOC	157775	157775
PORCENTAJE DE DENSIDAD COMENTARIOS	35.5%	42.6%

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Para tener una mejor perspectiva del análisis de la densidad de comentarios de todo el sistema, se lo represento mediante un gráfico de columnas que se visualiza a continuación en el **Gráfico 2-3**.

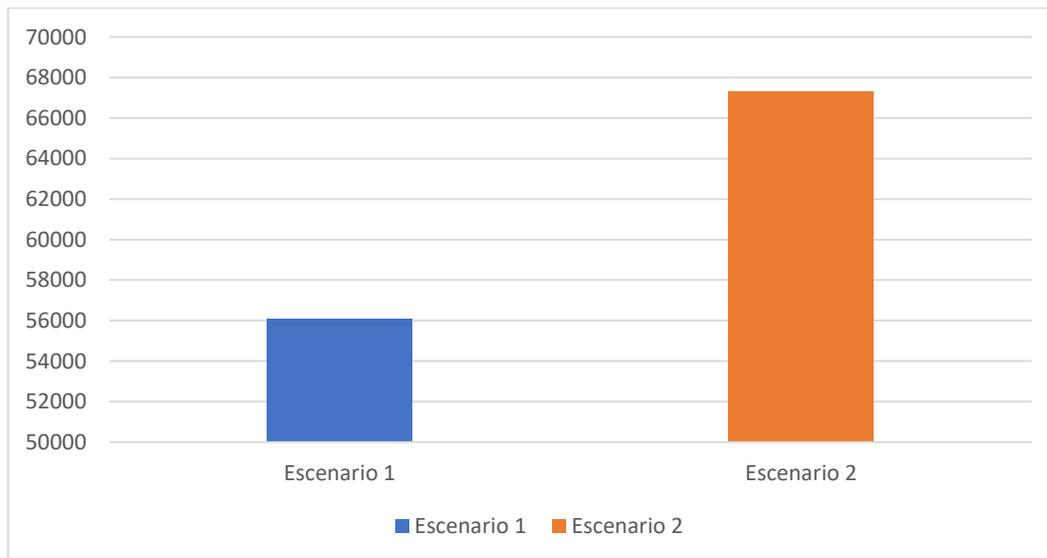


Gráfico 2-3: Densidad de Comentarios Resultados.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

De esta manera se puede realizar una comparación entre los 2 escenarios estudiados (escenario 1 y escenario 2). La densidad de comentarios en el sistema web se encuentra en un rango de **Bueno 36-70%**, el cual se especifica en la **Tabla 3-2**, por lo cual quiere decir que el sistema sería fácil de comprender si en un futuro deciden realizarle cambios.

3.5.2. Capacidad para ser cambiado

Cambiabilidad

En la **Tabla 3-3**, se muestran una recopilación de los datos obtenidos del escenario 1 que se realizó al culminar el software y del escenario 2 después de haber realizado las respectivas modificaciones en el sistema.

Tabla 3-3: Resultados Cambiabilidad

CAMBIABILIDAD		
DESCRIPCIÓN	Escenario 1	Escenario 2
NÚMERO DE DEFECTOS	220	212
MÓDULOS	17	17
PORCENTAJE DE CAMBIABILIDAD	12.94%	12.47%

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Para tener una mejor perspectiva del análisis de la cambiabilidad de todo el sistema, se lo representó mediante un gráfico de columnas que se visualiza a continuación en el **Gráfico 3-3**.

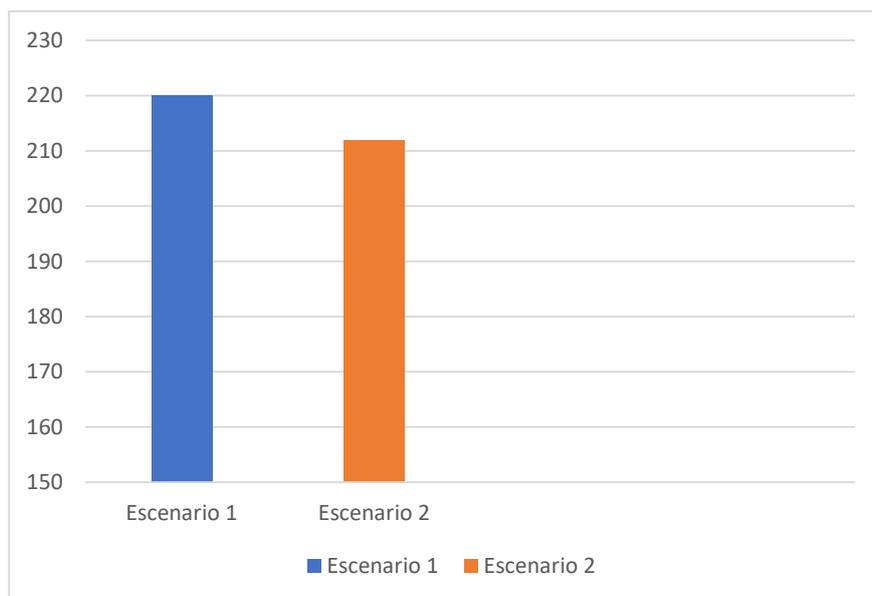


Gráfico 3-3: Cambiabilidad Resultados.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

De esta manera se puede realizar una comparación entre los 2 escenarios estudiados (escenario 1 y escenario 2). El sistema web se encuentra en un rango **Muy Bueno 1-25%**, de la cambiabilidad el cual se especifica en la **Tabla 4-2**.

3.5.3. Reusabilidad-Estabilidad

El análisis de la Reusabilidad-Estabilidad del sistema mostró resultados muy diferentes entre sí (Tabla 4-3). En la primera etapa del análisis que se realizó apenas se terminó de desarrollar el sistema, el porcentaje de estabilidad del sistema fue de 16.11% estando en el rango de **Muy Buena** (Tabla 4-3), pero de igual manera se realizó mantenibilidad del sistema en base a los resultados vistos, luego de los cambios realizados y volviendo a realizar el análisis el resultado fue de 10.05%, siguiendo la línea de resultados de **Muy Buena**. Esto da como conclusión que la capacidad del producto software de minimizar los efectos inesperados de las modificaciones es la mejor.

Tabla 4-3: Resultados Reusabilidad-Estabilidad

Reusabilidad-Estabilidad		
COMPONENTES	Escenario 1	Escenario 2
Acceso a Datos	236	161
Lógica de Negocio	11	6
Interfaz de Usuario	22	4
NÚMERO DE DEFECTOS	269	171
MÓDULOS	17	17
PORCENTAJE DE ESTABILIDAD	15.82%	10.05%

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

El siguiente **Gráfico 4-3**, se realizó en base a los datos obtenidos en **Tabla 4-3**.

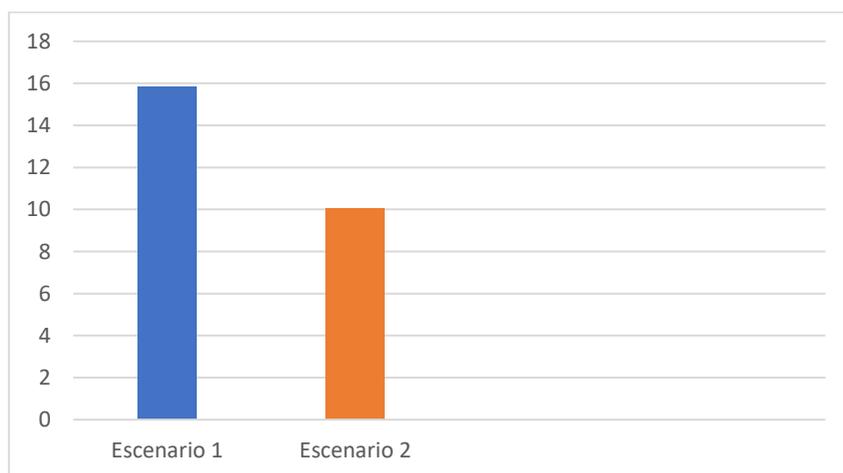


Gráfico 4-3: Reusabilidad-Estabilidad Resultados.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

3.5.4. Capacidad de ser probado

Densidad de defectos en pruebas unitarias

El análisis de la capacidad de ser probado dio resultados muy diferentes entre sí (**Tabla 5-3**).

En la primera etapa del análisis se determinó que se realizó una sola prueba unitaria, por esta razón dio un porcentaje de 1.01%, el cual está en un rango de **Muy Malo** (**Tabla 5-3**), a partir de esto se realizó mantenibilidad del sistema en base a los resultados vistos, luego de los cambios realizados y realizando más pruebas unitarias (17) el resultado fue de 15.17%, siguiendo la línea de resultados de **Muy Buena**. Esto da como conclusión que la capacidad del producto para ser probado es de suma importancia para comprobar que los datos sean iguales a los de la base de datos y no tener complicaciones a la hora de probar el sistema.

Tabla 5-3: Capacidad para ser probado

Capacidad para ser probado		
DESCRIPCIÓN	ANTES	AHORA
PRUEBAS UNITARIAS	1	17
MÓDULOS	17	17
PORCENTAJE DE CAPACIDAD PARA SER PROBADO	1.01%	15.17%

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

El siguiente **Gráfico 5-3**, se realizó en base a los datos obtenidos en **Tabla 5-3**.

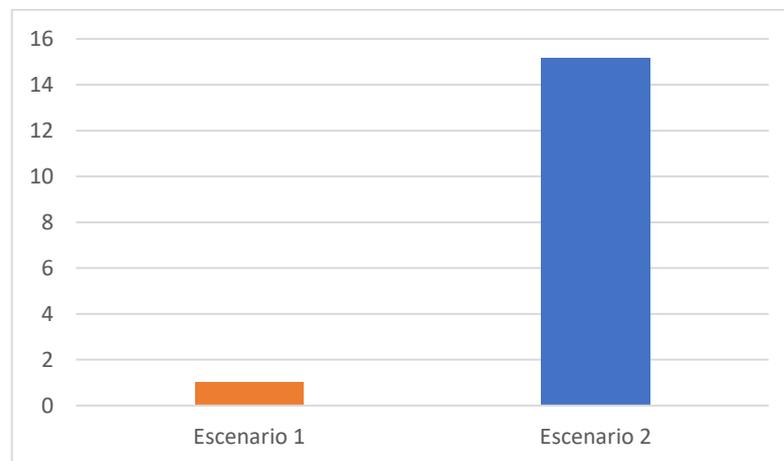


Gráfico 5-3: Capacidad para ser probado Resultados.

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Tabla 6-3: Resultados Escenario 1

Característica	Métrica	Ponderación	Porcentaje medido	Porcentaje con Ponderación
Analizabilidad	Densidad de complejidad ciclomática	15%	100%	15%
	Densidad De Comentarios	10%	2.5%	2.5%
Cambiabilidad	Densidad de defectos de la capacidad para ser cambiado	25%	87%	21.75%
Reusabilidad- Estabilidad	Densidad de defectos de estabilidad	25%	80%	20%
Capacidad de ser probado	Densidad de defectos en pruebas unitarias	25%	2%	0.5%
PROMEDIO TOTAL		100%	88.5%	59.75%

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Se puede observar en la **Tabla 6-3** el resultado del primer análisis de la mantenibilidad sin haber utilizado los indicadores de mantenibilidad, fue del 59.75%.

Tabla 7-3: Indicador de evaluación de la mantenibilidad

Calificación	Valor Cuantitativo
85%-100%	Altamente Mantenable
64%-84%	Moderadamente Mantenable
Menos de 63%	Dificultad de mantenibilidad

Fuente:(Coleman *et al.*, 1994).

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

De acuerdo con los resultados de la **Tabla 6-3**, actualmente se tiene un 59.75% de mantenibilidad, contrarrestando con los valores de la **Tabla 7-3**, el cual indica los indicadores de evaluación, el sistema estaría en un rango de menos del 63%, esto quiere decir que el sistema actualmente sería difícilmente mantenible.

Tabla 8-3: Resumen de resultados Escenario 2

Característica	Métrica	Ponderación	Porcentaje medido	Porcentaje con Ponderación
Analizabilidad	Densidad de complejidad ciclomática	15%	100%	15%
	Densidad De Comentarios	10%	5%	5%
Cambiabilidad	Densidad de defectos de la capacidad para ser cambiado	25%	90%	22.5%
Reusabilidad- Estabilidad	Densidad de defectos de estabilidad	25%	89%	22.25%
Capacidad de ser probado	Densidad de defectos en pruebas unitarias	25%	95%	23.75%
PROMEDIO TOTAL		100%	88.5%	88.5%

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

Podemos observar en la **Tabla 8-3** como resultados al ya utilizar los indicadores de mantenibilidad se obtuvo como resultado el 88.5% de nuestro sistema.

Tabla 9-3: Indicador de evaluación de la mantenibilidad

Calificación	Valor Cuantitativo
85%-100%	Altamente Mantenable
64%-84%	Moderadamente Mantenable
Menos de 63%	Dificultad de mantenibilidad

Fuente:(Coleman *et al.*, 1994).

Realizado por: Carlos Grefa y Bernabé Aragón, 2019

De acuerdo con los resultado de la **Tabla 8-3**, actualmente se tiene un 88.5% de mantenibilidad como resultado, contrarrestando con los valores de la **Tabla 9-3**, el cual indica los indicadores de evaluación, el sistema está en el rango de 85%-100%, lo cual quiere decir que es Altamente Mantenable.

3.6. Respuestas a preguntas de investigación

3.6.1. *¿Cómo se realizan actualmente los procesos relacionados la gestión de la veterinaria “Animal’s Vet”?*

La veterinaria dispone de algunos servicios que son ofrecidos al público, cómo: Cirugías para las mascotas, adopciones, ventas de insumos de mascotas. El control y registro de los procesos que realiza la clínica veterinaria, tales como: historia clínica, información del personal, información de las mascotas, registro de adopciones, alarmas, facturación, citas. Se han venido realizando de manera manual, ralentizando así el servicio hacía las personas y dificultando llevar un control preciso de los animales internados, el trámite para dar en adopción a alguna mascota, etc.

3.6.2. *¿Qué tecnologías se va a utilizar para el desarrollo del sistema web para la veterinaria “Animal’s Vet”?*

Para el desarrollo del sistema de la clínica veterinaria se utilizaron varias herramientas tecnológicas dedicadas a la producción de software, cómo el IDE de desarrollo Netbeans, el cual se utilizó para codificar el sistema, además de lenguajes de programación como Java, JavaScript, y el framework de desarrollo CSS GRID, la base de datos se la realizó en Postgresql. La arquitectura utilizada fue la de MVC (Modelo, Vista, Controlador), su principal característica es que cada componente de la arquitectura se la trata por separado, esto hace que cualquier cambio que se realice en el modelo se refleje automáticamente en las vistas. La metodología de desarrollo ágil que se utilizó fue SCRUM, la cual fomenta el surgimiento de equipo auto dirigidos y cooperativos y esto permite obtener el mejor resultado posible de un proyecto.

3.6.3. *¿Cómo son los parámetros para demostrar la mantenibilidad del sistema?*

Para determinar la mantenibilidad del sistema para la gestión de los procesos se utilizar ciertas características como analizabilidad, cambiabilidad, reusabilidad-estabilidad, capacidad de ser probado dando como resultado dio un total de 88.5% de mantenibilidad de sistema que se llegó a realizar. Se llegó a determinar que el software es altamente mantenible gracias a los parámetros anteriormente mencionados.

CONCLUSIONES

Una vez que se llegó a culminar el trabajo de titulación para la gestión de los procesos de la veterinaria “Animal’s Vet” se llegó idear las siguientes conclusiones.

- Para el desarrollo del sistema propuesto se analizó todos los procesos que realiza la Clínica veterinaria “Animal’s Vet”, en base a reuniones conjuntas y entrevistas realizadas a la Dra. Dina Paucar, gerente-propietaria. Se detalló las funcionalidades del sistema, definiendo los procesos generales como el registro de: las historias clínicas, personas (dueños de las mascotas), mascotas, adopciones, alarmas.
- En la fase de planificación y una vez determinado los procesos, se recopiló la lista de requerimientos de la Clínica Veterinaria en base a las necesidades del usuario y servicios, obteniendo un total de 70 requerimientos para el desarrollo del sistema web, los cuales se dividieron en 5 sprints, 58 historias de usuario y 12 historias técnicas, además se utilizó la metodología ágil SCRUM.
- Para realizar un análisis más detallado sobre la dimensión y de observar cómo estaba conformado nuestro sistema, se utilizó las métricas de la mantenibilidad el cual se obtuvo un porcentaje bajo del 59.75% al concluir el software, una vez utilizado las métricas ya localizamos donde se encontraban las fallas y se pudo mejorar el sistema a un porcentaje del 88.5%.
- Para evaluar la mantenibilidad se lo realizó a través de la norma ISO/25010, y se obtuvo como resultado un porcentaje del 88.5% de que es altamente mantenible el sistema de gestión de los procesos de la veterinaria “Animal’s Vet”.
- Uno de los aspectos importantes para tener una alta mantenibilidad del sistema, es la densidad de comentarios, ya que, a mayor cantidad de comentarios, más entendible será el proceso que se realizó para desarrollar el sistema.

RECOMENDACIONES

- Se recomienda leer el manual de usuario para instruirse y ver cómo se utiliza el sistema web correctamente, de tal modo que se haga un buen uso del software así mejora el desempeño de la veterinaria.
- Utilizar una metodología ágil como SCRUM es óptimo porque permite tener una integración entre el desarrollador y el cliente, de esta manera se pueden llegar a cumplir de manera más eficiente los objetivos planteados inicialmente.
- Para el correcto funcionamiento del sistema, es recomendable utilizar navegadores actualizados, debido a que, en las versiones antiguas de algunos navegadores como Internet Explorer, se necesitaría utilizar un Polyfills para que sean compatibles con el CSS GRID.
- Se recomienda utilizar otros programas que miden la mantenibilidad los cuales benefician al sistema y hacen que el software sea de mejor calidad.
- Se podría realizar un sistema de cobros de manera online, firma electrónica y venta de productos y que se entreguen a domicilio.

BIBLIOGRAFÍA

AcensTechnologies *'Framework para el desarrollo ágil de aplicaciones', 1, pp. 1–10.* [En línea] (Consulta: 5 November 2019). Disponible en: <https://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf>

DOMINGUEZ, A *Cuadro Comparativo Sistemas Gestores De Bases De Datos.* [En línea] (Consulta: 9 July 2019). Disponible en: <https://es.scribd.com/doc/137485506/Cuadro-Comparativo-Sistemas-Gestores-De-Bases-De-Datos#scribd>.

AYOZE, A. *CURSO DE PROGRAMACION WEB : javascript, ajax y jquery.* [En línea] (Consulta: 25 June 2019). CREATESPAC INDEPENDENT P. Disponible en: <https://www.amazon.es/Curso-Programación-Web-JavaScript-jQuery/dp/1542787408>.

CALABRESE, J. et al. *'Asistente para la evaluación de características de calidad de producto de software propuestas por ISO/IEC 25010 basado en métricas definidas usando el enfoque GQM', in XXIII Congreso Argentino de Ciencias de la Computación, pp. 660–671.* [En línea] (Consulta: 8 November 2019). Disponible en: <http://hdl.handle.net/10915/63778>.

CAMACHO, A *'HERRAMIENTA PARA EL ANÁLISIS DE REQUERIMIENTOS DENTRO DE LA PEQUEÑA EMPRESA DESARROLLADORA DE SOFTWARE EN BOGOTÁ', Известия Высших Учебных Заведений. Радиофизика, 48(9), pp. 1–16.* [En línea] (Consulta: 1 July 2019). Disponible en: <https://javeriana.edu.co/biblos/tesis/ingenieria/Tesis189.pdf>

CASTEJÓN, Ga, J. S. *'Arquitectura y Diseño De Sistemas Web Modernos',* [En línea] *Revista de Ingeniería Informática del CIIRM, pp. 1–6.*

COLEMAN, D. *'Using Metrics to Evaluate Software System Maintainability',* [En línea] *Computer, 27(8), pp. 44–49. doi: 10.1109/2.303623.*

VALLEJO, D. et al. *SISTEMA DE GESTIÓN PARA LA CLÍNICA VETERINARIA DE LA UNIVERSIDAD CENTRAL DEL ECUADOR UNIVERSIDAD CENTRAL DEL ECUADOR FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA CARRERA DE INGENIERÍA INFORMÁTICA* Trabajo de graduación previo a la obtención del Tí. [En línea] (Consulta: 18 July 2019). Disponible en: <http://www.dspace.uce.edu.ec/bitstream/25000/478/1/T-UCE-0011-27.pdf>

DEEMER, P. et al. *'Una introducción básica a la teoría y práctica de Scrum'.* [En línea] (Consulta: 24 August 2019). Disponible en: http://scrumprimer.org/primers/es_scrumprimer20.pdf.

PALMA, F J. E. *Historia del Diseño Web*. [En línea] (Consulta: 7 November 2019). Disponible en: https://www.dweb3d.com/blog/historia-del-diseno-web/?fbclid=IwAR37GJJgiVxy4gWhk8dUYIpPuO_npX494_sShaitRDHB5GC1bwDhq7EV79Q.

GIBERT, M. AND PÉREZ, O. *Fundamentos de bases de datos en PostgreSQL*. [En línea] (Consulta: 24 September 2019). Disponible en: https://s3.amazonaws.com/academia.edu.documents/35811174/P06_M2109_02152.pdf?response-content-disposition=inline%3Bfilename%3DBases_de_datos_en_PostgreSQL.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20190925%2Fus-east-1

GONZÁLEZ, M. *Herramienta de Desarrollo Netbeans*, p. 5. [En línea] (Consulta: 1 November 2019) Disponible en: http://www.consultorjava.com/wp/wp-content/uploads/2015/09/herramienta_desarrollo_netbeans.pdf.

GUTIÉRREZ, J. J. *¿Qué es un framework web?*, Universidad de Sevilla, pp. 1–4. [En línea] (Consulta: 5 November 2019) Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

LOPEZ, H et al. *3c tecnología., 3c Tecnología: glosas de innovación aplicadas a la pyme, ISSN-e 2254-4143, Vol. 6, N°. 2, 2017, págs. 17-31.* 3ciencias. [En línea] (Consulta: 8 September 2019) Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=6034898>.

HOUSE, C. *A Complete Guide to Grid | CSS-Tricks*, *CSS-Tricks*. [En línea] (Consulta: 9 July 2019). Disponible en: <https://css-tricks.com/snippets/css/complete-guide-grid/>

IRRAZÁBAL, E. *Medición De La Calidad : Resultados En Una Gran Empresa*, *XXI Congreso Argentino de Ciencias de la Computación*, 1, p. 11. [En línea] (Consulta: 9 September 2019) Disponible en: <https://pdfs.semanticscholar.org/77cf/a00cc74150cf3376998050e3dacb1f1e23a4.pdf>

ISO Norma ISO 25000, *Norma ISO25000: Calidad del Producto de Software*.

Iso25000.com 'ISO 25010', iso25000.com.

ARAUJO, J *¿Por qué CSS Grid es la mejor opción para crear diseños?.* [En línea] (Consulta: 9 July 2019). Disponible en: <https://platzi.com/tutoriales/1229-css-grid-layout/2071-por-que-css-grid-es-mejor-que-bootstrap-para-crear-disenos/>

LÓPEZ ECHEVERRY, A., VALENCIA AYALA, L. AND CABRERA, C. et al. *Introducción a la calidad de software.*, 'UNIVERSIDAD DE GUAYAQUIL ' *Scientia et Technica*, 2(39), pp. 326–331. [En línea] doi: 10.22517/23447214.3241.

MARTÍNEZ, J., HIGUERA, M. AND AGUILAR, E. ‘Enfoque metodológico para el diseño de interfaces durante el ciclo de vida del desarrollo de software’, *Gerencia Tecnológica Informática*, 12(34), pp. 59–74. [En línea] (Consulta: 7 November 2019). Disponible en: <https://revistas.uis.edu.co/index.php/revistagti/article/view/3846/4199>

MERA, J. And CUARAN, S. ‘Análisis sistemático de información de la Norma ISO 25010 como base para la implementación en un laboratorio de Testing de software en la Universidad Cooperativa de Colombia Sede Popayán’, *Memorias de Congresos UTP*, pp. 149–154. [En línea] (Consulta: 3 July 2019) Disponible en: <https://revistas.utp.ac.pa/index.php/memoutp/article/view/1483>.

MOZILLA *CSS básico - Aprende sobre desarrollo web | MDN*. [En línea] (Consulta: 5 November 2019). Disponible en: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics

NORMAS ISO 25000 *Información JAVA*.

BLANCARTE, O *Patrón de diseño Modelo Vista Controlador (MVC) - Oscar Blancarte - Software Architecture*. [En línea] (Consulta: 4 July 2019) Disponible en: <https://www.oscarblancarteblog.com/2014/07/21/patron-de-diseno-modelo-vista-controlador-mvc/>.

PAYARA SERVICES *Downloads – Payara Services Ltd*. [En línea] (Consulta: 25 June 2019) Disponible en: <https://www.payara.fish/documentation/>.

Qué es una hoja de estilo CSS. [En línea] (Consulta: 5 November 2019) Disponible en: <https://culturacion.com/que-es-una-hoja-de-estilo-css/>.

RIVERA, A. *Capítulo II. Arquitectura del Software*. [En línea] (Consulta: 4 July 2019) Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_1_a/capitulo2.pdf.

SCHWABER, K. and SUTHERLAND, J. ‘*La Guía definitiva de Scrum: Las Reglas de Juego*’, *Scrum.org*. [En línea] (Consulta: 5 September 2019) Disponible en: <https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf%0AScrum.org>.

SCRUM MANAGER *Scrum Manager*. doi: 1607208414838.

SDK, J. and NETBEANS, I. D. E. *Introducción básica a Netbeans para desarrollo Java*. [En línea] Disponible en: <http://marcial123.cubava.cu/files/2016/06/introduccion-basica-a-netbeans-para-desarrollo-java.pdf>.

SINGER, J ‘International Conference on Software Maintenance’, *IEEE International Conference on Software Maintenance, ICSM*. IEEE Computer Society Press, p. 400. [En línea]

(Consulta: 9 September 2019) Disponible en: <https://dl.acm.org/citation.cfm?id=853305>.

UNIWEBSIDAD *Capítulo 5. Frameworks (CSS avanzado)*. [Blog] (Consulta: 5 November 2019) Disponible en: <https://uniwebsidad.com/libros/css-avanzado/capitulo-5>.

VALENCIANO, J. *et al. AUDITORÍA MANTENIBILIDAD APLICACIONES SEGÚN LA ISO/IEC 25000 Autorización de difusión y utilización*. [En línea] (Consulta: 19 September 2019) Disponible en: [https://eprints.ucm.es/37485/1/AUDITORÍA MANTENIBILIDAD APLICACIONES SEGÚN LA ISO_IEC 25000.pdf](https://eprints.ucm.es/37485/1/AUDITORÍA_MANTENIBILIDAD_APLICACIONES_SEGÚN_LA_ISO_IEC_25000.pdf).

VIDAL, J. A. ‘El lenguaje de programación Awk/Gawk’, p. 294. [Blog] (Consulta: 4 July 2019). Disponible en: https://www.academia.edu/11413034/El_lenguaje_de_Programación_Java

WLADIMIR, W. and SIGUENCIA, L. *DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA ADMINISTRACIÓN DE LA CLÍNICA VETERINARIA DE LA FUNDACIÓN PROTECCIÓN ANIMAL ECUADOR*. ESCUELA POLITÉCNICA NACIONAL FACULTAD DE INGENIERÍA DE SISTEMAS [En línea] (Consulta: 18 July 2019) Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/5329/1/CD-4579.pdf>.

FERNÁNDEZ, Y ‘Patrón Modelo-Vista-Controlador.’, 11(1), pp. 47–57.

YUSOF, F., KANE, I. L. and YUSOP, Z. *Jurnal Teknologi Science and Engineering, Elektronische Ressource., Jurnal Teknologi*. Penerbit UTM Press. [En línea] (Consulta: 3 July 2019) Disponible en: <https://jurnalteknologi.utm.my/index.php/jurnalteknologi/article/view/6107/4050>

{Bibliography