

**“DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN DE LOS  
PROCESOS REALIZADOS POR LOS COMITÉS DE LA FUNDACIÓN  
VOLUNTARIOS ANIMALISTAS POPAYÁN”**



**DARLY VANNESSA FERNÁNDEZ LEÓN**

**ANGELA MARÍA MUÑOZ GARCÍA**

**CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POPAYÁN, SEPTIEMBRE 2016**

**“DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN DE LOS  
PROCESOS REALIZADOS POR LOS COMITÉS DE LA FUNDACIÓN  
VOLUNTARIOS ANIMALISTAS POPAYÁN”**



**CORPORACION UNIVERSITARIA  
AUTONOMA  
DEL CAUCA**

**DARLY VANNESSA FERNÁNDEZ LEÓN**

**ANGELA MARÍA MUÑOZ GARCÍA**

**Trabajo de grado para obtener el título de ingenieras de sistemas  
informáticos**

**Director**

**Esp. Julián Darío Bermúdez Trujillo**

**CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POPAYÁN, SEPTIEMBRE 2016**

## **NOTA DE ACEPTACIÓN**

Popayán 2016

El director y los jurados del trabajo de grado en modalidad trabajo de investigación denominado “DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN DE LOS PROCESOS REALIZADOS POR LOS COMITÉS DE LA FUNDACIÓN VOLUNTARIOS ANIMALISTAS POPAYÁN”, realizado por los estudiantes Angela María Muñoz García y Darly Vannessa Fernández León una vez revisado el informe final y aprobado la sustentación del mismo autorizan para que se realicen los trámites concernientes para optar por el título profesional de Ingeniería de Sistemas Informáticos.

---

Henry Córdoba Idrobo  
Jurado Evaluador

---

Gabriel Angel Osorio Hoyos  
Jurado Evaluador

---

Julian Dario Bermudez Trujillo  
Director

## **DEDICATORIA**

Primordialmente dedico este trabajo de grado a Dios y a mis padres, quienes me han dado fuerza para cada día superarme y afrontar las dificultades que se me presentan en la vida.

A mi hermano, que además de un apoyo total, ha sido un ejemplo a seguir como persona.

Compañeros y docentes de la Universidad, que gracias a sus aportes y conocimientos brindados día a día en el transcurso de estos cinco años permitieron el cumplimiento de una meta más en mi vida.

También darle gracias a nuestro director, quien con gran dedicación y tiempo nos guio durante todo el proyecto.

Darly Fernández León

## TABLA DE CONTENIDO

RESUMEN .....	1
ABSTRACT .....	2
INTRODUCCIÓN .....	3
1. PLANTEAMIENTO DEL PROBLEMA.....	5
1.1 PLANTEAMIENTO DEL PROBLEMA .....	5
1.1.1 Pregunta de investigación. ....	6
1.2 JUSTIFICACIÓN .....	7
1.2.1 Justificación técnica.....	7
1.2.2 Justificación Funcional.....	7
1.2.3 Justificación económica-financiera .....	8
1.2.4 Justificación social. ....	8
1.3 OBJETIVOS .....	9
1.3.1 Objetivo general.....	9
1.3.2 Objetivos específicos.....	9
2. MARCO TEÓRICO.....	11
2.1 ANTECEDENTES .....	11
2.1.1 Antecedentes.....	12
2.2 BASES TEÓRICAS .....	14
2.2.1 Sistema de información .....	14
2.2.2 Plataforma Web .....	14
2.2.3 Framework.....	15
2.2.4 Yii.....	15
2.2.5 Metodologías ágiles.....	16
2.2.6 Lean Canvas.....	18
2.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE .....	19
2.3.1 Extreme Programming.....	19

2.3.2	Patrón MVC .....	25
2.4	FRAMEWORK DE GESTIÓN DEL PROYECTO .....	26
2.4.1	Valores ágiles .....	26
2.4.2	Manifiesto ágil.....	26
2.4.3	Scrum .....	28
2.5	MARCO CONTEXTUAL.....	35
2.6	HERRAMIENTAS DE APOYO .....	36
2.6.1	Adobe Kuller .....	36
2.6.2	Bitbucket.....	36
2.6.3	Boostrap .....	37
2.6.4	Higcharts .....	37
2.6.5	Mysql Workbench .....	38
2.6.6	OpenShift.....	39
2.6.7	PhpMailer .....	39
2.6.8	PhpMyAdmin .....	39
2.6.9	Planning Poker .....	39
2.6.10	Servidor Apache.....	40
2.6.11	Sublime .....	41
2.6.12	StartUml .....	41
2.6.13	Trello .....	42
2.6.14	XAMPP.....	43
2.6.15	WakaTime .....	43
2.7	GLOSARIO .....	44
3.	METODOLOGÍA.....	48
3.1	IMPLEMENTACIÓN DE SCRUM.....	48
3.1.1	Planificación Sprint 0 .....	48
3.1.2	Planificación Sprint 1 .....	51
3.1.3	Planificación Sprint 2 .....	52
3.1.4	Planificación Sprint 3 .....	52
3.1.5	Planeación Daily .....	53

3.1.6 Sprint Review.....	53
3.2 IMPLEMENTACIÓN EXTREME PROGRAMMING.....	55
3.2.1 Planificación.....	57
3.2.2 Diseño.....	58
3.2.3 Desarrollo .....	58
3.2.4 Pruebas .....	60
4. INGENIERÍA DEL PROYECTO.....	61
4.1 IMPLEMENTACIÓN SPRINT 0.....	61
4.1.1 Diagrama bases de datos .....	61
4.1.2 Diagrama de Clases.....	63
4.1.3 Arquitectura del Sistema .....	65
4.1.4 Implementación Inception.....	66
4.2 IMPLEMENTACIÓN SPRINT 1.....	72
4.2.1 Planeación .....	73
4.2.2 Diseño .....	94
4.2.3 Desarrollo .....	95
4.2.4 Pruebas .....	96
4.2.5 Sprint Review.....	97
4.2.6 Retrospectiva.....	97
4.3 IMPLEMENTACIÓN SPRINT 2.....	98
4.3.1 Planeación .....	99
4.2.2 Diseño .....	129
4.2.3 Desarrollo .....	130
4.2.4 Pruebas .....	131
4.2.5 Sprint Review.....	135
4.2.6 Retrospectiva.....	135
4.4 IMPLEMENTACIÓN SPRINT 3.....	136
4.4.1 Planeación .....	137
4.4.2 Diseño .....	152
4.4.3 Desarrollo .....	153

4.4.4	Pruebas .....	154
4.4.5	Sprint Review.....	155
4.4.6	Retrospectiva.....	155
5.	RESULTADOS .....	157
6.	CONCLUSIONES.....	161
6.1	CONCLUSIONES .....	161
6.2	TRABAJO FUTURO.....	162
7.	BIBLIOGRAFÍA.....	163
	ANEXO .....	169
	INTERFACES GRÁFICAS DE LA APLICACIÓN WEB.....	169
	REUNIONES CON LA FUNDACIÓN “VAP” .....	195
	CARTA DE ENTREGA PLATAFORMA WEB .....	197



## LISTA DE FIGURAS

Figura 1. Metodologías ágiles .....	16
Figura 2. Prácticas extreme programming .....	19
Figura 3. Entorno de trabajo Scrum .....	28
Figura 4. Pilares Scrum .....	29
Figura 5. Scrum team .....	30
Figura 6. Artefactos Scrum .....	31
Figura 7. Eventos de Scrum .....	32
Figura 8. AdobeKuler .....	36
Figura 9. Bitbucket .....	37
Figura 10. MySQL Workbench .....	38
Figura 11. Planning Poker .....	40
Figura 12. Sublime Text2 .....	41
Figura 13. StarUML .....	42
Figura 14. Tablero Trello .....	42
Figura 15. Waka Time [Fuente propia] .....	43
Figura 16. Product backlog [Fuente propia] .....	51
Figura 17. Carta sprint review [Fuente propia] .....	54
Figura 18. Tablero Kanban [Fuente propia] .....	58
Figura 19. Diagrama de bases de datos [Fuente propia] .....	62
Figura 20. Diagrama de clases [Fuente propia] .....	64
Figura 21. Arquitectura cliente servidor .....	65
Figura 22. Capas de la aplicación .....	66
Figura 23. Diseña tu caja [Fuente propia] .....	69
Figura 24. Interfaz crear usuario [Fuente propia] .....	94
Figura 25. Prueba funcional interfaz crear usuario [Fuente propia] .....	96
Figura 26. Prueba funcional registrar usuario [Fuente propia] .....	97
Figura 27. Interfaz gráfica crear proceso [Fuente propia] .....	129
Figura 28. Interfaz gráfica búsqueda de mascotas [Fuente propia] .....	132
Figura 29. Interfaz resultado búsqueda de mascotas [Fuente propia] .....	132
Figura 30. Interfaz gráfica Historia clínica de la mascota [Fuente propia] .....	133
Figura 31. Interfaz listado de procesos por mascota [Fuente propia] .....	133
Figura 32. Interfaz gráfica crear proceso [Fuente propia] .....	134
Figura 33. Interfaz gráfica cuestionario [Fuente propia] .....	134
Figura 34. Prueba de funcionalidad crear proceso [Fuente propia] .....	135
Figura 35. Indicador resultado por evento [Fuente propia] .....	152
Figura 36. Interfaz gráfica perfil de usuario [Fuente propia] .....	154

Figura 37. Interfaz gráfica indicador donación [Fuente propia] ..... 155

## LISTA DE TABLAS

Tabla 1. Antecedentes .....	12
Tabla 2. Product owner (dueño del producto) .....	49
Tabla 3. Team development (equipo) desarrolladora e investigadora.....	49
Tabla 4. Team development (equipo) desarrolladora e investigadora.....	49
Tabla 5. Scrum Master (Director del proyecto).....	49
Tabla 6. Lean canvas del Proyecto VAP .....	56
Tabla 7. Cronograma .....	70
Tabla 8. Retrospectiva Sprint 0 .....	71
Tabla 9. Historia de usuario: ingreso al sistema.....	73
Tabla 10. Task card inicio de sesión .....	74
Tabla 11. Task card conexión servidor.....	74
Tabla 12. Task card validar campos.....	75
Tabla 13. Task card recuperación de contraseña .....	75
Tabla 14. Tarjeta de colaboración Usuario.....	76
Tabla 15. Historia de usuario registro e ingreso .....	76
Tabla 16. Task card configuración framework.....	77
Tabla 17. Task card crear atributos en la entidad usuario.....	77
Tabla 18. Task card validar rol de usuario .....	78
Tabla 19. Task card log sesión de usuarios .....	79
Tabla 20. Task card código de validación enviado al email .....	79
Tabla 21. Tarjeta de colaboración HU2 registro e ingreso .....	80
Tabla 22. Historia de usuario registro de usuario .....	80
Tabla 23. Task card crear interfaces graficas CRUD .....	81
Tabla 24. Task card conexión servidor.....	81
Tabla 25. Task card validación de campos .....	82
Tabla 26. Task card selección búsqueda .....	82
Tabla 27. Task card selección búsqueda por rol .....	83
Tabla 28. Tarjeta de colaboración HU5 registro de usuario .....	83
Tabla 29. Historia de usuario guardar usuario.....	84
Tabla 30. Task card validar petición.....	85
Tabla 31. Task card enviar notificación .....	85
Tabla 32. Task card enviar información de usuario.....	86
Tabla 33. Task card listar usuarios por rol .....	86
Tabla 34. Tarjeta de colaboración HU6 guardar usuario .....	87
Tabla 35. Historia de usuario registro de animales .....	87
Tabla 36. Task card crear interfaces gráficas CRUD .....	88
Tabla 37. Task card conexión al servidor .....	88

Tabla 38. Task card validación de campos .....	89
Tabla 39. Task card seleccionar búsqueda.....	89
Tabla 40. Task card visualizar galería de mascotas.....	90
Tabla 41. Task card seleccionar foto de galería.....	90
Tabla 42. Tarjeta de colaboración HU3 registro de animales.....	91
Tabla 43. Historia de usuario guardar animales .....	91
Tabla 44. Task card validar petición CRUD.....	92
Tabla 45. Task card generar galería de animales no adoptados.....	93
Tabla 46. Task card listar animales con el mismo nombre.....	93
Tabla 47. Tarjeta de colaboración HU4 guardar animales .....	94
Tabla 48. Evento programado registro de usuario .....	95
Tabla 49. Prueba funcional registro de usuario .....	96
Tabla 50. Retrospectiva Sprint 1 .....	98
Tabla 51. Historia de usuario registro de consulta .....	99
Tabla 52. Task card interfaz gráfica crear consulta.....	100
Tabla 53. Task card conexión servidor.....	101
Tabla 54. Task card validar campos.....	101
Tabla 55. Task card crear PDF de consultas por mascota.....	102
Tabla 56. Tarjeta colaboración HU19 Registro de consulta .....	102
Tabla 57. Historia de usuario: guardar consulta .....	102
Tabla 58. Task card validar peticiones CRUD en el servidor .....	103
Tabla 59. Task card generar PDF de consultas .....	104
Tabla 60. Tarjeta colaboración HU22 Guardar Consulta.....	105
Tabla 61. Historia de usuario: registro de vacunación.....	105
Tabla 62. Task card Interfaz gráfica vacunación .....	106
Tabla 63. Task card conexión servidor.....	107
Tabla 64. Task card validar campos.....	107
Tabla 65. Tarjeta colaboración HU17 registro de vacunación .....	108
Tabla 66. Historia de usuario: guardar vacunación .....	108
Tabla 67. Task card validación de peticiones CRUD .....	109
Tabla 68. Tarjeta colaboración HU20 guardar vacunas .....	109
Tabla 69. Historia de usuario: registro de esterilización .....	110
Tabla 70. Task card interfaz gráfica crear esterilización .....	110
Tabla 71. Task card conexión servidor.....	111
Tabla 72. Task card validar campos.....	111
Tabla 73. Tarjeta colaboración HU18 Registro de esterilización .....	112
Tabla 74. Historia de usuario: guardar esterilización.....	112
Tabla 75. Task card validar peticiones CRUD.....	113
Tabla 76. Tarjeta colaboración HU21 Guardar esterilización. ....	114

Tabla 77. Historia de usuario: Registro de proceso.....	114
Tabla 78. Task card CRUD para gestionar procesos .....	115
Tabla 79. Task card conexión servidor.....	116
Tabla 80. Task card validación de campos de acuerdo al tipo de dato .....	116
Tabla 81. Task card ingresar respuestas al cuestionario de adopción.....	117
Tabla 82. Task card descargar formato de adopción .....	117
Tabla 83. Task card ingresar respuestas al cuestionario de seguimiento .....	118
Tabla 84. Task card descargar formato de seguimiento .....	118
Tabla 85. Task card ingresar respuesta de postulación .....	119
Tabla 86. Task Card Gestionar contacto con la fundación.....	119
Tabla 87. Tarjeta de colaboración HU15 Proceso.....	120
Tabla 88. Historia de usuario: guardar proceso.....	120
Tabla 89. Task card validar peticiones CRUD.....	121
Tabla 90. Task card interfaz gráfica del cuestionario .....	122
Tabla 91. Task card capturar y guardar respuestas .....	122
Tabla 92. Task card generar formato de procesos.....	123
Tabla 93. Task card enviar notificación al email.....	123
Tabla 94. Tarjeta de colaboración HU16 guardar proceso.....	124
Tabla 95. Historia de usuario: registro de eventos .....	124
Tabla 96. Task card interfaz registro de eventos.....	125
Tabla 97. Task card conexión servidor.....	126
Tabla 98. Task card validar campos.....	126
Tabla 99. Task card interfaz gráfica calendario de eventos .....	127
Tabla 100. Task card información del evento.....	127
Tabla 101. Tarjeta de colaboración HU7 registro de eventos.....	128
Tabla 102. Evento Programado Registro de Proceso .....	130
Tabla 103. Pruebas de funcionalidad Proceso .....	131
Tabla 104. Retrospectiva Sprint 2 .....	135
Tabla 105. Historia de usuario guardar evento.....	137
Tabla 106. Task card validar peticiones CRUD.....	138
Tabla 107. Task card enviar información al calendario .....	138
Tabla 108. Task card Enviar información detallada sobre el evento .....	139
Tabla 109. Tarjeta colaboración HU8 Guardar eventos .....	139
Tabla 110. Historia de usuario: registro de productos .....	139
Tabla 111. Task card interfaz gráfica para crear productos .....	140
Tabla 112. Task card conexión servidor.....	141
Tabla 113. Task card validar campos.....	141
Tabla 114. Tarjeta colaboración HU9 Registro de productos.....	142
Tabla 115. Historia de usuario: guardar productos.....	142

Tabla 116. Task card validar peticiones CRUD.....	143
Tabla 117. Tarjeta colaboración HU10 Guardar Productos.....	143
Tabla 118. Historia de usuario: registro de productos Zoosanitarios.....	144
Tabla 119. Task card crear productos donados y zoosanitarios .....	145
Tabla 120. Task card conexión servidor.....	145
Tabla 121. Task card validar campos.....	146
Tabla 122. Tarjeta colaboración HU11 Registro de productos zoosanitarios .....	146
Tabla 123. Historia de usuario: guardar productos Zoosanitarios .....	146
Tabla 124. Task card validar peticiones CRUD.....	147
Tabla 125. Tarjeta colaboración HU10 Guardar Producto .....	148
Tabla 126. Historia de usuario: Seleccionar indicador .....	148
Tabla 127. Task card indicadores de gestión .....	149
Tabla 128. Historia de usuario: enviar gráfica indicador.....	150
Tabla 129. Task card indicador de gestión gráfico .....	151
Tabla 130. Evento Programado Indicador.....	153
Tabla 131. Prueba de funcionalidad indicador de eventos.....	154
Tabla 132. Retrospectiva Sprint 3 .....	156

## RESUMEN

Este proyecto está orientado a gestionar los procesos internos de la Fundación Voluntarios Animalistas Popayán, para la implementación de este proyecto se utilizaron algunas prácticas de Scrum como entorno de trabajo y Extreme Programming abreviado (XP) como metodología de desarrollo abarcando las fases planeación, diseño, desarrollo y pruebas. Tiene como objetivo el desarrollo de una plataforma web para la recopilación de información organizacional de manera fácil y centralizada. Por otra parte, se generan indicadores de gestión gráficos que determinan el estado actual de los procesos internos de cada comité basándose en hechos y datos proporcionando una herramienta para la toma de decisiones y estrategias organizacionales y comerciales.

Palabras claves: *plataforma web, xp, Scrum, fundación.*

## **ABSTRACT**

This Project is aimed to manage the processes of the Foundation “Voluntarios Animalistas Popayán”, for the implementation of this project were used some practices of Scrum as a framework and XP as development methodology covering with phases planning, design, development and testing. It aims to develop a web platform for gathering organizational information easily and centrally. Otherwise graphical management indicators are generated determine the current state of the internal processes of each Committee based on facts and data providing a tool for decision-making and organizational strategies and business.

Keywords: web platform, XP, Scrum, foundation.



## INTRODUCCIÓN

En la ciudad de Popayán (Cauca) existen tres fundación en pro del cuidado de la fauna domestica callejera, la “Fundación para la protección de los animales Vida Animal”, “Voluntarios Animalistas Popayán” y “Rescate y bienestar animal”, de ahí que el objeto de investigación es la fundación Voluntarios Animalistas Popayán conformada por cuatro comités: adopción, bienestar animal, comunicación y donación, cada uno proporciona información significativa para la fundación detallando datos básicos de los animales o actividades que están bajo la responsabilidad del representante de cada comité y a partir de la información recolectada permite a la fundación tomar decisiones, plantear estrategias y proponer soluciones ambientales a la comunidad.

Las estrategias, soluciones o decisiones tomadas por la fundación se basan en datos adquiridos de forma manual a través de formatos, lo que generaba pérdida de información, duplicidad, retrasos para realizar determinadas consultas e inconvenientes entre los integrantes de la fundación. Para contribuir tecnológicamente al problema anteriormente mencionado se propuso implementar un proyecto en el que se incluyó la planificación, diseño, desarrollo y pruebas de una plataforma web para la fundación Voluntarios Animalistas Popayán para la recopilación de información organizacional de la misma.

Es claro que la secretaría de salud ambiental no tiene como objetivo rescatar, rehabilitar y dar en adopción animales domésticos, pero sí en realizar jornadas de vacunación y esterilización, para disminuir los índices del virus de la rabia por sus altos índices de mortalidad transmitida de animales a seres humanos, además de prevenir la superpoblación canina evitando la propagación de enfermedades y animales domésticos en estado de abandono, por este motivo se determinó contribuir tecnológicamente tanto a la fundación con una plataforma web que permite la administración de información proporcionando indicadores de gestión

como a la comunidad payanesa interesada con una herramienta para informarse y participar en los eventos en pro del cuidado la fauna domestica callejera.

## 1. PLANTEAMIENTO DEL PROBLEMA

### 1.1 PLANTEAMIENTO DEL PROBLEMA

Las fundaciones para la protección de animales en la ciudad de Popayán (Cauca) tales como “Fundación para la protección de los animales vida Animal”, “Voluntarios Animalistas Popayán” y “Rescate y bienestar animal” se han conformado por el aumento de caninos y felinos en distintas condiciones de abandono y maltrato, contrarrestando la problemática ambiental a causa del alto grado de reproducción que tienen los animales, en promedio una pareja de perros y su descendencia en un año pueden originar 16 cachorros es decir que en seis años pueden originar 67.000 cachorros y a la gran variedad de enfermedades por las condiciones en las que viven; por consiguiente, cada una de las fundaciones constituidas legalmente tiene como objetivo rescatar, rehabilitar y dar en adopción para disminuir la Zoonosis que es la trasmisión del virus de la rabia de animales caninos a seres humanos a través de mordeduras, en la mayoría de casos es mortal si no es tratado, en promedio causa la muerte a 60.000 personas al año [1].

“Voluntarios Animalistas Popayán” es una Fundación sin ánimo de lucro que trabaja activamente en pro de la fauna callejera, buscando promover la tenencia responsable y respeto por la vida animal, colaborando en actividades tales como el rescate, recuperación, socialización, esterilización y adopción de animales que se encuentran abandonados, maltratados y son víctima de la indiferencia; en la ciudad de Bogotá se reportan más de 800.000 perros y gatos callejeros en condiciones totalmente de abandono [2], así mismo en la ciudad de Cali Zoonosis reporta 11.000 animales habitantes de la ciudad, en las mismas condiciones [3], por otra parte, según reportes de las fundaciones en Popayán bajo su protección hay alrededor de 300 animales exceptuando los que habitan en la calle para un aproximado de 500 perros callejeros [4], [5], [6].

La Fundación “VAP” se encuentra conformada por cuatro comités, el comité de adopción, bienestar animal, donaciones y comunicación, sus procesos son realizados de forma manual, el primero se encarga de llevar toda la documentación y requisitos necesarios para que las personas adopten una mascota, por lo tanto, genera información desactualizada y no se puede realizar el seguimiento ya que en muchas ocasiones el adoptante ha cambiado de domicilio o teléfono, por esta razón no se obtiene un control continuo del crecimiento de la mascota, el segundo se encarga de tener toda la documentación al día de los animales que se encuentran en el hogar de paso, al no tener la información sistematizada se dificulta llevar un control periódico de los animales, generando duplicidad de la información que no coinciden entre los comités, el comité de donación se encarga de recaudar todo tipo de donaciones, por ejemplo dinero en efectivo, alimento, medicamentos, cobijas, camas para mascotas entre otros, al igual el registro de ventas de sus productos como calendarios, lapiceros, vasos, cuadernos, camisetas, ropa para mascotas y collares; cada uno de los integrantes del comité y voluntarios se encargan de vender y recaudar donaciones que muchas veces no son registradas, es por esta razón que no se tiene un control de las ventas y donaciones. Finalmente, el comité de comunicación se encarga de divulgar las actividades como jornadas de esterilización, vacunación, adopción, mascotas perdidas y convenios con clínicas veterinarias, esta comunicación se hace a través de una página en Facebook de ahí que llega a un público sesgado de quienes tengan cuenta en Facebook o que revisen permanentemente la red social.

### **1.1.1 Pregunta de investigación.**

¿Cómo contribuir tecnológicamente en la planificación, control y gestión de procesos de adopción, donación, bienestar animal y comunicación de eventos liderados por los cuatro comités de la Fundación “VAP”?

## **1.2 JUSTIFICACIÓN**

### **1.2.1 Justificación técnica**

Debido a que los procesos de recolección de datos actualmente en la fundación son llevados manualmente desencadenan pérdida y duplicidad de información, altos tiempos de respuesta, generando que las decisiones tomadas no sean las acertadas.

Con la realización de este proyecto se busca brindar una solución tecnológica a partir de la implementación de una plataforma web adaptativa bajo los entornos de trabajo *Yii* y *Bootstrap*, el primero para la implementación de la funcionalidad y el segundo con la integración de *Html*, *CSS* y *JavaScript* en la parte visual de la plataforma, que sistematice los procesos internos, mejorando así el rendimiento, organización de procesos y tiempos de respuesta, asimismo servir de base para la implementación de nuevos sistemas de información que apoyen la problemática actual.

Por lo tanto, en el desarrollo de este trabajo se ponen en práctica todos los conocimientos adquiridos en programación e Ingeniería de software, además la gestión de proyectos, habilidades de comunicación y documentación, brindando una solución adecuada para mejorar los procesos administrativos de la fundación "VAP" dando a conocer sus actividades en pro de la fauna domestica callejera en la comunidad Caucana.

### **1.2.2 Justificación Funcional.**

Se propone desarrollar una plataforma web, que permita gestionar los procesos en cada comité, como el registro de las mascotas con su historia clínica con los datos característicos del animal su respectivo dueño, las consultas realizadas con sus correspondientes medicamentos teniendo en cuenta el trámite realizado si es

esterilización y/o vacunación, también las donaciones recibidas como dinero en efectivo, alimento, medicamentos, cobijas, camas para perros, productos zoonosanitarios, igualmente el registro de productos como por ejemplo calendarios, lapiceros, vasos, cuadernos, camisetas, ropa para perros y collares, así mismo jornadas de esterilización, voluntarios, cronograma de eventos, clínicas veterinarias vinculadas, seguimiento del adoptante y adoptado (mascota) e indicadores de actividades nombradas anteriormente; con un diseño web adaptativo a las diferentes condiciones de los dispositivos, permitiendo un control en los procesos administrativos de los animales domésticos que están bajo la protección de la Fundación “VAP”, además de ser un desarrollo escalable.

Esta implementación traerá tanto para los integrantes de la Fundación como para los usuarios de está grandes beneficios como, control de productos e ingresos, eliminar procesos manuales, información oportuna y actualizada permitiendo generar indicadores de gestión, rapidez en la consulta de registros, confiabilidad de los datos, centralización de la información, mantener informada a la Fundación sobre la situación de la mascota adoptada, divulgación de eventos a los usuarios y acceso adaptativo ajustado a diferentes dispositivos.

### **1.2.3 Justificación económica-financiera**

Al ser una plataforma web con los procesos de la Fundación “VAP” centralizados minimizará costos de transporte, operación y comunicación de los representantes de los comités, asimismo reducción de tiempos de respuesta para obtener información precisa y actualizada.

### **1.2.4 Justificación social.**

Desde el punto de vista jurídico en la actualidad en Colombia se ha aplicado la ley 1774 del 6 de Enero del 2016 que especifica “Los animales como seres sintientes

no son cosas, recibirán especial protección contra el sufrimiento y el dolor, en especial, el causado directa o indirectamente por los humanos, por lo cual en la presente ley se tipifican como punibles algunas conductas relacionadas con el maltrato a los animales, y se establece un procedimiento sancionatorio de carácter policivo y judicial” [7].

La fundación “VAP” tiene como objetivo rescatar, rehabilitar y dar en adopción para disminuir la Zoonosis ya que en promedio causa la muerte a 60.000 personas al año, la implementación del sistema de información apoyará la finalidad de la Fundación que afronta la problemática actual del incremento de los animales domésticos abandonados y maltratados en las calles, puesto que “la principal causa es la falta de recursos económicos, espacio en el hogar, enfermedades y agresividad” [8].

### **1.3 OBJETIVOS**

#### **1.3.1 Objetivo general.**

Contribuir tecnológicamente en la gestión de procesos administrativos de los cuatro comités de la Fundación “Voluntarios Animalistas Popayán” a través de una plataforma web.

#### **1.3.2 Objetivos específicos.**

- Facilitar los procesos administrativos realizados por los integrantes de los cuatro comités de adopción, donación, bienestar animal y comunicación en la Fundación “VAP”.
- Centralizar los procesos de adopción, donación, bienestar animal y comunicación en la Fundación “VAP” tecnificando el control administrativo para la toma de decisiones.

- Proveer a los integrantes de la Fundación, información con indicadores de gestión de los comités de adopción, donación, bienestar animal y comunicación en la Fundación “VAP”.



## **2. MARCO TEÓRICO.**

### **2.1 ANTECEDENTES**

A través de las fundaciones se ha empezado a concienciar a las personas sobre la tenencia responsable y el respeto por la vida animal, logrando que éstas participen en las diversas actividades realizadas por las fundaciones frente al cuidado de la fauna domestica callejera, de ahí que comenzaron a tomar fuerza las contribuciones tecnológicas. En la siguiente tabla se presentan seis (6) apoyos tecnológicos orientados a favorecer los procesos de fundaciones protectoras de animales:

### 2.1.1 Antecedentes

Tabla 1. Antecedentes

Título	Beneficiario	Principal aporte
Plataforma web para la administración clínica de los animales	Fundación Protección Animal Ecuador	Agilizar los reportes diarios de médicos y rescatistas [9].
Plataforma web “Voz Animal”	Fundación “Voz Animal”	Permite la administración de los procesos internos tales como promover campañas de esterilización, apadrinar, educación y promoción de adopciones [10].
Aplicativo web <b>Online</b>	Instituciones dedicadas a la fauna callejera en la ciudad de Guayaquil	Aplicativo web para el estudio de la problemática actual de adopción de mascotas en la ciudad de Guayaquil para agilizar el proceso de adopción Online, con el fin de establecer cláusulas de contrato, fechas de visitas y apadrinamientos [11].
Plataforma Animal Safe	Fundación “ <i>Animal Safe</i> ”	Ubicada en Ginebra, Valle del Cauca

		esta plataforma se utiliza para el seguimiento, proceso de adopción, apadrinamiento, protección, suministro de alimentos, medicamentos y albergue de aprox. de 650 perritos con el objetivo de mejorar su calidad de vida [12].
Plataforma web “Paraíso de la mascota”	Fundación “Paraíso de la mascota”	Localizada en la ciudad de Cali, Valle del Cauca, controla los procesos como búsqueda de mascotas perdidas, adopciones, atención clínica veterinaria, donaciones y eventos [13].
Plataforma web “Animal naturalis”	Organización “Animal naturalis”.	Trabaja en pro de todos los animales, en defensa de sus derechos; se encarga de difundir campañas, artículos y recibir donativos a través del portal web [14].

## **2.2 BASES TEÓRICAS**

### **2.2.1 Sistema de información**

“Para soportar los procesos misionales y de apoyo en una organización es importante contar con sistemas de información que se conviertan en fuente única de datos útiles para apoyar o argumentar las decisiones corporativas.

Como resultado de la gestión de sistemas de información se obtienen los sistemas de información de apoyo, sistemas de información misionales, servicios informativos digitales y sistemas de información de direccionamiento estratégico” [15].

Los sistemas de información proporcionan una gran variedad de opciones, que permite tomar decisiones acertadas ya que se cuenta con información útil que mejora la gestión de procesos dentro de una organización, por tanto contienen datos internos característicos de éstas, pero estos datos por si solos no brindan ningún significado, por esta razón los sistemas de información cobran importancia ya que los datos característicos del entorno de la empresa son debidamente modelados convirtiéndose así en información oportuna para el control, gestión y formulación de estrategias comerciales por parte de los directivos y/o administrativos enfocados al crecimiento organizacional.

### **2.2.2 Plataforma Web**

“Una aplicación Web es una aplicación informática que se ejecuta en un entorno Web. La arquitectura Web permite a múltiples clientes o usuarios establecer una comunicación a través de Internet con un servidor Web” [16].

Una plataforma web permite a varios usuarios acceder a una aplicación informática para realizar gestión de procesos desde cualquier tipo de dispositivo.

### 2.2.3 Framework

“Framework es una estructura de soporte definida en el que otro proyecto de software puede ser organizado y desarrollado. En programación, es un set de funciones o código genérico que realiza tareas comunes y frecuentes en todo tipo de aplicaciones (creación de objetos, conexión a base de datos, etc.). Esto brinda una base sólida para desarrollar aplicaciones concretas y permite obviar los componentes más triviales y genéricos del desarrollo y abre camino a que diseñadores y programadores puedan pasar más tiempo identificando requerimientos de software” [17].

Los *Framework* que son utilizados para el desarrollo web son un conjunto de librerías que dan soporte al desarrollo y la construcción de cualquier sitio web, es decir facilitan el acceso a la base de datos, uso de plantillas, mapeo de *URLs* entre otras cosas permiten una modularización de los componentes y promueve la reutilización de código puesto que es orientado a objetos, por ello algunos framework se basan en el patrón de arquitectura MVC que es la separación del modelo de datos, interfaz gráfica y controlador posibilitando agregar nuevas funcionalidades, además permitir agregar nuevos integrantes al grupo de desarrollo cómodamente, de ahí que usar un *Framework* para desarrollos reducen tiempos tanto en la elaboración como implementación [17].

### 2.2.4 Yii

“*Yii* es un *Framework* PHP basado en componentes de alta rendimiento para desarrollar aplicaciones Web de gran escala. Él mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo” [18].

*Yii* es un *Framework* que maneja el patrón modelo, vista, controlador abreviado MVC, es decir separa los datos, lógica del negocio y las interfaces de usuario,

permitiendo así que el software sea escalable y los desarrolladores puedan realizar sus diversas modificaciones, además de obtener un código reutilizable y limpio personalizándolo para adaptarse a las necesidades del desarrollo.

## 2.2.5 Metodologías ágiles



Figura 1. Metodologías ágiles

[19]

Las metodologías tradicionales de desarrollo se caracterizan “por ser rígidas y dirigidas por la documentación que se genera en cada una de las actividades desarrolladas” [20], rehusándose a los posibles cambios que se puedan generar a lo largo de la implementación de un proyecto *software* además de estar dirigidos por normas, políticas y contratos prefijados.

Como alternativa surgió el concepto “ágil” aplicable al desarrollo de software teniendo como objetivo “esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto” [20], algunas de las metodologías ágiles existentes son:

- **Agile Unified Process (AUP):** “Describe un enfoque simple y comprensible para la construcción de sistemas mediante la utilización de prácticas y técnicas ágiles sin dejar de cumplir con RUP” [21].
- **Dynamic Systems Development Method (DSDM):** “Se trata de un proceso de desarrollo ágil que divide el proyecto en tres fases: pre-proyecto, ciclo de vida y post-proyecto. La fase de ciclo de vida, a su vez, se divide en cinco instancias: estudio de factibilidad, iteración de modelo funcional, iteración de diseño y construcción, estudio de negocio y construcción e implementación” [21].
- **Extreme Programming (XP):** “La programación extrema se diferencia de las metodologías tradicionales, al igual que las metodologías ágiles en general, por ser un enfoque basado en la adaptabilidad más que en la previsibilidad” [21], en el capítulo III y IV se hará más énfasis del uso de esta metodología en el proyecto.
- **Essential Unified Process (EssUP):** “Las prácticas en *EssUP* integran los principios del éxito del proceso unificado, la agilidad y los conceptos de madurez de procesos o *Capability Maturity Model Integration (CMMI)*, aprovechando sus diferentes capacidades: estructura, agilidad y mejora de procesos” [21].
- **Scrum:** “*Scrum* más que una metodología es un entorno de trabajo para la gestión de proyectos complejos. Se basa principalmente en la premisa de ejecutar un proyecto en iteraciones de entre dos y cuatro semanas, llamado *Sprint*, y de duración fija” [21], en el capítulo III y IV se hará más énfasis del uso de este entorno de trabajo en el proyecto.

### 2.2.6 Lean Canvas

“*Lean Canvas* es una herramienta rápida y eficaz para crear y comunicar modelos de negocio para *Startups*. Propone un método híbrido basado en el modelo de negocio *Canvas* y la metodología *Lean*, surge una herramienta para diseñar modelos de negocio de la forma más eficaz” [22].

La forma de utilizar *Lean Canvas* es resolver los siguientes ítems:

- **Segmentos de clientes:** Identificar los posibles clientes del modelo de negocio a realizar
- **Problemas:** “Reconocer cuales son los 3 principales problemas que se solucionarían con el modelo de negocio a desarrollar, descubriendo cuales son las soluciones alternativas del producto” [22].
- **Proposición única de valor:** Especificar la viabilidad del proyecto.
- **Solución:** “Establecer cuáles son las 3 características más importantes del producto/servicio que resuelvan los problemas encontrados en el ítem “Problema” [22].
- **Canales:** Los canales de comunicación para hacer llegar el producto al mercado.
- **Flujos de ingreso:** “Plantear una estrategia sobre cómo se ganaría dinero con el desarrollo del producto” [22].
- **Estructura de costes:** Estimar gastos para el desarrollo del proyecto.
- **Métricas clave:** “Establecer las actividades a medir, generando un conjunto de indicadores para tomar decisiones” [22].



- **Ventaja diferencial:** Definir el valor agregado del producto a desarrollar.

## 2.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE

### 2.3.1 Extreme Programming

Metodología ágil que promueve el trabajo en equipo, retroalimentación continua y comunicación entre el cliente y el equipo de desarrollo, esta metodología permite responder al cambio expuesto por el cliente brindando soluciones simples, además la aplicación de test continuos asegurando la calidad y el continuo progreso del software, al ser una metodología basada en valores Extreme programming y cambio de roles permite al equipo de trabajo tener una visión más clara para obtener versiones estables y ejecutables que finalmente desencadenan el cumplimiento de los objetivos [23].

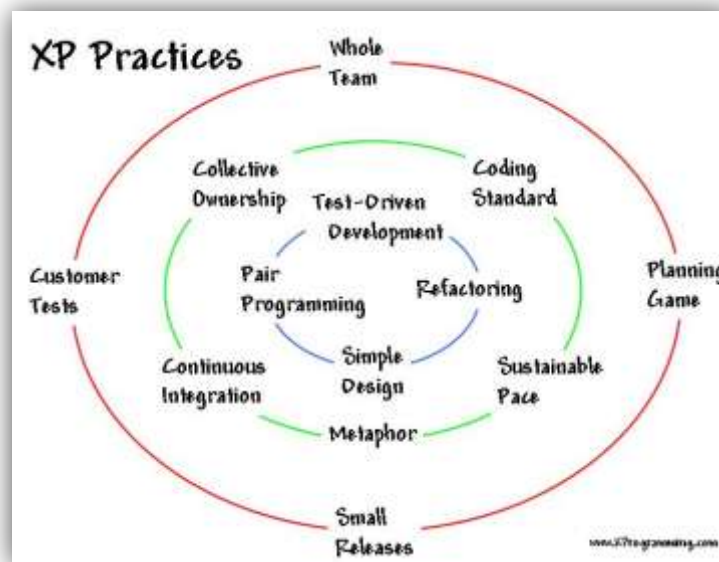


Figura 2. Prácticas extreme programming

[24]

### 2.3.1.1 Valores Extreme Programming

- **Comunicación:** “Extreme Programming se enfoca en la comunicación directa y fluida, permitiendo que el equipo de desarrollo tenga una visión conjunta de lo que se debe realizar en el proyecto de desarrollo, minimizando la documentación formal y aplicando la programación en parejas para el cumplimiento de este valor” [23].
- **Sencillez:** “Extreme programming busca la simplicidad en el desarrollo de un producto software, cumpliendo con las funcionalidades especificadas para el proyecto” [23].
- **Retroalimentación:** “Los programadores obtienen el estado del software a través de los test de unidad e integración continua, además el cliente consigue ver el estado del proyecto a lo largo de la iteración a través de los test de aceptación permitiendo una retroalimentación” [23].
- **Valentía o Coraje:** “El equipo de desarrollo debe establecer una comunicación que permita manifestar inquietudes, dudas e ideas acerca de las funcionalidades del proyecto” [23].

### 2.3.1.2 Variables Extreme Programming

La metodología Extreme Programming especifica cuatro variables aplicables a proyectos de desarrollo como **costo, tiempo, calidad y alcance**, [25] proporcionando un método de control. Cabe aclarar que en un equipo de trabajo el cliente o Jefe tienen la posibilidad de escoger tres de las cuatro variables asignando la descartada al equipo de desarrollo para que este la evalúe.

- **Costo:** Hace referencia a la cantidad de dinero que se debe invertir para el desarrollo de un proyecto, se deben de tener en cuenta varios factores como la cantidad de integrantes en el equipo de desarrollo, equipos, alcance del proyecto, entorno de desarrollo entre otros, cada una de estas variables permiten visualizar posibles situaciones que puedan favorecer o afectar el desarrollo del proyecto, para finalmente obtener el costo de éste [25].
- **Tiempo:** Realizar una estimación de tiempo para un proyecto de desarrollo, va ligado al alcance planteado, equipo de desarrollo y equipos asignados, por esta razón se debe establecer un tiempo prudencial ya que en exceso afecta el alcance del proyecto y a falta de tiempo puede afectar la calidad y el costo. [25]
- **Calidad:** La calidad del proyecto se visualiza de dos formas, la primera la calidad externa a través de las interfaces de usuario y la velocidad del sistema, el cliente percibe que en el producto obtenido se observen las especificaciones otorgadas por él, la segunda es la calidad interna esta hace referencia a los componentes internos del sistema y entre más alta sea se tiene un control adecuado del producto software [25].
- **Alcance:** Definir el alcance dentro de un proyecto de desarrollo es uno de los factores más importantes, ya que permite definir el costo, tiempo y calidad igualmente permite realizar una planificación para el desarrollo de éste [25].

### 2.3.1.3 Fases Extreme Programming

- **Planificación:** La planificación es la fase inicial de la metodología Extreme Programming, en esta fase se realizan una serie de entrevistas entre el cliente, equipo de desarrollo y gerentes permitiendo determinar el objetivo, integrantes del equipo de trabajo, tiempo y cronograma del proyecto. Para la ejecución de esta fase se aplican reglas y prácticas que se describirán a continuación:
  - ✓ **Historias de Usuario:** “sustituyen a los documentos de especificación funcional y a los casos de uso” [26] pero las historias de usuario deben especificar de manera puntual la funcionalidad a desarrollar según el cliente, para que sean entendibles y se realice la estimación de tiempo y calidad por parte del equipo de desarrollo, quienes posteriormente harán una entrevista al cliente para ampliar la información depositada, el tiempo de desarrollo de una historia de usuario es de 1 a 3 semanas.
  - ✓ **Plan de entregas:** Es la elaboración de un cronograma de entregas elaborado por clientes, equipo de desarrollo, gerentes entre otros, en el que se definirá la priorización de las historias de usuario según las necesidades del cliente, dadas las especificaciones de tiempo establecidas por el equipo de desarrollo [26].
  - ✓ **Plan de iteraciones:** Al inicio y cierre de los ciclos de desarrollo de las historias de usuario se realiza una planificación de iteraciones, se desarrolla y prueba la funcionalidad de cada una de las historias de usuario, verificando el correcto funcionamiento sin alterar las funcionalidades ya validadas; estas pruebas se denominan pruebas de aceptación en las que se evalúa la calidad y vulnerabilidad de la implementación de las historias de usuario [26].

- ✓ **Velocidad del proyecto:** La velocidad del proyecto hace referencia a emplear el tiempo estimado para cada historia de usuario, ya que si se cumple con los tiempos asignados se logra mantener al equipo de trabajo motivado.
- ✓ **Programación en parejas:** “XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador” [26], según XP aumenta la productividad y la calidad del producto final ya que frente a posibles errores abran más posibilidades de solución.
- ✓ **Reuniones diarias de seguimiento:** El equipo de desarrollo programa reuniones diarias para exponer las dificultades, soluciones e ideas manteniendo una comunicación fluida y proponer soluciones en equipo.
- **Diseño:** En esta fase se define el diseño del producto software a desarrollar, se deben aplicar conceptos de simplicidad y claridad, [26] para la ejecución de esta fase se aplican reglas y prácticas que se describirán a continuación:
  - ✓ **Simplicidad:** Los diseños a implementar en el producto software deben ser sencillos e intuitivos para el cliente, XP le apuesta a una implementación en la que se ejecute en menor tiempo, también se debe tener en cuenta que no se puede desarrollar una funcionalidad que no haga parte de la iteración.
  - ✓ **Recodificación:** XP le apuesta a reutilizar “código” funcional ya que sirve de base para ampliar la funcionalidad que se esté desarrollando mejorándolo para adaptarlo a las necesidades del producto a desarrollar, pues si se utilizara sin recodificar podría causar anomalías futuras representando pérdida de tiempo y aumento de costos.

- ✓ **Metáfora del sistema:** Se establecen pautas, sintaxis y nomenclatura permitiendo una comunicación fluida entre el equipo de trabajo y el cliente.
- ✓ **Tarjetas C.R.C:** (Clase, Responsabilidad y colaboración), ayudan al equipo de desarrollo a utilizar la programación orientada a objetos, estableciendo un orden lo que optimizan tiempos y costos.
- **Desarrollo:** “En esta fase se divide las funcionalidades en pequeñas unidades, siendo implementadas por separado para finalmente ser integradas cumpliendo con los objetivos planteados”, [26] para la ejecución de esta fase se aplican reglas y prácticas que se describirán a continuación:
  - ✓ **Disponibilidad del cliente:** Así como el equipo de desarrollo es uno de los factores más importantes, el cliente también lo es ya que desde el inicio del desarrollo del proyecto está presente con la especificación de las historias de usuario y a lo largo del desarrollo de estas proporciona información relevante, por las razones expuestas es fundamental la participación y el rol que desempeña.
  - ✓ **Programación dirigida por las pruebas:** Al implementar test de funcionamiento, se puede visualizar la funcionalidad que se está ejecutando para determinar posibles errores y así mismo corregirlos de manera inmediata previniendo posibles fallas que se puedan presentar más adelante, también permite la creación de un repositorio para acceder a las versiones estables.

- ✓ **Uso de estándares:** Al implementar estándares de codificación, proporciona al equipo de desarrollo escalabilidad y además facilita la comprensión de la implementación.
- **Prueba:** En esta fase se realizan “pruebas de funcionalidad que permiten identificar fallas del programa”, [26] para la ejecución de esta fase se aplican reglas y prácticas que se describirán a continuación:
  - ✓ **Pruebas unitarias:** Uno módulo puede ser una versión estable siempre y cuando cada uno de sus componentes hayan aprobado las pruebas unitarias, este tipo de pruebas son de utilidad para el equipo de desarrollo ya que pueden reutilizar el código y tener acceso a antiguas versiones.
  - ✓ **Detección y corrección de errores:** Al encontrar un error dentro de una iteración se debe corregir de manera inmediata ya que puede servir de base para otras fallas.
  - ✓ **Pruebas de aceptación:** “son consideradas como pruebas de caja negra” [26], en estas pruebas se somete al producto software en función de casos reales, verificando la funcionalidad de cada una de las historias de usuario, es decir que estando en un entorno real cumpla con el objetivo.

### 2.3.2 Patrón MVC

MVC el patrón de arquitectura de software propone tres componentes modelo, vista y controlador, permitiendo separar los datos y lógica de las interfaces de usuario. Proporcionando así reutilización de código que facilita el desarrollo de aplicaciones.

### **2.3.2.1 Modelo**

“El Modelo es el objeto que representa los datos del programa. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo” [27]

### **2.3.2.2 Vista**

“La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo” [27].

### **2.3.2.3 Controlador**

“El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista” [27].

## **2.4 FRAMEWORK DE GESTIÓN DEL PROYECTO**

### **2.4.1 Valores ágiles**

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación amplia.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

### **2.4.2 Manifiesto ágil**

- La prioridad más alta es satisfacer al cliente con entregas tempranas y continuas de software valorable.



- Bienvenido requisitos cambiantes, incluso en desarrollo avanzado. Los procesos ágiles utilizan el cambio para dar ventaja competitiva al cliente.
- Entregar software funcional frecuentemente, desde un par de semanas a un par de meses, con preferencia a la escala de tiempo más corta.
- Empresarios y desarrolladores trabajan juntos diariamente durante el proyecto.
- Construir proyectos con personas motivadas. Entregarles el entorno y soporte que necesitan y confiar que harán el trabajo.
- El método más efectivo y eficiente para transmitir información desde y hacia un equipo de desarrollo es la conversación cara a cara.
- El software funcional es la principal medida de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deberían ser capaces de mantener un ritmo constante de manera indefinida.
- La atención continua a la excelencia técnica y buen diseño mejoran la agilidad.
- Simplicidad – el arte de maximizar la cantidad de trabajo no hecho – es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- En intervalos regulares, el equipo reflexiona sobre cómo llegar a ser más eficaz, para luego ajustar su comportamiento de manera correspondiente [28].

### 2.4.3 Scrum

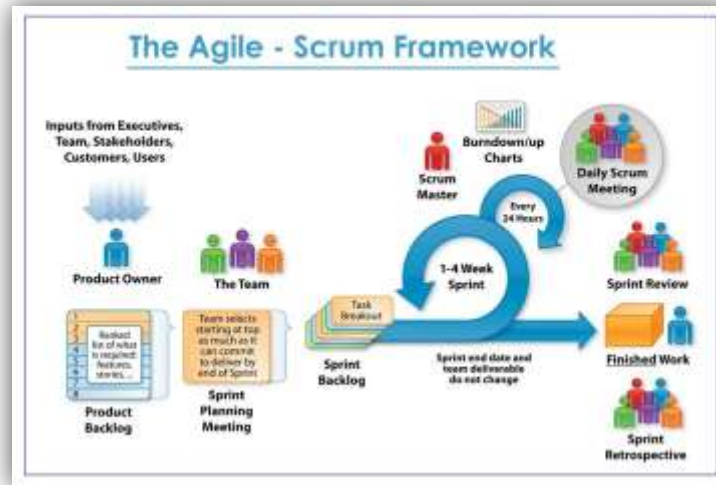


Figura 3. Entorno de trabajo Scrum

[29]

Según *Ken Schwaber y Jeff Sutherland* *Scrum* es un entorno de trabajo para la gestión del proyecto, basándose en el empirismo es decir la experiencia que tienen las personas para tomar decisiones puesto que se tiene un conocimiento previo para la gestión de un proyecto [30], también permite enfrentarse a problemas complejos empleando varias técnicas y procesos para el control de estos, *Scrum* motiva a los integrantes de un equipo de trabajo, debido al enfoque de documentación necesaria, entregas continuas y funcionales, reuniones diarias de sincronización del equipo y adaptabilidad a los cambios.

### 2.4.3.1 Pilares Scrum

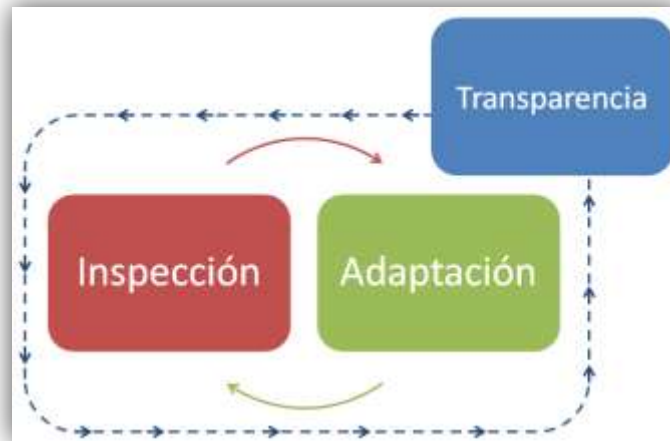


Figura 4. Pilares Scrum

[31]

*Scrum* busca el control de procesos empíricos, se basa en la experiencia de las personas en los diversos roles ejercidos en grupos de desarrollo para así tomar decisiones a partir de lo que se conoce, estimando un control del riesgo apoyándose en tres pilares: transparencia, inspección y adaptación.

- **Transparencia:** “Los aspectos significativos del desarrollo del proyecto deben ser visibles para el equipo de trabajo. De tal manera que compartan un entendimiento común de lo que se está realizando” [30].
- **Inspección:** “Los usuarios de *Scrum* deben inspeccionar frecuentemente los artefactos de *Scrum* y el progreso de éste, para detectar variaciones y realizar correcciones a tiempo” [30].
- **Adaptación:** “Acorde a los resultados obtenidos en la inspección se deben realizar los ajustes pertinentes de los problemas encontrados” [30].

### 2.4.3.2 Equipo Scrum

“Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener una retroalimentación continua” [30], ya que son equipos autoorganizados y multifuncionales lo que permite elegir diversas formas para trabajar, “el equipo Scrum consiste en un Dueño de Producto (*Product Owner*), el Equipo de Desarrollo (*Development Team*) y un *Scrum Master*” [30].

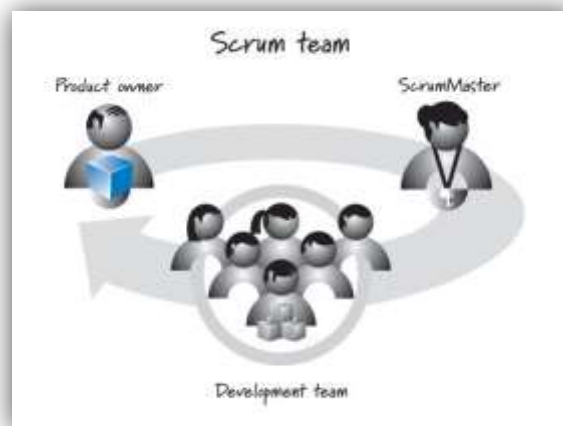


Figura 5. Scrum team

[32]

- **Dueño de Producto (Product Owner):** se ocupa de priorizar la lista del producto, de esa manera ordena y optimiza lo que se va a realizar durante el sprint de forma que sea entendible para el equipo de desarrollo.
- **Equipo de Desarrollo (Development Team):** Son los encargados de la implementación software, con características de organización, multifuncionales y auto organizados [30].
- **Scrum Master:** Es el encargado de verificar que el entorno de trabajo se cumpla a cabalidad ayudando tanto al *Product Owner* como al equipo de desarrollo a regirse bajo los valores y prácticas de Scrum.

### 2.4.3.3 Artefactos

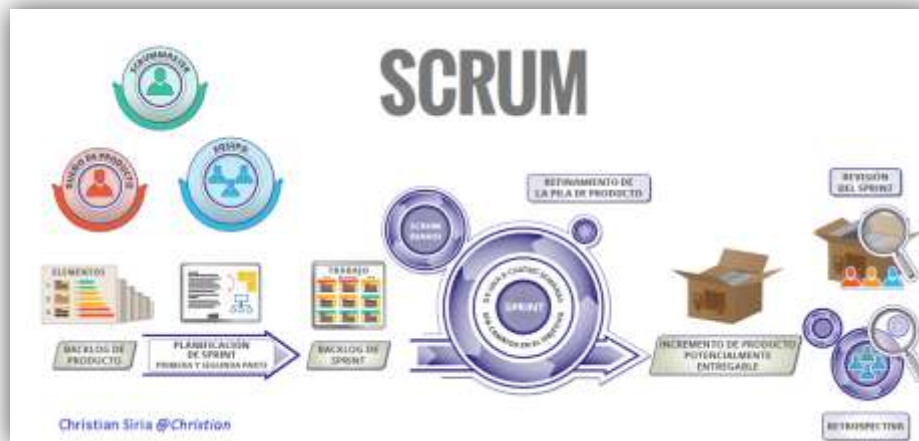


Figura 6. Artefactos Scrum

[33]

- **Product backlog:** “El *Product backlog* es simplemente una lista de ítems que representan trabajo pendiente. Cualquiera puede agregar ítems al *backlog*, pero sólo el *Product Owner* tiene el derecho de determinar el orden en el que serán desarrollados por el equipo” [30].

El *Product backlog* siempre está en constante cambio por tanto no se obtiene una lista final de las características del producto, así pues la lista debe estar lo más actualizada, esta lista contiene características, funcionalidades, requisitos y mejoras del producto software a ser entregado en un futuro.

- **Sprint backlog:** Sprint backlog es una representación física que especifica el orden de ejecución de las características del sistema a desarrollar en un *Sprint*, la lista de pendientes del *Sprint* es estimada por el equipo de desarrollo.
- **Incremento:** “El Incremento es la suma de todos los elementos de la lista de producto completados durante un *Sprint*, y al final de éste, el nuevo

incremento debe estar “Terminado”, esto significa que el dueño del producto puede liberarlo” [30].

#### 2.4.3.4 Eventos



Figura 7. Eventos de Scrum

[34]

“Todos los eventos son bloques de tiempo (*time-boxes*), que tienen una duración máxima. Una vez que comienza un *Sprint*, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que sea alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso” [30].

- **Sprint:** “es un bloque de tiempo (*time-box*) de un mes o menos durante el que se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable durante el Sprint se tienen las siguientes características: ” [30]
  - ✓ No se realizan cambios que puedan afectar al objetivo del *Sprint*.
  - ✓ Los objetivos de calidad no disminuyen.

- ✓ El alcance puede ser clarificado y renegociado entre el dueño de producto y el equipo de desarrollo.
  
- **Agile inception:** “Esencialmente se trata de técnicas de conceptualización que deberían emplearse en el proceso de iniciación de un proyecto para aumentar la probabilidad de éxito del producto resultante” [35]. Es decir al iniciar un proyecto todas las personas que lo integran o que están directamente implicadas en éste, deben clarificar las inquietudes con el fin de tener el producto que se desea. “El principal objetivo del agile inception es construir una visión completa sobre el concepto de producto y que además no caiga en sesgos personales, es decir, que esa visión sea compartida y comprendida de idéntica forma por los principales interesados” [35]. Por ello para preparar una inception se deben realizar las siguientes preguntas:
  - ✓ ¿Porque estamos aquí?: Describir por qué se está realizando el producto software.
  - ✓ Elevator Pitch: Cómo defender el proyecto si tuviera un tiempo determinado.
  - ✓ Diseña Tu caja: Permite explicar los beneficios que tendrían las personas con el producto.
  - ✓ Crea tu not list: describe que se va a hacer, lo que no está contemplado y lo que quizás se debe discutirse para ser realizado.
  - ✓ Que te quita el sueño: Permite determinar cuáles son los miedos y riesgos para minimizarlos y evitarlos.
  - ✓ Calcula tu tamaño: Determinar el tiempo que se usara para el cumplimiento del alcance.
  - ✓ Cuáles son tus prioridades: Permite indicar el nivel de prioridad de las facetas del producto es decir el alcance, presupuesto, tiempo o calidad [36].

- **Planificación sprint:** “La Reunión de Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para *Sprints* más cortos, el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito” [30].

En la planificación del sprint el *Product Owner* expone las funcionalidades que se requieren que sean implementadas, el equipo de desarrollo expresa sus inquietudes para comprender en detalle las funcionalidades a manera que el equipo determine cuánto puede entregar, la duración del sprint, el tamaño de integrantes del equipo de desarrollo de ahí que el equipo se compromete a entregar determinadas funcionalidades terminando el *Sprint*.

- **Daily:** “El *Scrum* diario es una reunión con un bloque de tiempo de 15 minutos para que el equipo de desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas” [30].

La reunión diaria es realizada para que el equipo establezca una comunicación y sincronicen su trabajo permitiendo así evitar bloqueos y un continuo progreso del desarrollo planeando estrategias para el cumplimiento del objetivo del *Sprint*.

- **Sprint review:** Al finalizar el *Sprint* se realiza una revisión para la aprobación de la entrega formal del producto, también tiene como objetivo hacer una retroalimentación de las actividades que se realizaron durante el *sprint* [30].

- **Retrospectiva:** “El propósito de la retrospectiva es:

- ✓ Inspeccionar cómo fue el último *Sprint* en cuanto a personas, relaciones, procesos y herramientas.



- ✓ Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras” [36].

La última reunión realizada en el sprint es la retrospectiva de manera que se enfoca en el análisis del proceso, es decir la forma en la que el equipo de desarrollo trabaja teniendo en cuenta las técnicas usadas, las herramientas, las habilidades y técnicas utilizadas por los integrantes del equipo *Scrum*, en esta reunión participa el *Product Owner*, *Scrum Master* y el equipo de desarrollo.

## 2.5 MARCO CONTEXTUAL

En la ciudad de Popayán (Cauca) existen varias Fundaciones sin ánimo de lucro en pro al cuidado de la fauna callejera, entre ellas nuestro objeto de estudio es la Fundación Voluntarios Animalistas Popayán, está se encarga del rescate, recuperación, socialización, esterilización y adopción de animales domésticos callejeros que se encuentran abandonados; entre los procesos internos realizados por la fundación a través de los comités son:

- ✓ Proceso de adopción
- ✓ Proceso de seguimiento
- ✓ Bienestar animal
- ✓ Comunicación

Teniendo en cuenta la cantidad de procesos realizados por la fundación desde el momento de su creación en la fecha 12 Noviembre de 2012, estos son registrados de forma manual, presentando problemas de organización, duplicidad de información e inconsistencia de datos, por consiguiente es necesario brindar una solución tecnológica que permita a los representantes del comité una adecuada administración de información, ejercer estrategias y tomar decisiones.

## 2.6 HERRAMIENTAS DE APOYO

### 2.6.1 Adobe Kuler



Figura 8. AdobeKuler

[37]

“*Adobe Kuler* es una aplicación alojada en la web de *Adobe Labs* para generar temas de color que puede inspirar cualquier proyecto. Permite navegar por miles de temas de la comunidad Kuler utilizando la tecnología de *Adobe Flash* para lograr que las herramientas en línea genere una paleta de colores para cualquier aplicación” [38].

### 2.6.2 Bitbucket

“*Bitbucket* es un servicio de alojamiento en la web, para los proyectos que utilizan el sistema de control de versiones *Mercurial* y *Git*, los repositorios pueden tener hasta cinco usuarios en el caso de las cuentas gratuitas y los repositorios privados no se muestran en las páginas de perfil” [39].



Figura 9. Bitbucket

[40]

### 2.6.3 Bootstrap

“*Bootstrap* es el *Frontend* más popular en los últimos tiempos. Es el primer marco *Frontend* móvil elegante, intuitivo y de gran alcance para el desarrollo de una web más fácil. Basado en los últimos estándares de desarrollo de Web *HTML*, *CSS*, *Javascript Plugins* de *jQuery* para validar entrada de datos, visualización de tablas y graficas” [41].

### 2.6.4 Highcharts

“*Highcharts* es una biblioteca de gráficos escrita en *JavaScript*, que ofrece una manera fácil de añadir gráficos interactivos a un sitio web o aplicación web. *Highcharts* ofrece tipos de graficas como líneas, áreas, columnas, barras, pastel, dispersión, indicadores angulares, rangos de área, rango de columna, burbuja, diagrama de caja, barras de error, cascada y tipos de gráfico polar” [42].

## 2.6.5 Mysql Workbench

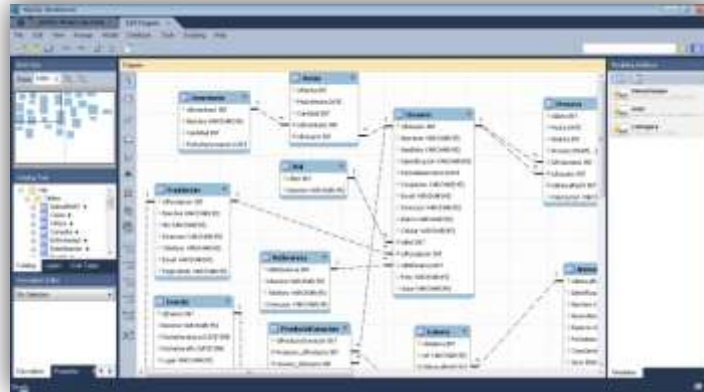


Figura 10. MySQL Workbench

[43]

“MySQL Workbench es una herramienta gráfica para trabajar con servidores y bases de datos MySQL. La funcionalidad de MySQL Workbench abarca cuatro temas principales” [43]:

- **Desarrollo de SQL:** “Permite crear y administrar las conexiones a los servidores de bases de datos. MySQL Workbench proporciona la capacidad de ejecutar consultas SQL” [43].
- **Modelado de datos (Diseño):** “Permite crear modelos de base de datos de forma gráfica” [43].
- **Administración del servidor:** “Permite administrar instancias del servidor MySQL mediante la administración de usuarios, la realización de copias de seguridad y recuperación” [43].

### 2.6.6 OpenShift

“*OpenShift* es una plataforma *PaaS (Platform as a Service)* es una categoría de servicios *Cloud* que proporciona una plataforma y un entorno que permiten a los desarrolladores crear aplicaciones y servicios que funcionen a través de internet. Los servicios *PaaS* se alojan en la nube, y los usuarios pueden acceder a ellos simplemente a través de su navegador web” [44].

### 2.6.7 PhpMailer

“*PHPMailer* es una clase desarrollada en PHP que trata de facilitar las funcionalidades asociadas a los envíos y manejos de emails. Se basa en el componente active server *ASPMail*. *PHPMailer* permite hacer de manera sencilla trabajos complejos, como adjuntar ficheros o realizar envíos de mails con formato *HTML*. Permite de una forma sencilla tareas complejas como por ejemplo:

- Enviar mensajes de correo con ficheros adjuntos.
- Enviar mensajes de correo en formato *HTML*” [45]

### 2.6.8 PhpMyAdmin

*PhpMyAdmin* es una herramienta de software libre escrito en PHP, con la intención de manejar la administración de *MySQL* a través de Internet. *PhpMyAdmin* es compatible con una amplia gama de operaciones en *MySQL*. Operaciones de uso frecuente: gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.” [46].

### 2.6.9 Planning Poker

“La técnica de *Planning poker* permite hacer una estimación inicial del proyecto rápida y fiable, dado que todos los miembros del equipo expresan su opinión sin

sentirse condicionados por el resto” [47], en la siguiente figura se presentara las barajas del *Planning poker*.



Figura 11. Planning Poker

[48]

“Cada número significa un peso / esfuerzo / complejidad para completar un objetivo (historia de usuario). La numeración de las cartas está basada en la sucesión de *Fibonacci*. La distancia entre números crece conforme se hacen mayores. De esta manera, se facilita la decisión sobre qué tamaño tiene un objetivo” [47].

### 2.6.10 Servidor Apache

“El servidor HTTP *Apache* es un servidor web de software libre desarrollado por *Apache Software Foundation* (ASF). El producto obtenido de este proyecto es un servidor de código fuente completo, descargable y gratuito. *Apache* es robusto y con un ciclo de desarrollo muy rápido gracias a la gran cantidad de colaboradores y voluntarios que dispone, *Apache* cuenta con las siguientes características: estable, flexible, eficiente y extensible” [49].

## 2.6.11 Sublime

“*Sublime Text* es un editor de texto pensado para escribir código en la mayoría de lenguajes de programación y formatos documentales de texto, utilizados en la actualidad: Java, Python, Perl, HTML, JavaScript, CSS, HTML, XML, PHP, C, C++, entre otros permite escribir todo tipo de documentos de código en formato de texto y es capaz de colorear el código, ayudar con la escritura, corregir mientras se escribe y usar abreviaturas” [50].

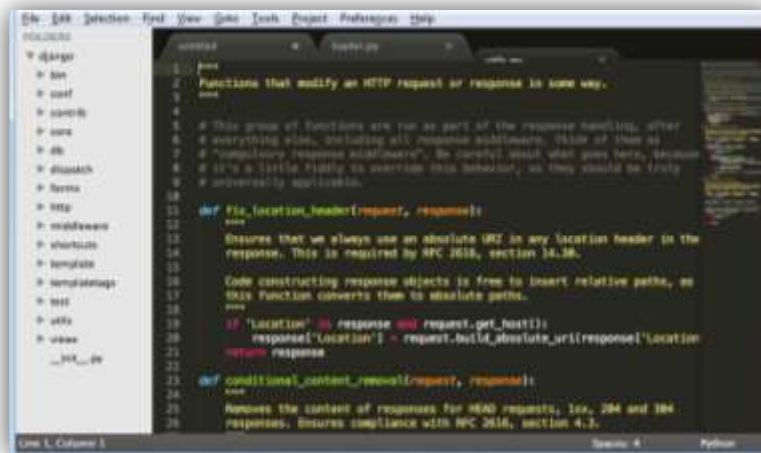


Figura 12. Sublime Text2

[51]

## 2.6.12 StartUml

“StarUML es una herramienta CASE libre y se encuentra bajo la licencia GPL (Licencia publica general). Es compatible con el modelado de sistemas usando los diagramas UML” [52], alguno de los diagramas que se pueden crear en *StarUML* son los casos de uso, diagramas de secuencia, colaboración y clases que facilitan el desarrollo de un producto software.



Figura 13. StarUML

[53]

### 2.6.13 Trello

“Trello facilita la visualización de un proyecto de arriba a abajo. Observar el progreso de tarjetas en un tablero proporciona un resumen de alto nivel sobre el progreso de un proyecto, mientras que cada tarjeta expone una parte del trabajo” [54].

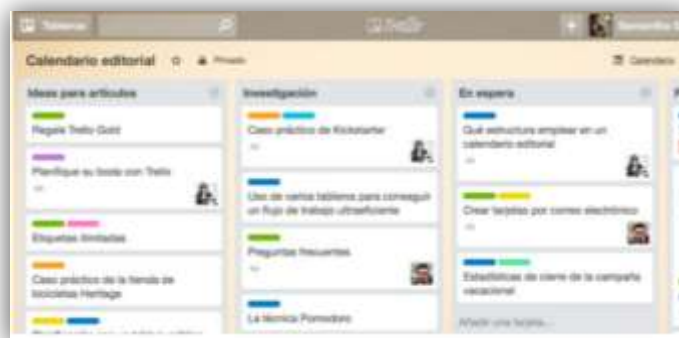


Figura 14. Tablero Trello

[54]

A través de la creación de un tablero en *Trello* permite enviar notificaciones a otros miembros de esta aplicación que sean vinculados a un tablero en específico, cada



miembro añadido puede crear, actualizar, eliminar y realizar comentarios sobre tareas y actividades que se postean con el objetivo de organizar un “proyecto creando notas rápidas en una pizarra blanca y que se pueden desplazar entre las columnas” [54].

#### 2.6.14 XAMPP

“XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar, su principal función es la gestión de base de datos” [55].

#### 2.6.15 WakaTime

WakaTime es un *plugin* que se configura en *Sublime Text* para determinar el tiempo en horas, días, semanas y mes de desarrollo que un desarrollador ha implementado sobre varios productos software, esta estimación se proporciona a través de graficas con indicadores en las que se puede observar lenguajes, editores y proyectos más usados, adicionalmente envía esta información al correo electrónico con el que un usuario se registró.



Figura 15. Waka Time [Fuente propia]

## 2.7 GLOSARIO

**Activista:** “Participante de un movimiento social, de una organización sindical o de un partido político que interviene activamente en la propaganda y el proselitismo de sus ideas” [56].

**Adopción:** “La adopción es el evento mediante el cual una persona mayor de edad que cumple ciertos requisitos (afecto por los animales, tiempo, espacio físico y dinero), escoge una mascota como compañero de actividades y parte de su familia” [57].

**Albergue:** Lugar donde se da resguardo provisional a “una persona o un animal” [58].

**Campañas:** Actividades que se realizan con el fin de dar información o hacer partícipes a las personas que estén interesadas.

**Donaciones:** “Liberalidad de alguien que transmite gratuitamente algo que le pertenece a favor de otra persona que lo acepta” [59].

**Esterilización:** Es un método quirúrgico que evita la propagación de enfermedades hereditarias como también disminuir la reproducción de animales no deseados. “La esterilización canina corresponde al método más eficiente y de resultados positivos en el control y reducción de la sobrepoblación canina en gran escala” [60].

**Fauna callejera:** Es aquel que no tiene dueño conocido, que se encuentre solo sin compañía de persona alguna que pueda demostrar su propiedad o custodia que atente contra su dignidad y sus derechos [61].

**Fundación:** Es una organización sin ánimo de lucro que se crea con el fin de realizar actividades de índole sociocultural [62].

**Hogar de paso:** Son todos aquellos lugares en donde un animalista o activista brinda por determinado tiempo vivienda y comida a un animal doméstico en condiciones de abandono.

**Mascota:** Animal doméstico de compañía. [63]

**Movimiento Animalista:** Aquellas personas que están a favor de la fauna callejera, buscando la manera de dar solución a esa problemática que les aflige [64].

**Daily:** Reuniones diarias entre el equipo de trabajo para determinar el progreso del *Sprint*.

**Development team:** Son los encargados de la implementación software, con características de organización, multifuncionales y auto organizados [65].

**Product backlog:** Lista de ítems de priorización para el desarrollo de un producto que pueden ser modificados en su orden por el *Product owner*.

**Product owner:** Se ocupa de priorizar la lista del producto, de esa manera ordena y optimiza lo que se va a realizar durante el *sprint* de forma que sea entendible para el equipo de desarrollo.

**Scrum:** Es un entorno de trabajo que permite enfrentarse a problemas complejos es decir emplea varias técnicas y procesos para el control de estos, basándose en el empirismo que tienen las personas para tomar decisiones en la gestión de un proyecto [65].

**Scrum master:** Encargado de verificar que el entorno de trabajo se cumpla a cabalidad ayudando tanto al *Product owner* como al equipo de desarrollo a regirse bajo *Scrum*.

**Sprint:** Tiempo fijo en el que se desarrollan unas actividades previamente definidas en el desarrollo de producto software [66].

**Iteración:** “Que indica repetición” [67], es un proceso repetitivo de instrucciones.

**Desarrollo:** Fase de la metodología de extreme programming en la que se divide las funcionalidades en pequeñas unidades, en el que se realiza programación en parejas a fin de integrarse nuevamente.

**Diseño:** Fase de la metodología extreme programming en la que se definen las funcionalidades y arquitectura del sistema.

**Framework:** “Es un conjunto de archivos y directorios que facilitan la creación de aplicaciones, puesto que incorpora funcionalidades ya desarrolladas, probadas e implementadas en un determinado lenguaje de programación” [68]

**MVC:** “Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos” [69].

**Metodología:** “Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal” [70].

**Planificación:** Fase de la metodología extreme programming que permite determinar el objetivo, integrantes del equipo de trabajo, tiempo y cronograma del proyecto.

**Prueba:** Fase de la metodología extreme programming en la que se realizan pruebas de funcionalidad que permiten identificar fallas del programa.

**Test:** “Prueba destinada a evaluar conocimientos o aptitudes, en la que se elige la respuesta correcta entre varias opciones previamente fijadas” [71], test son pruebas a las que se somete una persona, producto software, entre otros para verificar conocimiento y funcionalidad respectivamente.

**XP:** Metodología ágil que promueve el trabajo en equipo, realimentación continua y comunicación entre el cliente y el equipo de desarrollo. El cual consta de las siguientes fases planificación, diseño, desarrollo y pruebas.

**Plataforma web:** “Es una aplicación informática que se ejecuta en un entorno Web. La arquitectura Web permite a múltiples clientes o usuarios establecer una comunicación a través de Internet con un servidor Web” [72].

**Uml:** Notaciones gráficas que describen el desarrollo de un sistema software.

### 3. METODOLOGÍA

La implementación de la plataforma web se realizó utilizando algunas prácticas de *Scrum* como entorno de trabajo para la gestión del proyecto, que permitió motivar a los integrantes del grupo, debido al enfoque en la documentación necesaria, entregas continuas y funcionales, reuniones diarias de sincronización del equipo y adaptabilidad a los cambios, asimismo para el manejo del proyecto software se utilizó la metodología ágil *Extreme programming* porque promueve el trabajo en equipo, realimentación continua y comunicación entre el cliente y el equipo de desarrollo. Esta metodología se aplicó para los cuatro *sprint* del proyecto.

#### 3.1 IMPLEMENTACIÓN DE SCRUM

Para el desarrollo del proyecto se hizo necesario al finalizar cada Sprint, una reunión que permitiera al *Development team* y el *product owner* priorizar el *product backlog* y estimar la cantidad de historias de usuario que se comprometieron a desarrollar por sprint.

A continuación se presentarán los *Sprint* determinados por la reunión del *sprint planning*:

##### 3.1.1 Planificación Sprint 0

###### 3.1.1.1 Inception

Para dar inicio al Sprint 0 se estableció quienes serían parte del *Scrum Team* por esta razón, se realizó una asignación de roles ya que permite tener claridad en los objetivos, organización del trabajo, entregas frecuentes de funcionalidades, reuniones continuas para reflexionar y visualizar el progreso del desarrollo.

- **Asignación de roles**

**Tabla 2. Product owner (dueño del producto)**

<b>Nombre</b>	Carlos Andrés Duran
<b>Rol</b>	Product owner
<b>Responsable</b>	Validar producto final.

**Tabla 3. Team development (equipo) desarrolladora e investigadora**

<b>Nombre</b>	Darly Vannessa Fernández León
<b>Rol</b>	Investigadora principal
<b>Responsable</b>	Desarrolladora e Investigadora

**Tabla 4. Team development (equipo) desarrolladora e investigadora**

<b>Nombre</b>	Angela María Muñoz García
<b>Rol</b>	Investigadora principal
<b>Responsable</b>	Desarrolladora e Investigadora

**Tabla 5. Scrum Master (Director del proyecto)**

<b>Nombre</b>	Ing. Julián Bermúdez Trujillo
<b>Rol</b>	Scrum Master
<b>Responsable</b>	Verificar el cumplimiento de las prácticas aplicadas de <i>Scrum</i>

- **Product Backlog**

Para el desarrollo del producto software se realizó un lista priorizada de las características a implementar en la herramienta *Trello*, que permitió buscar el *MVP* (Mínimo, viable y producto), al priorizar la lista se obtuvieron las funcionalidades con mayor valor; primera versión del producto con la especificación de las historias de usuario tiempo, peso y actividades a desarrollar, así mismo estimar una inversión económica que sea viable para el desarrollo del proyecto, por tanto en la preparación del inception se respondió a los siguientes interrogantes:

- ¿Porque estamos aquí?
- Elevator Pitch
- Diseña tu caja
- Crea tu not list
- ¿Qué te quita el sueño?
- Calcula un tamaño
- ¿Cuáles son tus prioridades?





Figura 16. Product backlog [Fuente propia]

### 3.1.2 Planificación Sprint 1

Este sprint lo conforman 6 historias de usuario que hacen parte del *Frontend* y *Backend* con una estimación de 14 días laborables de 8 horas de trabajo, para este *Sprint* son:

- Ingreso el sistema
- Registro e ingreso
- Registro de Usuario
- Guardar Usuario
- Registro de Animales
- Guardar Animales

### **3.1.3 Planificación Sprint 2**

Este sprint lo constituyen 9 historias de usuario que hacen parte del *Frontend* y *Backend* con una estimación de 15 días laborables de 8 horas de trabajo, para este *Sprint* son:

- Registro de Consulta
- Guardar Consulta
- Registro de Vacunación
- Guardar Vacunación
- Registro de Esterilización
- Guardar Esterilización
- Registro de proceso
- Guardar proceso
- Registro de eventos

### **3.1.4 Planificación Sprint 3**

Este sprint lo componen 7 historias de usuario que hacen parte del *Frontend* y *Backend* con una estimación de 16 días laborables de 8 horas de trabajo, para este *Sprint* son:

- Guardar eventos

- Registro de Productos
- Guardar Productos
- Registro de Productos Zoosanitarios
- Guardar Productos Zoosanitarios
- Seleccionar Indicador
- Seleccionar Gráfica indicador

### **3.1.5 Planeación Daily**

Las reuniones se hicieron de manera presencial con una duración de 15 minutos que proporcionó conocer el progreso, velocidad, fallos y contratiempo en el desarrollo propiciando una retroalimentación continua, permitiendo así evitar bloqueos en el equipo con un progreso continuo en el desarrollo de cada sprint.

### **3.1.6 Sprint Review**

Los *sprint review* se realizaron con los integrantes de la Fundación “VAP”, con el fin de socializar las actividades en el sprint finalizado, exponiendo características y funcionalidades propias del *Sprint* a presentar, a continuación se da a conocer una carta de contextualización de uno de los *Sprint review* realizados.

Popayán, 24 Junio de 2016

Señores  
Fundación Voluntarios Animalistas Popayán  
Popayán (Cauca)

Cordial Saludo

Se inicia la presentación a las 6:30PM el día 24 de Junio de 2016 socializándose el proyecto de grado titulado "Desarrollo de una plataforma web para la gestión de los procesos realizados por los comités de la Fundación voluntarios animalistas Popayán" por Angela María Muñoz García y Darly Vanessa Fernández León , a los integrantes de los comités de la fundación, abarcando los temas de objetivos y funcionalidad para este proyecto, dando inicio a la etapa de pruebas.

Se finalizó la presentación con la contextualización del proyecto y para adquirir el grado de comprensión de lo expuesto se hace un foro con algunas preguntas realizados por parte de los representantes de la fundación.

En constancia firma:

Carlos A. Durán  
1.061.729.079. Pop.

Carlos Andrés Duran  
Secretario

Figura 17. Carta sprint review [Fuente propia]

### 3.1.7 Retrospectiva

La retrospectiva se realizó evaluando tiempos y peso de cada historia de usuario con el fin de estimar el tiempo realmente usado para la implementación de la historia, por otra parte al aplicar los valores ágiles, se hizo una reunión para exponer abiertamente cuáles fueron los inconvenientes, fallas encontradas,

historias de usuario no culminadas y elección incorrecta de herramientas finalizando cada uno de los sprint.

### **3.2 IMPLEMENTACIÓN EXTREME PROGRAMMING**

Para dar inicio al proyecto software se realizaron una serie de reuniones con los representantes de la “Fundación Voluntarios Animalistas Popayán”, en las que se expusieron las características que el software debería poseer, permitiendo así comprender la problemática presentada para la generación de ideas que proporcionen una solución tecnológica, a continuación en la tabla se presenta *Lean Canvas* del proyecto “VAP”.

**Tabla 6. Lean canvas del Proyecto VAP**

<b>LEAN CANVAS</b>				
<b>PROBLEMA</b>	<b>SOLUCIÓN</b>	<b>PROPUESTA DE VALOR ÚNICA</b>	<b>CANALES</b>	<b>SEGMENTO DE CLIENTES</b>
<ul style="list-style-type: none"> <li>• Duplicidad de información.</li> <li>• Pérdida de información.</li> </ul>	Desarrollo de una plataforma web para la gestión de procesos en la Fundación Voluntarios Animalistas Popayán.	Gestionar los procesos de la fundación de manera centralizada.	<ul style="list-style-type: none"> <li>• Voz a Voz.</li> <li>• Redes sociales.</li> <li>• Publicidad en empresas.</li> </ul>	<ul style="list-style-type: none"> <li>• Clínicas veterinarias.</li> <li>• Fundaciones protectoras de animales.</li> <li>• Secretaria de salud Zoonosis.</li> <li>• Policía Ambiental.</li> <li>• Protectores de animales</li> </ul>
<b>ALTERNATIVAS</b>				
<ul style="list-style-type: none"> <li>• Internet.</li> <li>• Redes Sociales.</li> </ul>				
<b>ESTRUCTURA DE COSTOS</b>			<b>FLUJO DE INGRESOS</b>	
<ul style="list-style-type: none"> <li>• Internet, celulares, servidor, mano de obra, portátiles.</li> </ul>			<ul style="list-style-type: none"> <li>• Mensualidad por publicidad en el aplicativo web.</li> </ul>	

### 3.2.1 Planificación

Dando inicio a la ejecución de las fases definidas por la metodología XP, en la primera fase a través de la definición de las historias de usuario descritas por el cliente, se realizaron una serie de entrevistas en las que se definieron los procesos actualmente empleados para la captura, manejo y visualización de la información, permitiendo definir su finalidad, tiempo, integrantes del equipo de trabajo y estimación del cronograma de entrega a lo largo del proyecto, como fechas de entrega y publicación evaluando el cumplimiento de objetivos [73], por cada historia de usuario se estimó un peso y tamaño usando la herramienta *planning póker* permitiendo evaluar puntos de historia con la baraja de cartas de *poker* con valores como 0, 1, 2, 3, 5, 8, 13, 20, 40 y 100 de acuerdo a la experiencia de los desarrolladores se escoge un valor que representa el peso de cada una de las historias de usuario; también se utilizó la herramienta *Trello* que sirve para el manejo de las tareas e identificar la priorización de actividades teniendo en cuenta los puntos de historia evaluados anteriormente, llevando un control adecuado de la implementación que se va realizando en el transcurso del proyecto [74].

Por otro lado XP recomienda la programación en parejas y reuniones diarias, por esta razón se empleó el repositorio *Bitbucket* para tener un control de versiones funcionales de acuerdo a las modificaciones realizadas al código por cada uno de los integrantes; las reuniones diarias se realizaron de manera presencial con una duración de 15 minutos conociendo así el progreso, velocidad y fallas en el desarrollo propiciando una retroalimentación continua en el equipo de desarrollo.

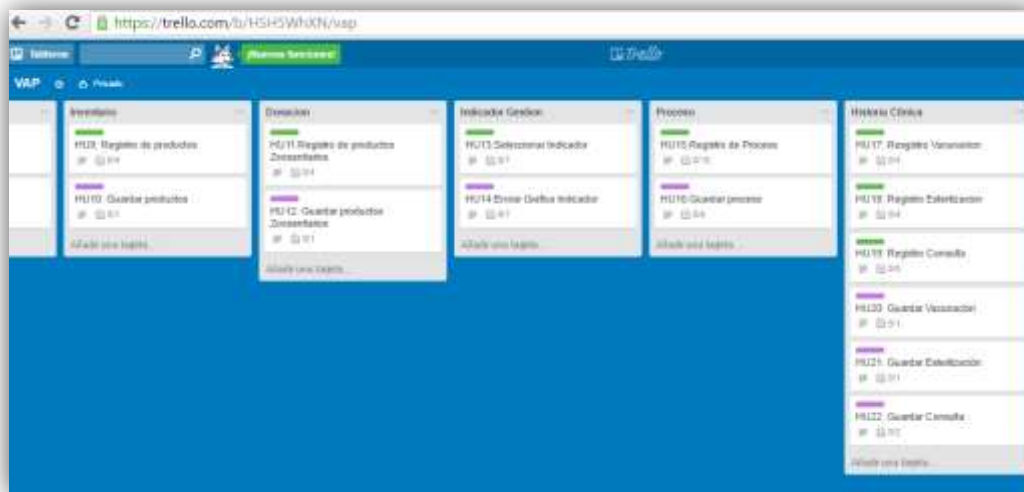


Figura 18. Tablero Kanban [Fuente propia]

### 3.2.2 Diseño

En la segunda fase se definieron las funcionalidades y la arquitectura del proyecto con un diseño simple facilitando dar un aproximado de la solución final, en esta fase se utilizó *UML* que “es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de *software*” [75] y como repositorio documental *Google Drive* para el manejo y control de versiones. Finalmente se aplicó la recodificación de extensiones brindadas por el *framework Yii* que se ajustaron a las necesidades de cada módulo a desarrollar.

### 3.2.3 Desarrollo

En la fase del desarrollo para la creación de la plataforma web se utilizaron herramientas como *mysql* que es un gestor de base de datos que respalda el modelo, vista y controlador; *php* como lenguaje de programación creado para el desarrollo web [76] junto con el uso del *Framework Yii* de *Php* que facilitó implementar la plataforma web debido a su escalabilidad ya que utiliza el modelo



de capas MVC permitiendo gestionar las reglas del negocio, interacciones a la base de datos y las interfaces de usuario creando así el *backend* y *frontend* en paralelo.

Se utilizó Apache para los servicios de *HTTP* y *JSON* que para gestionar, identificar e intercambiar información en distintas tecnologías siendo fácil de usar en *javascript* y reduciendo el tiempo de respuesta por parte del servidor [77].

Por consiguiente para poder manejar los lenguajes de programación, entornos de trabajo y librerías se utilizó el editor de texto *Sublime* [78] pues facilitó la codificación e implementación de productos software, que maneja una interfaz agradable, tiene una gran gama de plugins, es gratuito y posee una paleta de comandos para facilitar la codificación.

Por otra parte para el manejo de la capa de presentación se empleó *BOOTSTRAP* puesto que es adaptable para cualquier tipo de proyecto y provee estilos aplicables a la interfaz gráfica adaptativa es decir amigable para el usuario final, permitiendo así realizar una distribución adecuada de los contenidos a visualizar, también está incorporado por *HTML* y *CSS*. Con *HTML* se crea un esquema conformado por etiquetas para dar estructura a la páginas web, entre sus etiquetas se encuentran los scripts con funciones específicas delimitados con lenguajes como lo son *Javascript* y *Php*, adicionalmente *CSS* se utilizó para dar estilo a la plataforma web es decir que cumpla con los estándares de los colores, tipos de letra, tamaño, separaciones verticales y horizontales entre elementos definidos por *Bootstrap* [79] y para la creación del juego de colores se utilizó la aplicación en línea *Adobe Kuler* que es una herramienta que facilita escoger adecuadamente la gama de colores para el proyecto, es decir que estos sean agradables para lograr un impacto visual.

### 3.2.4 Pruebas

En la fase de pruebas se utilizó la herramienta *Postman* que es una herramienta que se utiliza con el propósito de sincronizar dispositivos comprobando las peticiones al *Backend*, además de las pruebas de caja blanca y negra ya que utilizando el entorno de trabajo *Scrum* se hace necesario realizar entregas periódicas funcionales, que motiven al equipo de trabajo y al cliente con los resultados obtenidos en cada tarea (pruebas de aceptación), posteriormente para tener un control de versiones funcionales se empleó la herramienta *Bitbucket*, una de las principales ventajas del uso de esta herramienta es la de permitir al grupo de desarrollo poder enviar y observar cambios realizados al código, sin la necesidad de disponer del tiempo de los desarrolladores para poder continuar con la implementación; igualmente en *Scrum* el *Development team* realizó estimaciones de tiempo sobre el sprint y para esto se utilizó la herramienta *WakaTime* para conocer el rendimiento de cada uno de los integrantes con las tareas asignadas a través de gráficas y estadísticas que motivan a la mejora continua en el proceso de desarrollo evaluando calidad, tiempo y costo.

## **4. INGENIERÍA DEL PROYECTO**

### **4.1 IMPLEMENTACIÓN SPRINT 0**

#### **4.1.1 Diagrama bases de datos**

En la figura 19 se presenta el diagrama de base de datos de la plataforma web para la fundación “Voluntarios Animalistas Popayán”.

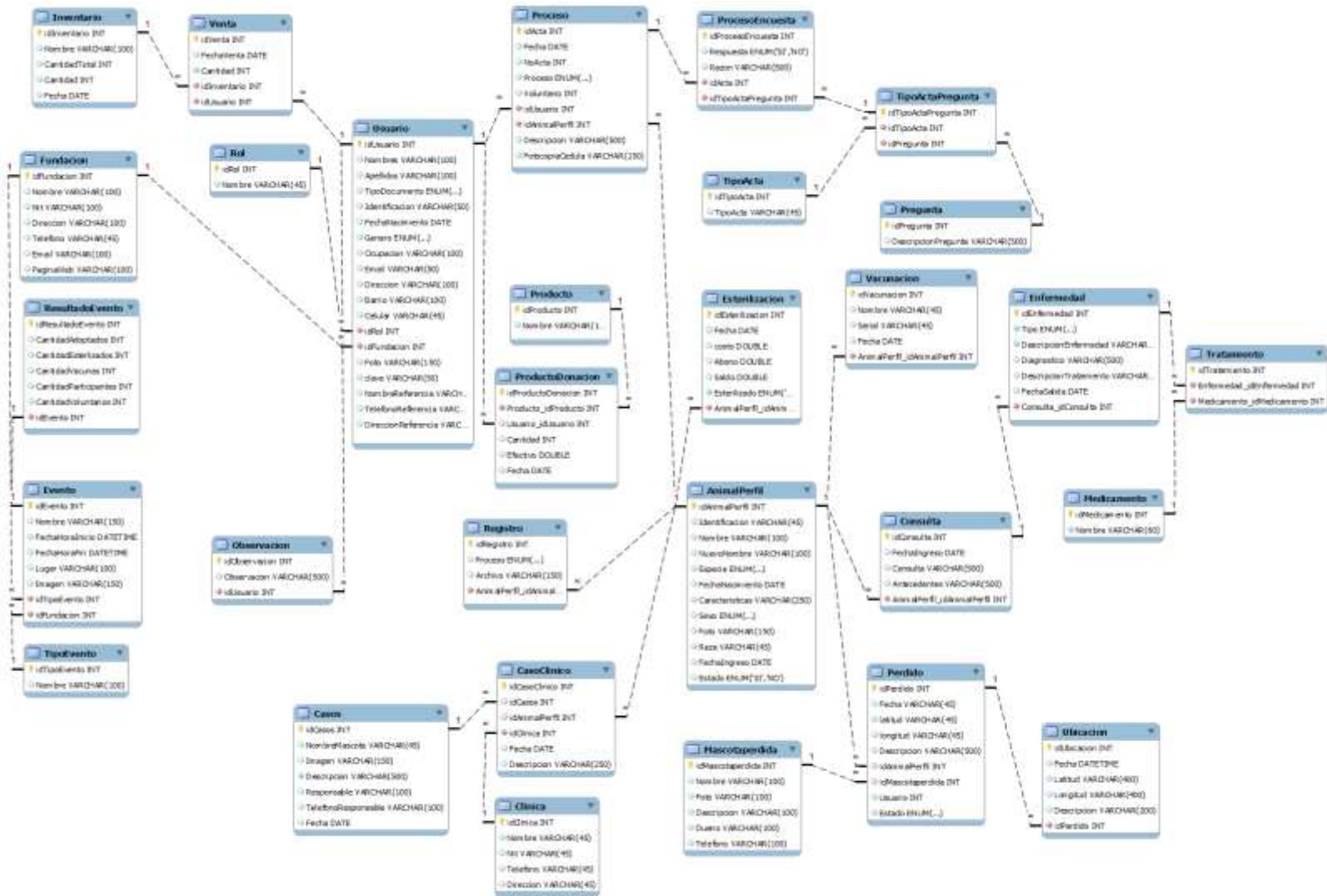


Figura 19. Diagrama de bases de datos [Fuente propia]

#### **4.1.2 Diagrama de Clases**

En la figura 20 se presenta el diagrama de clases de la plataforma web para la fundación “Voluntarios Animalistas Popayán”.

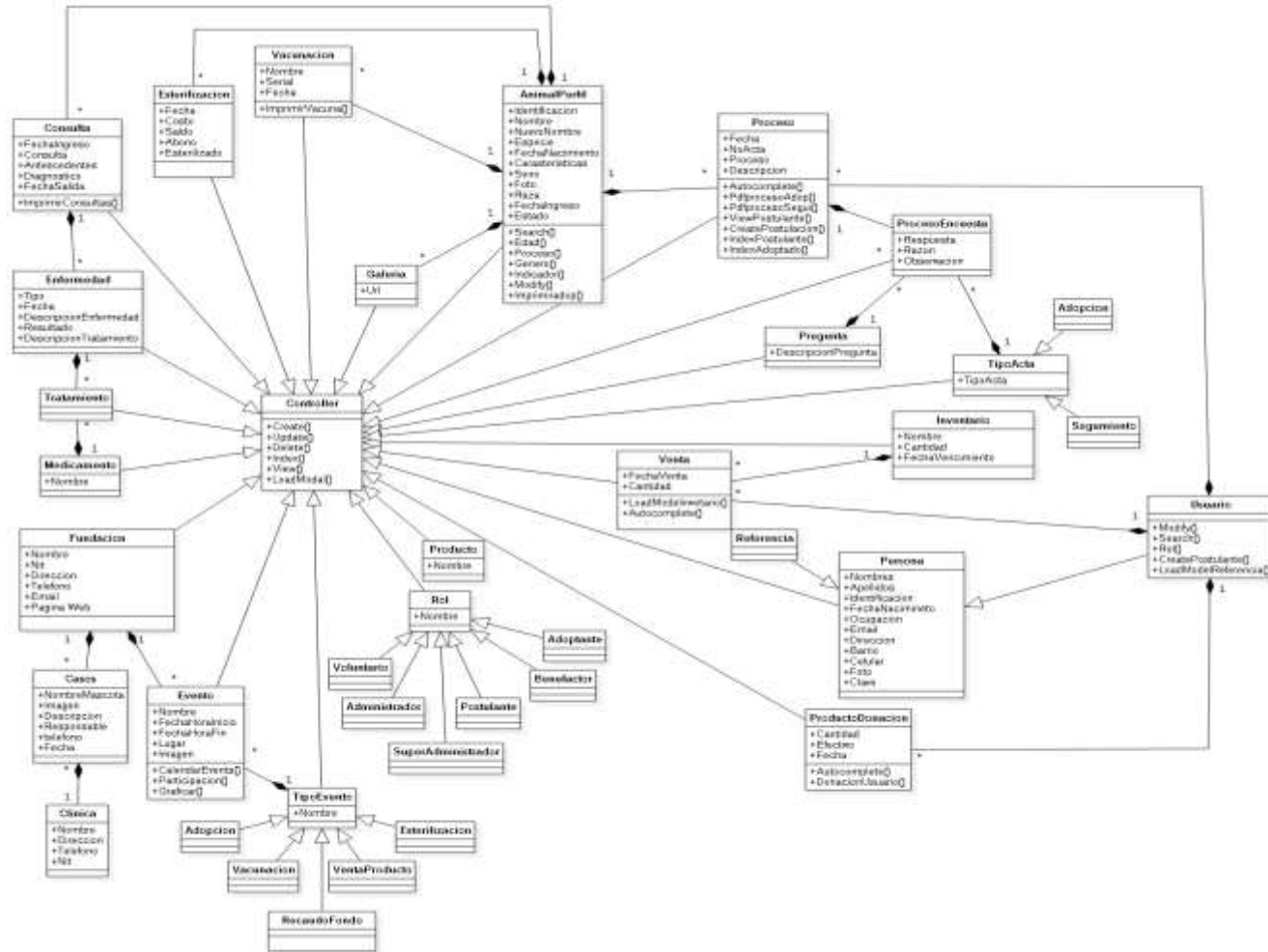


Figura 20. Diagrama de clases [Fuente propia]

### 4.1.3 Arquitectura del Sistema

La arquitectura empleada como soporte para el desarrollo de la plataforma web fue la arquitectura cliente/servidor, ya que permite el manejo de la concurrencia, seguridad y autonomía cumpliendo con las funcionalidades del producto software.

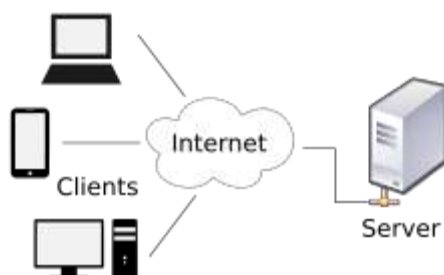


Figura 21. Arquitectura cliente servidor

[80]

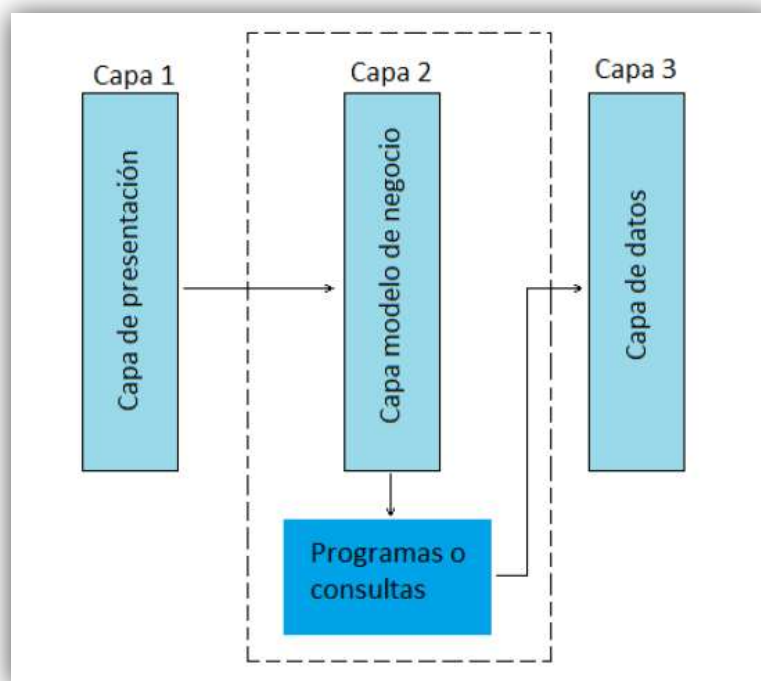
#### 4.1.3.1 Cliente

Los clientes en la arquitectura planteada para el desarrollo de este proyecto están representados por los integrantes de la fundación y la comunidad caucana interesada que a través de la plataforma web realizan consultas y búsquedas de información accediendo por medio de dispositivos móviles, tablets, portátiles y computadores.

#### 4.1.3.2 Servidor

En la arquitectura cliente servidor tanto los clientes como los servidores son conceptos abordados a nivel software, entre tanto el cliente es definido como el proceso que requiere un determinado servicio y el servidor es el proceso que valida la solicitud y proporciona el servicio que requiere el cliente, por ello el servidor brinda la comunicación entre el cliente y la información comprendida en el

sistema de información o aplicativo por medio del diseño web como *HTML*, *CSS* y *JavaScript* permite que los clientes trabajen en la interfaz gráfica del aplicativo *GUI* mientras el servidor se focaliza en realizar procesos centrales como almacenar información del aplicativo, peticiones de información y obtener información comprendida en el aplicativo usando el lenguaje de programación PHP y sentencias SQL para el acceso a la base de datos MYSQL que es usada en el proyecto actual.



**Figura 22. Capas de la aplicación**

[81]

#### **4.1.4 Implementación Inception**

Esta técnica permitió clarificar las estrategias construyendo así una visión completa sobre el concepto del producto a realizar, en esta clarificación participaron las personas implicadas en el proyecto, obteniendo el producto que se desea llevar a cabo con el análisis de las siguientes preguntas.



#### **4.1.4.1 ¿Porque se está aquí? ¿Por qué se está haciendo este producto?**

Siendo las 9:31 am del día 01 de febrero de 2016 para proveer una solución software que facilite a los integrantes de los comités de la “Fundación Voluntarios Animalistas Popayán” la gestión de los procesos internos y a la comunidad caucana obtener información acerca de la fundación. La plataforma debe ofrecer información acerca de la fundación, gestionar los proceso de adopción, seguimiento y postulación realizados por el comité de adopción y bienestar animal de la fundación “VAP”, además debe ofrecer información de los eventos que se realizaron y realizaran por la fundación, igualmente generar indicadores gráficos por cada comité.

#### **4.1.4.2 Elavator Pitch - ¿Cómo defender el proyecto, si se tiene un tiempo determinado?**

Es un producto software para facilitar la gestión de los proceso internos de los comités de adopción, bienestar animal, comunicación y donación realizados por los representantes de cada comité de la fundación “VAP”, por medio de una plataforma web que apoyé la finalidad de la fundación como rescatar, rehabilitar y dar en adopción animales domésticos en estado de abandono en la ciudad de Popayán.

#### **4.1.4.3 Not List**

**IN:**

- ✓ Adoptar animales domésticos.
- ✓ Inventario de productos donados.
- ✓ Ventas de productos de la fundación
- ✓ Historia clínica de las mascotas del hogar de paso
- ✓ Visualizar eventos de la fundación.
- ✓ Gestión de usuarios.

## OUT:

- ✓ Pasarela de pagos.

## DISCUTIR:

- ✓ Videos sobre casos clínicos de las mascotas.

### 4.1.4.4 Que te quita el sueño

- ¿Cuáles son los miedos?
  - ✓ La fundación no utilice la plataforma web
  - ✓ La no aceptación de la plataforma web
- ¿Cómo evitarlo?
  - ✓ La interacción del usuario con la aplicación deberá ser sencilla.
  - ✓ Garantizar al usuario final la confidencialidad de su información en la aplicación.
- ¿Cómo minimizar el riesgo?
  - ✓ La funcionalidad de la aplicación deberá permitir tener un usuario recurrente.
  - ✓ El usuario final encuentre lo que necesita en la aplicación.
- ¿Qué nos mantiene despiertos en las noches?
  - ✓ Riesgo1: No cumplir con las funcionalidades establecidas para la plataforma web.
  - ✓ Riesgo 2: El tiempo no sea suficiente para la ejecución del proyecto.
  - ✓ Riesgo 3: El cliente no quede a gusto con el producto desarrollado.
  - ✓ Impedimento1: Surjan modificaciones en la plataforma web que tomen más del tiempo establecido.
  - ✓ Impedimento 2: El equipo de trabajo no engranen su potencial intelectual para tener un desarrollo del proyecto equilibrado.

- ✓ Impedimento 3: Los recursos (herramientas) no sean suficientes para el equipo de trabajo.

#### 4.1.4.5 Diseña tu caja



Figura 23. Diseña tu caja [Fuente propia]

#### 4.1.4.6 Calcular un tamaño

Tabla 7. Cronograma

ACTIVIDADES	SEMANAS																			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Elección del Tema	■																			
Reunión con la Fundación		■																		
Selección entorno de trabajo			■																	
Formulación del Problema			■																	
Revisiones Bibliográficas				■	■	■	■	■	■	■	■	■	■	■	■	■				
Referencias Bibliográficas				■	■															
Estado del Arte o Marco Referencial					■	■														
Marco Teórico						■	■													
Selección de Metodología desarrollo								■												
Desarrollo Aplicación								■	■	■	■	■	■	■	■	■	■	■	■	■
Elaboración del Artículo de Investigación																	■	■	■	
Sustentación del Trabajo																				■
Publicación del Trabajo																				■
<b>Desarrollo Aplicación - Mediante Ágiles XP</b>																				
Planificación								■				■				■				■
Diseño									■			■				■				■
Desarrollo									■	■		■	■		■	■			■	■
Pruebas											■				■			■		■

#### 4.1.4.7 Cuáles son las prioridades

- Entregar un producto de calidad.
- Cumplir con los requisitos establecidos.
- Proporcionar el producto en los tiempos acordados.

- Tener un buen equipo de trabajo.
- Contar con los recursos necesarios.
- Posicionar la marca del producto realizado.
- Obtener un renombre como empresa o grupo de trabajo al desarrollar productos de calidad.



#### 4.1.4.8 Sprint Review



Finalizando el sprint 0 se presentó la planeación a realizar durante el proyecto a los integrantes de la fundación junto con el *product owner*, planteando soluciones frente a riesgos observados y las funcionalidades a desarrollar para el producto final, priorizar las historias de usuario estimadas por el product owner permitió planificar las historias de usuario por cada *Sprint* estimando tiempo, presupuesto y capacidades para cumplir con la entrega final en cada *Sprint*.

#### 4.1.4.9 Retrospectiva

Para las retrospectivas se aplicó la técnica matriz de aprendizaje puesto que permitió inspeccionar herramientas, procesos y personas identificando los elementos que salieron bien y mal creando un plan de mejoras para la ejecución del siguiente *Sprint*.

Tabla 8. Retrospectiva Sprint 0

 <ul style="list-style-type: none"> <li>• Acordar reuniones con la fundación.</li> <li>• Disponibilidad de la fundación para trabajar.</li> <li>• Elegir framework Yii para el desarrollo.</li> <li>• Elegir entorno de trabajo Scrum</li> </ul>	 <ul style="list-style-type: none"> <li>• Priorización historias de usuario.</li> <li>• Modelado base de datos.</li> <li>• Estimación de peso y tamaño por historia de usuario</li> </ul>
---	--

<ul style="list-style-type: none"> <li>• Escoger la metodología de desarrollo Extreme Programming.</li> <li>• Usar Trello</li> <li>• Modelo negocio <i>Lean Canvas</i></li> </ul>	
 <ul style="list-style-type: none"> <li>• Galería de imágenes.</li> <li>• Eventos en un calendario.</li> <li>• Inventario de productos.</li> <li>• Plataforma web.</li> <li>• Indicadores gráficos de Gestión</li> </ul>	 <ul style="list-style-type: none"> <li>• Agradecimiento a la fundación por brindar información y tiempo para el desarrollo de la aplicación.</li> </ul>

## 4.2 IMPLEMENTACIÓN SPRINT 1

Para el sprint 1 se aplicó un proceso de planeación, diseño, desarrollo y pruebas con el fin de desarrollar 6 historias de usuario con una duración de 8 horas diarias en 14 días, ejecutando la primera iteración obteniendo un producto mínimo.

Según el product backlog las historias de usuario asignadas al sprint 1 son:

- Ingreso al sistema
- Registro e ingreso
- Registro de Usuario
- Guardar Usuario
- Registro de Animales
- Guardar Animales

### 4.2.1 Planeación

A continuación se observan las historias de usuario, task card (tareas de ingeniería) y tarjetas de colaboración del *Sprint* 1.

**Tabla 9. Historia de usuario: ingreso al sistema**

<b>Historia de usuario</b>	
<b>Número:</b> HU1	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Ingreso al sistema	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 8 puntos de historia	
<b>Tiempo estimado:</b> 16 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario	
<b>Se requiere:</b> Acceder al sistema	
<b>Para:</b> Que los usuarios ingresen información en la plataforma	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• El email del usuario debe ser único.</li> <li>• Los usuarios pueden tener la misma contraseña.</li> <li>• Los usuarios deben estar validados antes de ingresar a la plataforma.</li> <li>• Recuperar contraseña por medio de código de validación enviado al email.</li> <li>• Dependiendo del rol de usuario que ingrese se habilitarán las diferentes opciones del menú</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaz gráfica de inicio de sesión</li> <li>• Conexión al servidor.</li> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> <li>• Recuperación de contraseña.</li> </ul>	<p><b>3</b></p> <p><b>4</b></p> <p><b>3</b></p> <p><b>6</b></p>

Tabla 10. Task card inicio de sesión

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU1:</b> Ingreso al sistema
<b>Nombre de Tarea:</b> Interfaz gráfica de inicio de sesión	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para iniciar sesión en la plataforma web, con los campos email y contraseña.	

Tabla 11. Task card conexión servidor

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU1:</b> Ingreso al sistema
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso el ingreso al sistema.	



Tabla 12. Task card validar campos

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU1:</b> Ingreso al sistema
<b>Nombre de Tarea:</b> Validar Campos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, etc)	

Tabla 13. Task card recuperación de contraseña

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU1:</b> Ingreso al sistema
<b>Nombre de Tarea:</b> Recuperación de contraseña	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Si selecciona la opción recuperar contraseña, digitar el correo electrónico ingresado en la plataforma para la recibir el código de validación.	

**Tabla 14. Tarjeta de colaboración Usuario**

<b>Usuario</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Authenticate</li> <li>• AuthManager</li> <li>• Crear usuario</li> <li>• Editar usuario</li> <li>• Eliminar usuario</li> <li>• Ver usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• UserIdentity</li> <li>• Persona</li> <li>• Referencia</li> <li>• Rol</li> </ul>

**Tabla 15. Historia de usuario registro e ingreso**

<b>Número:</b> HU2
<b>Programador Responsable:</b> Team Development
<b>Nombre de la historia:</b> Registro e ingreso
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 13 puntos de historia
<b>Tiempo estimado:</b> 20 Horas
<b>Perspectiva del producto</b>
<p><b>Como:</b> Servidor</p> <p><b>Se requiere:</b> Responder a la petición de acceso al sistema</p> <p><b>Para:</b> Que los usuarios ingresen información en la plataforma</p>
<b>Criterios de aceptación</b>
<ul style="list-style-type: none"> <li>• El usuario reciba la notificación del código de validación de restablecer contraseña.</li> <li>• Creación la entidad Usuario en la base de datos.</li> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Exista el usuario en el sistema.</li> <li>• Dependiendo del rol de usuario que ingrese se habilitaran las diferentes opciones del menú.</li> </ul>

<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Configuración de framework</li> <li>• Crear en la base de datos el atributo Email y contraseña en la tabla Usuario</li> <li>• Validar rol de usuario</li> <li>• Guardar Log de sesión de usuarios</li> <li>• Código de validación enviado al email</li> </ul>	<b>6</b> <b>4</b> <b>2</b> <b>2</b> <b>6</b>

**Tabla 16. Task card configuración framework**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU2:</b> Registro e Ingreso
<b>Nombre de Tarea:</b> Configuración Framework	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Se configura el Framework YII con la aplicación web para la gestión de información	

**Tabla 17. Task card crear atributos en la entidad usuario**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU2:</b> Registro e Ingreso
<b>Nombre de Tarea:</b> Crear en la base de datos el atributo Email y contraseña en la tabla Usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	

<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la base de datos con los respectivos atributos de la tabla Usuario del modelo Entidad-Relación y la conexión de la base de datos con el aplicativo web.	

Tabla 18. Task card validar rol de usuario

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU2:</b> Registro e Ingreso
<b>Nombre de Tarea:</b> Validar Rol de usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Cada usuario en el sistema está registrado con un rol de usuario y dependiendo de éste podrá visualizar las opciones que se le permitirá gestionar. Los eventos son visibles para usuarios internos y externos a la plataforma pero los únicos que pueden publicar eventos son los usuarios con rol de administrador y súper administrador. Si el usuario ingresa con el rol de SuperAdministrador, Administrador y voluntario podrá ingresar al menú de usuarios, animales, productos, inventario para crear, editar, visualizar y eliminar (a excepción del voluntario), en cada uno de los menús anteriormente nombrados, por otra parte los usuarios con el rol postulante al ingresar al sistema solo podrán visualizar y editar su información, también tendrá acceso a la galería de mascotas que aún no han sido adoptadas y podrá ver el listado de los animales a los que se ha postulado. Igualmente los usuarios que ingresen al sistema con un rol de benefactor podrán observar las donaciones que ha realizado a través de gráficas o un resumen si lo han realizado donaciones en efectivo, también tendrá acceso a la galería de mascotas. Finalmente los usuarios logueados con el rol de adoptante podrá tener acceso a la información de las mascotas bajo sobre su protección, además pueden ver datos	

sobre las donaciones y ventas que hayan realizado.

Tabla 19. Task card log sesión de usuarios

TAREA DE INGENIERÍA	
<b>Número de Tarea:</b> 4	<b>HU2:</b> Registro e Ingreso
<b>Nombre de Tarea:</b> Guardar Log de sesión de usuarios	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Mantener la sesión activa identificando el usuario.	

Tabla 20. Task card código de validación enviado al email

TAREA DE INGENIERÍA	
<b>Número de Tarea:</b> 5	<b>HU2:</b> Registro e Ingreso
<b>Nombre de Tarea:</b> Código de validación enviado al email	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Se le notifica un código de validación al solicitante a través del correo electrónico, de manera que podrá acceder de nuevo al sistema.	

Tabla 21. Tarjeta de colaboración HU2 registro e ingreso

<b>Usuario</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Authenticate</li> <li>• AuthManager</li> <li>• Crear usuario</li> <li>• Editar usuario</li> <li>• Eliminar usuario</li> <li>• Ver usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• UserIdentity</li> <li>• Persona</li> <li>• Referencia</li> <li>• Rol</li> </ul>

Tabla 22. Historia de usuario registro de usuario

<b>Historia de usuario</b>
<b>Número:</b> HU5
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Registro de Usuario
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 13 puntos de historia
<b>Tiempo estimado:</b> 20 Horas
<b>Perspectiva del producto</b>
<p><b>Como:</b> Usuario</p> <p><b>Se requiere:</b> Acceder al sistema</p> <p><b>Para:</b> Que los usuarios ingresen información en la plataforma</p>
<b>Criterios de aceptación</b>
<ul style="list-style-type: none"> <li>• El email del usuario debe ser único.</li> <li>• Los usuarios pueden tener la misma contraseña.</li> <li>• El número de identificación debe ser único.</li> <li>• El rol de usuario, referencia, foto, dirección, teléfono, email, contraseña debe ser obligatorios.</li> </ul> <p><b>Restricción:</b></p> <ul style="list-style-type: none"> <li>• Los usuarios solo podrán ser creados, eliminados por el usuario bajo el rol</li> </ul>

de Superadministrador o administrador, a excepción del postulante que para postularse a una mascota deberá digitar su información y luego finalizar ese proceso.	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> <li>• Crear interfaces gráficas del CRUD para gestionar Usuario</li> <li>• Conexión al servidor.</li> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> <li>• Seleccionar Búsqueda por identificación</li> <li>• Seleccionar búsqueda por rol</li> </ul>	<p style="text-align: center;"><b>5</b></p> <p style="text-align: center;"><b>4</b></p> <p style="text-align: center;"><b>4</b></p> <p style="text-align: center;"><b>3</b></p> <p style="text-align: center;"><b>4</b></p>

Tabla 23. Task card crear interfaces graficas CRUD

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU5:</b> Registro de Usuario
<b>Nombre de Tarea:</b> Crear interfaces gráficas del CRUD para gestionar Usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaces gráficas para crear, actualizar, ver y eliminar usuarios, y búsquedas de usuario por identificación o listado de usuarios por determinado rol.	

Tabla 24. Task card conexión servidor

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU5:</b> Registro de Usuario
<b>Nombre de Tarea:</b> Conexión servidor	

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso el ingreso al sistema.	

Tabla 25. Task card validación de campos

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU5:</b> Registro de Usuario
<b>Nombre de Tarea:</b> Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, etc)	

Tabla 26. Task card selección búsqueda

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU5:</b> Registro de Usuario
<b>Nombre de Tarea:</b> Seleccionar Búsqueda por identificación	



<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Seleccionar búsqueda por identificación mostrará la información de un usuario, solo podrá realizarla un usuario bajo el rol de superadministrador, administrador y voluntario.	

Tabla 27. Task card selección búsqueda por rol

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU5:</b> Registro de Usuario
<b>Nombre de Tarea:</b> Seleccionar búsqueda por rol	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Seleccionar búsqueda por rol mostrará el listado de usuarios que tengan el rol definido en la búsqueda, solo podrá realizarla un usuario bajo el rol de superadministrador, administrador y voluntario.	

Tabla 28. Tarjeta de colaboración HU5 registro de usuario

<b>Usuario</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear usuario</li> <li>• Editar usuario</li> <li>• Eliminar usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• UserIdentity</li> <li>• Persona</li> </ul>

<ul style="list-style-type: none"> <li>• Ver usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Referencia</li> <li>• Rol</li> </ul>
---	---

Tabla 29. Historia de usuario guardar usuario

<b>Historia de usuario</b>	
<b>Número:</b> HU6	
<b>Programador Responsable:</b> Team Development	
<b>Nombre de la historia:</b> Guardar Usuario	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 8 puntos de historia	
<b>Tiempo estimado:</b> 16 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor	
<b>Se requiere:</b> Responder a la petición de registro de usuario	
<b>Para:</b> Que los usuarios ingresen información en la plataforma	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Creación la entidad Usuario en la base de datos.</li> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Valide las peticiones del usuario</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz</li> </ul>	<b>6</b>
<ul style="list-style-type: none"> <li>• Enviar Notificación de usuario registrado</li> </ul>	<b>4</b>
<ul style="list-style-type: none"> <li>• Enviar Usuario por identificación</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• listar usuarios por rol seleccionado</li> </ul>	<b>3</b>

Tabla 30. Task card validar petición

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU6:</b> Guardar Usuario
<b>Nombre de Tarea:</b> El servidor valide la petición del CRUD enviada a través de la interfaz	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Validar cada una de las peticiones realizadas a través de la interfaz gráfica como crear, eliminar, actualizar y ver la información de un usuario o la lista de usuarios acorde a rol del usuario que accede al sistema.</p>	

Tabla 31. Task card enviar notificación

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU6:</b> Guardar Usuario
<b>Nombre de Tarea:</b> Enviar Notificación de usuario registrado	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Validar la creación de nuevo usuario registrado por el superadministrador o administrador en la base de datos, se le debe enviar al usuario creado una notificación al correo electrónico de bienvenida al aplicativo web. Así mismo al usuario que se postule a una mascota al finalizar el ingreso de información personal se le enviará el mismo correo de bienvenida.</p>	

Tabla 32. Task card enviar información de usuario

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU6:</b> Guardar Usuario
<b>Nombre de Tarea:</b> Enviar Usuario por identificación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Validar la petición de búsqueda de un usuario por su número de identificación realizada a través de la interfaz gráfica dará como respuesta la información del usuario con respectivos atributos encontrados en la base datos.</p>	

Tabla 33. Task card listar usuarios por rol

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU6:</b> Guardar Usuario
<b>Nombre de Tarea:</b> listar usuarios por rol seleccionado	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Validar la petición de búsqueda de usuarios que tienen el mismo rol, realizada a través de la interfaz gráfica dará como respuesta la lista de usuarios con ese rol podrán ser gestionados (CRUD) por el superadministrador, administrador y voluntario.</p>	

Tabla 34. Tarjeta de colaboración HU6 guardar usuario

Usuario	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> <li>• Crear usuario</li> <li>• Editar usuario</li> <li>• Eliminar usuario</li> <li>• Ver usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• UserIdentity</li> <li>• Persona</li> <li>• Referencia</li> <li>• Rol</li> </ul>

Tabla 35. Historia de usuario registro de animales

Historia de usuario
<b>Número:</b> HU3
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Registro de Animales
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 13 puntos de historia
<b>Tiempo estimado:</b> 20 Horas
<p style="text-align: center;"><b>Perspectiva del producto</b></p> <p><b>Como:</b> Usuario  <b>Se requiere:</b> Gestionar información de animales que pertenecen a la fundación VAP  <b>Para:</b> Que los usuarios ingresen información en la plataforma</p>
<p style="text-align: center;"><b>Criterios de aceptación</b></p> <ul style="list-style-type: none"> <li>• El número de identificación del animal debe ser único.</li> <li>• Los animales pueden tener el mismo nombre.</li> <li>• En la foto, especie, fecha de nacimiento, fecha de ingreso, estado adoptado (si ó no) debe ser obligatorios.</li> <li>• Filtro de búsqueda para la galería de mascotas</li> </ul> <p><b>Restricción:</b></p> <ul style="list-style-type: none"> <li>• Los animales solo podrán ser creados, eliminados o actualizados por el usuario bajo el rol de Superadministrador o administrador.</li> <li>• La foto y el nombre son los únicos atributos que podrá actualizar el</li> </ul>

adoptante y el voluntario.	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces gráficas del CRUD para gestionar Animal</li> <li>• Conexión al servidor.</li> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> <li>• Seleccionar Búsqueda por nombre del animal</li> <li>• Visualizar galería de mascotas</li> <li>• Seleccionar foto de la galería para información detallada del animal.</li> </ul>	<p style="text-align: center;"><b>5</b></p> <p style="text-align: center;"><b>3</b></p> <p style="text-align: center;"><b>2</b></p> <p style="text-align: center;"><b>2</b></p> <p style="text-align: center;"><b>5</b></p> <p style="text-align: center;"><b>3</b></p>

**Tabla 36. Task card crear interfaces gráficas CRUD**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU3:</b> Registro de Animales
<b>Nombre de Tarea:</b> Crear interfaces gráficas del CRUD para gestionar Animal	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaces gráficas para crear, actualizar, ver y eliminar animales, y búsquedas de animales por nombre.	

**Tabla 37. Task card conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU3:</b> Registro de Animales
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia

<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso gestión de animales.	

Tabla 38. Task card validación de campos

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU3:</b> Registro de Animales
<b>Nombre de Tarea:</b> Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, etc)	

Tabla 39. Task card seleccionar búsqueda

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU3:</b> Registro de Animales
<b>Nombre de Tarea:</b> Seleccionar Búsqueda por nombre del animal	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia

<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Seleccionar búsqueda por nombre de la mascota mostrará el listado de las mascotas con el mismo nombre, solo podrá realizarla un usuario bajo el rol de superadministrador, administrador y voluntario.	

**Tabla 40. Task card visualizar galería de mascotas**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU3:</b> Registro de Animales
<b>Nombre de Tarea:</b> Visualizar galería de mascotas	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Mostrar en una interfaz gráfica las fotos de los animales que se encuentran en el hogar de paso es decir que actualmente no se encuentran adoptados, para que los visitantes del aplicativo web se animen a adoptar postulándose en el aplicativo, además de permitir una búsqueda por filtros con datos comunes de las mascotas.	

**Tabla 41. Task card seleccionar foto de galería**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 6	<b>HU3:</b> Registro de Animales
<b>Nombre de Tarea:</b> Seleccionar foto de la galería para información detallada del animal.	



<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Seleccionar cualquier foto de la galería de mascotas que se encuentran en adopción y visualizar la información de la mascota y el visitante del aplicativo podrá “postularse”.	

Tabla 42. Tarjeta de colaboración HU3 registro de animales

<b>AnimalPerfil</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear Animal</li> <li>• Editar Animal</li> <li>• Eliminar Animal</li> <li>• Ver Animal</li> <li>• Search</li> <li>• Edad</li> <li>• Proceso</li> <li>• Género</li> <li>• Indicador</li> <li>• Modify</li> <li>• Imprimiradop</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• Galería</li> </ul>

Tabla 43. Historia de usuario guardar animales

<b>Historia de usuario</b>
<b>Número:</b> HU4
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Guardar Animales
<b>Nivel de riesgo de desarrollo:</b> Media

<b>Peso de la historia:</b> 8 puntos de historia	
<b>Tiempo estimado:</b> 16 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor	
<b>Se requiere:</b> Responder a la petición de registro de animales	
<b>Para:</b> Que los usuarios ingresen información en la plataforma	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Creación la entidad Animal en la base de datos.</li> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• valide las peticiones del usuario</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz</li> </ul>	<b>6</b>
<ul style="list-style-type: none"> <li>• Generar galería de animales no adoptados</li> </ul>	<b>6</b>
<ul style="list-style-type: none"> <li>• listar animales con el mismo nombre</li> </ul>	<b>4</b>

**Tabla 44. Task card validar petición CRUD**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU4:</b> Guardar Animales
<b>Nombre de Tarea:</b> El servidor valide la petición del CRUD enviada a través de la interfaz	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada una de las peticiones realizadas a través de la interfaz gráfica como crear, eliminar, actualizar y ver la información de un animal o la lista de animales con el mismo nombre.	

Tabla 45. Task card generar galería de animales no adoptados

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU4:</b> Guardar Animales
<b>Nombre de Tarea:</b> Generar galería de animales no adoptados	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Enviar a la interfaz gráfica información de las mascotas no adoptadas en forma de galería de imágenes que el visitante del aplicativo web o futuro postulante visualice en la interfaz.</p>	

Tabla 46. Task card listar animales con el mismo nombre

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU4:</b> Guardar Animales
<b>Nombre de Tarea:</b> listar animales con el mismo nombre	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 22/02/2016	<b>Fecha Fin:</b> 10/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Validar la petición de búsqueda de animales que tienen el mismo nombre, realizada a través de la interfaz gráfica dará como respuesta la lista de animales con ese nombre, que posteriormente serán gestionadas (CRUD) por el superadministrador, administrador y voluntario.</p>	

Tabla 47. Tarjeta de colaboración HU4 guardar animales

AnimalPerfil	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> <li>• Crear Animal</li> <li>• Editar Animal</li> <li>• Eliminar Animal</li> <li>• Ver Animal</li> <li>• Search</li> <li>• Edad</li> <li>• Proceso</li> <li>• Género</li> <li>• Indicador</li> <li>• Modify</li> <li>• Imprimiradop</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• Galería</li> </ul>

#### 4.2.2 Diseño

A continuación en la figura 24 se ilustra la interfaz de usuario crear usuario, para conocer más sobre la totalidad de las interfaces de usuario ver [ANEXO](#)

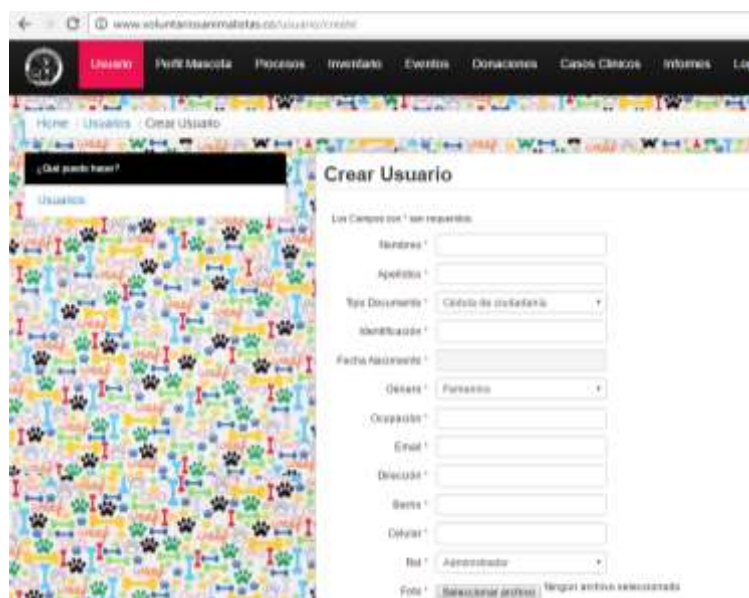
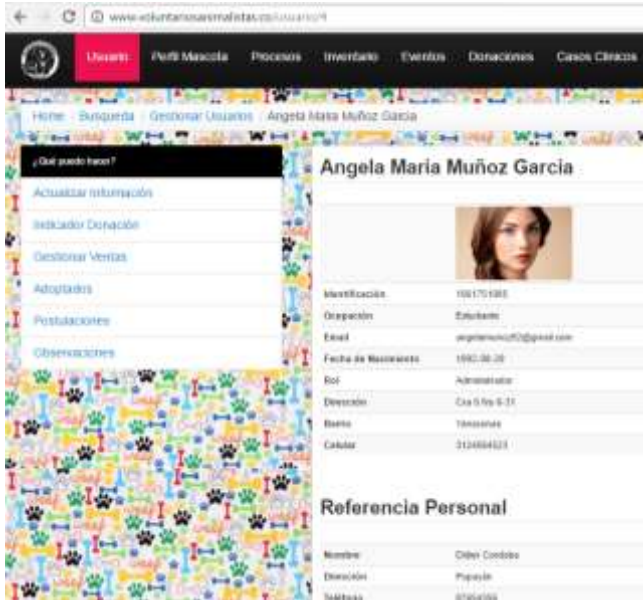


Figura 24. Interfaz crear usuario [Fuente propia]

### 4.2.3 Desarrollo

Tabla 48. Evento programado registro de usuario

Evento Programado Registro de usuario
Resultados del registro

Eventos Programados


## 4.2.4 Pruebas

Tabla 49. Prueba funcional registro de usuario

<b>Historia de usuario</b>	Registrar usuario
<b>Propósito</b>	Crear un usuario y asignar rol
<b>Requisito</b>	Ser administrador o súper administrador del sistema
<b>Pasos</b>	Se ingresa al menú usuarios y se selecciona la opción crear usuario, ahí aparecerá un formulario que pedirá información personal como nombres, apellidos, identificación, fecha de nacimiento, ocupación, email dirección, barrio, celular, rol, foto, clave y adicionalmente datos de una referencia personal tales como nombre, teléfono y dirección. Se presiona el botón crear para enviar la información al Backend
<b>Resultados esperados</b>	Un formulario con la información personal y su respectiva foto anteriormente registrada.

Los campos con \* son requeridos.

Nombre \* Angela Diana

Apellido \* Muñoz García

Tipo Documento \* Cédula de ciudadanía

Identificación \* 1001751085

Fecha nacimiento \* 1992-09-29

Género \* Femenino

Ocupación \* Estudiante

Email \* angeladiana25@gmail.com

Dirección \* Cra 9 No 9-31

Barrio \* Venezuela

Celular \* 3124564523

Rol \* Administrador

Foto \* Seleccionar archivo

Clave \*

Registra una referencia personal:

Figura 25. Prueba funcional interfaz crear usuario [Fuente propia]

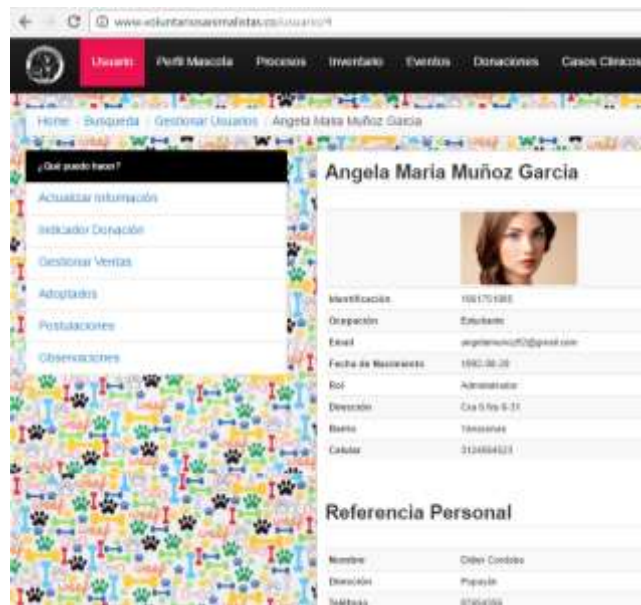


Figura 26. Prueba funcional registrar usuario [Fuente propia]





#### 4.2.5 Sprint Review

Finalizando el sprint 1 el equipo de desarrollo entregó las funcionalidades ingreso al sistema, registró e ingreso, registro de usuario, guardar usuario, registro de animales y guardar animales, con un estado de “terminado”, es decir software funcional al que se aplicaron pruebas de caja negra y blanca.

#### 4.2.6 Retrospectiva

Para la retrospectiva se aplicó la técnica matriz de aprendizaje puesto que permitió inspeccionar herramientas, procesos y personas identificando los elementos que salieron bien y mal creando un plan de mejoras para la ejecución del siguiente *Sprint*.

**Tabla 50. Retrospectiva Sprint 1**

 <ul style="list-style-type: none"> <li>• Generación de crud por parte del framework Yii.</li> <li>• Usar el repositorio remoto Bitbucket.</li> <li>• Usar sublime para la codificación.</li> <li>• Usar la herramienta Trello para la gestión de tareas.</li> </ul>	 <ul style="list-style-type: none"> <li>• Tiempo de aprendizaje.</li> <li>• Cambios en la base de datos.</li> <li>• Tiempo de aprendizaje Autoasigment framework Yii.</li> <li>• Estimación peso y tamaño.</li> </ul>
 <ul style="list-style-type: none"> <li>• Validación inicio de sesión por correo.</li> <li>• Notificación de usuario registrado enviando un email.</li> <li>• La mascota tenga una identificación.</li> </ul>	 <ul style="list-style-type: none"> <li>• Agradecimientos al product owner por mantener motivado al equipo</li> </ul>

### 4.3 IMPLEMENTACIÓN SPRINT 2

En el sprint 2 se aplicó un proceso de planeación, diseño, desarrollo y pruebas con el fin de desarrollar 9 historias de usuario con una duración de 8 horas diarias en 15 días, ejecutando la primera iteración obteniendo un producto mínimo.

Según el product backlog las historias de usuario asignadas al sprint 2 son:

- Registro de Consulta
- Guardar Consulta
- Registro de Vacunación
- Guardar Vacunación
- Registro de Esterilización
- Guardar Esterilización
- Registro de proceso
- Guardar proceso



- Registro de eventos

#### 4.3.1 Planeación

A continuación se observan las historias de usuario, task card (tareas de ingeniería) y tarjetas de colaboración del *Sprint 2*.

**Tabla 51. Historia de usuario registro de consulta**

<b>Historia de usuario</b>
<b>Número:</b> HU19
<b>Programador Responsable:</b> Team Development
<b>Nombre de la historia:</b> Registro de consultas
<b>Nivel de riesgo de desarrollo:</b> Alta
<b>Peso de la historia:</b> 21 puntos de historia
<b>Tiempo estimado:</b> 12 Horas
<b>Perspectiva del producto</b>
<p><b>Como:</b> Usuario</p> <p><b>Se requiere:</b> Registrar las enfermedades, medicamentos y tratamientos en las consultas realizadas.</p> <p><b>Para:</b> Anexar a la historia clínica de la mascota.</p>
<b>Criterios de aceptación</b>
<ul style="list-style-type: none"> <li>• Una mascota tiene consultas como enfermedades haya tenido o antiparasitarios se le aplicaron.</li> <li>• En las consultas debe ser obligatoria la fecha de ingreso, consulta, antecedentes y diagnóstico.</li> <li>• Por cada consulta se debe especificar la enfermedad, tratamiento y medicamentos.</li> <li>• La enfermedad debe tener como características la descripción, resultado y tratamiento a realizar.</li> <li>• La especificación de los medicamentos recetados para la consulta son obligatorios.</li> <li>• Descargar resumen de consultas.</li> </ul>
<b>Restricciones</b>

<ul style="list-style-type: none"> <li>Las consultas solo pueden ser creadas y eliminadas por los usuarios con rol de (Administrador, SuperAdministrador).</li> </ul>	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> <li>Crear interfaz gráfica para la creación de consulta, enfermedad y medicamento.</li> </ul>	2
<ul style="list-style-type: none"> <li>Conexión al servidor.</li> </ul>	2
<ul style="list-style-type: none"> <li>Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> </ul>	2
<ul style="list-style-type: none"> <li>Descargar el resumen de todas las consultas realizadas por mascota.</li> </ul>	4

**Tabla 52. Task card interfaz gráfica crear consulta**

TAREA DE INGENIERÍA	
<b>Número de Tarea:</b> 1	<b>HU19:</b> Registro de consultas
<b>Nombre de Tarea:</b> interfaz gráfica para la creación de consulta, enfermedad y medicamento.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>            Crear la interfaz gráfica para el registro de consultas por mascota del hogar de paso, con los campos fecha, antecedentes y diagnóstico, adicionalmente para seguir con la secuencia de la consulta crear las interfaces gráficas de enfermedad y tratamiento en la primera se especifica información relacionada con la sintomatología de la mascota en el campo de descripción de la enfermedad, también se debe determinar cuál es el diagnóstico a través del campo descripción del diagnóstico, por otra parte en la segunda interfaz seleccionar los medicamentos que van a ser aplicados, para finalmente seleccionar la opción de visualizar o descargar la consulta completa.            Este ingreso de datos se realiza dependiendo del rol de usuario (Administrador, SuperAdministrador).</p>	

Tabla 53. Task card conexión servidor

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU19:</b> Registro de consultas
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso la creación de eventos.	

Tabla 54. Task card validar campos

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU19:</b> Registro de consultas
<b>Nombre de Tarea:</b> Validar Campos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, )	

Tabla 55. Task card crear PDF de consultas por mascota

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU19:</b> Registro de consultas
<b>Nombre de Tarea:</b> Descargar el resumen de todas las consultas realizadas por mascota	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Al seleccionar la opción descargar, el usuario obtiene un resumen de todas las consultas realizadas por mascota con la descripción por consulta del motivo de consulta, diagnóstico y medicamentos para el tratamiento.</p>	

Tabla 56. Tarjeta colaboración HU19 Registro de consulta

<b>Consulta</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear consultas</li> <li>• Editar consultas</li> <li>• Actualizar consultas</li> <li>• Ver consultas</li> <li>• Administrar consultas</li> <li>• Imprimir consultas</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• AnimalPerfil</li> <li>• Enfermedad</li> <li>• Tratamiento</li> <li>• Enfermedad</li> </ul>

Tabla 57. Historia de usuario: guardar consulta

<b>Historia de usuario</b>
<b>Número:</b> HU22

<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Guardar Consulta	
<b>Nivel de riesgo de desarrollo:</b> Alta	
<b>Peso de la historia:</b> 21 puntos de historia	
<b>Tiempo estimado:</b> 12 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor	
<b>Se requiere:</b> Responder a la petición de registro de consulta	
<b>Para:</b> Que los usuarios ingresen información en la plataforma	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Creación la entidad Consulta, enfermedad, tratamiento y medicamento en la base de datos.</li> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Valide las peticiones del usuario</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz</li> <li>• Generar formato de las consultas realizadas a la mascota</li> </ul>	<p><b>7</b></p> <p><b>5</b></p>

Tabla 58. Task card validar peticiones CRUD en el servidor

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU22:</b> Guardar Consulta
<b>Nombre de Tarea:</b> El servidor valide la petición del CRUD enviada a través de la interfaz	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 7 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016

<b>Programador Responsable:</b> Team Development
<p><b>Descripción:</b>  Validar cada una de las peticiones realizadas a través de la interfaz gráfica como crear, eliminar, actualizar y ver la información de las consultas realizadas a un animal. Por otro lado cuando al crear una consulta se tienen en cuenta los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>• Una mascota tiene consultas como enfermedades haya tenido o antiparasitarios se le aplicaron.</li> <li>• En las consultas debe ser obligatoria la fecha de ingreso, consulta, antecedentes y diagnóstico.</li> <li>• Por cada consulta se debe especificar la enfermedad, tratamiento y medicamentos.</li> <li>• La enfermedad debe tener como características la descripción, resultado y tratamiento a realizar.</li> <li>• La especificación de los medicamentos recetados para la consulta son obligatorios.</li> <li>• Descargar resumen de consultas.</li> </ul> <p>Finalmente las consultas solo pueden ser creadas y eliminadas por los usuarios con rol de (Administrador, SuperAdministrador).</p>

Tabla 59. Task card generar PDF de consultas

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU22:</b> Guardar Consulta
<b>Nombre de Tarea:</b> Generar formato de las consultas realizadas a la mascota	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>  Validar la petición descargar consultas y enviar a la interfaz gráfica información de las consultas con su enfermedad, tratamiento, medicamentos, fecha y tipo de</p>	

enfermedad en un formato pdf, que podrá visualizar en la interfaz gráfica de la aplicación web.

**Tabla 60. Tarjeta colaboración HU22 Guardar Consulta**

<b>Consulta</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear Animal</li> <li>• Editar Animal</li> <li>• Eliminar Animal</li> <li>• Ver Animal</li> <li>• Search</li> <li>• Edad</li> <li>• Proceso</li> <li>• Género</li> <li>• Indicador</li> <li>• Modify</li> <li>• Imprimiradop</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• Galería</li> </ul>

**Tabla 61. Historia de usuario: registro de vacunación**

<b>Historia de usuario</b>
<b>Número:</b> HU17
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Registro de vacunación
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 8 puntos de historia
<b>Tiempo estimado:</b> 8 Horas

<b>Perspectiva del producto</b>	
<p><b>Como:</b> Usuario  <b>Se requiere:</b> Registrar las vacunas aplicadas a las mascotas del hogar de paso.  <b>Para:</b> Anexar a la historia clínica de la mascota.</p>	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Cada vacunación de tener nombre, serial y fecha.</li> <li>• El serial debe ser único.</li> <li>• Descargar resumen de vacunación.</li> </ul>	
<b>Restricciones</b>	
<ul style="list-style-type: none"> <li>• Las vacunas solo pueden ser creadas y eliminadas por los usuarios con rol de (Administrador, SuperAdministrador o Voluntario).</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaz gráfica para la creación de vacunación.</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• Conexión al servidor.</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> </ul>	<b>2</b>

**Tabla 62. Task card Interfaz gráfica vacunación**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU17:</b> Registro de vacunación
<b>Nombre de Tarea:</b> interfaz gráfica para la creación de vacunación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	



**Descripción:**

Crear interfaz gráfica para el registro de vacunas por mascota del hogar de paso o mascotas que han sido dadas en adopción, con los campos nombre, serial y fecha. Este ingreso de datos se realiza dependiendo del rol de usuario (Administrador, SuperAdministrador o Voluntario), adicionalmente tendrá la opción de imprimir el resumen de vacunas aplicadas.

**Tabla 63. Task card conexión servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU17:</b> Registro de vacunación
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/04/2016	<b>Fecha Fin:</b> 31/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso la creación de eventos.	

**Tabla 64. Task card validar campos**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU17:</b> Registro de vacunación
<b>Nombre de Tarea:</b> Validar Campos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, )

**Tabla 65. Tarjeta colaboración HU17 registro de vacunación**

<b>Vacunación</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear vacunas</li> <li>• Listado vacunas</li> <li>• Administrar vacunas</li> <li>• Imprimir vacunas</li> <li>• Editar vacunas</li> <li>• Eliminar vacunas</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• AnimalPerfil</li> </ul>

**Tabla 66. Historia de usuario: guardar vacunación**

<b>Historia de usuario</b>
<b>Número:</b> HU20
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Guardar vacunación
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 13 puntos de historia
<b>Tiempo estimado:</b> 12 Horas
<p style="text-align: center;"><b>Perspectiva del producto</b></p> <p><b>Como:</b> Servidor  <b>Se requiere:</b> Responder a la petición de vacunación.  <b>Para:</b> Que los usuarios visualicen la información de las vacunas aplicadas por mascota.</p>
<p style="text-align: center;"><b>Criterios de aceptación</b></p> <ul style="list-style-type: none"> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> </ul>

<ul style="list-style-type: none"> <li>• Creación de la entidad vacunas en la base de datos.</li> <li>• Validar las peticiones del cliente.</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz.</li> </ul>	<b>16</b>

**Tabla 67. Task card validación de peticiones CRUD**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU20:</b> Guardar vacunas
<b>Nombre de Tarea:</b> Validación de peticiones CRUD	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 16 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>  Validar las peticiones del usuario para la creación, edición, actualización y visualización de las vacunas por mascota, dependiendo del rol de usuario con el que esté logueado.  Si está logueado con rol de SuperAdministrador, Administrador podrá crear, editar y eliminar las vacunas por mascota, de lo contrario si el usuario está logueado con un rol de Adoptante, Benefactor y postulante solo podrá visualizar la información referente a las vacunas.</p>	

**Tabla 68. Tarjeta colaboración HU20 guardar vacunas**

<b>Vacunación</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear vacunas</li> <li>• Listado vacunas</li> <li>• Administrar vacunas</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• AnimalPerfil</li> </ul>

<ul style="list-style-type: none"> <li>• Imprimir vacunas</li> <li>• Editar vacunas</li> <li>• Eliminar vacunas</li> </ul>	
--	--

**Tabla 69. Historia de usuario: registro de esterilización**

<b>Historia de usuario</b>	
<b>Número:</b> HU18	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Registro de esterilización	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 8 puntos de historia	
<b>Tiempo estimado:</b> 8 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario	
<b>Se requiere:</b> Registrar el costo de la esterilización	
<b>Para:</b> Anexar a la historia clínica de la mascota.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• La esterilización debe contener datos como fecha, saldo, costo y abono.</li> <li>• Los datos saldo, costo y abono deben ser obligatorios.</li> <li>• Visualizar el saldo de la esterilización.</li> <li>• Especificar si la esterilización se ha realizado (SI/NO)</li> </ul>	
<b>Restricciones</b>	
<ul style="list-style-type: none"> <li>• Las vacunas solo pueden ser creadas y eliminadas por los usuarios con rol de (Administrador, SuperAdministrador o Voluntario).</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaz gráfica para la creación de vacunación.</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• Conexión al servidor.</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> </ul>	<b>2</b>

**Tabla 70. Task card interfaz gráfica crear esterilización**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU18:</b> Registro de esterilización
<b>Nombre de Tarea:</b> interfaz gráfica para la creación de esterilización	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>            Crear interfaz gráfica para el registro de esterilización por mascota del hogar de paso o de las mascotas que han sido dadas en adopción, con los campos fecha, saldo, abono y esterilizado. Este ingreso de datos se realiza dependiendo del rol de usuario (Administrador, SuperAdministrador o Voluntario), adicionalmente tendrá la opción de imprimir el resumen de vacunas aplicadas.</p>	

Tabla 71. Task card conexión servidor

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU18:</b> Registro de esterilización
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>            Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso el registro de esterilización.</p>	

Tabla 72. Task card validar campos

<b>TAREA DE INGENIERÍA</b>
----------------------------

<b>Número de Tarea:</b> 3	<b>HU18:</b> Registro de esterilización
<b>Nombre de Tarea:</b> Validar Campos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, )	

**Tabla 73. Tarjeta colaboración HU18 Registro de esterilización**

<b>Esterilización</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear esterilización</li> <li>• Editar esterilización</li> <li>• Eliminar esterilización</li> <li>• Listado esterilización</li> <li>• Administrar esterilización</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• AnimalPerfil</li> </ul>

**Tabla 74. Historia de usuario: guardar esterilización**

<b>Historia de usuario</b>
<b>Número:</b> HU21
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Guardar esterilización
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 13 puntos de historia

<b>Tiempo estimado:</b> 12 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor <b>Se requiere:</b> Responder a la petición de esterilización. <b>Para:</b> Que los usuarios visualicen la información de esterilización por mascota.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Creación de la entidad esterilización en la base de datos.</li> <li>• Validar las peticiones del cliente.</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz.</li> </ul>	<b>16</b>

**Tabla 75. Task card validar peticiones CRUD**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU21:</b> Guardar esterilización
<b>Nombre de Tarea:</b> Validación de peticiones CRUD	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 16 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>  Validar las peticiones del usuario para la creación, edición, actualización y visualización de la esterilización por mascota, dependiendo del rol de usuario con el que esté logueado.  Si está logueado con rol de SuperAdministrador, Administrador podrá crear, editar y eliminar la esterilización por mascota, de lo contrario si el usuario está logueado con un rol de voluntario, Benefactor y postulante solo podrá visualizar la información referente a la esterilización.</p>	

Tabla 76. Tarjeta colaboración HU21 Guardar esterilización.

<b>Esterilización</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear esterilización</li> <li>• Editar esterilización</li> <li>• Eliminar esterilización</li> <li>• Listado esterilización</li> <li>• Administrar esterilización</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• AnimalPerfil</li> </ul>

Tabla 77. Historia de usuario: Registro de proceso

<b>Historia de usuario</b>
<b>Número:</b> HU15
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Registro de Proceso
<b>Nivel de riesgo de desarrollo:</b> Alta
<b>Peso de la historia:</b> 21 puntos de historia
<b>Tiempo estimado:</b> 30 Horas
<b>Perspectiva del producto</b>
<p><b>Como:</b> Usuario</p> <p><b>Se requiere:</b> Gestionar los procesos realizados por la fundación</p> <p><b>Para:</b> Que los usuarios ingresen información en la plataforma</p>
<b>Criterios de aceptación</b>
<ul style="list-style-type: none"> <li>• El No de acta del proceso debe ser único.</li> <li>• El Animal, el voluntario, el adoptado, la Fecha y el proceso a realizar deben ser obligatorios en los procesos de adopción y seguimiento.</li> <li>• Realizar el cuestionario de preguntas asignadas al tipo de acta “adopción” y “seguimiento”</li> <li>• Descargar el proceso realizado en el aplicativo web.</li> </ul>



<b>Restricción:</b> <ul style="list-style-type: none"> <li>• Los procesos sólo podrán ser creados, actualizados y eliminados por el usuario bajo el rol de Superadministrador o administrador y voluntario.</li> <li>• Los procesos los podrán ver y descargar los roles voluntario, administrador y superadministrador del aplicativo web.</li> <li>• Los procesos de postulación no es obligatorio que el campo idVoluntario sea registrado en la entidad proceso.</li> </ul>	
Actividades	N° de horas de implementación
• Crear interfaces gráficas del CRUD para gestionar Procesos	2
• Conexión al servidor.	4
• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).	3
• Ingresar respuestas al cuestionario de adopción a través de un formulario	4
• Descargar formato de adopción	3
• Ingresar respuestas al cuestionario de seguimiento.	4
• Descargar formato de seguimiento	3
• Ingresar respuesta a la pregunta del proceso postulación	4
• Gestionar contacto con la fundación	3

**Tabla 78. Task card CRUD para gestionar procesos**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU15:</b> Registro de Proceso
<b>Nombre de Tarea:</b> Crear interfaces gráficas del CRUD para gestionar Procesos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Crear interfaces gráficas para crear, actualizar, ver y eliminar procesos, además de los formularios de preguntas de los procesos, cada vez que se cree un proceso será asignado a un voluntario de la fundación.

Tabla 79. Task card conexión servidor

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU15:</b> Registro de Proceso
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario.	

Tabla 80. Task card validación de campos de acuerdo al tipo de dato

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU15:</b> Registro de Proceso
<b>Nombre de Tarea:</b> Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	

<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, etc)	

**Tabla 81. Task card ingresar respuestas al cuestionario de adopción**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU15:</b> Registro de proceso
<b>Nombre de Tarea:</b> Ingresar respuestas al cuestionario de adopción	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ingresar y enviar respuestas en el cuestionario visualizado en la interfaz gráfica	

**Tabla 82. Task card descargar formato de adopción**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU15:</b> Registro de Proceso
<b>Nombre de Tarea:</b> Descargar formato de adopción	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia

<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Al seleccionar descargar formato de adopción se visualizará el formato con las respuestas dadas en el formulario de preguntas, características de la mascota adoptada, y el acuerdo que deberá firmar el adoptante a lo que se compromete con la mascota.	

**Tabla 83. Task card ingresar respuestas al cuestionario de seguimiento**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 6	<b>HU15:</b> Registro de Proceso
<b>Nombre de Tarea:</b> Ingresar respuestas al cuestionario de seguimiento.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ingresar y enviar respuestas en el cuestionario visualizado en la interfaz gráfica	

**Tabla 84. Task card descargar formato de seguimiento**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 7	<b>HU15:</b> Registro de Proceso
<b>Nombre de Tarea:</b> Descargar formato de seguimiento	

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Al seleccionar descargar formato de seguimiento se visualizará el formato con las respuestas dadas en el formulario de preguntas, características de la mascota adoptada, y su respectiva foto actual de la mascota.	

**Tabla 85. Task card ingresar respuesta de postulación**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 8	<b>HU15:</b> Registro de Proceso
<b>Nombre de Tarea:</b> Ingresar respuesta a la pregunta del proceso postulación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ingresar y enviar respuesta a la pregunta visualizada en la interfaz gráfica después de ingresar la información de usuario para una postulación.	

**Tabla 86. Task Card Gestionar contacto con la fundación**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 9	<b>HU15:</b> Registro de Proceso

<b>Nombre de Tarea:</b> Gestionar contacto con la fundación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaces gráficas para gestionar el contacto con los integrantes de cada comite de la fundación voluntarios animalistas	

Tabla 87. Tarjeta de colaboración HU15 Proceso

Proceso	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> <li>• Crear proceso</li> <li>• Editar proceso</li> <li>• Eliminar proceso</li> <li>• Ver proceso</li> <li>• Autocomplete</li> <li>• PdfProcesoAdop</li> <li>• PdfProcesoSegui</li> <li>• ViewPostulante</li> <li>• CreatePostulacion</li> <li>• IndexPostulante</li> <li>• IndexAdoptado</li> </ul>	<ul style="list-style-type: none"> <li>• Usuario</li> <li>• AnimalPerfil</li> <li>• Controller</li> </ul>

Tabla 88. Historia de usuario: guardar proceso

<b>Historia de usuario</b>
<b>Número:</b> HU16

<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Guardar Proceso	
<b>Nivel de riesgo de desarrollo:</b> Alta	
<b>Peso de la historia:</b> 13 puntos de historia	
<b>Tiempo estimado:</b> 20 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor	
<b>Se requiere:</b> Responder a la petición registro de procesos	
<b>Para:</b> Que los usuarios ingresen información en la plataforma	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Creación de las entidades Proceso, Tipo acta, pregunta, tipoactapregunta, procesoencuenta en la base de datos acorde al diagrama Entidad relación.</li> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Valide las peticiones del usuario</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz</li> </ul>	<b>6</b>
<ul style="list-style-type: none"> <li>• Enviar preguntas al cuestionario de la interfaz gráfica</li> </ul>	<b>4</b>
<ul style="list-style-type: none"> <li>• Capturar y guardar respuestas del cuestionario</li> </ul>	<b>5</b>
<ul style="list-style-type: none"> <li>• Generar formato de procesos</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• Enviar notificación al email.</li> </ul>	<b>2</b>

**Tabla 89. Task card validar peticiones CRUD**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU16:</b> Guardar Proceso
<b>Nombre de Tarea:</b> El servidor valide la petición del CRUD enviada a través de la interfaz	

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 11/04/2016	<b>Fecha Fin:</b> 31/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada una de las peticiones realizadas a través de la interfaz gráfica como crear, eliminar y actualizar la información de los procesos realizados ejecutados solamente bajo el rol de administrador, superadministrador y voluntario. Así mismo todos los usuarios podrán ver y descargar el proceso de adopción y seguimiento.	

**Tabla 90. Task card interfaz gráfica del cuestionario**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU16:</b> Guardar Proceso
<b>Nombre de Tarea:</b> Enviar preguntas al cuestionario de la interfaz gráfica	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Enviar las preguntas asignadas al formulario del proceso de adopción, seguimiento, y la pregunta de interés realizada en el proceso de postulación, a la interfaz gráfica de visualización del formulario.	

**Tabla 91. Task card capturar y guardar respuestas**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU16:</b> Guardar Proceso
<b>Nombre de Tarea:</b> Capturar y guardar respuestas del cuestionario	



<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Capturar las respuestas del proceso de adopción, seguimiento y postulación que serán almacenadas en la base de datos.	

**Tabla 92. Task card generar formato de procesos**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU16:</b> Guardar Proceso
<b>Nombre de Tarea:</b> Generar formato de procesos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar la petición “descargar” formato de adopción o seguimiento respectivamente en la visualización de cada proceso consultando en la base de datos la información y enviando a la vista de la aplicación web el resultado del formato de adopción o seguimiento.	

**Tabla 93. Task card enviar notificación al email.**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU16:</b> Guardar Proceso

<b>Nombre de Tarea:</b> Enviar notificación al email.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar la creación del proceso de postulación a un animal realizada por un visitante de la página en la base de datos, se le debe enviar al usuario creado una notificación al correo electrónico de bienvenida al aplicativo web.	

**Tabla 94. Tarjeta de colaboración HU16 guardar proceso**

<b>Proceso</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear proceso</li> <li>• Editar proceso</li> <li>• Eliminar proceso</li> <li>• Ver proceso</li> <li>• Autocomplete</li> <li>• PdfProcesoAdop</li> <li>• PdfProcesoSegui</li> <li>• ViewPostulante</li> <li>• CreatePostulacion</li> <li>• IndexPostulante</li> <li>• IndexAdoptado</li> </ul>	<ul style="list-style-type: none"> <li>• Usuario</li> <li>• AnimalPerfil</li> <li>• Controller</li> </ul>

**Tabla 95. Historia de usuario: registro de eventos**

<b>Historia de usuario</b>
----------------------------

<b>Número:</b> HU7	
<b>Programador Responsable:</b> Team Development	
<b>Nombre de la historia:</b> Registro de eventos	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 13 puntos de historia	
<b>Tiempo estimado:</b> 20 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario	
<b>Requiero:</b> Publicar eventos de la fundación.	
<b>Para:</b> Que los usuarios visualicen la información publicada sobre los eventos.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• El evento deben tener fecha de inicio y cierre del evento.</li> <li>• Cada evento tiene una imagen relacionada con las actividades a realizar.</li> <li>• Especificar la hora de inicio y cierre del evento.</li> <li>• Clasificar el evento.</li> <li>• Visualizar los eventos a través de un calendario.</li> <li>• Información detallada del evento.</li> </ul>	
<b>Restricciones</b>	
<ul style="list-style-type: none"> <li>• Los eventos solo pueden ser creadas y eliminadas por los usuarios con rol de (Administrador, SuperAdministrador).</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaz gráfica para la creación de eventos y tipos de eventos.</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• Conexión al servidor.</li> </ul>	<b>3</b>
<ul style="list-style-type: none"> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> </ul>	<b>2</b>
<ul style="list-style-type: none"> <li>• Interfaz gráfica para el calendario de eventos.</li> </ul>	<b>7</b>
<ul style="list-style-type: none"> <li>• Información del evento, evidenciando las características de este.</li> </ul>	<b>5</b>

**Tabla 96. Task card interfaz registro de eventos**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU7:</b> Registro de eventos
<b>Nombre de Tarea:</b> Interfaz gráfica para el registro de eventos y tipos de eventos.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para la creación de eventos, con los campos nombre de evento, fecha y hora de inicio, fecha y hora de cierre, lugar, dirección y con opción de subir una imagen, adicionalmente la interfaz tipo de evento, se podrá especificar el nombre de los posibles tipos de evento que se realizan.	

**Tabla 97. Task card conexión servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU7:</b> Registro de eventos
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/04/2016	<b>Fecha Fin:</b> 31/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso la creación de eventos.	

**Tabla 98. Task card validar campos**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU7:</b> Registro de eventos
<b>Nombre de Tarea:</b> Validar Campos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, )	

Tabla 99. Task card interfaz gráfica calendario de eventos

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU7:</b> Registro de eventos
<b>Nombre de Tarea:</b> Interfaz gráfica para el calendario de eventos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 7 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Se visualiza un calendario con los eventos creados durante el mes, especificando la duración por fechas y nombre.	

Tabla 100. Task card información del evento

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU7:</b> Registro de eventos
<b>Nombre de Tarea:</b> Información del evento	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 11/03/2016	<b>Fecha Fin:</b> 31/03/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>  A partir de la interfaz gráfica del calendario se podrá visualizar la información del evento al dar click sobre el nombre de este, desplegándose una ventana emergente especificando las características del evento como lugar, hora y fecha.</p>	

**Tabla 101. Tarjeta de colaboración HU7 registro de eventos**

<b>Evento</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear eventos</li> <li>• Editar eventos</li> <li>• Eliminar eventos</li> <li>• Listado eventos</li> <li>• Administrar eventos</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• Fundacion</li> <li>• TipoEvento</li> <li>• Adopción</li> <li>• Vacunación</li> <li>• RecaudaFondo</li> <li>• Esteriliza</li> <li>• VentaProducto</li> </ul>

## 4.2.2 Diseño

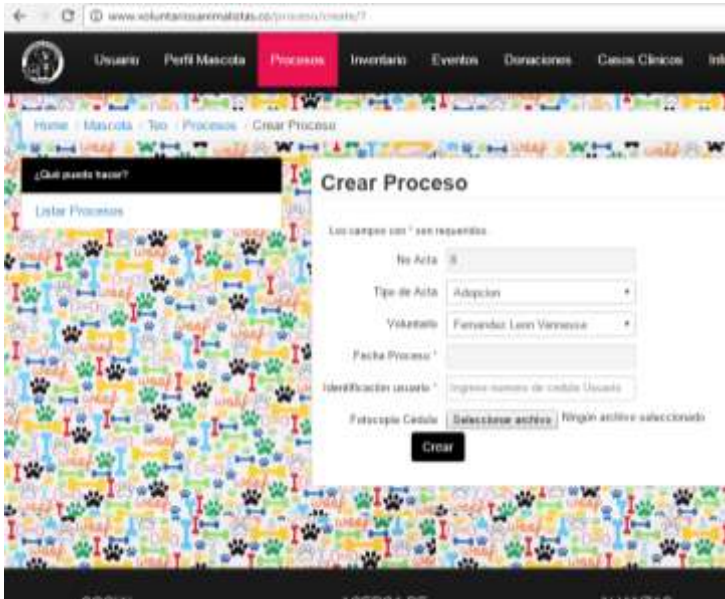
The image shows a web browser window with the URL [www.voluntariosanimalistas.co/proceso/crear/](http://www.voluntariosanimalistas.co/proceso/crear/). The page has a navigation menu with items: Usuario, Perfil Mascota, **Procesos**, Inventario, Eventos, Donaciones, Casos Clínicos, and In. Below the menu is a breadcrumb trail: Home / Mascota / Tipo / Procesos / Crear Proceso. A sidebar on the left contains a search bar with the text '¿Qué puedo hacer?' and a link 'Listar Procesos'. The main content area is titled 'Crear Proceso' and contains a form with the following fields and options:

- Label: 'Los campos con \* son requeridos'
- Field: 'No Acte' with a value of '0'.
- Field: 'Tipo de Acte' with a dropdown menu showing 'Adopción'.
- Field: 'Voluntario' with a dropdown menu showing 'Fernando Leon Vanessa'.
- Field: 'Fecha Proceso' with an asterisk indicating it is required.
- Field: 'Identificación usuario' with an asterisk and a placeholder text 'ingrese numero de cédula (usuario)'. Below this field is a small text 'Código de Verificación'.
- Field: 'Fotocopia Cédula' with a file selection button 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'.
- Button: 'Crear'.

Figura 27. Interfaz gráfica crear proceso [Fuente propia]

## 4.2.3 Desarrollo

Tabla 102. Evento Programado Registro de Proceso

Evento Programado Registro de Proceso
Resultados del registro

Eventos Programados
<pre data-bbox="310 1192 1302 1787">80 public function actionCreate(\$id) 81 { 82     \$model=new Proceso; 83     \$model-&gt;idAnimalPerfil=\$id; 84     \$user=new Usuario; 85     \$connection = Yii::app()-&gt;db; 86     \$sqlUsuario = "SELECT MAX(idActa) as tt FROM proceso"; 87     \$command = \$connection-&gt;createCommand(\$sqlUsuario); 88     \$dataReader = \$command-&gt;query(); 89     \$rows = \$dataReader-&gt;readAll(); 90     foreach (\$rows as \$key ) { 91         \$numeroActa=\$key['tt']; 92     } 93     \$model-&gt;NoActa=\$numeroActa+1; 94     if(isset(\$_POST['Proceso'])) 95     { 96         \$model-&gt;attributes=\$_POST['Proceso']; 97         \$idUsuario=\$_POST['Proceso']['idUsuario']; 98         \$idUsuarioint=(int)\$idUsuario; 99         /*Actualizar rol*/ 100         # \$user = \$this-&gt;loadModel(\$idUsuarioint, 'User'); 101         \$assigned_roles = Yii::app()-&gt;authManager-&gt;getRoles(\$idUsuarioint); 102         if(empty(\$assigned_roles)) //checks that there are assigned roles{ 103             \$auth=Yii::app()-&gt;authManager; //initializes the authManager 104             foreach(\$assigned_roles as \$n=&gt;\$role){ 105                 if(\$auth-&gt;revoke(\$n,\$idUsuarioint)) //remove each assigned role for this user 106                 { 107                     Yii::app()-&gt;authManager-&gt;save(); //again always save the result} 108                 } 109             } 110         /*Fin Actualizar rol*/ 111     } 112 }</pre>



#### 4.2.4 Pruebas

**Tabla 103. Pruebas de funcionalidad Proceso**

<b>Historia de usuario</b>	Registrar proceso
<b>Propósito</b>	Crear un proceso y guardarlo
<b>Requisito</b>	Ser administrador o súper administrador del sistema
<b>Pasos</b>	<p>Se ingresa al menú en la opción perfil mascota y seguidamente en el campo de búsqueda se registra el nombre de la mascota, a continuación aparecerá un listado de las mascotas con el nombre ingresado, se selecciona la mascota y se presenta su información. Se presiona en la opción proceso y se visualizarán los procesos ya creados, presionar la opción crear proceso y aparecerá un formulario con los siguientes campos fecha, número de acta, seleccionar proceso, voluntario, el número de identificación del adoptante y se presiona el botón crear para enviar la información al backend.</p> <p>Al presionar el botón crear redirecciona al formulario de preguntas necesarios para llevar a cabo un determinado proceso al concluir nos llevará a la vista del proceso con toda la información consignada para dicho proceso.</p>
<b>Resultados esperados</b>	Un formulario con la información de la mascota adoptada, adoptante y voluntario, además de las opciones ver y descargar PDF del proceso.



Figura 28. Interfaz gráfica búsqueda de mascotas [Fuente propia]

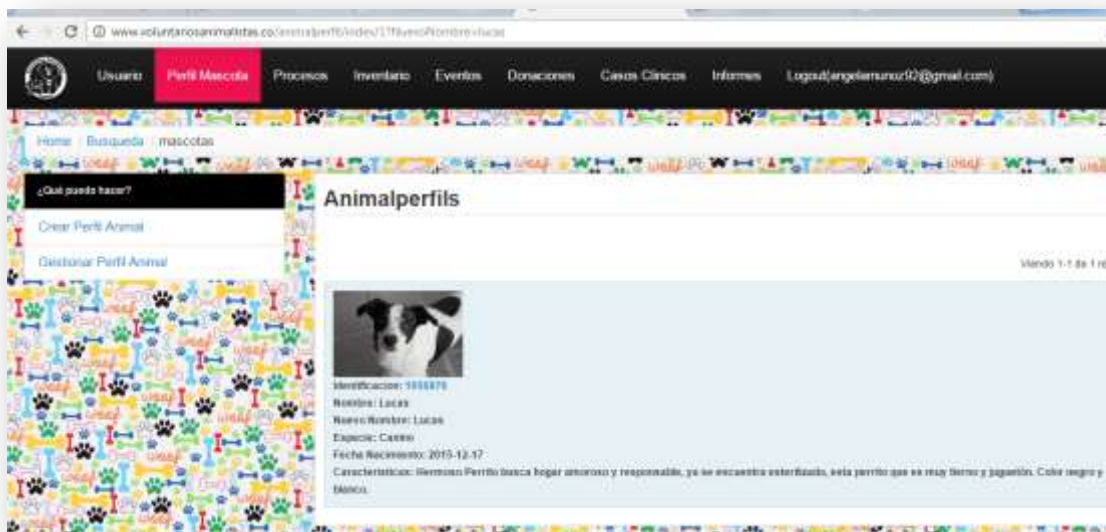


Figura 29. Interfaz resultado búsqueda de mascotas [Fuente propia]

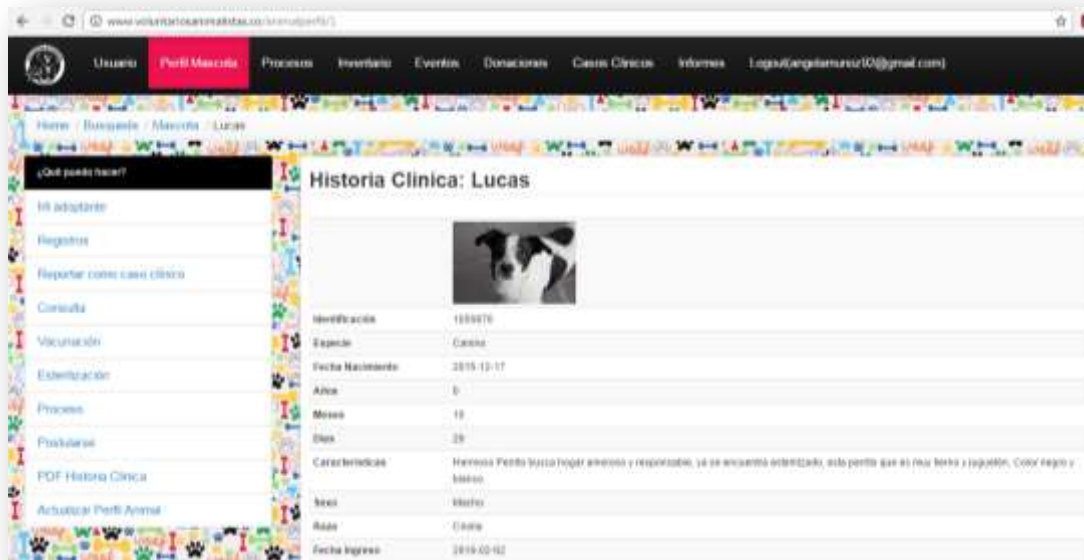


Figura 30. Interfaz gráfica Historia clínica de la mascota [Fuente propia]

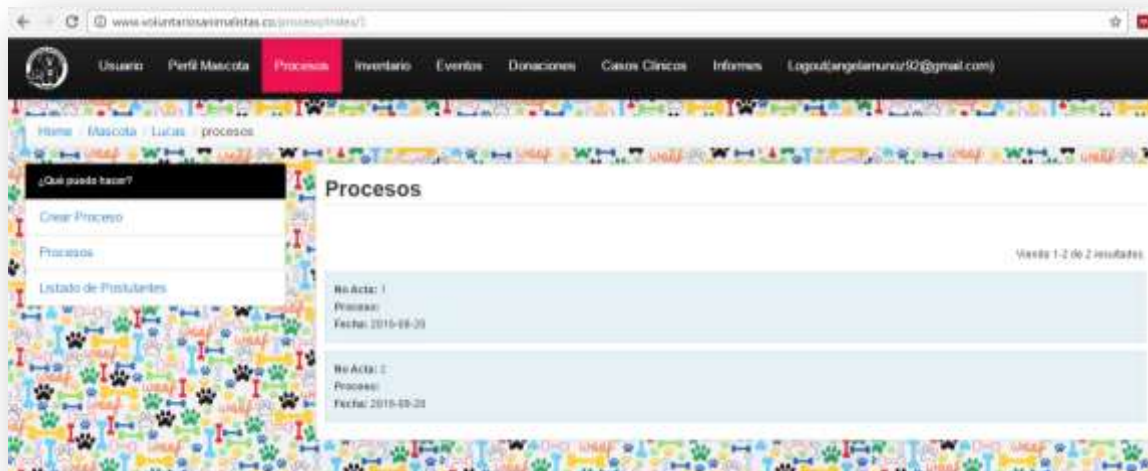


Figura 31. Interfaz listado de procesos por mascota [Fuente propia]

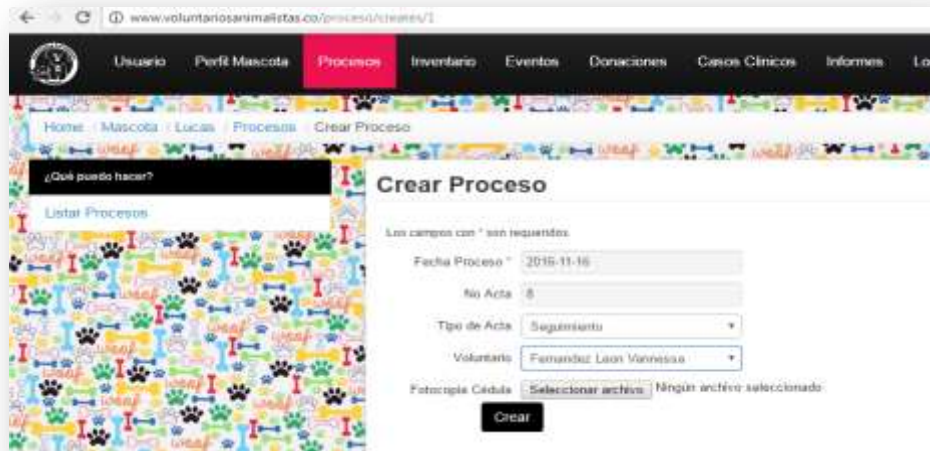


Figura 32. Interfaz gráfica crear proceso [Fuente propia]

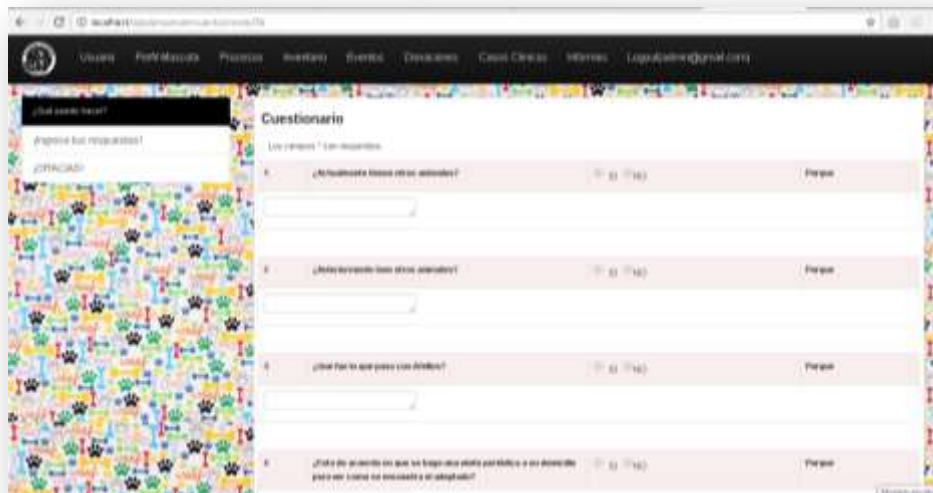


Figura 33. Interfaz gráfica cuestionario [Fuente propia]



Figura 34. Prueba de funcionalidad crear proceso [Fuente propia]





#### 4.2.5 Sprint Review

Finalizando el sprint 2 el equipo de desarrollo entregó las funcionalidades: registro de consulta, guardar consulta, registro de vacunación, guardar vacunación, registro de esterilización, guardar esterilización, registro de proceso, guardar proceso y registro de eventos, con un estado de “terminado” es decir software funcional al que se aplicaron pruebas de caja negra y blanca.

#### 4.2.6 Retrospectiva

Para las retrospectivas se aplicó la técnica matriz de aprendizaje puesto que permitió inspeccionar herramientas, procesos y personas identificando los elementos que salieron bien y mal creando un plan de mejoras para la ejecución del siguiente *Sprint*.

Tabla 104. Retrospectiva Sprint 2

 <ul style="list-style-type: none"> <li>• Generación de CRUD por parte del framework Yii.</li> <li>• Usar el repositorio remoto Bitbucket.</li> <li>• Usar sublime para la codificación.</li> <li>• Usar la herramienta Trello para la gestión de tareas.</li> <li>• Se redujo el tiempo de aprendizaje</li> </ul>	 <ul style="list-style-type: none"> <li>• Tiempo de aprendizaje.</li> <li>• Cambios en la base de datos.</li> <li>• Estimación peso y tamaño.</li> <li>• Guardar proceso.</li> <li>• Registro de proceso.</li> </ul>
 <ul style="list-style-type: none"> <li>• Auto complementar campos.</li> <li>• Implementar historia clínica básica de las mascotas del hogar de paso</li> </ul>	 <ul style="list-style-type: none"> <li>• Agradecimientos al product owner por mantener motivado al equipo.</li> <li>• Al médico veterinario por facilitar el formato de las historias clínicas</li> </ul>

#### 4.4 IMPLEMENTACIÓN SPRINT 3

Durante la iteración del sprint 3 se aplicó un proceso de planeación, diseño, desarrollo y pruebas con el fin de desarrollar 7 historias de usuario con una duración de 8 horas diarias en 16 días, ejecutando la primera iteración obteniendo un producto mínimo.

Según el product backlog las historias de usuario asignadas al sprint 3 son:

- Guardar eventos
- Registro de Productos
- Guardar Productos
- Registro de Productos Zoosanitarios
- Guardar Productos Zoosanitarios

- Seleccionar Indicador
- Enviar Gráfica indicador

#### 4.4.1 Planeación

A continuación se observan las historias de usuario, task card (tareas de ingeniería) y tarjetas de colaboración del *Sprint 3*.

**Tabla 105. Historia de usuario guardar evento**

<b>Historia de usuario</b>	
<b>Número:</b> HU8	
<b>Programador Responsable:</b> Team Development	
<b>Nombre de la historia:</b> Guardar eventos	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 8 puntos de historia	
<b>Tiempo estimado:</b> 16 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor	
<b>Se requiere:</b> Responder a la petición de publicación de eventos	
<b>Para:</b> Que los usuarios visualicen la información de los eventos a realizar.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Creación de la entidades evento y tipo de evento en la base de datos.</li> <li>• Validar las peticiones del cliente.</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz.</li> </ul>	<b>6</b>
<ul style="list-style-type: none"> <li>• Enviar información al calendario sobre el evento creado.</li> </ul>	<b>8</b>
<ul style="list-style-type: none"> <li>• Enviar información detallada sobre el evento</li> </ul>	<b>2</b>

seleccionado.	
---------------	--

Tabla 106. Task card validar peticiones CRUD

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU8:</b> Guardar eventos
<b>Nombre de Tarea:</b> Validación de peticiones CRUD	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar las peticiones del usuario para la creación, edición, actualización y visualización de los eventos y tipo de eventos, dependiendo del rol de usuario con el que esté logueado.	

Tabla 107. Task card enviar información al calendario

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU8:</b> Guardar eventos
<b>Nombre de Tarea:</b> Enviar información al calendario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 8 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Al momento que el usuario crear un evento, se debe enviar esta información a la interfaz gráfica del calendario, para ser visualizada por nombre y fechas	



correspondientes al mes.

Tabla 108. Task card Enviar información detallada sobre el evento

TAREA DE INGENIERÍA	
<b>Número de Tarea:</b> 3	<b>HU8:</b> Guardar eventos
<b>Nombre de Tarea:</b> Enviar información detallada sobre el evento	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Al momento que el usuario seleccione el evento por el nombre, se debe desplegar una ventana emergente con la información detallada como lugar, fecha y hora.	

Tabla 109. Tarjeta colaboración HU8 Guardar eventos

Evento	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"><li>• Crear eventos</li><li>• Editar eventos</li><li>• Eliminar eventos</li><li>• Listado eventos</li><li>• Administrar eventos</li><li>• Crear calendario</li></ul>	<ul style="list-style-type: none"><li>• Controller</li><li>• Fundacion</li><li>• TipoEvento</li><li>• Adopción</li><li>• Vacunación</li><li>• RecaudaFondo</li><li>• Esteriliza</li><li>• VentaProducto</li></ul>

Tabla 110. Historia de usuario: registro de productos

<b>Historia de usuario</b>
----------------------------

<b>Número:</b> HU9	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Registro de productos	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 13 puntos de historia	
<b>Tiempo estimado:</b> 20 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario	
<b>Se requiere:</b> Registrar productos	
<b>Para:</b> llevar un inventario de productos, con sus respectivas ventas.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Un producto debe tener un nombre, cantidad y fecha de vencimiento.</li> <li>• Registrar las ventas con fecha y cantidad del producto.</li> <li>• Asociar a un usuario la venta de un producto.</li> </ul>	
<b>Restricciones</b>	
<ul style="list-style-type: none"> <li>• Los productos solo pueden ser creados y eliminados por los usuarios con rol de (Administrador, SuperAdministrador).</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaz gráfica para la creación de productos y ventas.</li> </ul>	<b>9</b>
<ul style="list-style-type: none"> <li>• Conexión al servidor.</li> </ul>	<b>5</b>
<ul style="list-style-type: none"> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> </ul>	<b>6</b>

Tabla 111. Task card interfaz gráfica para crear productos

<b>TAREA DE INGENIERÍA</b>
----------------------------

<b>Número de Tarea:</b> 1	<b>HU9:</b> Registro de productos
<b>Nombre de Tarea:</b> Interfaz gráfica para la creación de productos.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 9 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para la creación de productos, con los campos nombre, fecha y cantidad, adicionalmente la interfaz para la venta de productos, se debe adicionar la información del usuario que realizó la venta y los productos asociados a esa venta.	

Tabla 112. Task card conexión servidor

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU9:</b> Registro de productos
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso la creación de productos y ventas.	

Tabla 113. Task card validar campos

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU9:</b> Registro de productos
<b>Nombre de Tarea:</b> Validar Campos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, )	

Tabla 114. Tarjeta colaboración HU9 Registro de productos

<b>Inventario</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear productos</li> <li>• Listado productos</li> <li>• Administrar productos</li> <li>• Autocompletar productos</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• Venta</li> <li>• Usuario</li> </ul>

Tabla 115. Historia de usuario: guardar productos

<b>Historia de usuario</b>
<b>Número:</b> HU10
<b>Programador Responsable:</b> Team Development
<b>Nombre de la historia:</b> Guardar productos
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 8 puntos de historia

<b>Tiempo estimado:</b> 16 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor <b>Se requiere:</b> Responder a la petición de productos y ventas <b>Para:</b> Que los usuarios visualicen la información de los productos y ventas realizadas.	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Creación de la entidades venta e inventario en la base de datos.</li> <li>• Validar las peticiones del cliente.</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz.</li> </ul>	<b>16</b>

**Tabla 116. Task card validar peticiones CRUD**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU10:</b> Guardar productos
<b>Nombre de Tarea:</b> Validación de peticiones CRUD	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 16 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar las peticiones del usuario para la creación, edición, actualización y visualización de los productos y ventas, dependiendo del rol de usuario con el que esté logueado.	

**Tabla 117. Tarjeta colaboración HU10 Guardar Productos**

Inventario	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> <li>• Crear productos</li> <li>• Listado productos</li> <li>• Administrar productos</li> <li>• Autocompletar productos</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• Venta</li> <li>• Usuario</li> </ul>

Tabla 118. Historia de usuario: registro de productos Zoosanitarios

Historia de usuario	
<b>Número:</b> HU11	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Registro de productos zoosanitarios	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 13 puntos de historia	
<b>Tiempo estimado:</b> 20 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario	
<b>Se requiere:</b> Registrar productos zoosanitarios.	
<b>Para:</b> llevar un inventario de productos zoosanitarios	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• El producto zoosanitario debe tener nombre, cantidad, efectivo y fecha.</li> <li>• Registrar las donaciones con fecha y cantidad del producto zoosanitario.</li> <li>• Asociar a un usuario la donación de productos zoosanitarios</li> </ul>	
<b>Restricciones</b>	
<ul style="list-style-type: none"> <li>• Los productos zoosanitarios solo pueden ser creados y eliminados por los usuarios con rol de (Administrador, SuperAdministrador).</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>

<ul style="list-style-type: none"> <li>• Crear interfaz gráfica para la creación de productos donados por usuario y productos zoonosanitarios.</li> </ul>	<b>9</b>
<ul style="list-style-type: none"> <li>• Conexión al servidor.</li> </ul>	<b>5</b>
<ul style="list-style-type: none"> <li>• Validación de campos de acuerdo al tipo de dato (int, string, double, float, etc).</li> </ul>	<b>6</b>

**Tabla 119. Task card crear productos donados y zoonosanitarios**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU11:</b> Registro de productos zoonosanitarios
<b>Nombre de Tarea:</b> Interfaz gráfica para la creación de productos donados por usuario y productos zoonosanitarios.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 9 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para la creación de productos zoonosanitarios, con los campos nombre, fecha, efectivo y cantidad, adicionalmente la interfaz para la donación por usuario, se debe adicionar la información del usuario que realizó la donación y los productos asociados a esa donación.	

**Tabla 120. Task card conexión servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU11:</b> Registro de productos zoonosanitarios
<b>Nombre de Tarea:</b> Conexión servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso la creación de productos y ventas.

Tabla 121. Task card validar campos

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU11:</b> Registro de productos zoonosanitarios
<b>Nombre de Tarea:</b> Validar Campos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, )	

Tabla 122. Tarjeta colaboración HU11 Registro de productos zoonosanitarios

<b>Producto</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear productos donados</li> <li>• Listado productos donados</li> <li>• Administrar productos donados</li> <li>• Graficar indicadores</li> <li>• Autocompletar productos donados</li> <li>• Donación por usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• ProductoDonacion</li> <li>• Usuario</li> </ul>

Tabla 123. Historia de usuario: guardar productos Zoonosanitarios



<b>Historia de usuario</b>	
<b>Número:</b> HU12	
<b>Programador Responsable:</b> Team Development	
<b>Nombre de la historia:</b> Guardar productos	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 8 puntos de historia	
<b>Tiempo estimado:</b> 16 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor	
<b>Se requiere:</b> Responder a la petición de registrar productos zoonosanitarios	
<b>Para:</b> Que los usuarios visualicen la información de los productos donados	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Exista la conexión entre la base de datos y el aplicativo web.</li> <li>• Creación de las entidades Producto y Productodonacion en la base de datos.</li> <li>• Validar las peticiones del cliente.</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• El servidor valide la petición del CRUD enviada a través de la interfaz.</li> </ul>	<b>16</b>

**Tabla 124. Task card validar peticiones CRUD**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU10:</b> Guardar productos
<b>Nombre de Tarea:</b> Validación de peticiones CRUD	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 16 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Validar las peticiones del usuario para la creación, edición, actualización y visualización de los productos donados, así mismo el dinero donado, dependiendo del rol de usuario con el que esté logueado.

**Tabla 125. Tarjeta colaboración HU10 Guardar Producto**

<b>Producto</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
<ul style="list-style-type: none"> <li>• Crear productos donados</li> <li>• Listado productos donados</li> <li>• Administrar productos donados</li> <li>• Graficar indicadores</li> <li>• Autocompletar productos donados</li> <li>• Donación por usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Controller</li> <li>• ProductoDonacion</li> <li>• Usuario</li> </ul>

**Tabla 126. Historia de usuario: Seleccionar indicador**

<b>Historia de usuario</b>
<b>Número:</b> HU13
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Seleccionar Indicador
<b>Nivel de riesgo de desarrollo:</b> Alta
<b>Peso de la historia:</b> 13 puntos de historia
<b>Tiempo estimado:</b> 20 Horas
<p style="text-align: center;"><b>Perspectiva del producto</b></p> <p><b>Como:</b> Usuario</p> <p><b>Se requiere:</b> visualizar los indicadores de gestión de los cuatro comités de la fundación, bienestar animal, adopción, comunicación y donación.</p> <p><b>Para:</b> analizar el estado actual de los procesos realizados en la fundación VAP</p>

<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Se deben generar los siguientes indicadores internos a los cuatros comités.</li> <li>• Indicador del porcentaje de animales no adoptados por género</li> <li>• Indicador del número de animales no adoptados por edad y género</li> <li>• Número de procesos de animales adoptados al año</li> <li>• Número de animales adoptados por género al año</li> <li>• Indicador del resultado de c/u de los eventos realizados por la fundación</li> <li>• Indicador de productos más donados</li> <li>• Indicador de ventas por usuario</li> <li>• Indicador de donaciones por usuario</li> </ul>	
<b>Restricciones</b>	
<ul style="list-style-type: none"> <li>• Los indicadores pueden ser vistos por los usuarios con rol de (Administrador, SuperAdministrador)</li> <li>• Los indicadores pertenecientes al comité de donación los usuarios que inicie sesión podrán ver su propio indicador de venta por usuario o donaciones por usuario.</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaz gráfica de los indicadores de gestión</li> </ul>	<b>20</b>

**Tabla 127. Task card indicadores de gestión**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU13:</b> Seleccionar Indicador
<b>Nombre de Tarea:</b> Crear interfaz gráfica de los indicadores de gestión	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 13 puntos de historia
<b>Tiempo de implementación estimado:</b> 9 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Seleccionar y enviar petición al sistema del indicador a visualizar teniendo en cuenta el rol de acceso al sistema.

Tabla 128. Historia de usuario: enviar gráfica indicador

<b>Historia de usuario</b>	
<b>Número:</b> HU14	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Enviar gráfica Indicador	
<b>Nivel de riesgo de desarrollo:</b> Alta	
<b>Peso de la historia:</b> 21 puntos de historia	
<b>Tiempo estimado:</b> 30 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Servidor	
<b>Se requiere:</b> responder a la petición de indicadores de gestión de los cuatro comités de la fundación, bienestar animal, adopción, comunicación y donación.	
<b>Para:</b> analizar el estado actual de los procesos realizados en la fundación VAP	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>Exista la conexión entre la base de datos y el aplicativo web.</li> </ul>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>Enviar el indicador de gestión a la interfaz gráfica.</li> </ul>	<b>30</b>

Tabla 129. Task card indicador de gestión gráfico

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU14:</b> Enviar gráfica Indicador
<b>Nombre de Tarea:</b> Enviar el indicador de gestión a la interfaz gráfica.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 21 puntos de historia
<b>Tiempo de implementación estimado:</b> 30 Horas	
<b>Fecha Inicio:</b> 01/04/2016	<b>Fecha Fin:</b> 22/04/2016
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Recibir y consultar la petición al sistema del indicador a visualizar teniendo en cuenta el rol de acceso al sistema.                      Se deben generar los siguientes indicadores internos a los cuatros comités.</p> <ul style="list-style-type: none"> <li>• Indicador del porcentaje de animales no adoptados por género</li> <li>• Indicador del número de animales no adoptados por edad y género</li> <li>• Número de procesos de animales adoptados al año</li> <li>• Número de animales adoptados por género al año</li> <li>• Indicador del resultado de c/u de los eventos realizados por la fundación</li> <li>• Indicador de productos más donados</li> <li>• Indicador de ventas por usuario</li> <li>• Indicador de donaciones por usuario</li> </ul>	

#### 4.4.2 Diseño

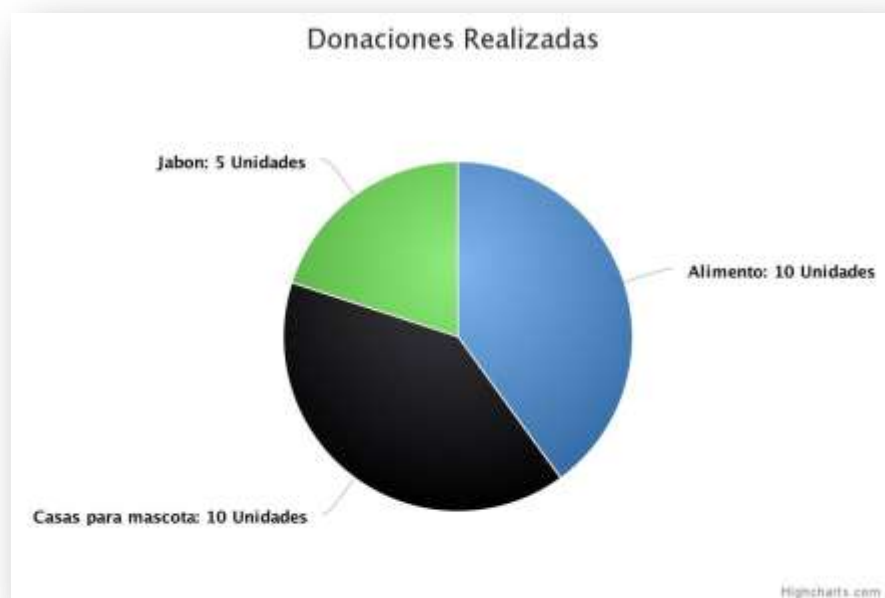
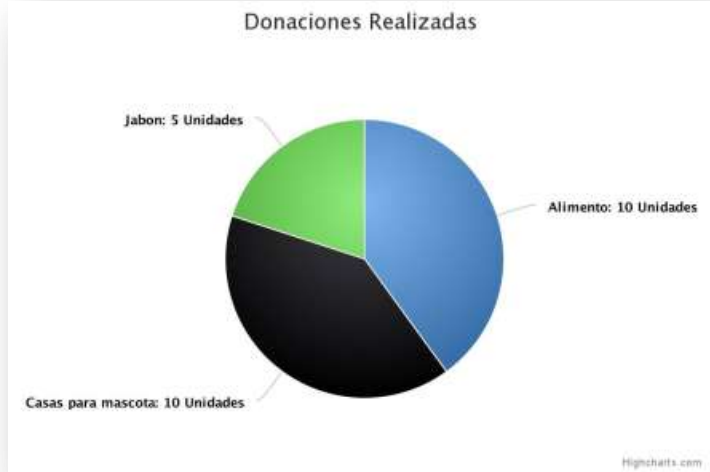


Figura 35. Indicador resultado por evento [Fuente propia]

### 4.4.3 Desarrollo

Tabla 130. Evento Programado Indicador

Evento Programado Indicador								
<b>Resultados del indicador del evento</b>								
 <p>A pie chart titled "Donaciones Realizadas" (Donations Made) showing the distribution of donations. The chart is divided into three segments: a blue segment representing "Alimento: 10 Unidades" (Food: 10 Units), a black segment representing "Casas para mascota: 10 Unidades" (Pet houses: 10 Units), and a green segment representing "Jabon: 5 Unidades" (Soap: 5 Units). The total number of units is 25. The chart is credited to Highcharts.com.</p> <table border="1"><thead><tr><th>Categoría</th><th>Unidades</th></tr></thead><tbody><tr><td>Alimento</td><td>10</td></tr><tr><td>Casas para mascota</td><td>10</td></tr><tr><td>Jabon</td><td>5</td></tr></tbody></table>	Categoría	Unidades	Alimento	10	Casas para mascota	10	Jabon	5
Categoría	Unidades							
Alimento	10							
Casas para mascota	10							
Jabon	5							
<b>Eventos Programados</b>								
<pre>260: public function actionindicacionuario(\$id) 261: { 262:     \$conex= new Productobase(); 263:     \$conexcion = Yii::app()-&gt;db; 264:     \$sql = "SELECT 265:     producto.Nombre, 266:     Sum(productobase.Cantidad) AS Val 267: FROM 268:     productobase 269: WHERE 3000 producto IN productobase.Producto_idProducto = 270:     producto_idProducto 271: AND 00 Usuario_idUsuario = '\$id' 272: GROUP BY producto.Nombre"; 273:     \$command = \$conexcion-&gt;createCommand(\$sql); 274:     \$dataReader = \$command-&gt;query(); 275:     \$rows = \$dataReader-&gt;readAll(); 276: 277:     \$conexcion = Yii::app()-&gt;db; 278:     \$sqlfeccion = "SELECT 279:     productobase.Fecha, 280:     productobase.Efectivo 281: FROM 282:     productobase 283: WHERE 3000 producto IN productobase.Producto_idProducto = 284:     producto_idProducto 285: AND 00 286:     productobase.Usuario_idUsuario = '\$id' AND productobase.Efectivo IS NOT NULL 287: GROUP BY 288:     productobase.Fecha"; 289:     \$command = \$conexcion-&gt;createCommand(\$sqlfeccion); 290:     \$dataReader = \$command-&gt;query(); 291:     \$rowsfeccion = \$dataReader-&gt;readAll(); 292: 293:     \$this-&gt;render('indicacionuario',array( 294:         'rows'=&gt;\$rows, 295:         'rowsfeccion'=&gt;\$rowsfeccion, 296:     )); 297: }</pre>								

#### 4.4.4 Pruebas

Tabla 131. Prueba de funcionalidad indicador de eventos

<b>Historia de usuario</b>	Indicadores
<b>Propósito</b>	Visualizar indicador de eventos
<b>Requisito</b>	Ser súper administrador, administrador, benefactor, voluntario, postulante, adoptante.
<b>Pasos</b>	Se ingresa al menú usuario en la opción perfil y se selecciona la opción indicador donación a continuación se desplegará el indicador con las donaciones mas realizadas y efectivo donado.
<b>Resultados esperados</b>	Un formulario con el indicador gráfico del resultado del evento.

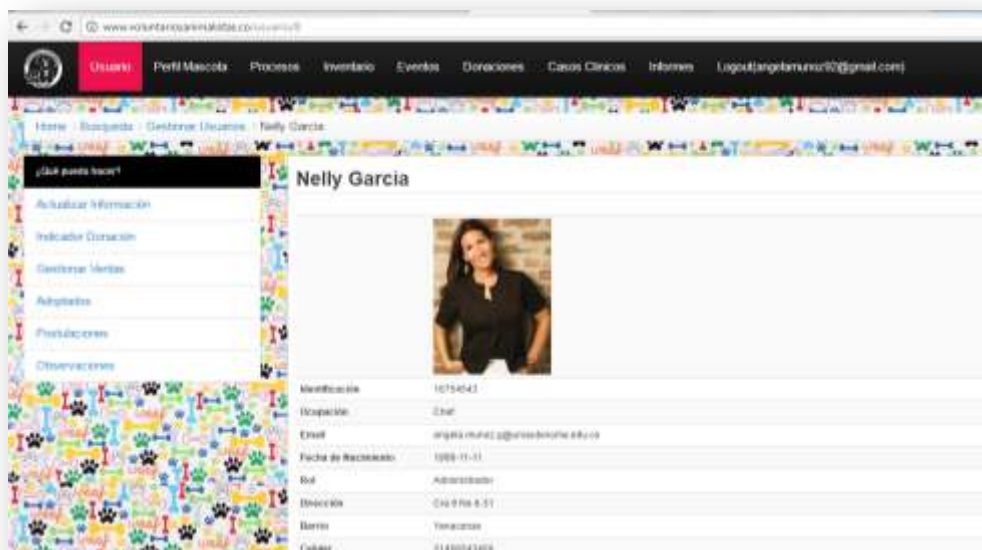


Figura 36. Interfaz gráfica perfil de usuario [Fuente propia]



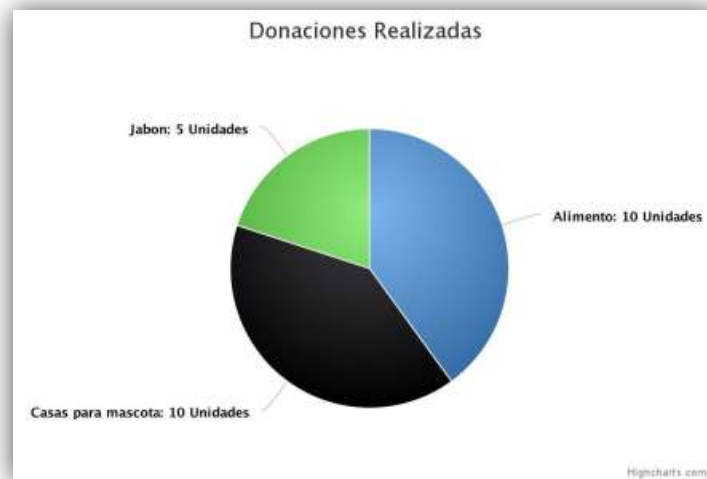


Figura 37. Interfaz gráfica indicador donación [Fuente propia]





#### 4.4.5 Sprint Review

Finalizando el sprint 3 el equipo de desarrollo entregó las funcionalidades guardar eventos, registro de productos, guardar productos, registro de productos zosanitarios, guardar productos zosanitarios, seleccionar indicador y seleccionar indicador gráfico, con un estado de “terminado” es decir software funcional al que se aplicaron pruebas de caja negra y blanca.

#### 4.4.6 Retrospectiva

Para las retrospectivas se aplicó la técnica matriz de aprendizaje puesto que permitió inspeccionar herramientas, procesos y personas identificando los elementos que salieron bien y mal creando un plan de mejoras para la ejecución del siguiente *Sprint*.

Tabla 132. Retrospectiva Sprint 3

 <ul style="list-style-type: none"><li>• Generación de CRUD por parte del framework Yii.</li><li>• Usar el repositorio remoto Bitbucket.</li><li>• Usar sublime para la codificación.</li><li>• Usar la herramienta Trello para la gestión de tareas.</li><li>• Se redujo el tiempo de aprendizaje.</li></ul>	 <ul style="list-style-type: none"><li>• Tiempo de aprendizaje.</li><li>• Uso de la librería Highcharts.</li><li>• Uso de MPDF52</li></ul>
 <ul style="list-style-type: none"><li>• Determinar el indicador gráfico a utilizar como barras y tortas.</li></ul>	 <ul style="list-style-type: none"><li>• Agradecimientos al <i>product owner</i> por mantener motivado al equipo.</li></ul>

## 5. RESULTADOS

La implementación de la plataforma web proporciona a los integrantes de la fundación “VAP” la automatización de los procesos internos de adopción, bienestar animal, comunicación y donación, mejorando la toma de decisiones y creación de estrategias a partir de la recolección y procesamiento de datos que permiten obtener información significativa.

Así pues se detalla la mejora significativa por cada uno de los comités de la fundación.

- **Comité adopción:** en el comité de adopción se manejaban los procesos internos de adopción y seguimiento de las mascotas de forma manual a través de formatos, por esta razón se encontraban formatos con la misma información, datos faltantes, incorrectos, información de contacto desactualizada por eso impedía realizar un seguimiento a las mascotas en adopción, debido a que estos procesos eran manipulados por varios integrantes de este comité en efecto dificultó la gestión. Por lo tanto se adicionaron las funcionalidades registrar proceso, usuario y animal con el propósito de permitir al usuario crear, editar, ver y eliminar procesos de manera automatizada y centralizada.
- **Comité bienestar animal:** en el comité de bienestar animal se manejaban formatos con datos característicos de los animales del hogar de paso, además de la historia clínica, casos, esterilizaciones y vacunas, por esta razón se encontraban formatos con la misma información, datos faltantes, incorrectos, información de contacto desactualizada por eso impedía realizar un seguimiento a las mascotas en adopción, debido a que estos procesos eran manipulados por varios integrantes de este comité en consecuencia obstaculizó la búsqueda y gestión de mascotas. Por lo tanto se adicionaron las funcionalidades registrar animal, consulta, vacunas,

esterilización y casos con el propósito de permitir al usuario crear, editar, ver y eliminar procesos internos del comité de manera automatizada y centralizada.

- **Comité comunicación:** el comité de comunicación se maneja a través de las redes sociales como *Facebook* usando una *fanpage* así pues la información llega a un público sesgado de quienes tengan cuenta en *Facebook* o revisen permanentemente la red social. Por tanto se adicionaron las funcionalidades registrar evento con el propósito de permitir al usuario crear, editar, ver y eliminar eventos publicados por la fundación a través de un calendario en el que se visualiza la duración, lugar, fecha y hora del evento en el aplicativo web de la fundación.
- **Comité donación:** En el comité de donación se manejaban formatos manuales con la información de la persona quien realiza la donación, el tipo de donación por ejemplo dinero en efectivo o productos zoosanitarios entre otros, como también la venta de productos por parte de los integrantes de la fundación. Por tanto se adicionaron las funcionalidades registrar producto donado, ventas e inventario con el propósito de permitir al usuario crear, editar, ver y eliminar donaciones y productos vendidos por usuario.

Con el desarrollo de las funcionalidades anteriormente mencionadas se da cumplimiento a los objetivos específicos:

- Facilitar los procesos administrativos realizados por los integrantes de los cuatros comités de adopción, donación, bienestar animal y comunicación en la Fundación “VAP”.

- Centralizar los procesos de adopción, donación, bienestar animal y comunicación en la Fundación “VAP” tecnificando el control administrativo para la toma de decisiones.

Por otro lado se agregó la funcionalidad indicador de gestión proporcionando a cada comité, información a través de indicadores gráficos como herramienta para establecer pautas y parámetros en la gestión de cada uno de los comités, evaluando las estrategias establecidas, el cumplimiento de metas, seguimiento de procesos y establecer mejoras para un control de los procesos críticos que permiten asumir responsabilidades en la toma de decisiones observando la situación actual. Por esta razón los indicadores propuestos por los integrantes de la fundación “VAP” e implementados en el aplicativo web son:

#### **Comité de Adopción**

- Número de procesos de animales adoptados al año
- Número de animales adoptados por género al año

#### **Comité de Bienestar Animal**

- Indicador del número de animales no adoptados por edad y género
- Indicador del porcentaje de animales no adoptados por género

#### **Comité de Comunicación**

- Indicador del resultado de c/u de los eventos realizados por la fundación

#### **Comité de Donación**

- Indicador de productos más donados
- Indicador de ventas por usuario

- Indicador de donaciones por usuario

Con el desarrollo de la funcionalidad anteriormente mencionada se da cumplimiento al objetivo específico:

- Proveer a los integrantes de la Fundación, información con indicadores de gestión de los comités de adopción, donación, bienestar animal y comunicación en la Fundación “VAP”.

Finalmente con el cumplimiento de los tres objetivos específicos, se ha alcanzado el objetivo general de este proyecto que es contribuir tecnológicamente en la gestión de procesos administrativos de los cuatro comités de la Fundación “Voluntarios Animalistas Popayán” a través de una plataforma web.

## 6. CONCLUSIONES

### 6.1 CONCLUSIONES

La fundación Voluntarios Animalistas Popayán, utilizaba herramientas que evidenciaban el control manual de los procesos internos, generando pérdida de tiempo al existir doble información sobre un mismo proceso o sin registro total de datos, procesos sin actualizar originando retrasos, información poco fiable e inexacta para la creación de estrategias, por esta razón surge la idea de automatizar los procesos a través del desarrollo de una plataforma web para la gestión de los procesos realizados por los comités de adopción, bienestar animal, comunicación y donación de la Fundación Voluntarios Animalistas Popayán.

Para la gestión del proyecto se utilizó el entorno de trabajo y Extreme programming como metodología de desarrollo que nos permitió trabajar en equipo aplicando el conjunto de buenas prácticas *Scrum*, también realizar entregas a tiempo aunque los requisitos fueron cambiantes y poco definidos para obtener los mejores resultados que cumplan con las necesidades del proyecto, además permite estar enfocados en el avance continuo que permita la maduración de un prototipo funcional y escalable empleando las fases de planificación, diseño, desarrollo y pruebas de la metodología Extreme programming y *Scrum* para realizar las retrospectivas.

Por otra parte el tiempo de aprendizaje para la implementación en la fase de desarrollo de la metodología Extreme Programming, con el uso del “framework Yii” no permitió cumplir con la estimación de peso y tamaño en el *Sprint 1*, por esta razón implicó aumentar las horas laborales por día para la ejecución del *Sprint 2* que debería durar en promedio 15 días calendario, para realizar la etapa de pruebas se utilizó el servidor web Openshift que incluyó un tiempo de aprendizaje y el tiempo empleado para la corrección de *bugs* presentadas en este proceso.

La plataforma suministra un valor agregado con la implementación de gráficas estadísticas que apoyan la gestión de cada uno de los comités puesto que permite analizar la situación actual de los procesos internos de cada uno en base a hechos y datos proporcionando una herramienta para la toma de decisiones y estrategias organizacionales y comerciales.

Finalmente se puede concluir que la plataforma web gestiona los procesos de adopción, bienestar animal, comunicación y donación de la fundación “VAP”, mejorando el tiempo de respuesta, centralizando la información, fácil de usar y un análisis del estado actual a través de indicadores gráficos.

## **6.2 TRABAJO FUTURO**

En nuestro proyecto de investigación se automatizaron los procesos de los cuatro comités de la fundación “Voluntarios Animalistas Popayán”, como trabajo futuro se podrá adicionar al sistema de información una pasarela de pagos o donaciones puesto que maneja el patrón de arquitectura MVC facilitando el desarrollo de nuevas funcionalidades.

Por otra parte como modelo de negocio entorno al desarrollo como propuesta a las clínicas veterinarias de la región caucana es permitirles manejar publicidad de sus servicios veterinarios en la plataforma web.



## 7. BIBLIOGRAFÍA

- [1] OMS, "Rabia." [Online]. Available: <https://goo.gl/ak5kCP>. [Accessed: 04-Apr-2016].
- [2] El tiempo, "El trágico destino de las mascotas que se regalan en Navidad," 15/01/2015. [Online]. Available: <https://goo.gl/ijtyIW>. [Accessed: 04-Feb-2016].
- [3] Y. Ospina, "El drama que viven los animales callejeros en Cali," 24/11/2014. [Online]. Available: <https://goo.gl/dqRCmY>. [Accessed: 01-Feb-2016].
- [4] A. Manzano, "Reporte 2015 Fundación VAP," 2015.
- [5] C. Villaquiran, "Reporte 2015 Fundación Vida Animal," 2015.
- [6] M. R. Rengifo, "Reporte 2015 Fundación Rescate y bienestar Animal," 2015.
- [7] L. F. Velasco, Por medio de la cual se modifican el código civil, la ley 1774 de 2016, el código penal, el código de procedimiento penal y se dictan otras disposiciones.
- [8] A. C. and A. Riera, "Estudio de la problemática actual de adopción de mascotas en la ciudad de Guayaquil y desarrollo de un aplicativo web para agilizar este proceso," Universidad Católica de Santiago de Guayaquil, 2014
- [9] W. L. Sigüencia, "Desarrollo e implementación de un sistema web para la administración de la clínica veterinaria de la Fundación Protección Animal en Ecuador," Universidad Politécnica Nacional, 2012.
- [10] M. Calderón, "Fundación voz animal." [Online]. Available: <http://www.fundacionvoanimal.org>. [Accessed: 03-Feb-2016].
- [11] A. C. and A. Riera, "Estudio de la problemática actual de adopción de mascotas en la ciudad de Guayaquil y desarrollo de un aplicativo web para agilizar este proceso," Universidad Católica de Santiago de Guayaquil, 2014.
- [12] E. Tascon, "Fundación Animal Safe." [Online]. Available: <http://www.fundacionanimalsafe.com>. [Accessed: 03-Feb-2016].
- [13] O. Villegas, "Fundación Paraíso de Mascota." [Online]. Available: <http://paraisodelamascota.org/>. [Accessed: 01-Feb-2016].

- [14] Animanaturalis, “Adopción de perros y gatos en Colombia.” [Online]. Available: <https://goo.gl/ZCHyuL>. [Accessed: 04-Feb-2016].
- [15] MINTIC, “Sistemas de Información.” [Online]. Available: <https://goo.gl/sgTK4Y>. [Accessed: 29-Jun-2016].
- [16] V. Moreno, “Análisis y diseño de una plataforma web para un sistema de gestión de usuarios,” Universidad Calor III de Madrid, 2012.
- [17] G. Martínez, G. Camacho and D. Biancha, “Diseño de framework web para el desarrollo dinámico de aplicaciones,” sci. tech., vol. 44, no. issn 0122–1701, p. 178, 2010
- [18] Q.Xue, “Yii framework.” [Online]. Available: <https://goo.gl/FcCG18> [Accessed: 04-Feb-2016].
- [19] Sinergia, “Dirección Ágil de Proyectos,” 2016. [Online]. Available: <https://goo.gl/14ng8s>. [Accessed: 20-Jul-2016].
- [20] P. L. and M. C. P. J.H.Canos, “Metodologías Ágiles en el Desarrollo de Software,” Universidad Politécnica de Valencia
- [21] M. Alaimo, “Introducción a la Agilidad y Scrum,” KLEER. pp. 7–19.
- [22] J. Megias, “Lean canvas, un lienzo de modelos de negocio para startups,” 30 OCT 2012. [Online]. Available: <https://goo.gl/mly0SW>. [Accessed: 20-Jul 2016].
- [23] A. Goyén and Fros, “Problemas y soluciones en la implementación de extreme programming,” universidad católica del Uruguay, 2001.
- [24] [1] Rae, “Donación,” 2016. [Online]. Available: <http://dle.rae.es/?id=E7oF9KZ>. [Accessed: 20-Jul-2016].
- [25] Kent Beck, Extreme Programming Explained. 1999.
- [26] J. Joskowicz, “Reglas y Prácticas en eXtremeProgramming,” 2008.
- [27] Y. Fernández Romero and Y. Díaz Gonzales “Patrón Modelo-Vista-Controlador.,” Rev. Telem@tica, vol. 11, no. 1, pp. 47–57, 2012.
- [28] M. Fowler and J. Highsmith, “The agile manifesto,” Softw. Dev. vol. 9, no. August, pp. 28–35, 2001.

- [29] "Scrum," 09/17/2014. [Online]. Available:<https://goo.gl/sj76bs>. [Accessed: 20-Jul-2016]
- [30] K. Schwaber and J. Sutherland, "La Guía de Scrum," 2013.
- [31] Agile & Scrum, "Los tres pilares de Scrum." [Online]. Available: <https://goo.gl/846QPW> [Accessed: 20-Jul-2016].
- [32] Tecnología, "Scrum roles y responsabilidades," marzo, 2015. [Online]. Available: <https://goo.gl/OT6hEJ>. [Accessed: 20-Jul-2016].
- [33] J. Párraga, "Gestión de proyectos, metodología de desarrollo ágil: Scrum," 7 Julio 2014. [Online]. Available: <https://goo.gl/frjNGW>. [Accessed: 20-Jul-2016].
- [34] [1] Rae, "Donación," 2016. [Online]. Available: <http://dle.rae.es/?id=E7oF9KZ>. [Accessed: 20-Jul-2016].
- [35] "Agile inception: una buena práctica en la iniciación y reorientación de proyectos y productos." [Online]. Available:<https://goo.gl/FLmYvj>. [Accessed: 20-Jul-2016].
- [36] P. Hundermark, "Un Mejor Scrum," 2009.
- [37] Adobe, "Adobe Color," 2016. [Online]. Available: <https://goo.gl/4Rm9Rj> [Accessed: 07-Jul-2016].
- [38] FAQs, "Adobe® kuler explore+create+share," 2011. [Online]. Available: <https://goo.gl/cXtcQu>. [Accessed: 20-Jul-2016]
- [39] Excentia, "Bitbucket y Stash," 2016. [Online]. Available: <https://goo.gl/dlzbzt>. [Accessed: 20-Jul-2016].
- [40] Atlassian, "Bitbucket," 2016. [Online]. Available: <https://bitbucket.org/>. [Accessed: 20-Jul-2016].
- [41] Tutorialspoint, "Bootstrap." 2014.
- [42] Highcharts, "What is Highcharts?," 2016. [Online]. Available: <http://www.highcharts.com/products/highcharts>. [Accessed: 20-Jul-2016]
- [43] MySQL Workbench Manual, "Chapter 1 General Information," 2016. [Online]. Available:<https://goo.gl/bu8o4G>. [Accessed: 20-Jul-2016]

- [44] Openshift, “Develop, Host, and Scale Your Apps in the Cloud,” 2016. [Online]. Available: <https://www.openshift.com/>. [Accessed: 20-Jul-2016].
- [45] Madeja, “PHPMailer,” 2015. [Online]. Available: <https://goo.gl/GEx6Of>. [Accessed: 20-Jul-2016].
- [46] K. Lira, O. López, and R. Gómez, “Aplicación móvil para el seguimiento del control prenatal médico-paciente,” Universidad Nacional Autónoma de Nicaragua UNAN- león.
- [47] [1] Rae, “Donación,” 2016. [Online]. Available: <http://dle.rae.es/?id=E7oF9KZ>. [Accessed: 20-Jul-2016].
- [48] Agiles, “Estimación y planificación ágil – Resultados del quinto encuentro ágil en Barcelona,” 1 julio, 2009. [Online]. Available: <https://goo.gl/9iJLqh>. [Accessed: 20-Jul-2016].
- [49] E. Misuf, “Apache,” Aula mentor, no. 978–84–369–5443–2.
- [50] J. Sanchez, “Sublime text 2,” 2012.
- [51] J. Skinner, “Sublime Text3,” 2008. [Online]. Available: <https://www.sublimetext.com/>. [Accessed: 03-May-2016].
- [52] [1] Rae, “Donación,” 2016. [Online]. Available: <http://dle.rae.es/?id=E7oF9KZ>. [Accessed: 20-Jul-2016].
- [53] StarUml, “About starUml,” 2005. [Online]. Available: <https://goo.gl/MCcpnS> [Accessed: 04-Feb-2016].
- [54] Trello, “Organización de un proyecto con Trello,” 2016. [Online]. Available: <https://trello.com/guide/organize.html>. [Accessed: 20-Jul-2016].
- [55] [1] Rae, “Donación,” 2016. [Online]. Available: <http://dle.rae.es/?id=E7oF9KZ>. [Accessed: 20-Jul-2016].
- [56] Asociación De Academias De La Lengua Española, “Activista,” 2016. [Online]. Available: <http://dle.rae.es/?id=0clYNCh>. [Accessed: 01-Feb-2015].
- [57] Secretaria de Salud, “Programa de Adopción,” 2012. [Online]. Available: <https://goo.gl/ZTpcHa>. [Accessed: 05-Feb-2016].
- [58] Asociación de Academias de la lengua Española, “Albergue.” [Online]. Available: <http://dle.rae.es/?id=1XOT8ne>. [Accessed: 04-Feb-2016].
- [59] Rae, “Donación,” 2016. [Online]. Available: <http://dle.rae.es/?id=E7oF9KZ>.

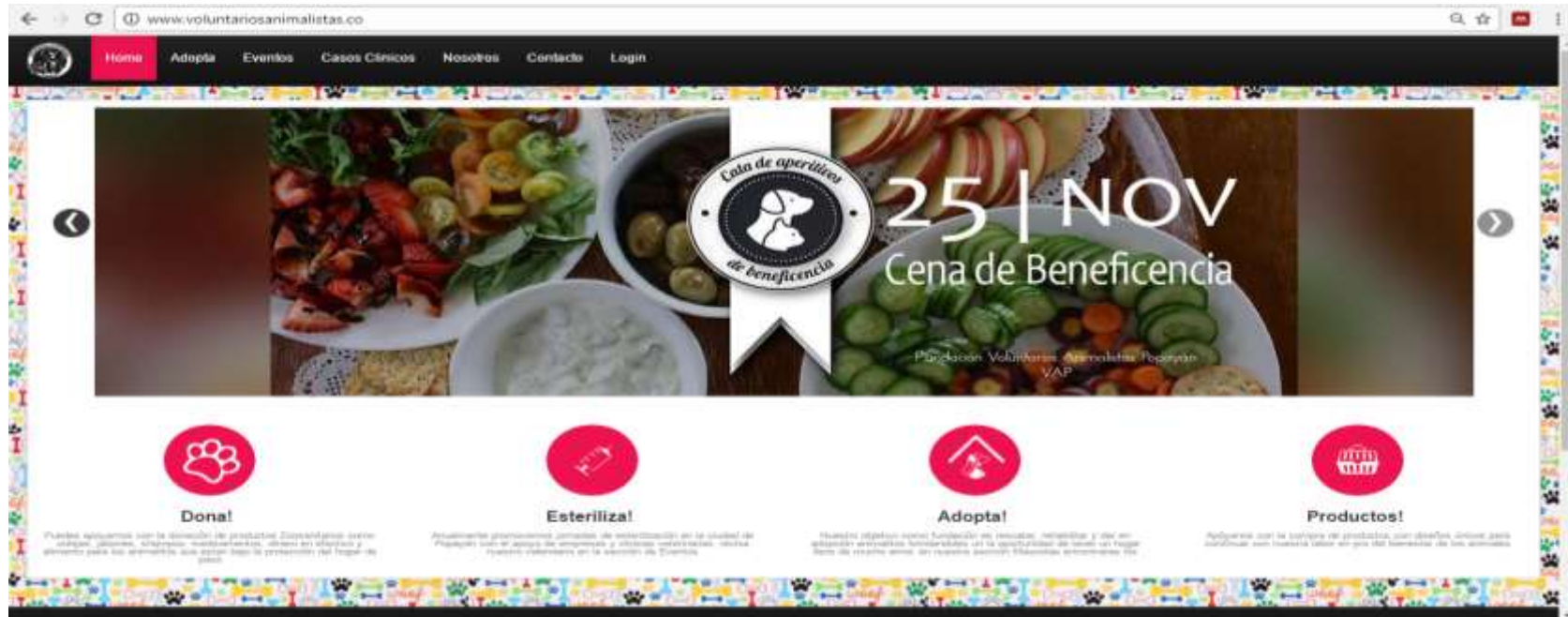
[Accessed: 20-Jul-2016].

- [60] L. J. Guamán, “La esterilización para evitar la sobrepoblación canina en el sector de la fragata de la ciudad de Guayaquil año 2014,” Universidad de Guayaquil, 2014.
- [61] M. T. FRANCO, Por medio del cual se establece una Política Pública para la protección integral de la fauna del Municipio de Medellín y se adicionan los Acuerdos números 32 de 1997; 25 y 42 de 2002. 2007.
- [62] Asociación de Academias de la lengua Española, “Fundación.” [Online]. Available: <http://dle.rae.es/?id=lbGShUM>. [Accessed: 04-Feb-2016]
- [63] Asociación de Academias de la lengua Española, “Mascota.” [Online]. Available: <http://dle.rae.es/?id=OWI1ptF>. [Accessed: 04-Jun-2016]
- [64] M. Ivanovic, “Un decálogo animalista”, Revista de bioética y derecho, vol. 22, pp. 56-66, Mayo 2011
- [65] K. Schwaber and J. Sutherland, “La Guía de Scrum,” 2013.
- [66] P. Hundermark, “Un Mejor Scrum,” 2009.
- [67] Asociación de Academias de la lengua Española, “Iteración.” [Online]. Available: <https://goo.gl/WC26M6>. [Accessed: 04-Jun-2016]
- [68] AcensTechnologies, “Framework para el desarrollo ágil de aplicaciones.” [Online]. Available: <https://goo.gl/QEtCTC>. [Accessed: 04-Jun-2016].
- [69] Universidad de Alicante, “Modelo vista controlador (MVC).” [Online]. Available: <https://goo.gl/MnzOJE>. [Accessed: 26-Jun-2016].
- [70] Asociación de Academias de la lengua Española, “Metodología.” [Online]. Available: <http://dle.rae.es/?id=P7eTCPD>. [Accessed: 04-Jun-2016].
- [71] Asociación de Academias de la lengua Española, “Test.” [Online]. Available: <http://dle.rae.es/?id=ZeapftD>. [Accessed: 04-Jun-2016].
- [72] V. Moreno, “Análisis y diseño de una plataforma web para un sistema de gestión de usuarios,” Universidad Calor III de Madrid, 2012.
- [73] J. H. Canos, P Letelier and M.C. Penadés, “Metodologías Ágiles en el Desarrollo de Software,” Universidad Politécnica de Valencia, 2013.

- [74] TrelloInc, "Trello," 2016. [Online]. Available: <https://trello.com/>. [Accessed: 01-Feb-2016].
- [75] StarUml, "StarUml," 2005. [Online]. Available: <http://staruml.io/>. [Accessed: 04-Feb-2016].
- [76] PHP Group, "PHP," 2016. [Online]. Available: <https://goo.gl/49OySd>. [Accessed: 01-Feb-2016].
- [77] Librosweb.es, "JSON," 2016. [Online]. Available: <https://goo.gl/q2hXml>. [Accessed: 04-Feb-2016].
- [78] J. Skinner, "Sublime Text 2," 2012. [Online]. Available: <https://www.sublimetext.com/>. [Accessed: 03-May-2016].
- [79] Librosweb.es, "CSS," 2016. [Online]. Available: <https://goo.gl/kxQ489>. [Accessed: 04-Feb-2016].
- [80] S. Ramos, "MODELO CLIENTE SERVIDOR," 2015. [Online]. Available: <http://clienteservidorrr.weebly.com/>. [Accessed: 20-Jul-2016]
- [81] M. Villamil and C. Vidal, "Diseño e implementación del sistema de información web y canal de distribución para la gestión ágil y presentación de la información socioeconómica de los 42 municipios del departamento del cauca en la oficina asesora de planeación departamental," 2015

## ANEXO

### INTERFACES GRÁFICAS DE LA APLICACIÓN WEB



Interfaz gráfica 1. Inicio

Interfaz gráfica 1: Esta interfaz gráfica es la página principal del sistema, brinda al usuario una barra de navegación con las funcionalidades del sistema que podrá ser usado tanto por los integrantes del comité de la fundación como la comunidad caucana interesada en el cuidado de la fauna callejera promoviendo la adopción de caninos y felinos en estado de abandono, donaciones e información acerca de los eventos que realizara la fundación con el fin de gestionar todos los procesos realizados por los cuatro comités de la “Fundación Voluntarios Animalistas Popayán”

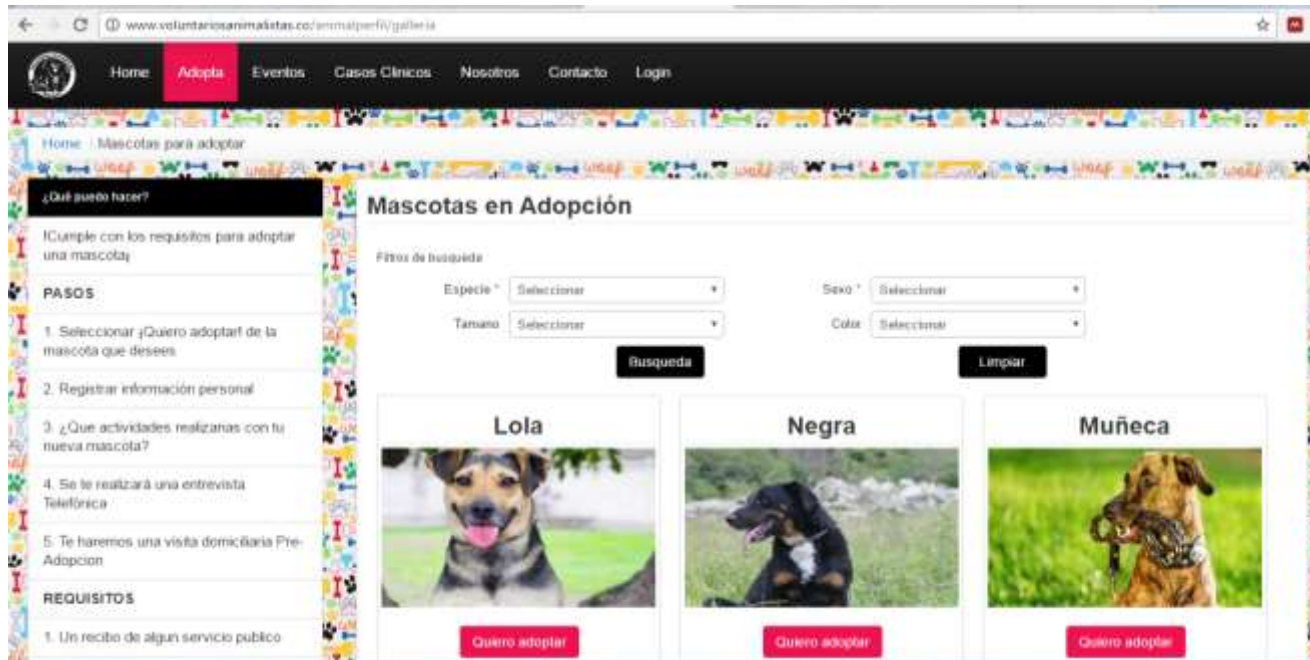
además de brindar al usuario una plataforma con interfaces responsivas, es decir adaptable a cualquier tipo de dispositivo.



**Interfaz gráfica 2. Información fundación “VAP”**

Interfaz gráfica 2: En esta interfaz se encuentra información relacionada con la fundación “Voluntarios Animalistas Popayán”, como el equipo de trabajo, misión, visión y algunas formas de apoyo para contribuir con los animales.



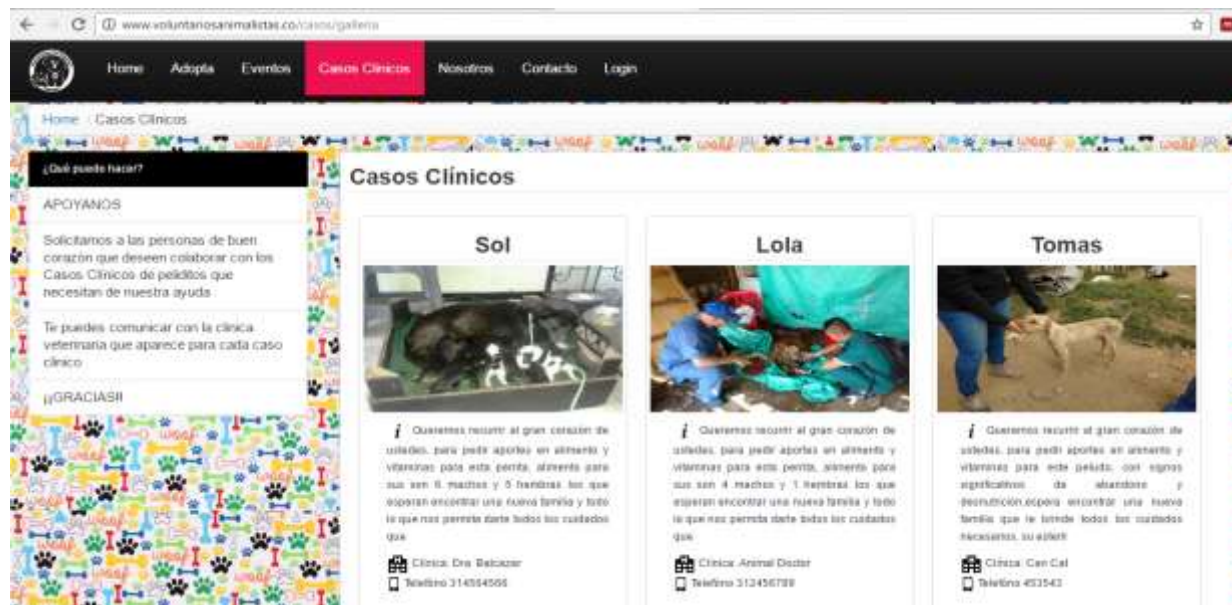


**Interfaz gráfica 3. Interfaz mascotas en adopción**

Interfaz gráfica 3: En esta interfaz se presentan las mascotas en adopción que se encuentran bajo la protección de la fundación “Voluntarios Animalistas Popayán” con información básica de la mascota, además de los requisitos para dar inicio al proceso de adopción entre los requisitos se encuentran:

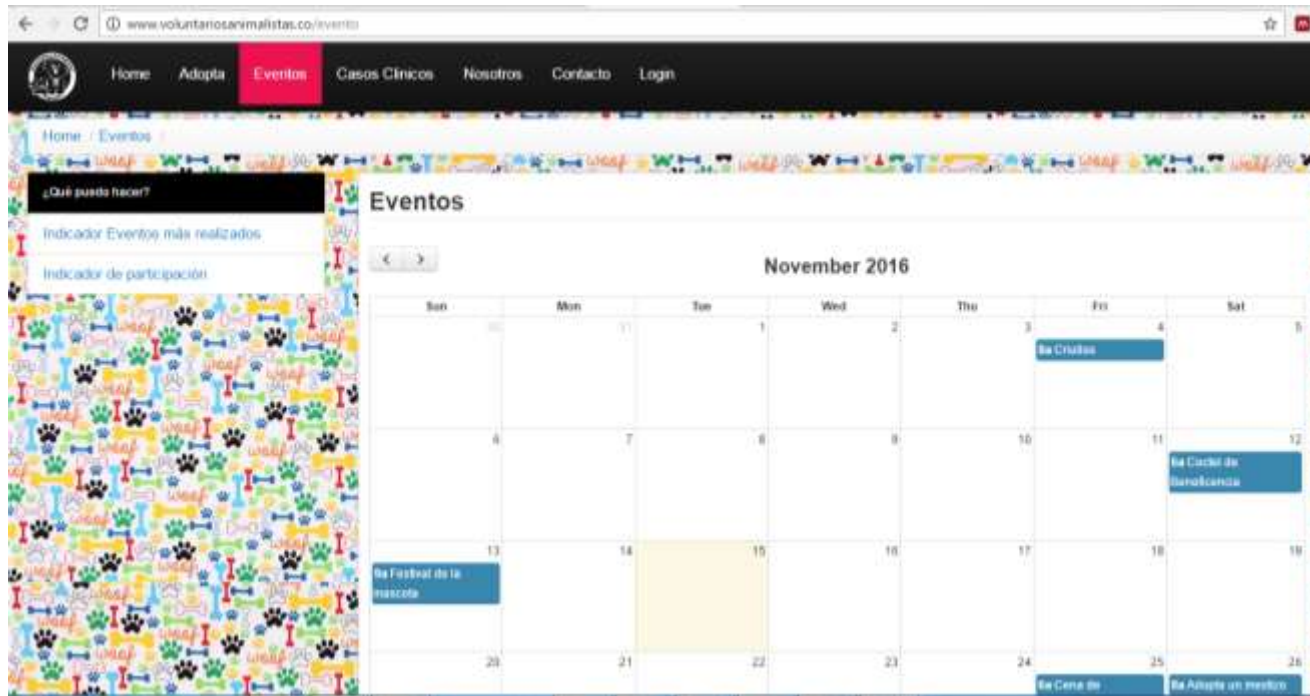
1. Ser mayor de edad
2. Postularse
3. Entrevista telefónica
4. Visita domiciliaria pre-adopción

5. Un recibo de algún servicio publico
6. Fotocopia de cedula
7. Esterilización
8. Donación



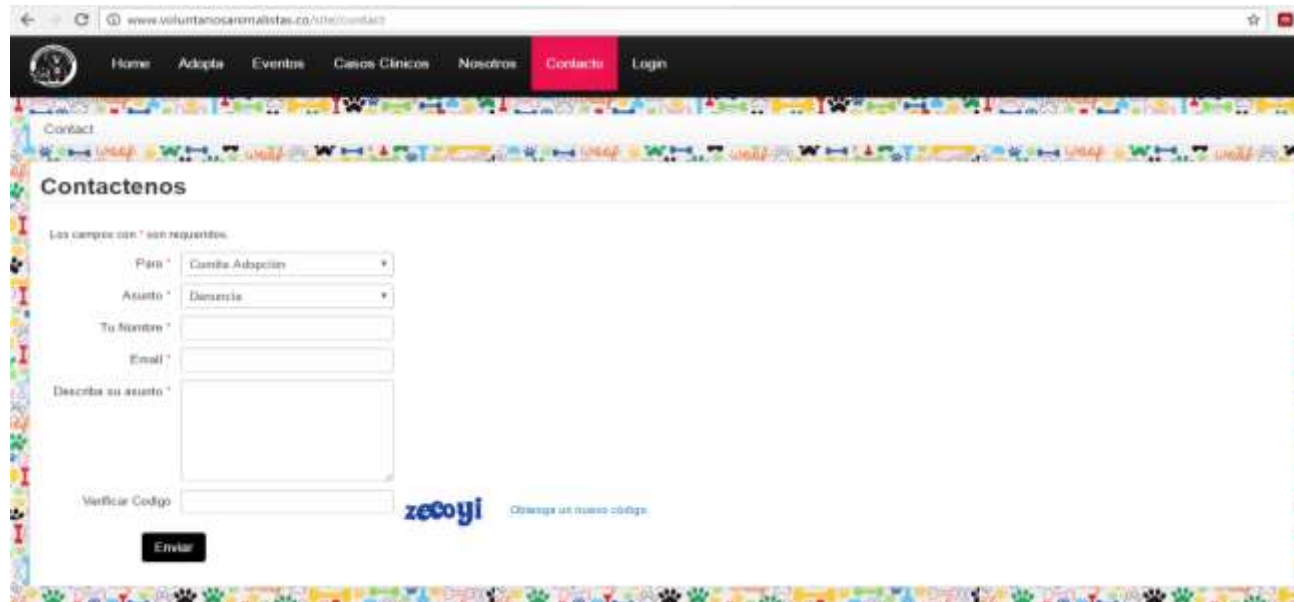
**Interfaz gráfica 4. Interfaz gráfica casos clínicos**

Interfaz gráfica 4: En esta interfaz gráfica se evidencian los casos clínicos que la fundación “Voluntarios Animalistas Popayán” apoya y que están directamente relacionados con clínicas veterinarias.



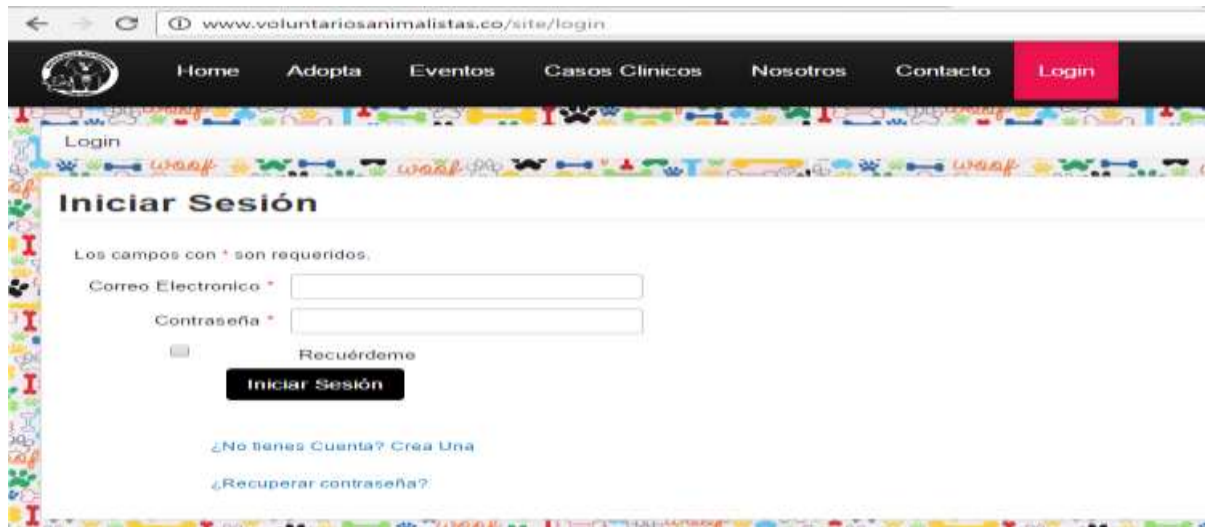
**Interfaz gráfica 5. Eventos “VAP”**

Interfaz gráfica 5: En esta interfaz se pueden visualizar los eventos patrocinados por la fundación, especificando fecha, hora y lugar donde se realizara el evento.



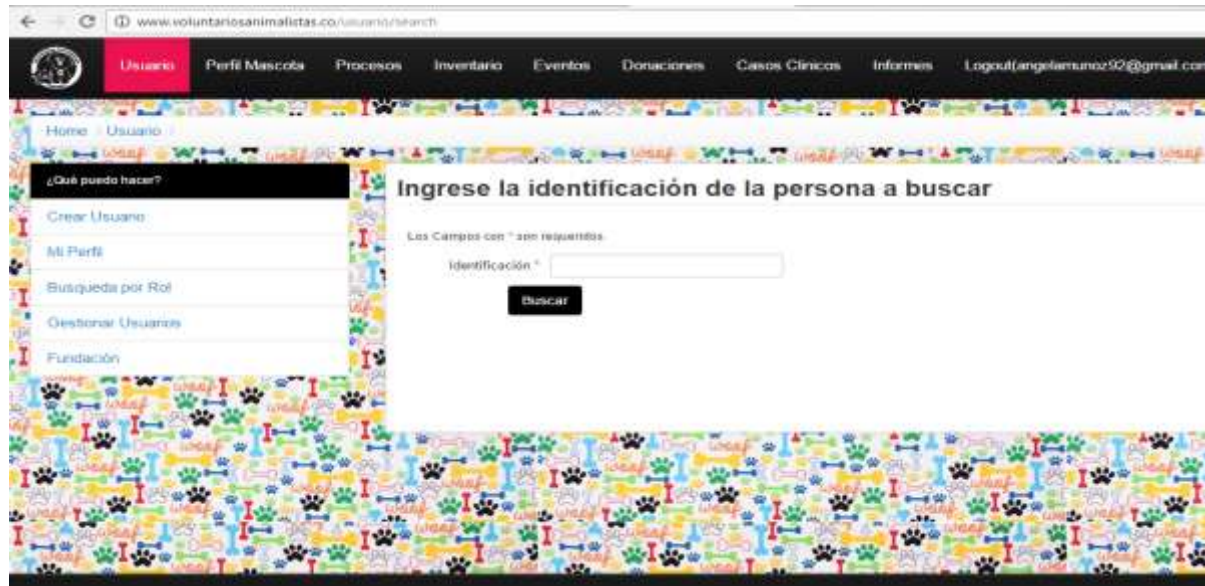
**Interfaz gráfica 6. Interfaz contáctanos**

Interfaz gráfica 6: A través de esta interfaz los usuarios podrán comunicarse con la fundación “Voluntarios Animalistas Popayán”



**Interfaz gráfica 7. Interfaz de logueo**

Interfaz gráfica 7: Los usuarios registrados en la plataforma tendrán acceso de acuerdo al rol asignado



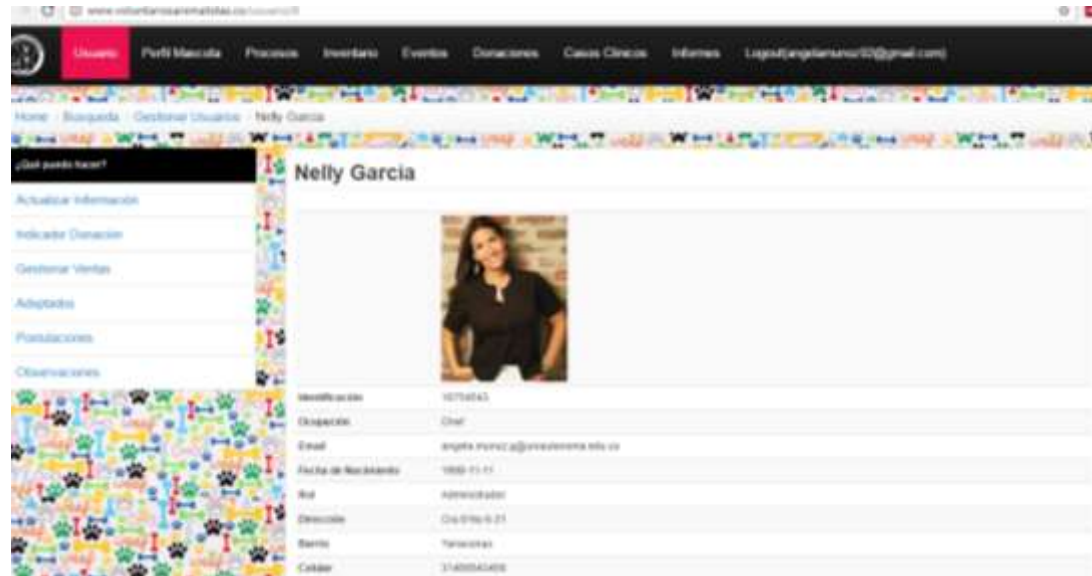
**Interfaz gráfica 8. Interfaz búsqueda usuario**

Interfaz gráfica 8: A través de esta interfaz gráfica se podrán buscar los usuarios registrados en la plataforma por el número de identificación, también tendrá acceso para crear un usuario, ver su perfil y realizar búsquedas por rol estas acciones están limitadas al rol de usuario.



**Interfaz gráfica 9. Interfaz crear usuario**

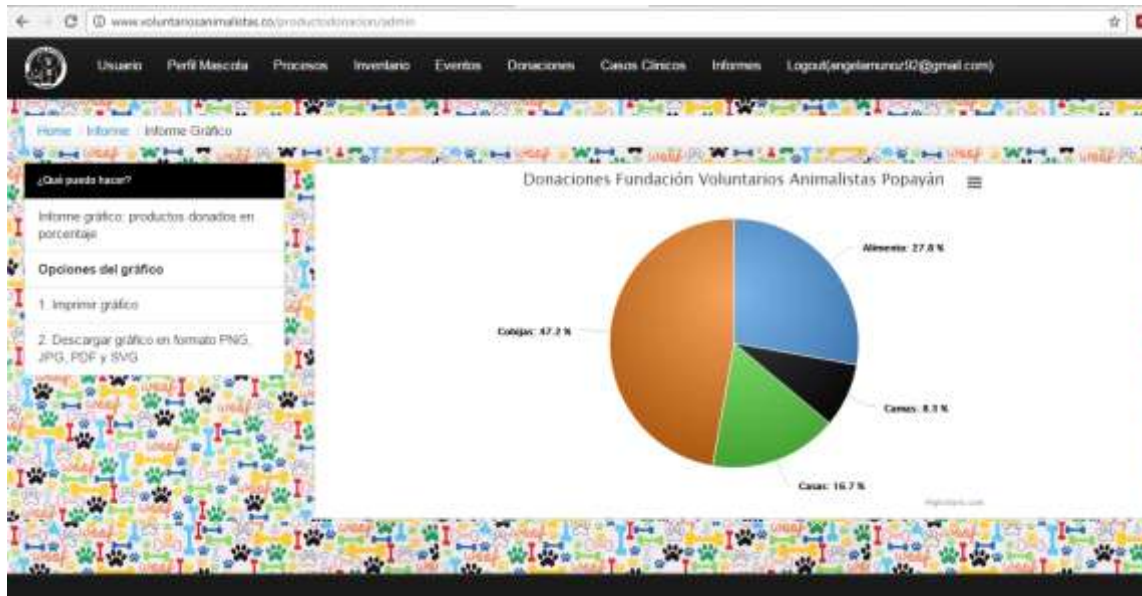
Interfaz gráfica 9: En esta interfaz se puede crear un usuario con información básica además de registrar una referencia



**Interfaz gráfica 10. Perfil de usuario**

Interfaz gráfica 10: En esta interfaz el usuario podrá ver su perfil con su información básica registrada, también los datos de la referencia personal, además de las opciones de indicador de productos donados y ventas.





**Interfaz gráfica 11. Indicador de donación por usuario**

Interfaz gráfica 11: En esta interfaz el usuario podrá observar a través de un indicador gráfico las donaciones tanto en productos como la cantidad de dinero en efectivo donado.



**Interfaz gráfica 12. Interfaz venta**

Interfaz gráfica 12: El usuario podrá registrar las ventas de los productos de la fundación

www.voluntariosanimalistas.co/venta/administrador/4

Usuario Perfil Mascota Procesos Inventario Eventos Donaciones Casos Clínicos Informes Logout(angelamunoz82@gmail.com)

Home Perfil Usuario / Angela Maria Muñoz Garcia / Ventas

¿Qué puedo hacer?

Crear Venta

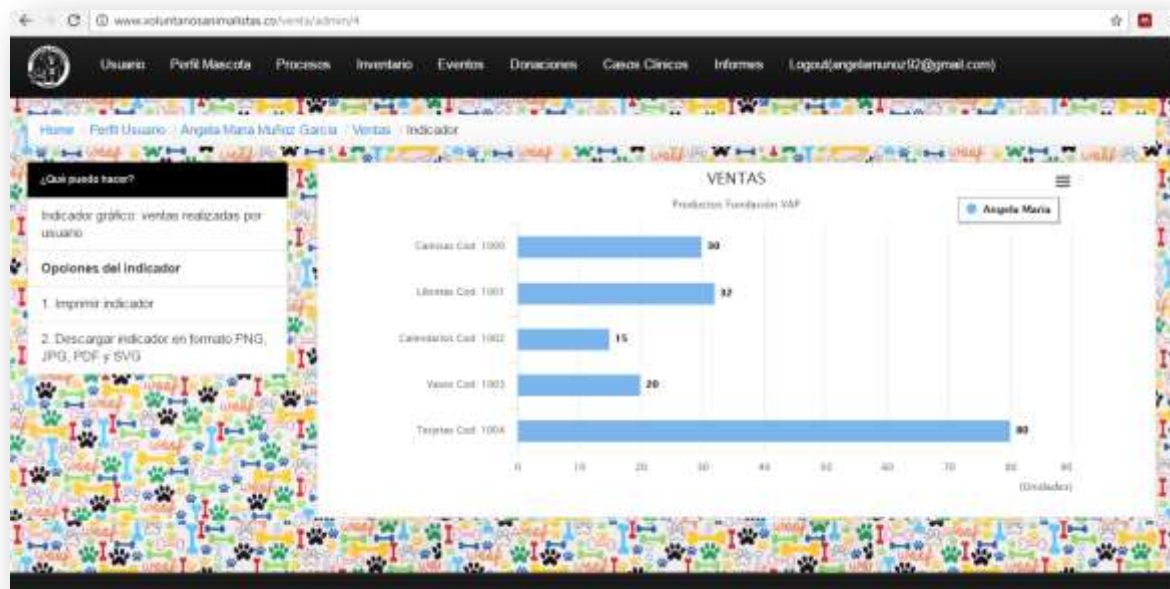
Indicador de venta

### Gestionar Ventas de: Angela Maria Muñoz Garcia

No	Fecha Venta	Cantidad Vendida	Codigo Artículo	Nombre Artículo	
1	2016-01-20	15	1002	Calendarios	👁️ ✎️ 🗑️
2	2016-05-01	32	1001	Libretas	👁️ ✎️ 🗑️
3	2016-07-06	80	1004	Tarjetas	👁️ ✎️ 🗑️
4	2016-06-15	20	1003	Vests	👁️ ✎️ 🗑️
5	2016-01-20	35	1000	Carnitas	👁️ ✎️ 🗑️

**Interfaz gráfica 13. Interfaz ventas por usuario**

Interfaz gráfica 13: En esta interfaz el usuario podrá visualizar las ventas realizadas, además del indicador de ventas.



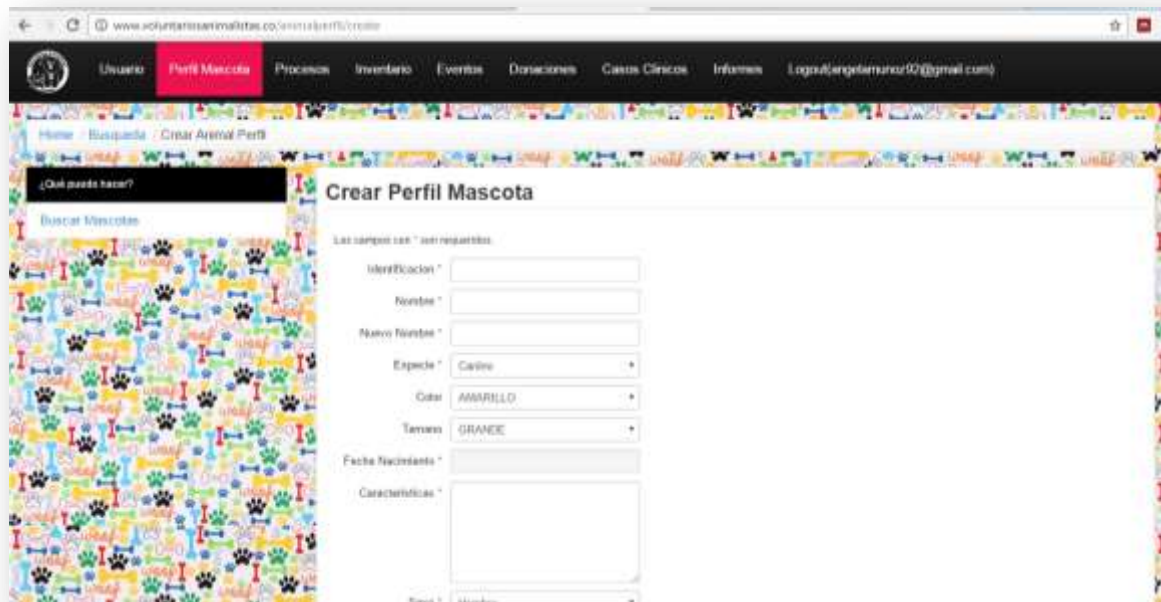
**Interfaz gráfica 14. Indicador gráfico de venta por usuario**

Interfaz gráfica 14: El usuario podrá visualizar las ventas realizadas a través de un indicador gráfico.



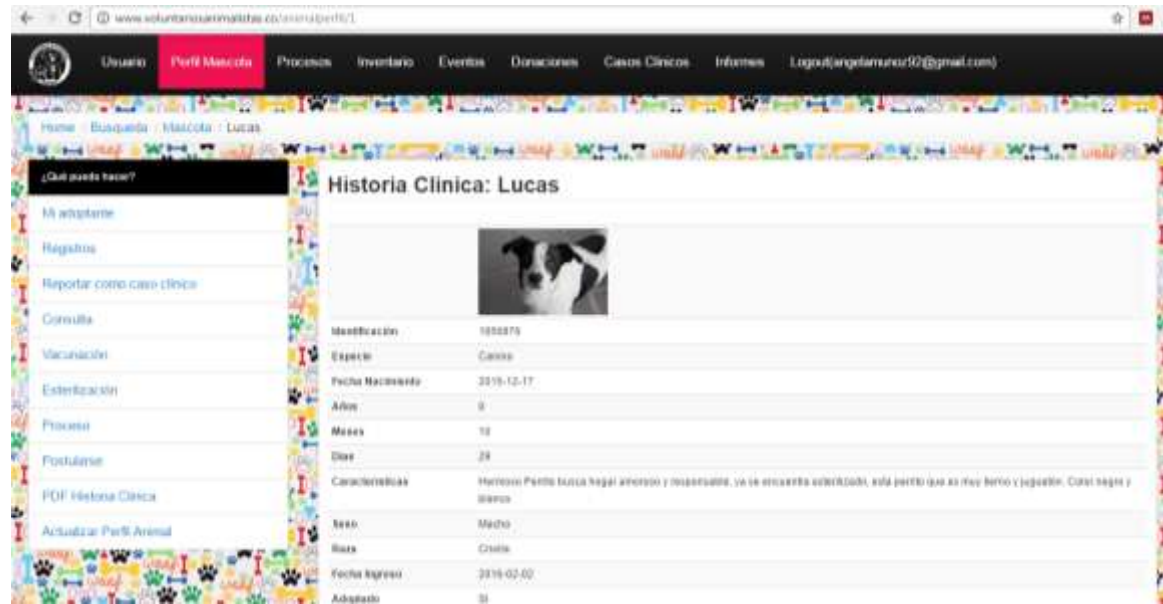
**Interfaz gráfica 15. Búsqueda de mascotas**

Interfaz gráfica 15: En esta interfaz el usuario podrá buscar los animales registrados por el nombre, además de consultar indicadores de adopción, edad, proceso y sexo; también podrá actualizar la información de la mascota y crear mascota.



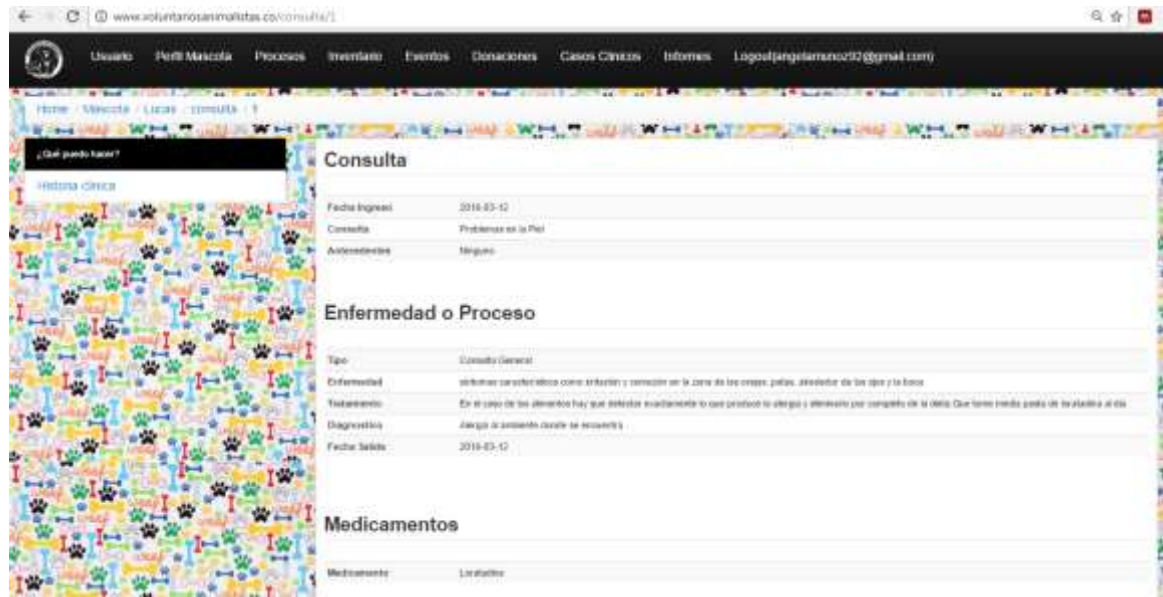
**Interfaz gráfica 16. Interfaz crear perfil mascota**

Interfaz gráfica 16: En esta interfaz el usuario podrá crear un perfil para las mascotas que se encuentran bajo el cuidado de la fundación “Voluntarios Animalistas Popayán” registrando información básica.



**Interfaz gráfica 17. Historia clínica de la mascota**

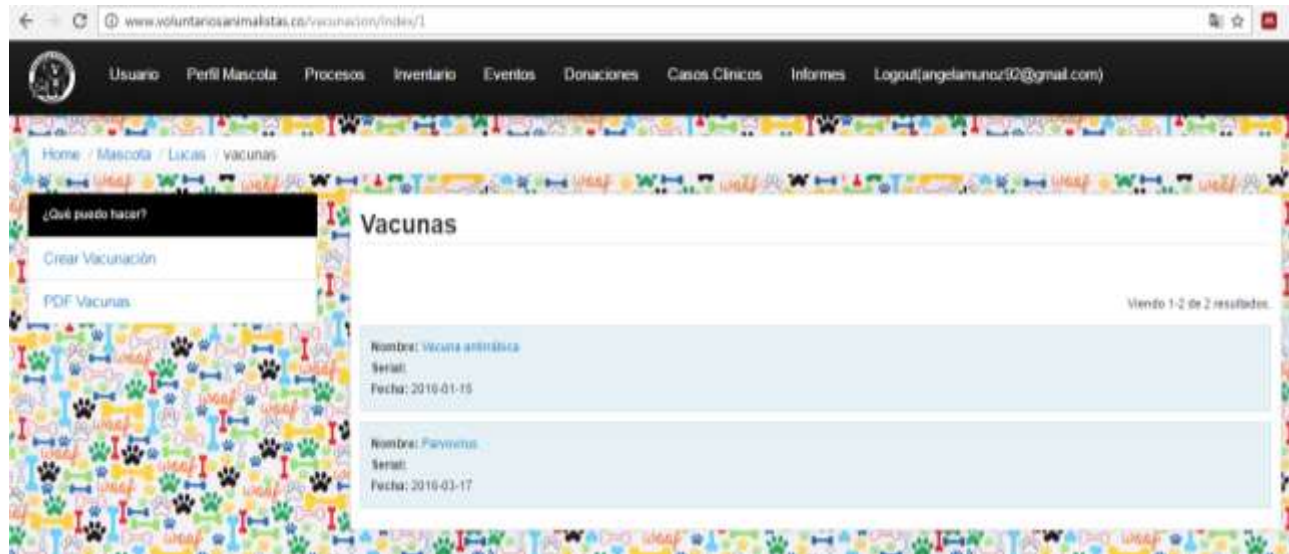
Interfaz gráfica 17: En esta interfaz el usuario podrá visualizar la información de la mascota creada o buscada, además de visualizar la información de la mascota podrá seleccionar las opciones de consulta, vacunación, esterilización, procesos en este caso de adopción, seguimiento y postulación, postularse a esta mascota si se encuentra en estado “NO” es decir que no está adoptada, actualizar la información de la mascota, como también eliminar o imprimir la información de la mascota teniendo en cuenta que algunas opciones solo serán ejecutadas por el administrador de la plataforma.



**Interfaz gráfica 18. Interfaz consultas de mascotas**

Interfaz gráfica 18: En esta interfaz gráfica el usuario administrador y adoptante podrá visualizar las consultas al veterinario con información básica registrada, como descripción de la consulta, la enfermedad el tratamiento que se le receto, los medicamentos además de la opción eliminar la consulta opción exclusiva para el administrador





**Interfaz gráfica 19. Listar Vacunas**

Interfaz gráfica 19: En esta interfaz gráfica el usuario podrá visualizar la lista de vacunas realizadas a la mascota consultada

www.voluntariosanimalistas.co/esterilizacion/1

Usuario Perfil Mascota Procesos Inventario Eventos Donaciones Casos Clínicos Informes Log

Home / Mascota / Lucas / Esterilización /

¿Qué puedo hacer?

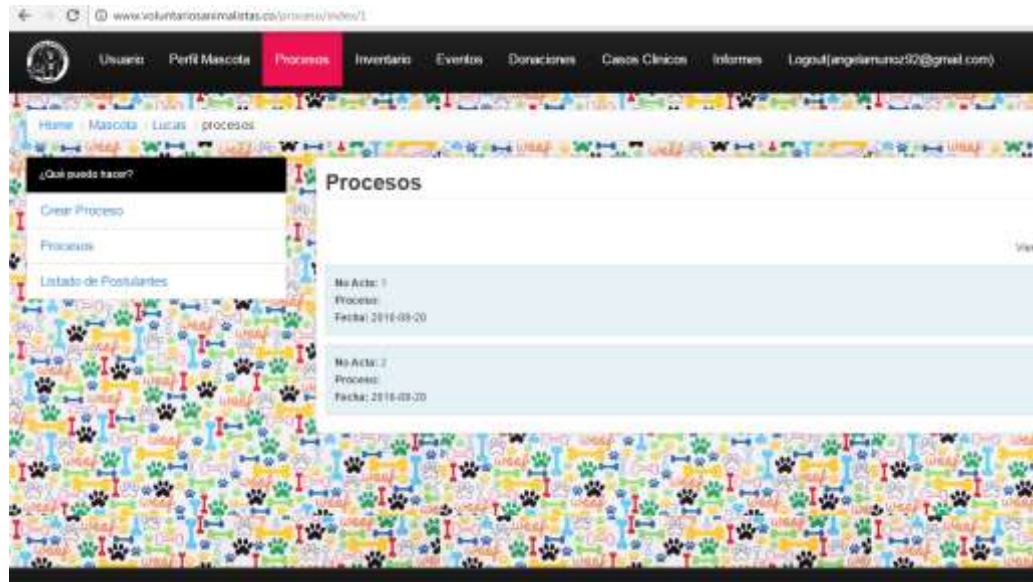
- Actualizar Esterilización
- Abonos

### Esterilización: Lucas

Fecha Esterilización	2016-08-10
costo	70000
Abonos	30000
Saldo	40000
Esterilizado	Si

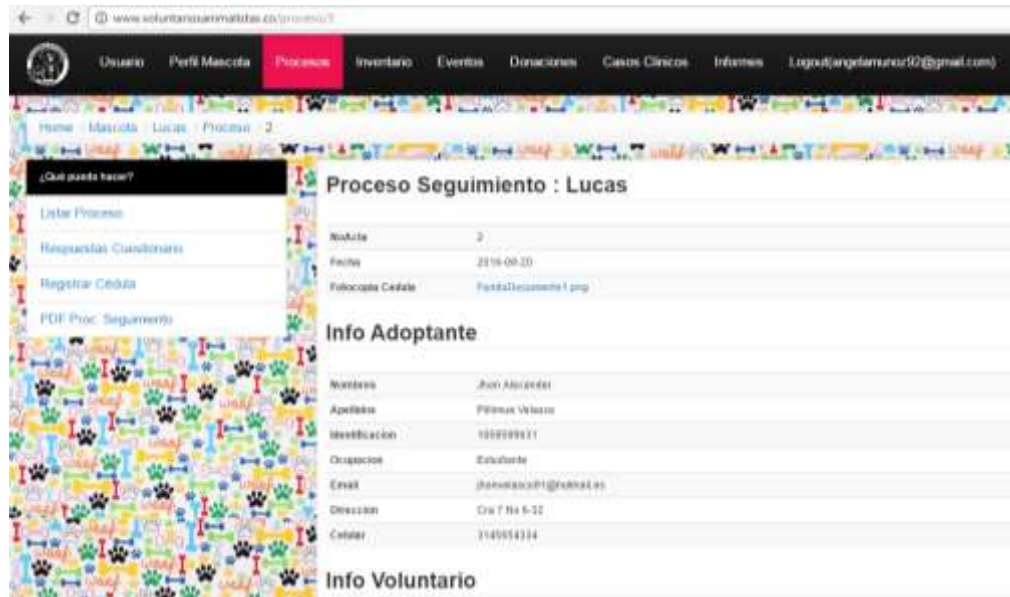
#### Interfaz gráfica 20. Visualizar esterilización

Interfaz gráfica 20: En esta interfaz gráfica el usuario administrador podrá visualizar información acerca de la esterilización realizada a la mascota consultada.



**Interfaz gráfica 21. Visualizar procesos**

Interfaz gráfica 21: En esta interfaz gráfica el usuario administrador podrá visualizar la lista de procesos realizados a la mascota consultada, como procesos de adopción, seguimiento y postulación, además de las opciones de crear un proceso y listar las postulaciones realizadas a la mascota.



**Interfaz gráfica 22. Visualizar proceso**

Interfaz gráfica 22: En esta interfaz gráfica el usuario administrador podrá visualizar un proceso con información registrada del adoptante, mascota y voluntario, además de la opción “respuestas cuestionario” que permite visualizar las respuestas ingresadas por el usuario al momento de realizar un proceso, “PDF Proc. Seguimiento” o “PDF Proc. Adopción”, con la opción de generar un formulario en pdf usado por los integrantes de la fundación para realizar procesos que acepta el adoptante como un acuerdo.

www.solucionemmalita.com/sistema/admin

Usuario: Perfil Mascota Procesos Inventario Eventos Donaciones Casos Clínicos Informes Logout(angelamunoz10@gmail.com)

Home Inventario

¿Qué puede hacer?

Crear Articulo

Informe de Ventas

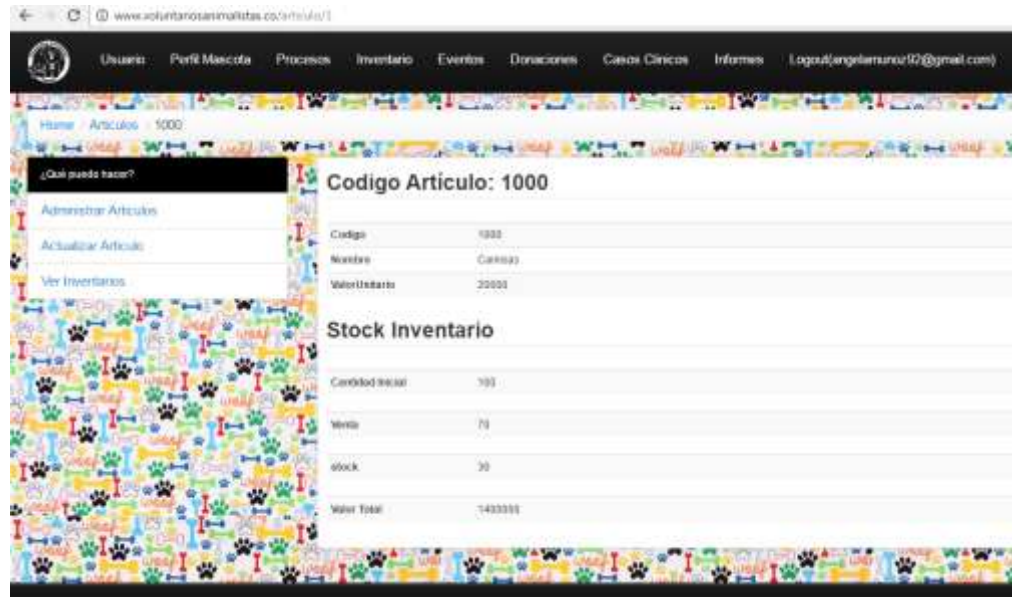
### Gestionar Articulos

Viendo 1-10 de 11 resultados

No.	Nombre	Código	Valor unitario	
1	Cereales	1000	20000	
2	Libros	1001	5000	
3	Calendarios	1002	10000	
4	Vinos	1003	5000	
5	Tarjetas	1004	3000	
6	Lapiceros	1005	1000	
7	Cebollas	1006	7000	
8	Pizzas	1007	3000	
9	Cuadernos	1008	5000	
10	Ropa Dog	1009	15000	

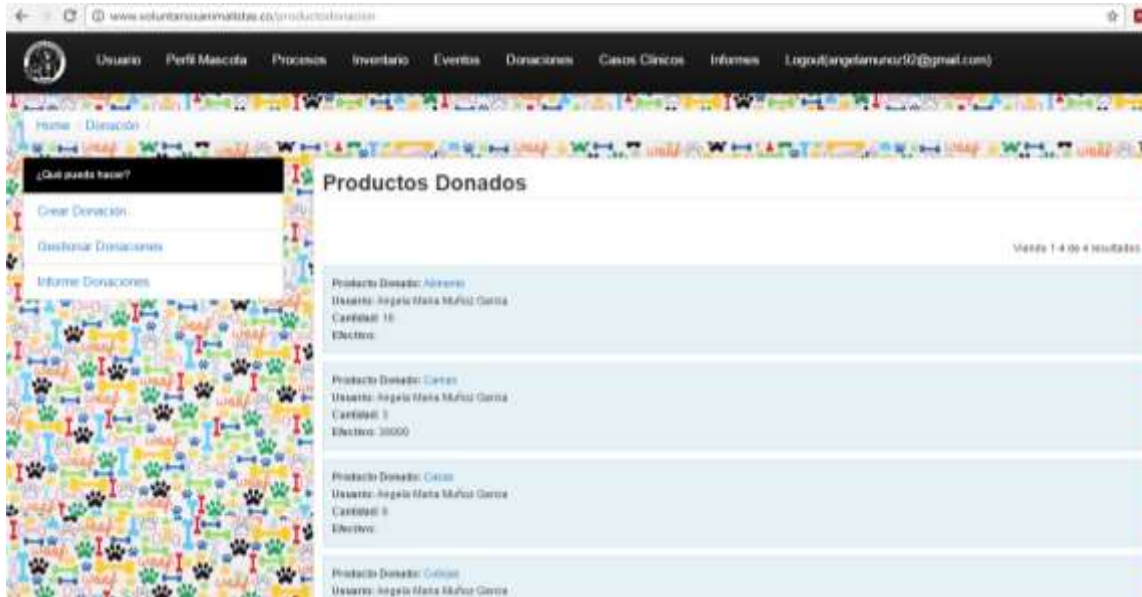
**Interfaz gráfica 23. Listar Inventario de productos**

Interfaz gráfica 23: En esta interfaz gráfica el usuario administrador podrá visualizar la lista de inventario de productos, con opciones de crear o actualizar inventario.



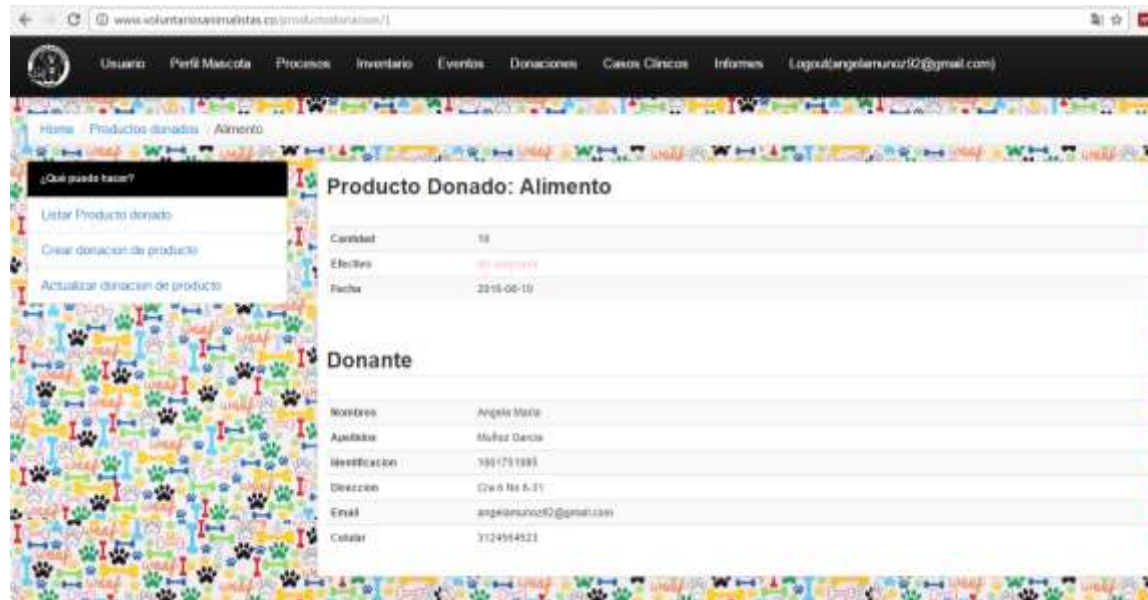
**Interfaz gráfica 24. Visualizar Inventario producto**

Interfaz gráfica 24: Interfaz gráfica en la que el usuario podrá obtener información del producto en venta como el nombre, cantidad total del producto, la cantidad actual del producto, y fecha de creación del inventario. Además de la opción de crear y actualizar inventario



**Interfaz gráfica 25. Lista de productos donados**

Interfaz gráfica 25: En esta interfaz gráfica el usuario administrador podrá visualizar la lista de productos donados. Con opciones de crear o actualizar producto donado.



**Interfaz gráfica 26. Visualizar producto donado**

Interfaz gráfica 26: a través de esta interfaz el usuario podrá obtener información del producto donado como el nombre y cantidad así mismo la información del usuario que realizo la donación.



## REUNIONES CON LA FUNDACIÓN “VAP”



Ilustración 1. Reunión fundación Voluntarios Animalistas Popayán



**Ilustración 2. Reunión entrega producto software**

## CARTA DE ENTREGA PLATAFORMA WEB



**Ilustración 3. Carta entrega formal producto software**