

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA DE SISTEMAS**

### **DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE REPORTES DE ANIMALES CALLEJEROS DOMÉSTICOS EN TIEMPO REAL DENTRO DEL DISTRITO METROPOLITANO DE QUITO**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**CHRISTOPHER DAVID PEÑAHERRERA ESPINOSA**

**[christopher.penaherrera@epn.edu.ec](mailto:christopher.penaherrera@epn.edu.ec)**

**DIRECTOR: PhD. Tania Elizabeth Calle Jiménez**

**[tania.calle@epn.edu.ec](mailto:tania.calle@epn.edu.ec)**

**Quito, marzo 2019**

## **DECLARACIÓN**

Yo, Christopher David Peñaherrera Espinosa, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Christopher David Peñaherrera Espinosa**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Christopher David Peñaherrera Espinosa, bajo mi supervisión.

---

**PhD. Tania Elizabeth Calle Jiménez**  
**DIRECTOR DE PROYECTO**

## **AGRADECIMIENTOS**

A Dios, por darme la salud, oportunidades y capacidades para poder mantenerme en el camino correcto y así poder cumplir con mis objetivos y retos.

A mis padres, por apoyarme en todo aspecto a lo largo de mi vida, siempre a pesar de momentos buenos y malos han sabido guiarme y ayudarme a tomar las mejores decisiones para lograr ser una mejor persona.

A mis amigos, quienes de alguna u otra forma han sabido apoyarme y brindarme su ayuda de manera incondicional.

A los profesores, que, con su perseverancia han logrado transmitirme su conocimiento y colaborar en mi formación profesional.

## **DEDICATORIA**

A mis padres gracias a quienes hoy soy una mejor persona. Gracias por estar en todo momento de mi vida.

# CONTENIDO

1. INTRODUCCIÓN .....	1
1.1. Descripción del problema .....	1
1.2. Planteamiento de la solución.....	1
1.3. Objetivos del proyecto .....	2
1.3.1. Objetivo general .....	2
1.3.2. Objetivos específicos.....	2
2. CONCEPTOS RELACIONADOS .....	2
2.1. Animales callejeros en Quito .....	2
2.2. Abandono animal .....	3
2.3. Animal doméstico.....	4
3. CONCEPTOS TÉCNICOS.....	4
3.1. Geolocalización en la web .....	4
3.2. Geotargeting .....	5
3.3. Google Maps.....	7
3.4. REST.....	8
3.5. SOAP .....	8
3.6. AJAX .....	9
4. METODOLOGÍA.....	11
4.1. Selección de metodología .....	12
4.2. Selección de metodologías ágiles por criterios de presencia.....	12
4.3. Aplicación de la metodología.....	13
4.3.1. SCRUM.....	13
4.4. Arquitectura del proyecto.....	16
4.5. Herramientas de desarrollo .....	18
5. DESARROLLO DE LA APLICACIÓN .....	18
5.1. Definición del proyecto .....	19
5.1.1. Alcance .....	19
5.1.2. Requerimientos.....	19
5.2. Historias de usuario épicas.....	20
5.3. Product Backlog.....	21
5.4. Release Planning.....	23
5.5. Desarrollo de Sprints .....	23
5.5.1. SPRINT 0.....	23
5.5.2. SPRINT 1.....	25

5.5.3.	SPRINT 2.....	37
5.5.4.	SPRINT 3.....	49
5.5.5.	SPRINT 4.....	55
6.	APLICACIÓN CASOS DE PRUEBA.....	74
6.1.	Prueba en módulo 1 – Gestión de Usuarios .....	74
6.2.	Prueba en módulo 2 – Reportes de Casos de Animales .....	78
6.3.	Prueba en módulo 3 – Perfil de Usuario / Galerías .....	78
6.4.	Prueba en módulo 4 – Estadísticas Casos Reportados .....	78
6.5.	Prueba en módulo 5 – Perfil Administrativo .....	79
7.	EVALUACIÓN DE USO DE LA APLICACIÓN .....	79
7.1.	Estructura de la encuesta .....	79
7.2.	Análisis de los resultados de la encuesta .....	81
7.2.1.	Resultados Parciales.....	81
7.2.2.	Resultado General.....	82
8.	EVALUACIÓN DE RENDIMIENTO DE LA APLICACIÓN.....	83
9.	CONCLUSIONES.....	85
10.	REFERENCIAS BIBLIOGRÁFICAS.....	86
11.	ANEXOS .....	89

## FIGURAS

Figura 1. Esquema aplicación de SOAP [20] .....	9
Figura 2. Funcionamiento de AJAX [23] .....	10
Figura 3. Tecnologías agrupadas bajo el concepto de AJAX [23] .....	11
Figura 4. Resumen marco de trabajo SCRUM [30] .....	16
Figura 5. Arquitectura MVC para el actual proyecto.....	17
Figura 6. Prototipo de interface - Pantalla Principal .....	31
Figura 7. Prototipo de interface - Registro de Usuario .....	32
Figura 8. Prototipo de interface - Inicio de Sesión.....	32
Figura 9. Prototipo de interface - Recuperación de Cuenta .....	33
Figura 10. Prototipo de interface - Formulario Reporte Animal en Adopción, Rescate, Perdido y Encontrado .....	33
Figura 11. Burndown Chart del Sprint 1.....	36
Figura 12. Prototipo de interface - Galería de imágenes para casos registrados de animales en Adopción, Perdidos y Encontrados .....	44
Figura 13. Prototipo de interface - Perfil de Usuario / Información de Usuario .....	44
Figura 14. Prototipo de interface - Perfil de Usuario / Historial de registros de casos reportados.....	45
Figura 15. Prototipo de interface - Perfil de Usuario / Casos cercanos de registros de reportes de rescate animal.....	45
Figura 16. Burndown Chart del Sprint 2.....	48
Figura 17. Prototipo de interface - Estadísticas perfil usuario.....	52
Figura 18. Burndown Chart del Sprint 3.....	55
Figura 19. Prototipo de interface - Inicio perfil administrador.....	66
Figura 20. Prototipo de interface - Inicio configuraciones generales .....	66
Figura 21. Prototipo de interface - Gestión usuarios / administradores / suscriptores .....	67
Figura 22. Prototipo de interface - Gestión reportes adopciones / rescate / encontrados / perdidos .....	67
Figura 23. Prototipo de interface - Estadísticas adopciones / rescate / encontrados / perdidos .....	68
Figura 24. Burndown Chart del Sprint 4.....	74
Figura 25. Resultados preguntas eficiencia y desempeño de la aplicación .....	81
Figura 26. Resultados preguntas experiencia y satisfacción del usuario .....	82
Figura 27. Resultados generales de la encuesta .....	83



## TABLAS

Tabla 1. Comparación de metodologías [25].....	11
Tabla 2. Comparación de metodologías ágiles por criterios de presencia [26] .....	13
Tabla 3. Roles Equipo Scrum.....	14
Tabla 4. Historia de Usuario Épica HUE01.....	20
Tabla 5. Historia de Usuario Épica HUE02.....	20
Tabla 6. Historia de Usuario Épica HUE03.....	20
Tabla 7. Historia de Usuario Épica HUE04.....	20
Tabla 8. Historia de Usuario Épica HUE05.....	21
Tabla 9. Historia de Usuario Épica HUE06.....	21
Tabla 10. Product Backlog .....	21
Tabla 11. Release Planning .....	23
Tabla 12. Tablero Kanban inicial Sprint 0 .....	24
Tabla 13. Tablero Kanban inicial Sprint 1 .....	25
Tabla 14. Historia de Usuario 01.....	25
Tabla 15. Historia de Usuario 02.....	26
Tabla 16. Historia de Usuario 03.....	26
Tabla 17. Historia de Usuario 04.....	27
Tabla 18. Historia de Usuario 05.....	27
Tabla 19. Puntos de esfuerzo Sprint 1.....	28
Tabla 20. Product Backlog del Sprint 1.....	28
Tabla 21. Tablero Kanban final Sprint 1 .....	31
Tabla 22. Revisión de Criterios de Aceptación del Sprint 1 .....	34
Tabla 23. Tablero Kanban inicial Sprint 2 .....	37
Tabla 24. Historia de Usuario 09.....	37
Tabla 25. Historia de Usuario 12.....	38
Tabla 26. Historia de Usuario 13.....	38
Tabla 27. Historia de Usuario 14.....	39
Tabla 28. Puntos de esfuerzo Sprint 2.....	40
Tabla 29. Product Backlog del Sprint 2.....	40
Tabla 30. Tablero Kanban final Sprint 2 .....	43
Tabla 31. Revisión de Criterios de Aceptación del Sprint 2 .....	46
Tabla 32. Tablero Kanban inicial Sprint 3 .....	49
Tabla 33. Historia de Usuario 15.....	49
Tabla 34. Puntos de esfuerzo Sprint 3.....	50
Tabla 35. Product Backlog del Sprint 3.....	50
Tabla 36. Tablero Kanban final Sprint 3 .....	52
Tabla 37. Revisión de Criterios de Aceptación del Sprint 3 .....	52
Tabla 38. Tablero Kanban inicial Sprint 4 .....	56
Tabla 39. Historia de Usuario 19.....	56
Tabla 40. Historia de Usuario 20.....	57
Tabla 41. Historia de Usuario 21.....	57
Tabla 42. Historia de Usuario 22.....	58
Tabla 43. Historia de Usuario 26.....	59
Tabla 44. Historia de Usuario 27.....	59
Tabla 45. Puntos de esfuerzo Sprint 4.....	60
Tabla 46. Product Backlog del Sprint 4.....	60
Tabla 47. Tablero Kanban final Sprint 4 .....	65
Tabla 48. Revisión de Criterios de Aceptación del Sprint 4 .....	68

Tabla 49. Casos de prueba Módulo 1 - Pantalla de inicio .....	75
Tabla 50. Casos de prueba Módulo 1 - Inicio de sesión .....	75
Tabla 51. Casos de prueba Módulo 1 - Registro de usuario .....	76
Tabla 52. Casos de prueba Módulo 1 - Recuperación de cuentas .....	77
Tabla 53 Tablero Kanban final .....	79
Tabla 54. Encuesta eficiencia y desempeño de la aplicación .....	80
Tabla 55. Encuesta experiencia y satisfacción del usuario .....	80
Tabla 56. Resultados pruebas de rendimiento .....	84
Tabla 57. Códigos de respuesta y latencias .....	84

## RESUMEN

En la actualidad las aplicaciones web son herramientas que optimizan procesos y tiempo llegando a ser de gran ayuda para las personas, razón por la cual, el presente trabajo de titulación se enfoca en el desarrollo de una aplicación web que gestiona reportes en tiempo real de animales que se encuentran en las calles del Distrito Metropolitano de Quito.

La aplicación web brinda al usuario la opción de enviar reportes de rescate, adopción y búsqueda de animales incluyendo el envío de localizaciones a través de la API de Google Maps, dichos reportes son utilizados para crear un medio de información sobre los problemas que atraviesan los animales callejeros y a través de la API de Highcharts se representa gráficamente la información recolectada.

La aplicación web fue desarrollada bajo la metodología de desarrollo ágil Scrum, se incluyen documentos tales como Lista del producto, Historias de usuarios, tableros Kanban, entre otros, los cuales nos ayudan a visualizar el desarrollo incremental - funcional que plantea la metodología y como se ha llevado a cabo el proceso de desarrollo de este proyecto.

En base a pruebas funcionales aplicadas al sistema, se obtuvo resultados satisfactorios que cumplen con los criterios de aceptación planteados. Finalmente, a través de encuestas realizadas a usuarios finales se evidencia un correcto desempeño y eficiencia del software desarrollado.

**Palabras clave:** Animales callejeros, Geolocalización, Scrum, Highcharts.

## **ABSTRACT**

Nowadays, web apps are tools that optimize processes and time, becoming a great help for people. That's the main reason of this title work, focusing on the development of a web application that manages animals' reports in real time, animals that are found in the streets of the Metropolitan District of Quito.

The web application allows the user to send reports of rescue, adoption and search of animals, including the locations through the Google Maps API. These reports will be used to create means of information of the problems that animals are going through in the streets, and through the Highcharts API, the information collected is graphically represented.

The web application was developed under the Scrum agile development methodology. It includes documents such as Product List, User Stories, Kanban Boards, among others, which help us to visualize the incremental - functional development that the methodology raises and how has carried out the development process of this project.

Based on functional tests applied to the system, was successfully achieved obtained the proposed criteria. Finally, through surveys carried out to beta users, the correct performance and efficiency of the developed software is evident.

**Keywords:** Stray animals, Geolocation, Scrum, Highcharts.

# **1. INTRODUCCIÓN**

## **1.1. Descripción del problema**

Actualmente, en la ciudad de Quito, existe una problemática social enfocada en los animales callejeros, la cual se ha ido propagando desde hace años atrás. Debido a factores económicos, familiares, sociales, entre otros. En la ciudad de Quito se estiman 122.280 animales en las calles [1].

Los animales que viven en las calles se encuentran en malas condiciones, lo que afecta a su bienestar y salud. La falta de concientización y responsabilidad en las personas, han sido factores que han impulsado el crecimiento de este problema. Por otra parte, animales que tienen un lugar de hospedaje, no siempre se encuentran en las mejores condiciones debido a que existe maltrato en ciertos casos. A consecuencia de lo mencionado, se realizan denuncias en el Municipio, pero por falta de recursos no es posible dar respuesta a todas [2] [3].

Diversas instituciones, fundaciones, entre otras organizaciones que prestan ayuda a animales callejeros en Quito, en muchas ocasiones son incapaces de atender los problemas que atraviesan los animales por algunos factores, uno de ellos es la falta de comunicación o conocimiento sobre lo que ocurre en las calles.

## **1.2. Planteamiento de la solución**

Por medio de la descripción del problema planteado, se propone desarrollar una solución en el ámbito tecnológico. La cual, gestionará reportes realizados por los usuarios acerca de animales callejeros domésticos que se encuentren en estado de adopción, extraviados, encontrados o que atraviesen algún caso de peligro, estos reportes serán en tiempo real vía web y además aplicando el concepto de geolocalización se podrá determinar ubicaciones.

El sistema web abarca el Distrito Metropolitano de Quito, con el objetivo de saber los lugares con más recurrencia de animales callejeros, lo cual servirá de apoyo a los animales en casos específicos y mediante el análisis de los reportes e información recolectados se puede determinar los lugares con más recurrencia.

## **1.3. Objetivos del proyecto**

### **1.3.1. Objetivo general**

- Desarrollar una aplicación web para la gestión de reportes de animales callejeros domésticos en tiempo real dentro del Distrito Metropolitano de Quito.

### **1.3.2. Objetivos específicos**

- Investigar y analizar procesos de rescate enfocados a animales domésticos con entidades relacionadas a la protección animal en el Distrito Metropolitano de Quito.
- Implementar una aplicación web para la gestión de reportes de animales callejeros domésticos en tiempo real dentro del Distrito Metropolitano de Quito en base a la metodología de desarrollo de software SCRUM.
- Diseñar formularios para la recepción de información de animales callejeros domésticos por medio de la web y la geolocalización.
- Generar reportes estadísticos públicos en base a la información almacenada acerca de animales callejeros domésticos.
- Realizar pruebas de rendimiento y usabilidad para determinar errores y posibles mejoras.

## **2. CONCEPTOS RELACIONADOS**

### **2.1. Animales callejeros en Quito**

En el Distrito Metropolitano de Quito, es complejo determinar la población de animales callejeros debido a distintos factores tales como falta de recursos económicos y humanos, falta de control de reproducción animal, entre otros. Por medio de proyecciones y estudios demográficos, se tiene como aproximación que existe alrededor de un perro por cada seis habitantes de la ciudad de Quito [1].

En el año 2013, se lleva a cabo el primer estudio poblacional de animales callejeros con muestras reales y aplicando un modelo estadístico, dicho estudio tenía el objetivo de analizar tres parámetros y relacionarlos, la economía, bienestar animal y prevalencia de parásitos en animales callejeros. Como resultado, se obtuvo que, en dieciséis parroquias urbanas y rurales de Quito existe un perro por cada cuarenta y nueve habitantes [1].

El Instituto Nacional de Estadística y Censos (INEC) para el año 2013, determinó un total de 41.676 animales callejeros en Quito, en 2018 estiman 122.280 animales callejeros, es decir, un crecimiento de casi el triple en un periodo de 5 años. El crecimiento es alarmante y teniendo en cuenta las consecuencias que conlleva gran cantidad de animales en las calles. Instituciones afines al rescate y bienestar animal como Urbandog, trabajan por medio de campañas para mantener el control de la población animal, el método más adecuado es la esterilización [4].

Desde 2011 se han hecho efectivas gran cantidad de campañas de esterilización logrando esterilizar alrededor de 42.844 mascotas. Se realizará un nuevo estudio en 5 años, para determinar cómo ha evolucionado la población de animales callejeros. Los animales en las calles provocan diversos problemas como un impacto ambiental con la fauna nativa, enfermedades, basura en espacios públicos, entre otros [5].

En el presente año, se ha optimizado el estudio realizado en 2013 utilizando un nuevo modelo estadístico con más precisión, tomando la misma muestra se tiene que en promedio entre zonas rurales y urbanas se tiene un perro por cada cincuenta y nueve habitantes [1].

## **2.2. Abandono animal**

La falta de educación, cultura y la irresponsabilidad son las causas principales para que el abandono animal crezca en el país. A pesar de las campañas de sensibilización y concientización efectuadas por instituciones / fundaciones afines al rescate animal, el problema de abandono sigue presente en las calles y en aumento.

En el año 2015, por medio de un análisis realizado se logró determinar que existen alrededor de 100.000 animales abandonados en todo el Ecuador. Según Carolina Redín, fundadora de ENDA [6], los lugares con mayor número de denuncias por abandono son el Sur de Quito, Pomasqui, Calderón y Carcelén. Entre las causas más comunes de abandono animal se presentan las siguientes [7] [8]:

- El propietario no tiene tiempo para atender a la mascota.
- Camadas no deseadas.
- El propietario ya no dispone de recursos para mantener a la mascota.
- Criadores irresponsables en búsqueda de dinero a cambio de mascotas.
- Llegada de un bebé a la familia.

- Cambio de domicilio.
- Enfermedad de la mascota.
- El propietario fallece.
- Vejez de la mascota.
- Comportamiento inapropiado de la mascota.

Para evitar abandonar a un animal se recomienda educar y esterilizar, con esto se va a contribuir al decrecimiento de animales en las calles.

### **2.3. Animal doméstico**

Un animal doméstico es aquel que puede llegar a ser amaestrado y que se ha acostumbrado a vivir junto a un ser humano. Regularmente, un animal doméstico es adquirido o adoptado por personas con las cuales comparten un mismo espacio [9] [10].

Las mascotas, como también se los conoce, son animales domésticos enfocados a brindar compañía a las personas. Los perros y gatos son los animales domésticos más comunes, pero existe una clasificación para estos animales como se muestra a continuación:

- De granja
- De compañía

#### **Animal doméstico de compañía**

Estos animales proporcionan compañía a los seres humanos, por lo general no requieren un hábitat específico para sobrevivir, sino que se adaptan al mismo medio que los humanos. Las aves y los mamíferos son dos grupos de animales que destacan en animales de compañía.

Estos animales son seleccionados por su comportamiento, adaptabilidad e interacción con los humanos, se caracterizan por ser amigables, además, poseer un animal de compañía puede traer consecuencias positivas para su propietario debido a que ayudan a reducir la depresión, preocupaciones, entre otros [11].

## **3. CONCEPTOS TÉCNICOS**

### **3.1. Geolocalización en la web**

Geolocalización, también conocido como georreferenciación, permite determinar la posición real de un objeto espacial en un sistema de coordenadas latitud (x), longitud (y) y



altitud (z). Se relaciona al término Geolocalización frecuentemente con los sistemas de posicionamiento [12].

La geolocalización en la web, es una tecnología que permite utilizar los datos obtenidos por medio de dispositivos conectados a la web para determinar la ubicación física real de los usuarios. A menudo se utiliza, para identificar la localización o ubicación de una persona que requiere de un servicio vía web manteniendo su privacidad [12].

### **Usos de Geolocalización Web**

Por medio de dispositivos conectados a la web se obtiene información en tiempo real, con dicha información se consigue localizar objetos en un mapa con alta precisión y en determinados periodos de tiempo. La obtención de ubicaciones se puede lograr de diferentes formas, tales como [13]:

- Navegación web por medio de direcciones IP.
- Teléfonos móviles.
- Dispositivos GPS.
- Identificación de Radio Frecuencias.

### **Seguridad y Privacidad en Geolocalización Web**

La gran demanda de aplicaciones web implementadas con la tecnología de geolocalización conlleva problemas relacionados con la seguridad de la información y la privacidad de los usuarios [13].

La obtención y generación de los datos de geolocalización se puede dar de dos formas:

- **Modo activo**, se basa en el dispositivo del usuario.
- **Modo pasivo**, se realiza una búsqueda de datos basada en servidores correlacionados [12].

## **3.2. Geotargeting**

Permite personalizar el contenido digital que se presenta a un usuario vía web dependiendo de la localización geográfica del mismo. Actualmente, el geotargeting es una técnica de marketing utilizada a nivel mundial, utiliza para poder potenciar el contenido que se brinda al usuario. Las empresas utilizan esta técnica para poder además de tener contenido dinámico, segmentar a su mercado y target [14].

Buenas prácticas de esta técnica, son las redes sociales o tiendas de venta online, que ofrecen a sus clientes información de sitios cercanos a su ubicación o promociones que estén en locaciones cercanas del cliente. Algunos ámbitos relevantes de la aplicación de esta técnica se presentan a continuación [14]:

- Páginas web en distintos idiomas.
- Comercio electrónico – Ecommerce.
- Publicidad regional.
- Servicios basados localización.
- Estudio de mercados.
- Protección de la propiedad intelectual.
- Seguridad de pagos.

La técnica del geotargeting puede ser utilizada a partir de diferentes recursos, comúnmente se utiliza la ubicación proporcionada por el navegador, pero se puede utilizar otros recursos para personalizar el contenido [15].

**A partir de datos proporcionados por el usuario**, se basa en utilizar la información que el usuario manualmente proporciona a la aplicación por medio de formularios, es decir, al momento de realizar un registro o creación de cuenta, seleccionar país, idioma, preferencias, etc. Esta información permite aplicar geotargeting, pero no en forma segura debido a que la información proporcionada por el usuario puede ser errónea o falsa [15].

**Ubicación del navegador web**, la API de geolocalización de HTML5 permite por medio de programación en la aplicación, recopilar información sobre la ubicación que tiene un usuario específico. Las coordenadas que se pueden obtener por medio del navegador son relativamente precisas y brindan fiabilidad para generar contenido adecuado [15].

La información de las coordenadas puede variar dependiendo la manera en la que se obtienen, por medio de GPS, direcciones IP, WLAN, direcciones MAC, etc. Dicha variación puede ser de pocos metros y en el peor de los casos kilómetros. En este caso, para que el navegador tenga acceso a la ubicación, se necesita el permiso o consentimiento del usuario, además, la aplicación web debe contar con una conexión segura HTTPS [15].

**GPS targeting**, obtiene la información de una terminal por medio del satélite, para poder determinar una localización se utiliza de 6 a 12 señales del satélite, en la actualidad, en su

mayoría los dispositivos móviles tienen integrado un receptor GPS a diferencia de los ordenadores de escritorio que carecen de dicho receptor [15].

### 3.3. Google Maps

Google Maps es un servidor de aplicaciones de mapas en la web, ofrece gran variedad de servicios a sus usuarios los principales se muestran a continuación [16]:

- Mapas desplazables.
- Fotografías satelitales.
- Rutas de ubicaciones (Google Street View).
- Tráfico en tiempo real.
- Cálculos de rutas.
- Navegador GPS.

Google Maps tuvo su lanzamiento el 8 de febrero de 2005, y solo era soportado en ciertos navegadores web, paso un periodo de 6 meses para terminar con su fase beta al público. Desde su desarrollo inicial se apoyaron de la técnica AJAX para lograr llamadas asíncronas a sus servidores y conseguir mejores resultados para sus usuarios.

La localización que ofrece google maps es por medio de coordenadas, las mismas están en el sistema WGS84, un sistema de coordenadas mundial que permite localizar cualquier punto sobre la tierra por medio de tres unidades dadas x, y, z [16].

**API Google Maps**, desarrollada con lenguaje JavaScript permite personalizar mapas con contenido propio, este contenido puede ser visualizado en dispositivos móviles o sitios web. Los mapas que ofrece la API de Google Maps son de cuatro tipos, estos mapas pueden ser personalizados dependiendo de la necesidad del usuario. Tipos de mapas [16]:

- Hojas de ruta.
- Satélite.
- Híbrido.
- Terreno.

Para la implementación de la API de Google Maps, se debe utilizar una etiqueta script con diferentes parámetros [17]:

- **Atributo src**, contiene la URL que indica la ubicación del archivo JS para la carga de los componentes de la API, este atributo contiene un clave la cual puede ser gratuita con ciertas restricciones o de pago dependiendo del cliente.

- **Async**, permite cargar a la web de manera normal sin retrasos, cuando la API esté listo con su contenido se carga de forma asíncrona.

Actualmente toda la API de Google Maps, funciona por medio del protocolo Https, esto con el objetivo de proporcionar sitios seguros y evitar la manipulación de terceros [17].

### 3.4. REST

REST (Representational State Transfer), es una arquitectura que comprende un conjunto de principios y técnicas para el diseño de software y el desarrollo de servicios. Se rige bajo el esquema de Cliente-Servidor con las siguientes particularidades [18] [19]:

- **Separa responsabilidades**, establecen una interface de comunicación para la separación completa de responsabilidades.
- **No maneja estados**, no mantienen estados asociados al cliente, es decir, cada petición que se realiza es independiente de las demás. Además, todas las peticiones que se realicen a un servicio son idénticas.
- **Permite caché** (respuestas cacheables en el cliente), el contenido que se genera por el servicio web, puede ser almacenado temporalmente, con el objetivo que cuando se genere por primera vez una petición la misma pueda ser consultada de cache en el caso de ser necesario.
- **Interfaz uniforme** (API estandarizada), todos los servicios serán invocados de igual forma, pero por medio de distintos métodos, se utilizará las palabras reservadas GET, POST, UPDATE y DELETE.
- **Trabajo en capas**, REST plantea que se utilicen capas intermedias para su buen funcionamiento, aportando así escalabilidad, rendimiento y seguridad. Antes de llegar al servidor las peticiones, pueden pasar por otras capas como por ejemplo un balanceador de cargas.

### 3.5. SOAP

Soap (Simple Object Access Protocol), se define como un protocolo estándar que por medio del lenguaje XML proporciona de manera simple y ligera intercambio de información entre dos puntos. Debido a sus características, este protocolo puede ser utilizado en servidores de aplicaciones que utilicen el modelo de comunicación RPC (Remote Procedure Call) [20] [21]

SOAP se compone de tres partes principales [20]:

- **SOAP envelope**, define la gestión de los mensajes, es decir, el contenido del mensaje y si es opcional u obligatorio.
- **Reglas de codificación**, estas definen el mecanismo de serialización a utilizar para encapsular mensajes de distintos tipos.
- **SOAP RPC**, se enfoca en el modo en que se realizan llamadas a procedimientos remotos y los resultados de los mismos.

Al momento de implementar SOAP, se debe tener en cuenta estándares para poder potenciar las funcionalidades que brinda este protocolo, se deben cumplir los objetivos principales que se mencionan a continuación [22]:

- Establecer protocolos de invocación a servicios remotos basado en los protocolos HTTP (Híper Text Transport Protocol) para la transmisión y XML (eXtensible Markup Language) para la codificación de datos.
- Debe ser independiente de la plataforma de implementación hardware y del lenguaje de programación.

A continuación, se presenta una implementación del protocolo SOAP:

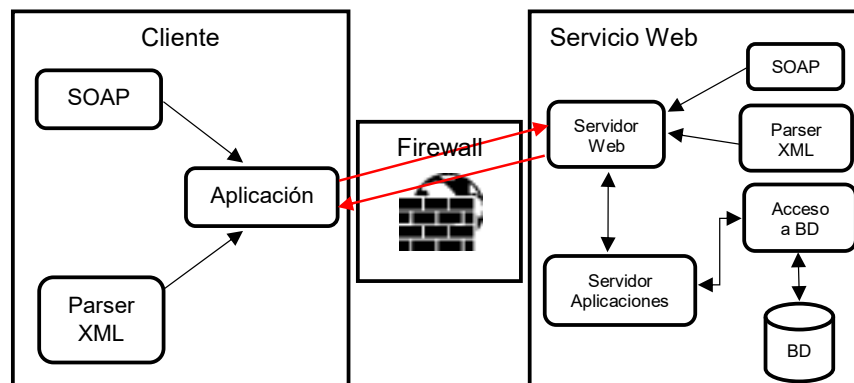


Figura 1. Esquema aplicación de SOAP [20]

### 3.6. AJAX

AJAX (JavaScript y XML Asíncrono), es una técnica de desarrollo web que se enfoca en aplicaciones, sitios o páginas que son interactivas o RIA (Rich Internet Applications) [23].

Las aplicaciones que utilizan esta técnica se ejecutan en el cliente, es decir, en el navegador del usuario, el usuario realiza una petición de datos adicionales hacia el

servidor, estos se cargan en segundo plano sin intervenir en el comportamiento de la página o sitio web por medio de una comunicación asíncrona con el servidor [23].

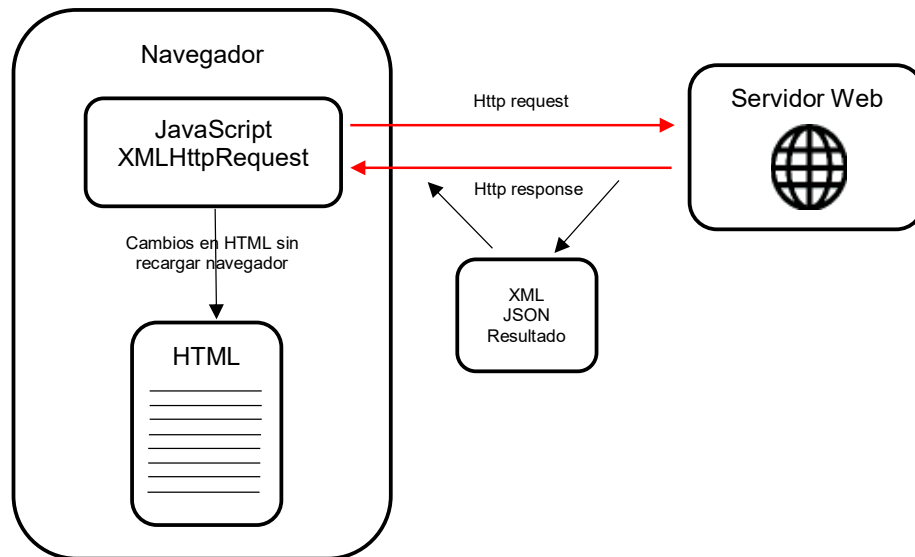


Figura 2. Funcionamiento de AJAX [23]

#### Tecnologías incluidas en AJAX [23]

- **XHTML o HTML y CSS**, utilizadas para la estructura y el diseño de la aplicación o página web que van en conjunto con la información.
- **Document Object Model (DOM)**, es posible manipular el DOM por medio de lenguaje JavaScript y así cambiar de forma dinámica la estructura de las aplicaciones o páginas web.
- **XMLHttpRequest**, utilizado para el intercambio de información de manera asíncrona entre el cliente y el servidor, PHP es un lenguaje de programación web orientado al desarrollo web con contenido dinámico utilizado con AJAX.
- **XML (eXtensible Markup Language)**, formato utilizado para la transferencia de objetos requeridos del servidor.
- **JSON (JavaScript Object Notation)**, formato ligero para intercambio de datos, fácil de analizar y generar. Se basa en lenguaje JavaScript y el Estándar ECMA-262 y está conformado por dos estructuras:
  - Colección de pares de nombres.
  - Lista ordenada de valores.

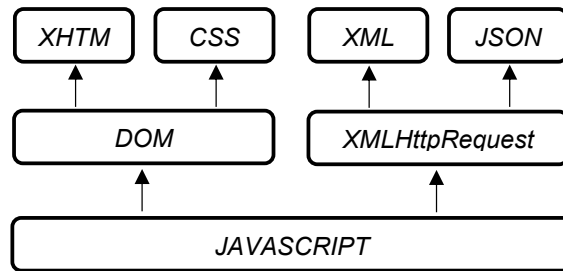


Figura 3. Tecnologías agrupadas bajo el concepto de AJAX [23]

## 4. METODOLOGÍA

Actualmente, existe diversidad de opciones con respecto al uso de metodologías para el desarrollo de software, metodologías tradicionales y metodologías ágiles ponen a disposición gran cantidad de recursos para conseguir productos de software satisfactorios.

Las metodologías tradicionales, han permaneciendo en uso y han sido exitosas desde hace tiempo, pero de igual manera teniendo constantemente un problema de fondo, es decir, estas metodologías no se adaptan adecuadamente a entornos en donde los requerimientos no son predecibles o pueden cambiar en el proceso de desarrollo.

Las metodologías ágiles a diferencia de las tradicionales, se enfocan en conseguir un equilibrio entre el proceso y el esfuerzo en el desarrollo del producto software, estas metodologías logran procesos de adaptación cuando surgen cambios [24].

A continuación, se detallan relevantes diferencias entre las metodologías tradicionales y ágiles.

Tabla 1. Comparación de metodologías [25]

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso muchas más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.

Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura de software.	La arquitectura del software es esencial y se expresa mediante modelos.

Debido a lo antes mencionado, para el desarrollo de este proyecto es necesario implementar una metodología ágil que permita un cronograma flexible y que se adapte a las necesidades del mismo, de igual manera, conseguir la entrega continua de funcionalidades que generen valor para el producto software deseado.

#### **4.1. Selección de metodología**

Varios autores como Dijkstra, plantean bases para el desarrollo de metodologías y establecen criterios de aceptación para lograr éxito en el desarrollo de software, algunas de estas se mencionan a continuación [26]:

- El coste del desarrollo inicial debe ser relativamente bajo.
- El software debe ser fácil de mantener.
- El software debe ser portable a nuevo hardware.
- El software debe hacer lo que el cliente quiere.

Con los criterios de éxito planteados, se desarrollan gran cantidad de metodologías ágiles entre las más importantes se tiene [26]:

- Agile Project Management (APM).
- Dynamic System Development Methods (DSDM).
- Scrum.
- Test Driven Development
- Extreme Programming (XP).

#### **4.2. Selección de metodologías ágiles por criterios de presencia**

Al momento de escoger una metodología ágil, se busca que esta tenga documentación clara y amplia para su aplicación, además, que la metodología tenga alguna certificación y training que garantice el éxito al momento de su uso.

Bajo este argumento, se han determinado seis indicadores para determinar la viabilidad y aplicación de una metodología, los indicadores son los siguientes [26]:



- Metodología con mayor presencia en internet.
- Metodología mejor documentada.
- Metodología certificada y con training.
- Metodología con comunidades.
- Metodología más utilizada por empresas.
- Presencia Empresarial.
- Metodología más utilizada en proyectos software.

Tabla 2. Comparación de metodologías ágiles por criterios de presencia [26]

Metodología	Mayor presencia en internet	Mejor documentación	Certificadas y con training	Comunidades	Presencia empresarial.	Proyectos de Software.	TOTAL
APM	2	1	3	5	1	1	11
DSDM	1	3	5	5	4	4	22
Scrum	5	2	5	5	5	5	27
Test Driven Development	3	4	3	2	2	2	16
XP	4	5	3	2	3	3	19
<b>TOTAL</b>	15	15	19	19	15	15	15

En la Tabla 2, se puede apreciar que, bajo los criterios de presencia planteados, SCRUM, es una metodología de desarrollo de software ágil viable para implementar, por lo tanto, el presente proyecto se va a desarrollar con la metodología SCRUM debido a que es la que más se adapta las necesidades del mismo.

### 4.3. Aplicación de la metodología

#### 4.3.1. SCRUM

Scrum es un marco de trabajo que permite desarrollar productos complejos, dentro de este marco existen roles, eventos y artefactos los cuales se relacionan por medio de reglas propias de Scrum. Este marco de trabajo, además, se enfoca en entregas que tengan el máximo valor posible, con el fin de lograr productividad y crecimiento en el desarrollo del producto de software. Scrum se caracteriza por ser [27]:

- Liviano.
- Fácil de entender.
- Difícil de dominar.

##### 4.3.1.1. Roles de Scrum

Los roles principales Scrum que se definen para este proyecto están conformados por las siguientes partes: Dueño del Producto (Product Owner), Equipo de Desarrollo

(Development Team) y Scrum Master. Se detalla a continuación el enfoque de cada una de las partes [27]:

- **Dueño del Producto:** miembro responsable de gestionar la Lista del Producto o Product Backlog, es decir, es una única persona no es un comité, que puede cambiar la prioridad de elementos de la Lista del Producto si se lo requiere.
- **Equipo de Desarrollo:** grupo de personas que se encargan de generar incrementos del producto terminados funcionales, los cuales pueden ser puestos en producción. Este grupo de trabajo debe lo necesariamente pequeño para ser ágil y lo necesariamente grande para cumplir con incrementos del producto.
- **Scrum Master:** miembro responsable de guiar al grupo o equipo bajo un marco de trabajo Scrum, haciendo cumplir sus reglas. Además, ayuda a entender a las partes externas al equipo que interacciones pueden o no tener con el mismo.

Tabla 3. Roles Equipo Scrum

<b>Rol</b>	<b>Cargo</b>	<b>Responsable</b>
Dueño del Producto	Director de Tesis	PhD. Tania Calle
Equipo de Desarrollo	Autor	David Peñaherrera
Scrum Master	Director de Tesis	PhD. Tania Calle

#### **4.3.1.2. Artefactos de Scrum**

Los artefactos que plantea Scrum representan el trabajo o valor que se va obteniendo durante el desarrollo de los proyectos y son útiles para mostrar de forma transparente el avance de los mismos. Se tienen los siguientes artefactos [27]:

- **Lista de producto (Product Backlog),** es una lista ordenada o priorizada de los requerimientos planteados para el desarrollo del producto, esta lista es responsabilidad del dueño del producto, y es el único que puede realizar cambios sobre la misma a medida que se desarrolle el proyecto, en inicio tiene requerimientos conocidos, pero no completos aún.
- **Lista de pendientes del Sprint (Sprint Backlog),** esta lista es realizada por el equipo de trabajo y abarca un conjunto de requerimientos tomados de la lista del producto, este conjunto de requerimientos se enfocará en cubrir los objetivos planteados para el Sprint y será desarrollado y entregado como incremento funcional.
- **Gráfico de trabajo pendiente (Burndown Chart),** es una herramienta que muestra a lo largo del tiempo el trabajo pendiente, el propósito de este gráfico es permitir visualmente controlar que el proyecto tenga un desarrollo correcto para realizar entregas dentro de un calendario planteado [28].

- Historia de Usuario, es una breve descripción de los requerimientos de un cliente, se enfocan en describir funcionalidades que se van a incorporar en un sistema de software y cuya implementación aporta valor al cliente, son utilizadas por lo general en entornos de trabajo ágiles como Scrum, se conforman por lo general de las siguientes partes [28]:
  - Nombre descriptivo.
  - Descripción de la funcionalidad.
  - Criterios de validación y verificación.
  - Puntos de esfuerzo o estimación.
  - Prioridad.
- Historias Épicas, se denomina Épica a una historia de usuario que es de gran tamaño, una historia épica se descompone en historias de usuario con tamaños más adecuados para que puedan ser gestionadas con los principios y técnicas ágiles, es decir, la estimación y seguimiento cercano [28].

#### **4.3.1.3. Eventos de Scrum**

En la metodología Scrum, se tiene eventos los cuales son definidos para poder minimizar la pérdida de tiempo, estos eventos tienen duraciones específicas, es decir, no pueden extenderse o acortarse. Un Sprint permite contener eventos, los mismos que finalizan siempre y cuando se haya cumplido con el o los objetivos planteados para un Sprint.

Los eventos de Scrum permiten llevar a cabo inspecciones de aspectos que se llevan a cabo en el desarrollo de un producto [27].

- Sprint, es el eje central de Scrum sobre el cual giran una serie de eventos previamente programados. Por lo general un Sprint dura alrededor de un mes o menos, al momento de la finalización del Sprint se tiene como resultado un incremento al producto final, el cual debe ser funcional, además inmediatamente al finalizar un Sprint comienza otro.

Un Sprint puede llegar a ser cancelado antes que llegue a su máximo de duración, por lo tanto, el objetivo del Sprint llega a quedar en estado obsoleto.

- Planificación del Sprint (Sprint Planning), se enfoca en la planificación del Sprint, este trabajo se realiza en conjunto de todo el equipo. Por lo general, dura 8 horas la planificación para un Sprint aproximado de un mes. Con este evento de Scrum se responde a las siguientes preguntas:
  - ¿Qué puede entregarse en el incremento resultante del Sprint que comienza?
  - ¿Cómo se conseguirá hacer el trabajo necesario para entregar el incremento?

- Objetivo del Sprint (Sprint Goal), sirve como una guía para que el equipo trabaje de manera conjunta, además proporciona un enfoque holístico al equipo pueda lograr un incremento al finalizar el Sprint.
- Scrum Diario (Daily Scrum), este evento tiene una duración de 15 minutos, se lo realiza diariamente por parte del equipo de desarrollo durante el Sprint y sirve para la planificación del trabajo a realizar durante las próximas 24 horas.
- Revisión del Sprint (Sprint Review), esta reunión es de aproximadamente 4 horas para un Sprint de un mes, y tiene como objetivo la revisión del incremento logrado y la adaptación de la lista del producto en el caso de ser necesario.
- Retrospectiva de Sprint (Sprint Retrospective), se la realiza después de la revisión del Sprint y antes de la planificación del siguiente sprint, su objetivo es que el equipo Scrum tenga una retroalimentación de su propio trabajo en búsqueda de mejoras para el futuro.

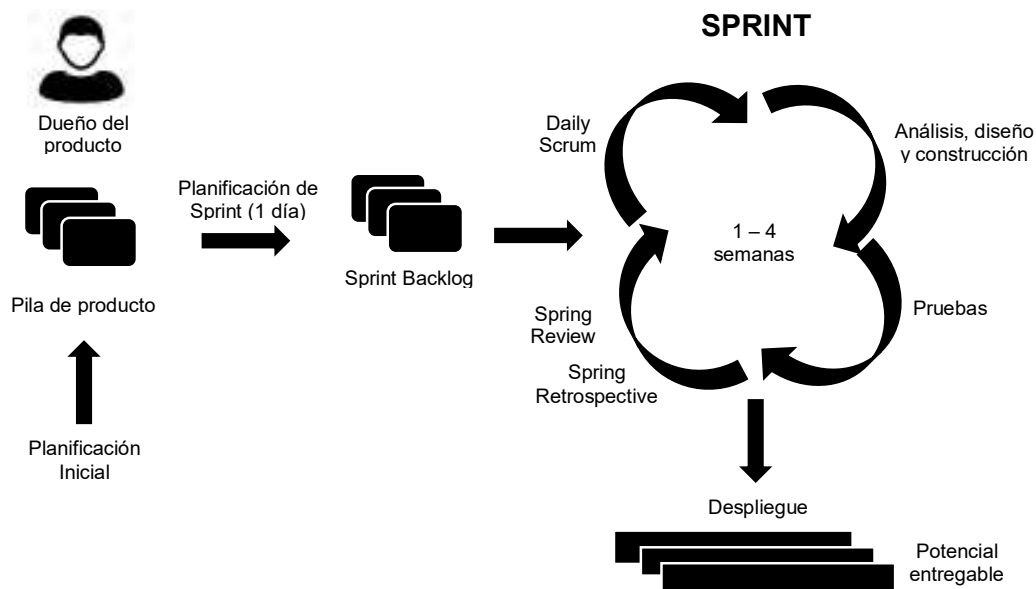


Figura 4. Resumen marco de trabajo SCRUM [30]

#### 4.4. Arquitectura del proyecto

Un patrón de diseño consiste en una solución probada para un problema específico que se presenta al momento de desarrollar software. Las ventajas de utilizar un patrón de diseño son varias, las más importantes se mencionan a continuación [29]:

- El código de la aplicación permanece organizado y mantenible.
- Permite la reutilización de código.
- Se acopla a proyectos que presentan escalabilidad.

Para la arquitectura de este proyecto, se utiliza el patrón de diseño denominado Modelo-Vista-Controlador (MVC).

MVC permite independizar la aplicación por medio de capas con el fin de organizar y hacer proyectos escalables. Además, permite diferenciar la presentación o interfaz de usuario de la lógica de la aplicación. Para el desarrollo de este proyecto se aplicará el patrón de diseño mencionado. Se detalla a continuación la función de cada capa [29]:

- **Modelo:** se encarga de la interacción y gestión con los datos de la aplicación, además contiene la lógica del negocio.
- **Vista:** presenta la información del negocio al usuario y presenta la interfaz con la que interactúa el usuario.
- **Controlador:** gestiona las peticiones del usuario, es el nexo entre el modelo y la vista.

A continuación, se presenta un esquema de la aplicación del MVC para este proyecto:

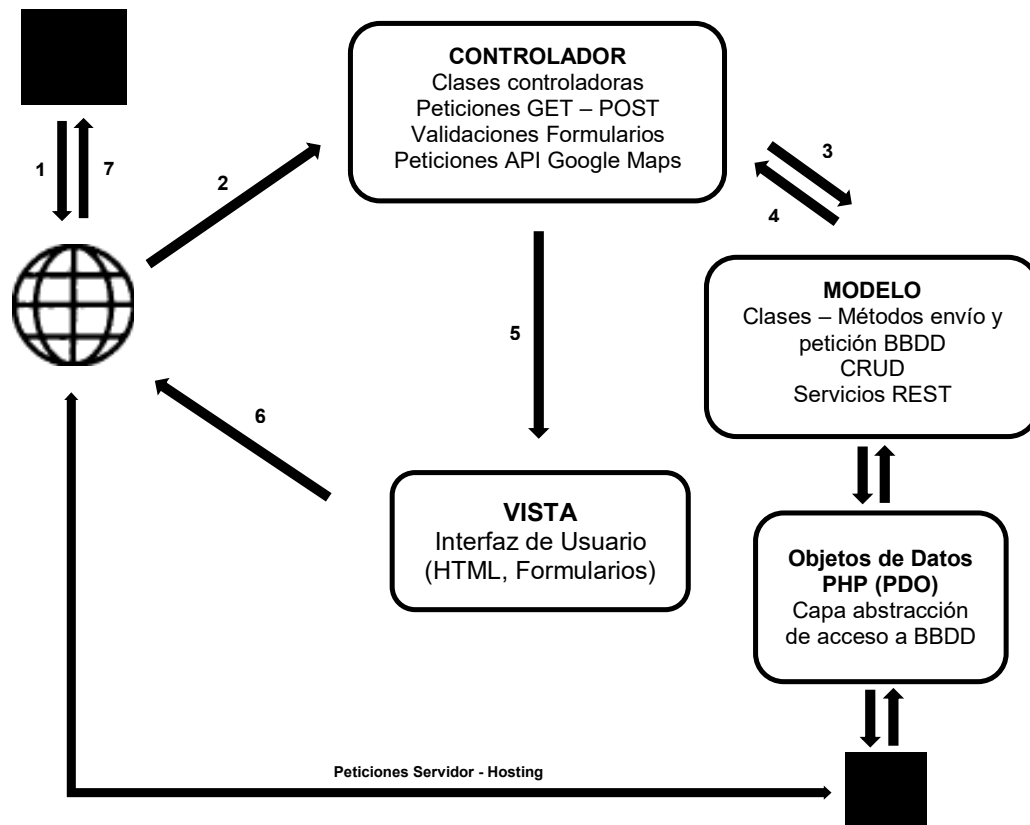


Figura 5. Arquitectura MVC para el actual proyecto

1. Petición del usuario hacia Servidor.
2. Servidor envía petición a Controlador y este recibe y gestiona la misma.
3. Controlador prepara información y llama al Modelo.
4. Controlador recibe los resultados y los prepara.
5. Controlador envía a la Vista los resultados preparados.
6. Vista genera interfaz con resultados y envía al servidor.
7. Servidor muestra al cliente la interfaz.

En la Figura 5, se puede apreciar la utilidad de MVC y el orden en el cual se ejecuta cada parte del patrón de diseño para las peticiones que realiza un usuario a la aplicación web.

#### **4.5. Herramientas de desarrollo**

Este proyecto se enfoca en el ámbito web, por tanto, como principal herramienta de desarrollo se hace uso del servidor web HostGator que proporciona la infraestructura para alojar completamente la aplicación.

El administrador de base de datos utilizado es phpMyAdmin, el mismo que proporciona soporte para el uso de base de datos por medio de la web, para este proyecto MySQL almacenará la información de la aplicación.

En lo que se refiere a lenguajes de programación, PHP se encargará de gestionar servicios en la parte del Modelo, mientras que, en el controlador se utiliza el lenguaje JavaScript. Tanto para la parte del Modelo y Controlador también se hace uso de frameworks basados en JavaScript tales como JQuery, AJAX y AngularJS.

Para el uso y gestión de servicios de geolocalización se hace uso de la API de Google Maps y para generar reportes gráficos se utiliza la API de Highcharts. La parte visual de la aplicación utiliza HTML5 como lenguaje de marcado, y para el diseño de la aplicación se utiliza CSS3 y el framework Bootstrap.

### **5. DESARROLLO DE LA APLICACIÓN**

En esta sección se describe el alcance que tendrá el proyecto y los requerimientos que se van a desarrollar para la aplicación web, los requerimientos estarán organizados por medio de módulos. Se detallan también las Historias Épicas a desarrollar, el Product Backlog para determinar los puntos de estimación y prioridad de cada historia de usuario y por último el

Release Planning en donde se especificará el grupo de Historias de Usuario que corresponden a cada Sprint.

## **5.1. Definición del proyecto**

### **5.1.1. Alcance**

El desarrollo de esta aplicación web, tiene como alcance la generación de reportes en tiempo real sobre animales callejeros en base a la información que proporcionen los usuarios que se encuentren dentro del Distrito Metropolitano de Quito. El proceso a seguir por la aplicación tiene como inicio el registro de usuarios, creación de perfil de usuario, registro de información sobre reportes de animales callejeros por parte de usuarios y finalmente la generación de reportes de forma inmediata.

La aplicación web tendrá dos perfiles, administrador para tener el control total sobre el contenido que se publique en la aplicación, y un perfil de usuario, el mismo que entregará información a la aplicación, la información podrá ser modificada o eliminada por ambos perfiles.

Todo el sistema web será desarrollado con el uso de las herramientas detalladas en el punto 4.5 Herramientas de Desarrollo, por último, la aplicación está enfocada para la Web, sin embargo, se podrá visualizar con facilidad y accesibilidad en dispositivos móviles.

### **5.1.2. Requerimientos**

La aplicación web se conforma de los siguientes módulos:

- Módulo Gestión de Usuarios.
- Módulo Reportes Casos de Animales.
- Módulo Perfil de Usuario y Galerías de Animales Reportados.
- Módulo Estadísticas.
- Módulo Perfil Administrativo.

La aplicación debe estar desplegada en la web para poder ser utilizada con todos los servicios que contiene, la principal causa de ello se debe al uso de API's, es decir, tanto la API de Google Maps como la de Highcharts necesitan trabajar en línea para poder mostrar resultados.

Además, la aplicación debe ser alojada en un servidor web que disponga de un certificado de seguridad, este requerimiento es específicamente para cumplir con las condiciones de uso que propone la API de Google Maps.

Los datos que maneja la aplicación serán obtenidos por medio de servicios REST, los mismos que generarán archivos JSON, a continuación, se muestra el funcionamiento de los servicios REST en la aplicación web.

## 5.2. Historias de usuario épicas

En base a los requerimientos y el alcance planteados para este proyecto, se detallan las siguientes historias de usuario épicas.

Tabla 4. Historia de Usuario Épica HUE01

HISTORIA DE USUARIO ÉPICA	HUE01
<b>Título:</b> Implementar página principal y registro de usuario.	
<b>Descripción:</b> Visualizar una página de inicio, menú y realizar el registro de usuario, ingresando la información requerida para crear una cuenta.	

Tabla 5. Historia de Usuario Épica HUE02

HISTORIA DE USUARIO ÉPICA	HUE02
<b>Título:</b> Realizar envío de reportes de animales.	
<b>Descripción:</b> Ingresar información textual, visual y geográfica para registrar formularios de reporte animal.	

Tabla 6. Historia de Usuario Épica HUE03

HISTORIA DE USUARIO ÉPICA	HUE03
<b>Título:</b> Implementar galerías informativas.	
<b>Descripción:</b> Visualizar galerías de animales separadas por casos y con filtros de búsqueda.	

Tabla 7. Historia de Usuario Épica HUE04

HISTORIA DE USUARIO ÉPICA	HUE04
<b>Título:</b> Implementar perfil de usuario.	
<b>Descripción:</b> Modificar la información personal, los casos reportados y consultar casos cercanos a la ubicación actual.	



Tabla 8. Historia de Usuario Épica HUE05

HISTORIA DE USUARIO ÉPICA	HUE05
<b>Título:</b> Consultar información clasificada de los reportes registrados.	
<b>Descripción:</b> Consultar reportes registrados por periodos de tiempo y por sectores.	

Tabla 9. Historia de Usuario Épica HUE06

HISTORIA DE USUARIO ÉPICA	HUE06
<b>Título:</b> Implementar perfil de administrador y gestión de la aplicación.	
<b>Descripción:</b> Gestionar usuarios, reportes, contenido en general y visualizar estadísticas específicas.	

### 5.3. Product Backlog

Las historias de usuario épicas son de alta complejidad por lo tanto se las puede descomponer en historia de usuario específicas como se detalla a continuación:

Tabla 10. Product Backlog

PRODUCT BACKLOG				
HU ÉPICA	CÓDIGO HU	HISTORIA DE USUARIO	PRIORIDAD	ESTIMACIÓN
HUE01	HU01	Construir página principal y menú personalizado.	Alta	8
	HU02	Implementar registro de usuario	Alta	8
	HU03	Implementar inicio de sesión	Alta	5
	HU04	Implementar recuperación de cuentas	Media	3
HUE02	HU05	Desarrollar registro reportes animales en rescate	Alta	8
	HU06	Desarrollar registro reportes animales en adopción	Alta	8

	HU07	Desarrollar registro reportes animales encontrados	Alta	8
	HU08	Desarrollar registro reportes animales perdidos	Alta	8
HUE03	HU09	Visualizar galería de animales perdidos	Media	5
	HU10	Visualizar galería de animales encontrados	Media	5
	HU11	Visualizar galería de animales en adopción	Media	5
HUE04	HU12	Visualizar y modificar información en perfil de usuario.	Alta	5
	HU13	Gestionar historial de reportes de usuario.	Alta	8
	HU14	Implementar búsqueda de reportes de rescate por área.	Alta	5
HUE05	HU15	Visualizar información estadística dinámica para reportes Adopción	Alta	8
	HU16	Visualizar información estadística dinámica para reportes Rescate	Alta	8
	HU17	Visualizar información estadística dinámica para reportes Encontrados	Alta	8
	HU18	Visualizar información estadística dinámica para reportes Perdidos	Alta	8
HUE06	HU19	Gestión de Usuarios Finales	Alta	6
	HU20	Gestión de Administradores	Alta	6
	HU21	Gestión de Suscriptores	Media	3
	HU22	Gestión de Reportes Adopciones	Alta	6

	HU23	Gestión de Reportes Rescate	Alta	6
	HU24	Gestión de Reportes Perdidos	Alta	6
	HU25	Gestión de Reportes Encontrados	Alta	6
	HU26	Visualización de Estadísticas Específicas	Alta	8
	HU27	Gestión información general de aplicación	Media	3

## 5.4. Release Planning

Por medio del Release Planning, se determinará cuales Historias de Usuario o actividades serán desarrolladas dentro de cada Sprint como se muestra en la siguiente tabla:

Tabla 11. Release Planning

RELEASE PLANNING			
SPRINT 1	SPRINT 2	SPRINT 3	SPRINT 4
HU01	HU09	HU15	HU19
HU02	HU10	HU16	HU20
HU03	HU11	HU17	HU21
HU04	HU12	HU18	HU22
HU05	HU13		HU23
HU06	HU14		HU24
HU07			HU25
HU08			HU26
			HU27

## 5.5. Desarrollo de Sprints

### 5.5.1. SPRINT 0

#### 5.5.1.1. PLANIFICACIÓN DEL SPRINT

El Sprint 0 aplicado en este proyecto se enfocará en la definición y preparación del entorno de desarrollo para la aplicación web, lo cual implica lo que se detalla a continuación:

- Instalación del servidor local MAMP.
- Configuración de puertos (Apache, Nginx, MySQL).

- Diseño e implementación de base de datos MySQL.
- Instalación y configuración editor de código Brackets.
- Descarga e implementación de librerías y API's.

El detalle del diseño de la base de datos se muestra se puede visualizar en el Anexo B.

Para tener un control de las tareas que se realiza en el desarrollo de cada Sprint, en este proyecto se utilizará el Tablero Kanban. En el Sprint 0, se tiene el tablero de la siguiente forma:

Tabla 12. Tablero Kanban inicial Sprint 0

PENDIENTE				EN DESARROLLO	EN PRUEBAS	FINALIZADO
HU01	HU09	HU15	HU19			
HU02	HU10	HU16	HU20			
HU03	HU11	HU17	HU21			
HU04	HU12	HU18	HU22			
HU05	HU13		HU23			
HU06	HU14		HU24			
HU07			HU25			
HU08			HU26			
			HU27			

#### • SPRINT REVIEW

Para el Sprint 0, se ha podido cumplir de manera satisfactoria con la preparación del entorno de desarrollo. No se ha presentado ningún inconveniente al momento de instalar un servidor local y sus respectivas configuraciones.

En lo que se refiere al uso de librerías externas, estas han sido descargadas y almacenadas para evitar inconvenientes o actualizaciones al momento de utilizarlas. Por último, la base de datos ha sido diseñada e implementada directamente con el gestor phpMyAdmin por medio de MySQL.

#### • SPRINT RETROSPECTIVE

El desarrollo del Sprint 0 se llevó a cabo sin inconvenientes, no se ha estimado puntos de esfuerzo para este Sprint ya que se enfoca solo en un proceso de preparación previo al desarrollo, todo el ambiente de desarrollo se ha establecido localmente.

## 5.5.2. SPRINT 1

### 5.5.2.1. PLANIFICACIÓN DEL SPRINT

- **OBJETIVO DEL SPRINT**

Construir una interfaz para el registro de usuarios, acceso a la aplicación web e implementar la opción para que el usuario puede recuperar su contraseña. Además, construir formularios para el registro y envío de reportes sobre animales por parte de los usuarios.

#### Tablero Kanban Inicial Sprint 1

Tabla 13. Tablero Kanban inicial Sprint 1

PENDIENTE			EN DESARROLLO	EN PRUEBAS	FINALIZADO
HU09	HU15	HU19	HU01		
HU10	HU16	HU20	HU02		
HU11	HU17	HU21	HU03		
HU12	HU18	HU22	HU04		
HU13		HU23	HU05		
HU14		HU24	HU06		
		HU25	HU07		
		HU26	HU08		
		HU27			

- **HISTORIAS DE USUARIO DEL SPRINT**

Tabla 14. Historia de Usuario 01

HISTORIA DE USUARIO		HU01
<b>Título:</b> Construir página principal y menú personalizado.		
<b>Descripción:</b> Visualizar una página principal con contenido referente a animales. Además, quiero tener un menú para poder acceder a todas las páginas que contiene la aplicación.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 8	
<b>Criterios de aceptación:</b> 1. La página principal debe estar estructurada con etiquetas semánticas propias de HTML5, además contener al menos un Slider de imágenes y una sección dinámica. 2. Visualizar información sobre los servicios que ofrece la aplicación y su funcionamiento. 3. El menú y pie de página deben ser iguales tanto para la página principal como para las demás páginas, además el menú debe cambiar de opciones dependiendo del usuario que inicie una sesión (usuario final o administrador).		

4. El menú debe tener la foto del usuario en miniatura y como máximo cada opción debe ser de nivel 2.

Tabla 15. Historia de Usuario 02

<b>HISTORIA DE USUARIO</b>		<b>HU02</b>
<b>Título:</b> Implementar registro de usuario.		
<b>Descripción:</b> Registrar mi información por medio de un formulario para la creación de una cuenta en la aplicación web. También quiero poder visualizar Términos y Condiciones de uso de la aplicación antes de enviar el formulario.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 8	
<b>Criterios de aceptación:</b>		
1. Por defecto al momento de crear una cuenta de usuario, se asignará el rol de usuario sin privilegios.		
2. Los datos ingresados en el formulario son validados dependiendo del tipo de dato que se solicite.		
3. El campo correo electrónico debe ser único y solo puede tener asignado una cuenta de usuario.		
4. Los términos y condiciones de uso deben ser visualizadas por medio de una ventana emergente y deben ser aceptados para la creación de una cuenta de usuario.		

Tabla 16. Historia de Usuario 03

<b>HISTORIA DE USUARIO</b>		<b>HU03</b>
<b>Título:</b> Implementar inicio de sesión.		
<b>Descripción:</b> Ingresar a la aplicación web por medio de un formulario que solicite un correo y una contraseña. Además, quiero visualizar mensajes informativos de error en el caso que el ingreso no sea exitoso.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 5	
<b>Criterios de aceptación:</b>		
1. En el formulario de inicio de sesión, el campo correo electrónico y contraseña, serán validados dependiendo del formato que se solicite en la aplicación.		

2. Los mensajes de error al iniciar sesión, deben desplegar información del tipo de error que ha ocurrido.
3. Al momento de iniciar sesión, la aplicación determinará el tipo de usuario que accedió y se creará la una sesión de tipo usuario o administrador mostrando los perfiles y accesos permitidos respectivamente.
4. La sesión de administrador tendrá acceso a todas las páginas de la aplicación.

Tabla 17. Historia de Usuario 04

<b>HISTORIA DE USUARIO</b>		<b>HU04</b>
<b>Título:</b> Implementar recuperación de cuentas.		
<b>Descripción:</b> Recuperar la contraseña de mi cuenta en la aplicación. Quiero poder generar una contraseña nueva y que la misma sea enviada al correo electrónico registrado en la aplicación.		
<b>Prioridad:</b> Media	<b>Esfuerzo:</b> 3	
<b>Criterios de aceptación:</b>		
1. En el formulario de recuperación de cuentas, el campo correo electrónico será validado dependiendo del formato que se solicite en la aplicación.		
2. Al momento de restablecer una contraseña se enviará un correo electrónico al usuario con una contraseña generada automáticamente.		

Tabla 18. Historia de Usuario 05

<b>HISTORIA DE USUARIO</b>		<b>HU05</b>
<b>Título:</b> Desarrollar registro reportes animales en rescate.		
<b>Descripción:</b> Realizar reportes de animales en rescate por medio de un formulario en la aplicación. Quiero poder enviar información en forma de texto, imagen y ubicación geográfica.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 8	
<b>Criterios de aceptación:</b>		
1. Los campos de entrada textual en el formulario deben ser validados dependiendo de la información que se solicite.		
2. Los campos SELECT, deben ser escogidos obligatoriamente.		

- 3. Los campos SELECT que permitan la opción “Otro”, deben presentar un cuadro emergente para entrada de texto.
- 3. Se deberá subir una sola foto de los formatos permitidos (jpg, png, jpeg) y de un tamaño máximo (2MB).
- 4. En el formulario debe haber un mapa que permita seleccionar al usuario una ubicación.

Las Historias de Usuario 06, 07 y 08 son similares a la Historia de Usuario 05, tienen la misma prioridad, requieren el mismo esfuerzo, pero difieren en algunos campos de información que se solicitan por medio de formularios. El resto de información solicitada es la misma (texto, imagen y ubicación). El contenido de las HU 06, 07 y 08 se encuentra detallado en el Anexo A.

• **PUNTOS DE ESFUERZO PARA EL SPRINT 1**

Tabla 19. Puntos de esfuerzo Sprint 1

N° HISTORIA DE USUARIO	PUNTOS DE ESFUERZO
HU01	8
HU02	8
HU03	5
HU04	3
HU05	8
HU06	8
HU07	8
HU08	8
Total Puntos de Esfuerzo	<b>56</b>

• **SPRINT BACKLOG**

Tabla 20. Product Backlog del Sprint 1

CÓDIGO	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	ACTIVIDADES
HU01	Construir página principal y menú personalizado	La página principal debe estar estructurada con etiquetas semánticas propias de HTML5, además contener al menos un Slider de imágenes y una sección dinámica.	<ul style="list-style-type: none"> <li>▪ Crear un nuevo proyecto en el editor de código seleccionado.</li> <li>▪ Diseñar la interface de la página de inicio.</li> <li>▪ Implementar Slider por medio de Bootstrap.</li> <li>▪ Crear controlador para obtener contenido dinámico.</li> </ul>



		Visualizar información sobre los servicios que ofrece la aplicación y su funcionamiento.	<ul style="list-style-type: none"> <li>▪ Agregar texto plano y contenido multimedia por medio de etiquetas HTML5.</li> </ul>
		El menú y pie de página deben ser iguales tanto para la página principal como para las demás páginas, además el menú debe cambiar de opciones dependiendo del usuario que inicie una sesión (usuario final o administrador).	<ul style="list-style-type: none"> <li>▪ Crear método para implementación de menú personalizado.</li> <li>▪ Crear método para visualizar información en pie de página.</li> <li>▪ Crear método para registro de suscriptores.</li> </ul>
		El menú debe tener la foto del usuario en miniatura y como máximo cada opción debe ser de nivel 2.	<ul style="list-style-type: none"> <li>▪ Mostrar foto en miniatura del usuario o administrador por medio de sesiones.</li> </ul>
HU02	Implementar registro de usuario.	Por defecto al momento de crear una cuenta de usuario, se asignará el rol de usuario sin privilegios.	<ul style="list-style-type: none"> <li>▪ Crear método para la creación de usuario y asignación de rol.</li> </ul>
		Los datos ingresados en el formulario son validados dependiendo del tipo de dato que se solicite.	<ul style="list-style-type: none"> <li>▪ Crear controlador para implementación de validación de formulario.</li> </ul>
		El campo correo electrónico debe ser único y solo puede tener asignado una cuenta de usuario.	<ul style="list-style-type: none"> <li>▪ Crear método para la búsqueda y comprobación de usuarios.</li> </ul>
		Los términos y condiciones de uso deben ser visualizadas por medio de una ventana emergente y deben ser aceptados para la creación de una cuenta de usuario.	<ul style="list-style-type: none"> <li>▪ Crear método para la lectura de información.</li> <li>▪ Implementar ventana emergente por medio de Bootstrap.</li> </ul>
HU03	Implementar inicio de sesión.	En el formulario de inicio de sesión, el campo correo electrónico y contraseña, serán validados dependiendo del formato que se solicite en la aplicación.	<ul style="list-style-type: none"> <li>▪ Crear controlador para implementación de validación de formulario.</li> </ul>
		Los mensajes de error al iniciar sesión, deben desplegar información del tipo de error que ha ocurrido.	<ul style="list-style-type: none"> <li>▪ Crear método para validación de registro de usuario desde base de datos.</li> </ul>
		Al momento de iniciar sesión, la aplicación determinará el tipo de usuario que accedió y se creará la una sesión de tipo usuario o administrador	<ul style="list-style-type: none"> <li>▪ Crear método para la implementación de una sesión activa.</li> </ul>

		mostrando los perfiles y accesos permitidos respectivamente.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener y almacenar información de usuario.</li> </ul>
		La sesión de administrador tendrá acceso a todas las páginas de la aplicación	<ul style="list-style-type: none"> <li>▪ Implementar una sesión activa en todas las páginas de la aplicación.</li> </ul>
HU04	Implementar recuperación de cuentas.	En el formulario de recuperación de cuentas, el campo correo electrónico será validado dependiendo del formato que se solicite en la aplicación.	<ul style="list-style-type: none"> <li>▪ Crear controlador para implementación de validación de formulario.</li> </ul>
		Al momento de restablecer una contraseña se enviará un correo electrónico al usuario con una contraseña generada automáticamente.	<ul style="list-style-type: none"> <li>▪ Crear método para envío de correo.</li> <li>▪ Crear método para generación de clave nueva.</li> </ul>
HU5	Desarrollar registro reportes animales en rescate	Los campos de entrada textual en el formulario deben ser validados dependiendo de la información que se solicite.	<ul style="list-style-type: none"> <li>▪ Crear controlador para validación cada campo requerido en formulario.</li> </ul>
		Los campos SELECT, deben ser escogidos obligatoriamente.	<ul style="list-style-type: none"> <li>▪ Crear controlador para validar campos que no han sido manipulados.</li> </ul>
		Los campos SELECT que permitan la opción "Otro", deben presentar un cuadro emergente para entrada de texto.	<ul style="list-style-type: none"> <li>▪ Crear controlador para desplegar entrada de texto para la opción "Otro"</li> </ul>
		En el formulario debe haber un mapa que permita seleccionar al usuario una ubicación.	<ul style="list-style-type: none"> <li>▪ Implementar mapa por medio de API Google Maps.</li> <li>▪ Crear controlador para permitir manipulación de mapas.</li> </ul>

De igual manera que son similares las Historias de Usuario 06, 07 y 08 con la Historia de Usuario 05, los criterios de aceptación y actividades que se plantean para la implementación de las mismas son iguales. Por tanto, se detallan los criterios de aceptación y actividades solo para la Historia de Usuario 05.

### **Tablero Kanban Final Sprint 1**

Todas las actividades planificadas para este Sprint se llevaron a cabo con éxito, por lo tanto, al finalizar el Sprint 1 pasan a estar en fase de prueba.

Tabla 21. Tablero Kanban final Sprint 1

PENDIENTE			EN DESARROLLO	EN PRUEBAS	FINALIZADO
HU09	HU15	HU19		HU01	
HU10	HU16	HU20		HU02	
HU11	HU17	HU21		HU03	
HU12	HU18	HU22		HU04	
HU13		HU23		HU05	
HU14		HU24		HU06	
		HU25		HU07	
		HU26		HU08	
		HU27			

### • PROTOTIPO

Para el desarrollo e implementación de este Sprint, se presentan el prototipo de las siguientes interfaces:



Figura 6. Prototipo de interface - Pantalla Principal

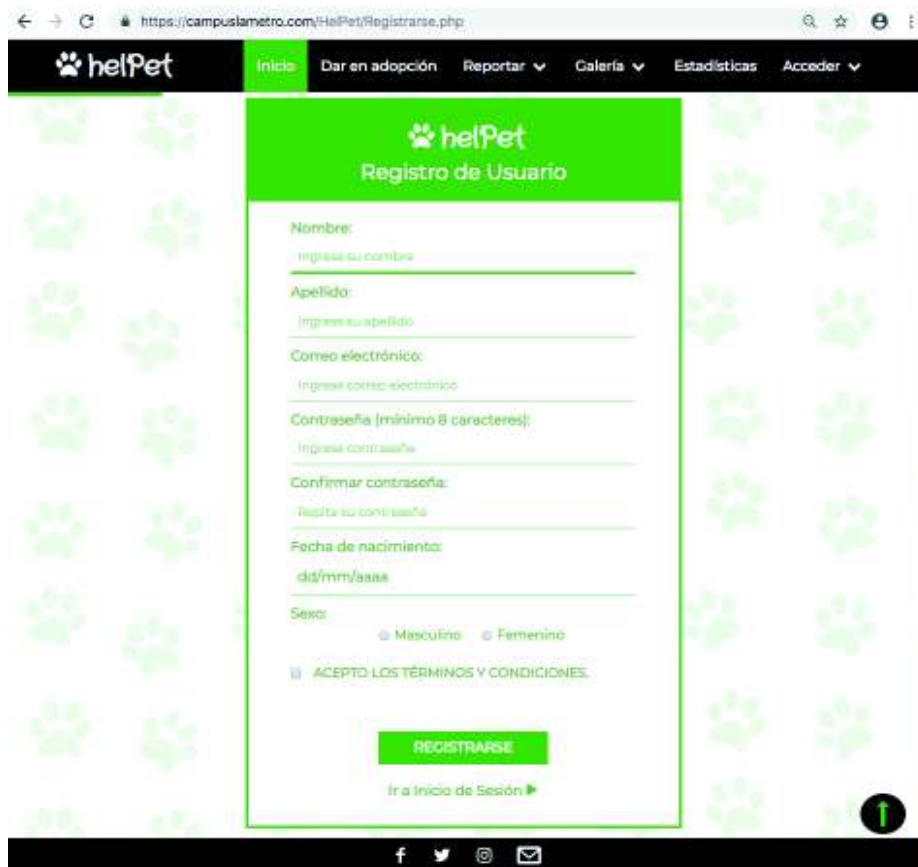


Figura 7. Prototipo de interface - Registro de Usuario



Figura 8. Prototipo de interface - Inicio de Sesión

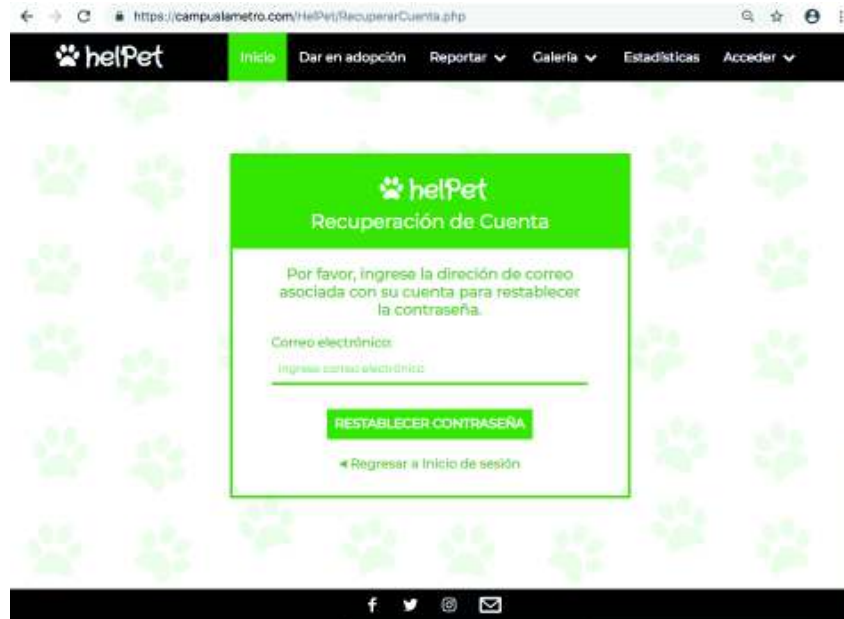


Figura 9. Prototipo de interface - Recuperación de Cuenta



Figura 10. Prototipo de interface - Formulario Reporte Animal en Adopción, Rescate, Perdido y Encontrado

- **SPRINT REVIEW**

Para el Sprint 1, se pudo cumplir con el objetivo planteado el cual consistía en el desarrollo del registro, ingreso y recuperación de contraseñas por parte de los usuarios, también se implementó el registro y envío de reportes sobre animales en cuatro casos diferentes.

Tabla 22. Revisión de Criterios de Aceptación del Sprint 1

CÓDIGO	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	CUMPLIDO
HU01	Construir página principal y menú personalizado	La página principal debe estar estructurada con etiquetas semánticas propias de HTML5, además contener al menos un Slider de imágenes y una sección dinámica.	SI
		Visualizar información sobre los servicios que ofrece la aplicación y su funcionamiento.	SI
		El menú y pie de página deben ser iguales tanto para la página principal como para las demás páginas, además el menú debe cambiar de opciones dependiendo del usuario que inicie una sesión (usuario final o administrador).	SI
		El menú debe tener la foto del usuario en miniatura y como máximo cada opción debe ser de nivel 2.	SI
HU02	Implementar registro de usuario.	Por defecto al momento de crear una cuenta de usuario, se asignará el rol de usuario sin privilegios.	SI
		Los datos ingresados en el formulario son validados dependiendo del tipo de dato que se solicite.	SI
		El campo correo electrónico debe ser único y solo puede tener asignado una cuenta de usuario.	SI
		Los términos y condiciones de uso deben ser visualizadas por medio de una ventana emergente y deben ser aceptados para la creación de una cuenta de usuario.	SI
HU03	Implementar inicio de sesión.	En el formulario de inicio de sesión, el campo correo electrónico y contraseña, serán validados dependiendo del formato que se solicite en la aplicación.	SI
		Los mensajes de error al iniciar sesión, deben desplegar información del tipo de error que ha ocurrido.	SI
		Al momento de iniciar sesión, la aplicación determinará el tipo de usuario que accedió y se creará la una sesión de tipo usuario o administrador mostrando los perfiles y accesos permitidos respectivamente.	SI
		La sesión de administrador tendrá acceso a todas las páginas de la aplicación	SI

HU04	Implementar recuperación de cuentas.	En el formulario de recuperación de cuentas, el campo correo electrónico será validado dependiendo del formato que se solicite en la aplicación.	SI
		Al momento de restablecer una contraseña se enviará un correo electrónico al usuario con una contraseña generada automáticamente.	SI
HU05	Desarrollar registro reportes animales en rescate	Los campos de entrada textual en el formulario deben ser validados dependiendo de la información que se solicite.	SI
		Los campos SELECT, deben ser escogidos obligatoriamente.	SI
		Los campos SELECT que permitan la opción "Otro", deben presentar un cuadro emergente para entrada de texto.	SI
		En el formulario debe haber un mapa que permita seleccionar al usuario una ubicación.	SI
HU06	Desarrollar registro reportes animales en adopción.	Los campos de entrada textual en el formulario deben ser validados dependiendo de la información que se solicite.	SI
		Los campos SELECT, deben ser escogidos obligatoriamente.	SI
		Los campos SELECT que permitan la opción "Otro", deben presentar un cuadro emergente para entrada de texto.	SI
		En el formulario debe haber un mapa que permita seleccionar al usuario una ubicación.	SI
HU07	Desarrollar registro reportes animales encontrados.	Los campos de entrada textual en el formulario deben ser validados dependiendo de la información que se solicite.	SI
		Los campos SELECT, deben ser escogidos obligatoriamente.	SI
		Los campos SELECT que permitan la opción "Otro", deben presentar un cuadro emergente para entrada de texto.	SI
		En el formulario debe haber un mapa que permita seleccionar al usuario una ubicación.	SI
HU08	Desarrollar registro reportes animales perdidos.	Los campos de entrada textual en el formulario deben ser validados dependiendo de la información que se solicite.	SI

		Los campos SELECT, deben ser escogidos obligatoriamente.	SI
		Los campos SELECT que permitan la opción "Otro", deben presentar un cuadro emergente para entrada de texto.	SI
		En el formulario debe haber un mapa que permita seleccionar al usuario una ubicación.	SI

- **GRÁFICO BURNDOWN CHART SPRINT 1**

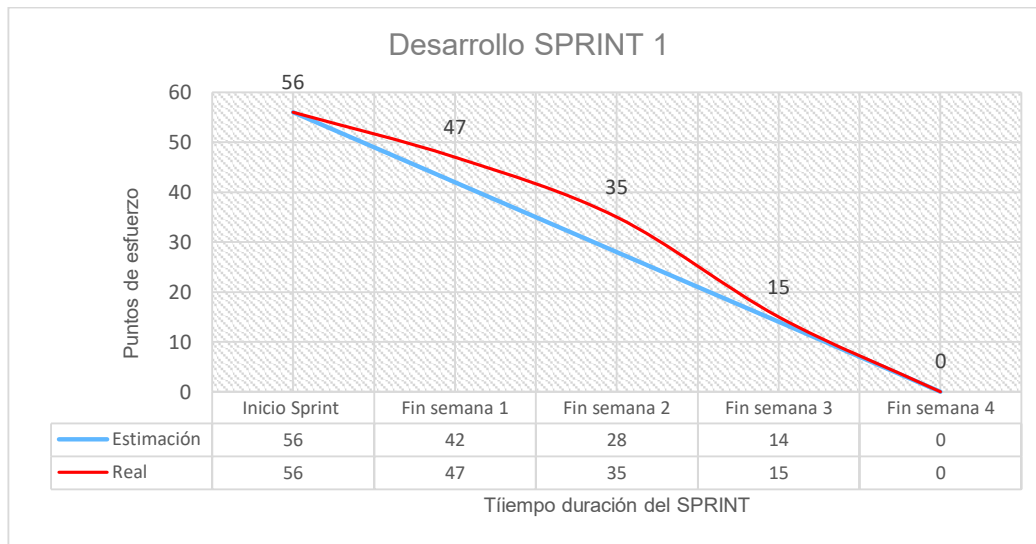


Figura 11. Burndown Chart del Sprint 1

- **SPRINT RETROSPECTIVE**

El desarrollo en el Sprint 1 se llevó a cabo dentro del tiempo establecido logrando cumplir con todos los criterios de aceptación planteados. Como se puede observar en el gráfico Burndown Chart, en la semana 1 hubo ciertos retrasos a la estructuración completa de la página principal y en la semana 3 también se tuvo un retraso en las actividades a causa de problemas con la manipulación de mapas por medio de formularios, en la semana 4 se pudo compensar el tiempo demorado.



### 5.5.3. SPRINT 2

#### 5.5.3.1. PLANIFICACIÓN DEL SPRINT

- **OBJETIVO DEL SPRINT**

Construir galerías de imágenes que permitan visualizar a los animales reportados en los casos de adopción, encontrados y perdidos, con su respectiva información. Además, desarrollar el perfil de usuario que permita modificar la información personal, gestionar reportes realizados por el usuario e implementar geotargeting con los registros de casos de rescate.

#### Tablero Kanban Inicial Sprint 2

Tabla 23. Tablero Kanban inicial Sprint 2

PENDIENTE		EN DESARROLLO	EN PRUEBAS	FINALIZADO
HU15	HU19	HU09		HU01
HU16	HU20	HU10		HU02
HU17	HU21	HU11		HU03
HU18	HU22	HU12		HU04
	HU23	HU13		HU05
	HU24	HU14		HU06
	HU25			HU07
	HU26			HU08
	HU27			

- **HISTORIAS DE USUARIO DEL SPRINT**

Tabla 24. Historia de Usuario 09

HISTORIA DE USUARIO		HU09
<b>Título:</b> Visualizar galería de animales perdidos.		
<b>Descripción:</b> Visualizar una galería de imágenes de animales perdidos y su respectiva información en una ventana emergente. Además, quiero tener filtros de búsqueda y paginación.		
<b>Prioridad:</b> Media	<b>Esfuerzo:</b> 5	
<b>Criterios de aceptación:</b> 1. Las imágenes de la galería deben desplegar una ventana emergente al pulsar sobre ellas, dicha ventana contendrá respectivamente la información de la mascota. 2. La información que se muestra sobre la mascota debe contener texto, imagen y ubicación geográfica por medio de un mapa.		

3. En la ventana emergente con información de la mascota, también se debe visualizar la información de contacto de la persona que realizó el reporte del caso.
4. Los filtros de búsqueda deben incluir parámetros específicos y una opción de entrada de texto que permita buscar por cualquier parámetro de la mascota.
5. Debe haber la opción de paginar las galerías de imágenes, dicha paginación debe actuar conjuntamente con los filtros de búsqueda.

Las Historias de Usuario 10 y 11 son similares a la Historia de Usuario 09, tienen la misma prioridad, requieren el mismo esfuerzo, pero difieren en la información que muestran y los filtros de búsqueda que se presentan para el usuario. El contenido de las HU 10 y 11 se encuentra detallado en el Anexo A.

Tabla 25. Historia de Usuario 12

<b>HISTORIA DE USUARIO</b>		<b>HU12</b>
<b>Título:</b> Visualizar y modificar información en perfil de usuario.		
<b>Descripción:</b> Visualizar y modificar información personal, foto de perfil de usuario, contraseña y ubicación.		
<b>Prioridad:</b> Alta		<b>Esfuerzo:</b> 5
<b>Criterios de aceptación:</b> <ol style="list-style-type: none"> <li>1. La información personal, foto de perfil, contraseña y ubicación deben ser modificadas por medio de formularios visualizados en ventanas emergentes.</li> <li>2. Los formularios deben ser validados dependiendo del tipo de dato que se solicite en los mismos.</li> <li>3. La foto de perfil debe ser de un máximo tamaño (2MB) y un tipo específico de imagen (jpg, jpeg y png).</li> <li>4. Al momento de cambiar la contraseña, se debe solicitar la contraseña actual, y una vez cambiada la contraseña de manera exitosa, se debe cerrar la sesión actual automáticamente.</li> </ol>		

Tabla 26. Historia de Usuario 13

<b>HISTORIA DE USUARIO</b>		<b>HU13</b>
<b>Título:</b> Gestionar historial de reportes de usuario.		

<b>Descripción:</b> Visualizar, modificar y eliminar los registros de reportes de animales en rescate, adopción, perdidos y encontrados que yo he reportado.	
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 8
<b>Criterios de aceptación:</b> <ol style="list-style-type: none"> <li>1. Los reportes realizados por el usuario, deben estar agrupados por el tipo de reporte.</li> <li>2. Por medio de una ventana emergente, se debe cargar la información de cada reporte respectivamente para poder ser visualizada y actualizada.</li> <li>3. El formulario de reporte a ser modificado debe ser validado dependiendo del tipo de dato que se solicite.</li> <li>4. En el formulario se debe presentar un mapa para poder escoger la ubicación geográfica que necesite el usuario.</li> <li>5. Al pulsar el botón eliminar un reporte se debe presentar una ventana emergente que solicite confirmación o cancelación de la acción eliminar.</li> </ol>	

Tabla 27. Historia de Usuario 14

<b>HISTORIA DE USUARIO</b>	<b>HU14</b>
<b>Título:</b> Implementar búsqueda de reportes de rescate por área.	
<b>Descripción:</b> Visualizar reportes de rescate animal registrados en la aplicación en un radio determinado a mi ubicación actual.	
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 5
<b>Criterios de aceptación:</b> <ol style="list-style-type: none"> <li>1. Debe haber un elemento SELECT que permita escoger el radio de búsqueda de casos de rescate, dicho radio de búsqueda debe ser en kilómetros.</li> <li>2. Al momento de cambiar el radio de búsqueda, un mapa debe mostrar de forma dinámica la ubicación de los casos de rescate encontrados y la ubicación actual.</li> <li>3. Se debe visualizar la información básica de los casos de rescate encontrados y la información de la persona que realizo la publicación del caso.</li> <li>4. Se debe calcular la distancia aproximada a la cual se encuentran los casos encontrados con respecto a la ubicación actual.</li> </ol>	

• **PUNTOS DE ESFUERZO PARA EL SPRINT 2**

Tabla 28. Puntos de esfuerzo Sprint 2

N° HISTORIA DE USUARIO	PUNTOS DE ESFUERZO
HU09	5
HU10	5
HU11	5
HU12	5
HU13	8
HU14	5
Total Puntos de Esfuerzo	<b>33</b>

• **SPRINT BACKLOG**

Tabla 29. Product Backlog del Sprint 2

CÓDIGO	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	ACTIVIDADES
HU09	Visualizar galería animales perdidos.	Las imágenes de la galería deben desplegar una ventana emergente al pulsar sobre ellas, dicha ventana contendrá respectivamente la información de la mascota.	<ul style="list-style-type: none"> <li>▪ Implementar ventanas modales por medio de Bootstrap.</li> <li>▪ Crear método para obtener información de cada mascota.</li> <li>▪ Crear controlador para carga de información en ventana modal.</li> </ul>
		La información que se muestra sobre la mascota debe contener texto, imagen y ubicación geográfica por medio de un mapa.	<ul style="list-style-type: none"> <li>▪ Crear método para cargar imágenes por medio de formularios.</li> <li>▪ Crear método para visualizar y manipular mapa.</li> </ul>
		En la ventana emergente con información de la mascota, también se debe visualizar la información de contacto de la persona que realizó el reporte del caso.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener información de contacto.</li> <li>▪ Crear controlador para carga de información en ventana modal.</li> </ul>
		Los filtros de búsqueda deben incluir parámetros específicos y una opción de entrada de texto que permita buscar por cualquier parámetro de la mascota.	<ul style="list-style-type: none"> <li>▪ Crear controlador para implementación de búsqueda general.</li> <li>▪ Crear controlador para búsqueda específica.</li> <li>▪ Crear controlador para visualizar resultados de búsqueda.</li> </ul>
		Debe haber la opción de paginar las galerías de imágenes, dicha paginación	<ul style="list-style-type: none"> <li>▪ Implementar controlador para paginación de registros.</li> </ul>

		debe actuar conjuntamente con los filtros de búsqueda.	<ul style="list-style-type: none"> <li>▪ Crear controlador para filtrar registros conjuntamente con controlador de paginación.</li> <li>▪ Crear controlador para visualizar resultados de búsqueda.</li> </ul>
HU12	Visualizar y modificar información en perfil de usuario.	La información personal, foto de perfil, contraseña y ubicación deben ser modificadas por medio de formularios visualizados en ventanas emergentes.	<ul style="list-style-type: none"> <li>▪ Implementar ventanas modales por medio de Bootstrap.</li> <li>▪ Crear controlador para obtener información de usuario.</li> <li>▪ Crear controlador para cargar información en modal.</li> </ul>
		Los formularios deben ser validados dependiendo del tipo de dato que se solicite en los mismos.	<ul style="list-style-type: none"> <li>▪ Crear método para envío de correo.</li> <li>▪ Crear controlador para validar cada campo solicitado en formulario.</li> </ul>
		La foto de perfil debe ser de un máximo tamaño (2MB) y un tipo específico de imagen (jpg, jpeg y png).	<ul style="list-style-type: none"> <li>▪ Crear método para control de formatos y tamaños de imágenes.</li> <li>▪ Crear método para envío de imágenes por medio de formularios.</li> <li>▪ Crear método para almacenar imágenes en un directorio predefinido.</li> </ul>
		Al momento de cambiar la contraseña, se debe solicitar la contraseña actual, y una vez cambiada la contraseña de manera exitosa, se debe cerrar la sesión actual automáticamente.	<ul style="list-style-type: none"> <li>▪ Crear método para encriptar contraseña.</li> <li>▪ Crear método para actualizar contraseña.</li> <li>▪ Crear método</li> </ul>
HU13	Gestionar historial de reportes de usuario.	Los reportes realizados por el usuario, deben estar agrupados por el tipo de reporte.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener reportes registrados en la base de datos.</li> <li>▪ Crear controlador para visualizar y agrupar registros de reportes por el tipo de caso.</li> </ul>
		Por medio de una ventana emergente, se debe cargar la información de cada reporte respectivamente para poder ser visualizada y actualizada.	<ul style="list-style-type: none"> <li>▪ Implementar ventanas modales por medio de Bootstrap.</li> <li>▪ Crear método para obtener información de un registro específico.</li> </ul>

			<ul style="list-style-type: none"> <li>▪ Crear controlador para cargar texto, imagen y ubicación geográfica en un formulario dentro de una ventana modal.</li> </ul>
		El formulario de reporte a ser modificado debe ser validado dependiendo del tipo de dato que se solicite.	<ul style="list-style-type: none"> <li>▪ Crear controlador para validar cada campo requerido en el formulario de actualización de reporte.</li> </ul>
		En el formulario se debe presentar un mapa para poder escoger la ubicación geográfica que necesite el usuario.	<ul style="list-style-type: none"> <li>▪ Crear controlador para cargar mapa en formulario,</li> <li>▪ Crear controlador para permitir manipulación del mapa.</li> </ul>
		Al pulsar el botón eliminar un reporte se debe presentar una ventana emergente que solicite confirmación o cancelación de la acción eliminar.	<ul style="list-style-type: none"> <li>▪ Implementar ventana moda por medio de Bootstrap.</li> <li>▪ Crear controlador para eliminación de registro de reporte.</li> </ul>
HU14	Implementar búsqueda de reportes de rescate por área.	Debe haber un elemento SELECT que permita escoger el radio de búsqueda de casos de rescate, dicho radio de búsqueda debe ser en kilómetros.	<ul style="list-style-type: none"> <li>▪ Crear controlador para establecer ubicación de usuario.</li> <li>▪ Implementar elemento HTML SELECT que permita escoger un número (1, 2, 5, 10, 20, 25).</li> <li>▪ Cargar elemento SELECT por defecto en la opción 1.</li> </ul>
		Al momento de cambiar el radio de búsqueda, un mapa debe mostrar de forma dinámica la ubicación de los casos de rescate encontrados y la ubicación actual.	<ul style="list-style-type: none"> <li>▪ Crear método para calcular distancias por teorema de cosenos de los Registros de Rescate.</li> <li>▪ Crear método para listar registros de rescate dentro del radio escogido en elemento SELECT.</li> <li>▪ Crear controlador para mostrar ubicaciones de registros de rescate en mapa.</li> </ul>
		Se debe visualizar la información básica de los casos de rescate encontrados y la información de la persona	<ul style="list-style-type: none"> <li>▪ Crear controlador que muestre la información de registros de rescate cercanos.</li> </ul>

		que realice la publicación del caso.	<ul style="list-style-type: none"> <li>▪ Crear controlador que muestre la información de usuario que realizó registro de rescate.</li> </ul>
		Se debe calcular la distancia aproximada a la cual se encuentran los casos encontrados con respecto a la ubicación actual.	<ul style="list-style-type: none"> <li>▪ Crear método para calcular distancias cercanas en base a ubicación de usuario.</li> <li>▪ Crear controlador para desplegar distancias calculadas.</li> </ul>

De igual manera que son similares las Historias de Usuario 10 y 11 con la Historia de Usuario 09, los criterios de aceptación y actividades que se plantean para la implementación de las mismas son iguales, por tanto, se detallan los criterios de aceptación y actividades solo para la Historia de Usuario 09.

### **Tablero Kanban Final Sprint 2**

Todas las actividades planificadas para este Sprint se llevaron a cabo con éxito, por lo tanto, al finalizar el Sprint 2 pasan a estar en fase de prueba.

Tabla 30. Tablero Kanban final Sprint 2

PENDIENTE		EN DESARROLLO	EN PRUEBAS	FINALIZADO
HU15	HU19		HU09	HU01
HU16	HU20		HU10	HU02
HU17	HU21		HU11	HU03
HU18	HU22		HU12	HU04
	HU23		HU13	HU05
	HU24		HU14	HU06
	HU25			HU07
	HU26			HU08
	HU27			

### **• PROTOTIPO**

Para el desarrollo e implementación de este Sprint, se presentan el prototipo de las siguientes interfaces:

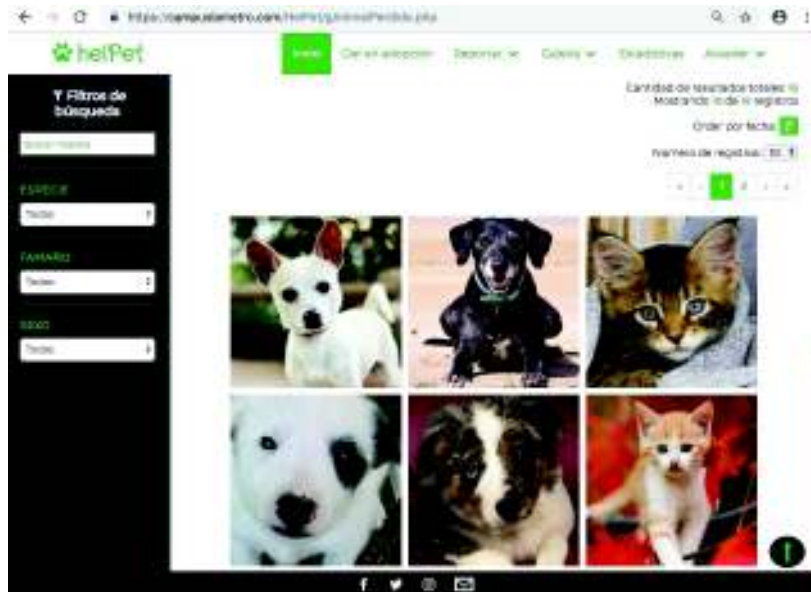


Figura 12. Prototipo de interface - Galería de imágenes para casos registrados de animales en Adopción, Perdidos y Encontrados



Figura 13. Prototipo de interface - Perfil de Usuario / Información de Usuario





Figura 14. Prototipo de interface - Perfil de Usuario / Historial de registros de casos reportados

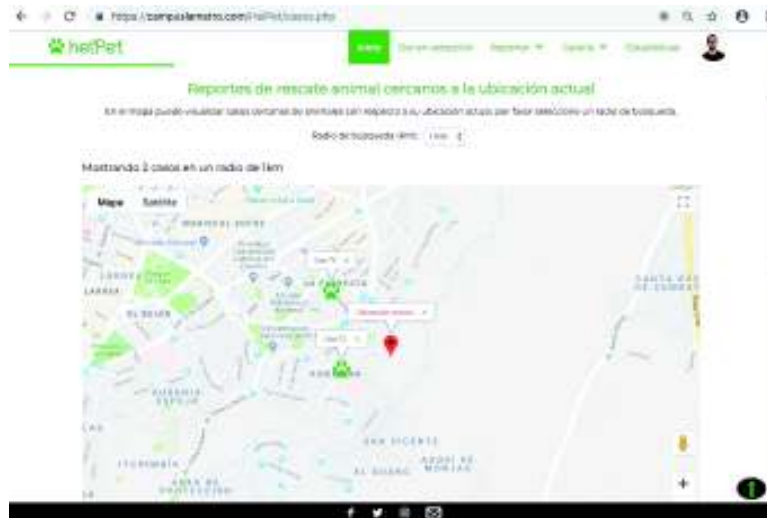


Figura 15. Prototipo de interface - Perfil de Usuario / Casos cercanos de registros de reportes de rescate animal

- **SPRINT REVIEW**

Para el Sprint 2, se cumplió con el objetivo planteado el cual consistía la construcción de galerías de imágenes para la visualización de registro de casos reportados por los usuarios, se implementaron filtros de búsqueda. También se concluyó con la implementación del módulo de perfil de usuario en su totalidad. A continuación, se muestra como se ha podido cumplir con el objetivo planteado para este Sprint.

Tabla 31. Revisión de Criterios de Aceptación del Sprint 2

CÓDIGO	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	CUMPLIDO
HU09	Visualizar galería animales perdidos.	de Las imágenes de la galería deben desplegar una ventana emergente al pulsar sobre ellas, dicha ventana contendrá respectivamente la información de la mascota.	SI
		La información que se muestra sobre la mascota debe contener texto, imagen y ubicación geográfica por medio de un mapa.	SI
		En la ventana emergente con información de la mascota, también se debe visualizar la información de contacto de la persona que realizó el reporte del caso.	SI
		Los filtros de búsqueda deben incluir parámetros específicos y una opción de entrada de texto que permita buscar por cualquier parámetro de la mascota.	SI
		Debe haber la opción de paginar las galerías de imágenes, dicha paginación debe actuar conjuntamente con los filtros de búsqueda.	SI
HU10	Visualizar galería animales encontrados.	de Las imágenes de la galería deben desplegar una ventana emergente al pulsar sobre ellas, dicha ventana contendrá respectivamente la información de la mascota.	SI
		La información que se muestra sobre la mascota debe contener texto, imagen y ubicación geográfica por medio de un mapa.	SI
		En la ventana emergente con información de la mascota, también se debe visualizar la información de contacto de la persona que realizó el reporte del caso.	SI
		Los filtros de búsqueda deben incluir parámetros específicos y una opción de entrada de texto que permita buscar por cualquier parámetro de la mascota.	SI
		Debe haber la opción de paginar las galerías de imágenes, dicha paginación debe actuar conjuntamente con los filtros de búsqueda.	SI
HU11	Visualizar galería de	Las imágenes de la galería deben desplegar una ventana emergente al pulsar sobre ellas, dicha ventana	SI

	animales en adopción.	contendrá respectivamente la información de la mascota.	
		La información que se muestra sobre la mascota debe contener texto, imagen y ubicación geográfica por medio de un mapa.	SI
		En la ventana emergente con información de la mascota, también se debe visualizar la información de contacto de la persona que realizó el reporte del caso.	SI
		Los filtros de búsqueda deben incluir parámetros específicos y una opción de entrada de texto que permita buscar por cualquier parámetro de la mascota.	SI
		Debe haber la opción de paginar las galerías de imágenes, dicha paginación debe actuar conjuntamente con los filtros de búsqueda.	SI
HU12	Visualizar y modificar información en perfil de usuario.	La información personal, foto de perfil, contraseña y ubicación deben ser modificadas por medio de formularios visualizados en ventanas emergentes.	SI
		Los formularios deben ser validados dependiendo del tipo de dato que se solicite en los mismos.	SI
		La foto de perfil debe ser de un máximo tamaño (2MB) y un tipo específico de imagen (jpg, jpeg y png).	SI
		Al momento de cambiar la contraseña, se debe solicitar la contraseña actual, y una vez cambiada la contraseña de manera exitosa, se debe cerrar la sesión actual automáticamente.	SI
HU13	Gestionar historial de reportes de usuario.	Los reportes realizados por el usuario, deben estar agrupados por el tipo de reporte.	SI
		Por medio de una ventana emergente, se debe cargar la información de cada reporte respectivamente para poder ser visualizada y actualizada.	SI
		El formulario de reporte a ser modificado debe ser validado dependiendo del tipo de dato que se solicite.	SI
		En el formulario se debe presentar un mapa para poder escoger la ubicación geográfica que necesite el usuario.	SI
		Al pulsar el botón eliminar un reporte se debe presentar una ventana emergente que solicite confirmación o cancelación de la acción eliminar.	SI

HU14	Implementar búsqueda de reportes de rescate por área.	Debe haber un elemento SELECT que permita escoger el radio de búsqueda de casos de rescate, dicho radio de búsqueda debe ser en kilómetros.	SI
		Al momento de cambiar el radio de búsqueda, un mapa debe mostrar de forma dinámica la ubicación de los casos de rescate encontrados y la ubicación actual.	SI
		Se debe visualizar la información básica de los casos de rescate encontrados y la información de la persona que realizó la publicación del caso.	SI
		Se debe calcular la distancia aproximada a la cual se encuentran los casos encontrados con respecto a la ubicación actual.	SI

- **GRÁFICO BURNDOWN CHART SPRINT 2**

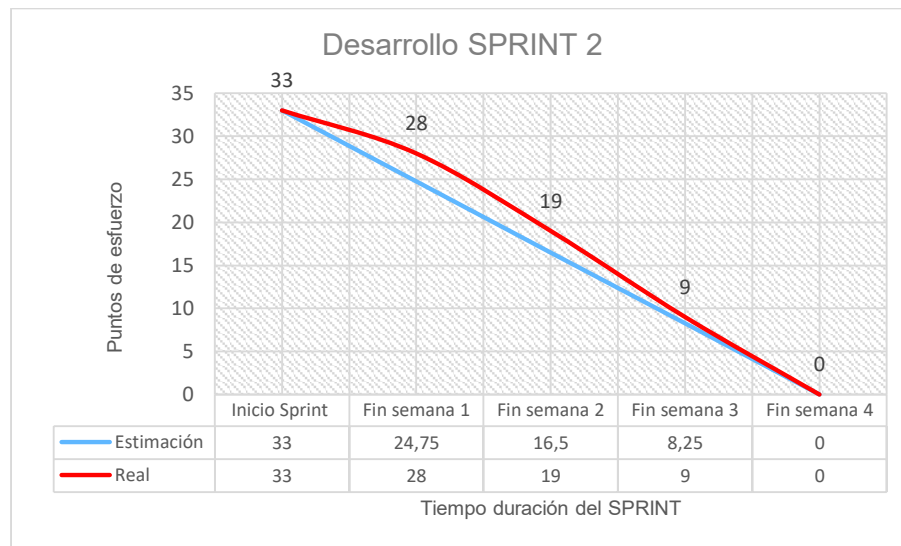


Figura 16. Burndown Chart del Sprint 2

- **SPRINT RETROSPECTIVE**

El desarrollo en el Sprint 2 se llevó a cabo dentro del tiempo establecido logrando cumplir con todos los criterios de aceptación planteados. Como se puede observar en el gráfico Burndown Chart, en la semana 1 y 2 hubo ciertos retrasos debido a errores en los filtros de búsqueda implementados en las galerías de imágenes. En la semana 3 y 4 se consiguió implementar lo planteado de forma correcta recuperando el tiempo de retraso.

### 5.5.4. SPRINT 3

#### 5.5.4.1. PLANIFICACIÓN DEL SPRINT

- **OBJETIVO DEL SPRINT**

Construir una interfaz que permita al usuario visualizar información estadística básica calculada en base a los registros realizados en la aplicación web de los cuatro tipos de casos de reportes disponibles. Además, implementar gráficos que se relacionen con la información mostrada.

#### Tablero Kanban Inicial Sprint 3

Tabla 32. Tablero Kanban inicial Sprint 3

PENDIENTE	EN DESARROLLO	EN PRUEBAS	FINALIZADO	
HU19	HU15		HU01	HU09
HU20	HU16		HU02	HU10
HU21	HU17		HU03	HU11
HU22	HU18		HU04	HU12
HU23			HU05	HU13
HU24			HU06	HU14
HU25			HU07	
HU26			HU08	
HU27				

- **HISTORIAS DE USUARIO DEL SPRINT**

Tabla 33. Historia de Usuario 15

HISTORIA DE USUARIO		HU15
<b>Título:</b> Visualizar información estadística dinámica para reportes Adopción.		
<b>Descripción:</b> Generar y visualizar en un periodo de tiempo datos estadísticos generales sobre animales en adopción. Además, quiero visualizar un mapa, para poder escoger cada una de las parroquias y visualizar datos estadísticos específicos sobre la parroquia escogida.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 8	
<b>Criterios de aceptación:</b> 1. Debe haber dos campos, fecha de inicio y fecha fin, que permitan generar información estadística de forma dinámica. 2. Al cargar la página estadística, debe contener información de todos los registros de reportes que tenga la aplicación web hasta la fecha actual. 3. La información textual que se visualice debe ser numérica y porcentual previamente calculada.		

4. La información que se visualice debe estar separada por distintos parámetros de los registros de adopción (especie, sexo, tamaño, esterilizado y edad).
5. Debe haber un mapa en el cual se diferencien los dos tipos de parroquias (urbanas y rurales). Cada parroquia debe tener una leyenda con el porcentaje de casos existentes en dicha parroquia.
6. En una tabla se debe visualizar el porcentaje y número de casos de cada parroquia.
7. Debe haber información detallada de cada parroquia por medio de texto y gráficos.

Las Historias de Usuario 16, 17 y 18 son similares a la Historia de Usuario 15, tienen la misma prioridad, requieren el mismo esfuerzo, pero difieren en la información y gráficos que se presentan para el usuario. El contenido de las HU 16, 17 y 18 se encuentra detallado en el Anexo A.

- **PUNTOS DE ESFUERZO PARA EL SPRINT 3**

Tabla 34. Puntos de esfuerzo Sprint 3

N° HISTORIA DE USUARIO	PUNTOS DE ESFUERZO
HU15	8
HU16	8
HU17	8
HU18	8
Total Puntos de Esfuerzo	<b>32</b>

- **SPRINT BACKLOG**

Tabla 35. Product Backlog del Sprint 3

CÓDIGO	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	ACTIVIDADES
HU15	Visualizar información estadística dinámica para reportes Adopción.	Debe haber dos campos, fecha de inicio y fecha fin, que permitan generar información estadística de forma dinámica.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener información de registros que este dentro un periodo de tiempo escogido.</li> </ul>
		Al cargar la página estadística, debe contener información de todos los registros de reportes que tenga la aplicación web hasta la fecha actual.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener registros completos hasta la fecha actual.</li> <li>▪ Crear controlador para implementar información por medio de gráficos dinámicos.</li> <li>▪ Crear controlador para implementar</li> </ul>

			información por medio de mapas dinámicos.
		La información textual que se visualice debe ser numérica y porcentual previamente calculada.	<ul style="list-style-type: none"> <li>▪ Crear método para calcular porcentajes y números totales en base a la información de los registros.</li> </ul>
		La información que se visualice debe estar separada por distintos parámetros de los registros de adopción (especie, sexo, tamaño, esterilizado y edad).	<ul style="list-style-type: none"> <li>▪ Crear controladores para independizar y visualizar la información.</li> </ul>
		Debe haber un mapa en el cual se diferencien los dos tipos de parroquias (urbanas y rurales). Cada parroquia debe tener una leyenda con el porcentaje de casos existentes en dicha parroquia.	<ul style="list-style-type: none"> <li>▪ Crear controlador para separar visual y funcionalmente parroquias en el mapa.</li> <li>▪ Crear controlador para implementar etiquetas con información sobre cada parroquia.</li> </ul>
		En una tabla se debe visualizar el porcentaje y número de casos de cada parroquia.	<ul style="list-style-type: none"> <li>▪ Crear controlador para implementar una tabla con datos numéricos y porcentuales.</li> </ul>
		Debe haber información detallada de cada parroquia por medio de texto y gráficos	<ul style="list-style-type: none"> <li>▪ Crear controlador para obtener información sobre cada parroquia.</li> <li>▪ Crear controlador para visualizar datos textuales, visuales y gráficos.</li> </ul>

De igual manera que son similares las Historias de Usuario 16, 17 y 18 con la Historia de Usuario 15, los criterios de aceptación y actividades que se plantean para la implementación de las mismas son iguales, por tanto, se detallan los criterios de aceptación y actividades solo para la Historia de Usuario 15.

### **Tablero Kanban Final Sprint 3**

Todas las actividades planificadas para este Sprint se llevaron a cabo con éxito, por lo tanto, al finalizar el Sprint 3 pasan a estar en fase de prueba.

Tabla 36. Tablero Kanban final Sprint 3

PENDIENTE	EN DESARROLLO	EN PRUEBAS	FINALIZADO	
HU19		HU15	HU01	HU09
HU20		HU16	HU02	HU10
HU21		HU17	HU03	HU11
HU22		HU18	HU04	HU12
HU23			HU05	HU13
HU24			HU06	HU14
HU25			HU07	
HU26			HU08	
HU27				

- **PROTOTIPO**

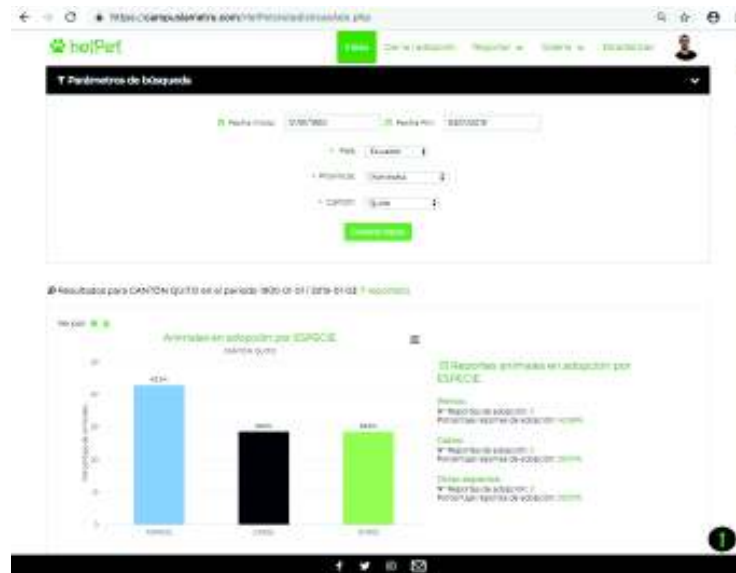


Figura 17. Prototipo de interface - Estadísticas perfil usuario

- **SPRINT REVIEW**

Tabla 37. Revisión de Criterios de Aceptación del Sprint 3

CÓDIGO	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	CUMPLIDO
HU15	Visualizar información estadística dinámica para reportes Adopción.	Debe haber dos campos, fecha de inicio y fecha fin, que permitan generar información estadística de forma dinámica.	SI
		Al cargar la página estadística, debe contener información de todos los registros de reportes que tenga la aplicación web hasta la fecha actual.	SI



		La información textual que se visualice debe ser numérica y porcentual previamente calculada.	SI
		La información que se visualice debe estar separada por distintos parámetros de los registros de adopción (especie, sexo, tamaño, esterilizado y edad).	SI
		Debe haber un mapa en el cual se diferencien los dos tipos de parroquias (urbanas y rurales). Cada parroquia debe tener una leyenda con el porcentaje de casos existentes en dicha parroquia.	SI
		En una tabla se debe visualizar el porcentaje y número de casos de cada parroquia.	SI
		Debe haber información detallada de cada parroquia por medio de texto y gráficos	SI
HU16	Visualizar información estadística dinámica para reportes Rescate.	Debe haber dos campos, fecha de inicio y fecha fin, que permitan generar información estadística de forma dinámica.	SI
		Al cargar la página estadística, debe contener información de todos los registros de reportes que tenga la aplicación web hasta la fecha actual.	SI
		La información textual que se visualice debe ser numérica y porcentual previamente calculada.	SI
		La información que se visualice debe estar separada por distintos parámetros de los registros de adopción (especie, sexo, tamaño, esterilizado y edad).	SI
		Debe haber un mapa en el cual se diferencien los dos tipos de parroquias (urbanas y rurales). Cada parroquia debe tener una leyenda con el porcentaje de casos existentes en dicha parroquia.	SI
		En una tabla se debe visualizar el porcentaje y número de casos de cada parroquia.	SI
		Debe haber información detallada de cada parroquia por medio de texto y gráficos	SI
HU17	Visualizar información estadística dinámica para	Debe haber dos campos, fecha de inicio y fecha fin, que permitan generar información estadística de forma dinámica.	SI.

	reportes Encontrados.	Al cargar la página estadística, debe contener información de todos los registros de reportes que tenga la aplicación web hasta la fecha actual.	SI
		La información textual que se visualice debe ser numérica y porcentual previamente calculada.	SI
		La información que se visualice debe estar separada por distintos parámetros de los registros de adopción (especie, sexo, tamaño, esterilizado y edad).	SI
		Debe haber un mapa en el cual se diferencien los dos tipos de parroquias (urbanas y rurales). Cada parroquia debe tener una leyenda con el porcentaje de casos existentes en dicha parroquia.	SI
		En una tabla se debe visualizar el porcentaje y número de casos de cada parroquia.	SI
		Debe haber información detallada de cada parroquia por medio de texto y gráficos	SI
HU18	Visualizar información estadística dinámica para reportes Perdidos.	Debe haber dos campos, fecha de inicio y fecha fin, que permitan generar información estadística de forma dinámica.	SI
		Al cargar la página estadística, debe contener información de todos los registros de reportes que tenga la aplicación web hasta la fecha actual.	SI
		La información textual que se visualice debe ser numérica y porcentual previamente calculada.	SI
		La información que se visualice debe estar separada por distintos parámetros de los registros de adopción (especie, sexo, tamaño, esterilizado y edad).	SI
		Debe haber un mapa en el cual se diferencien los dos tipos de parroquias (urbanas y rurales). Cada parroquia debe tener una leyenda con el porcentaje de casos existentes en dicha parroquia.	SI
		En una tabla se debe visualizar el porcentaje y número de casos de cada parroquia.	SI
		Debe haber información detallada de cada parroquia por medio de texto y gráficos	SI

- **GRÁFICO BURNDOWN CHART SPRINT 3**

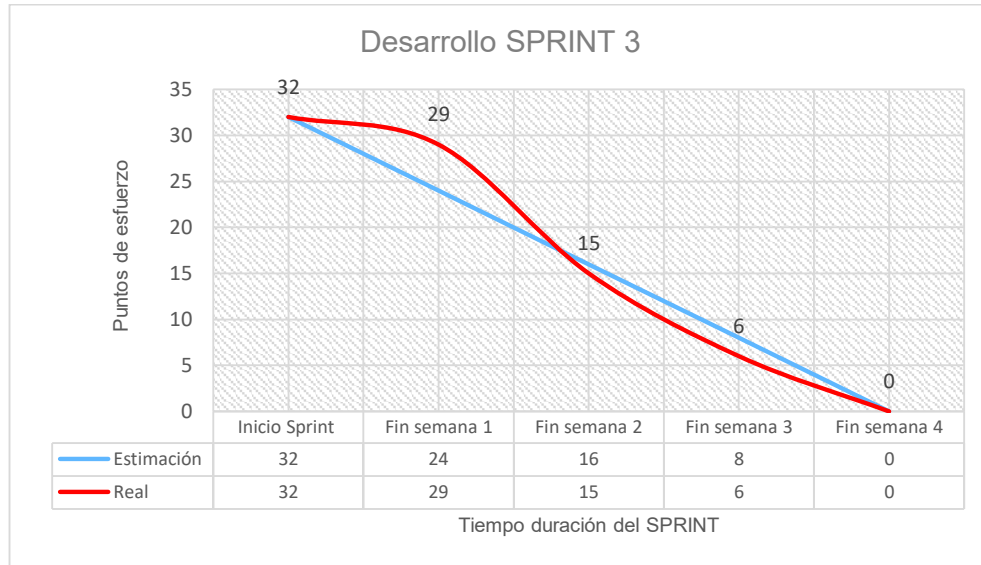


Figura 18. Burndown Chart del Sprint 3

- **SPRINT RETROSPECTIVE**

El desarrollo en el Sprint 3 se llevó a cabo dentro del tiempo establecido logrando cumplir con todos los criterios de aceptación planteados. Como se puede observar en el gráfico Burndown Chart, en la semana 1 y 2 se presentaron inconvenientes al momento de trabajar con la API de Highcharts, los inconvenientes fueron solucionados para la semana 3 y se pudo optimizar el tiempo de desarrollo gracias a la reutilización de ciertos servicios ya implementados.

### 5.5.5. SPRINT 4

#### 5.5.5.1. PLANIFICACIÓN DEL SPRINT

- **OBJETIVO DEL SPRINT**

Desarrollar el módulo administración para gestionar los registros de usuarios, reportes e información general de la aplicación. Además, implementar estadísticas específicas que contengan tanto información textual como gráfica.

## Tablero Kanban Inicial Sprint 4

Tabla 38. Tablero Kanban inicial Sprint 4

PENDIENTE	EN DESARROLLO	EN PRUEBAS	FINALIZADO		
	HU19		HU01	HU09	HU15
	HU20		HU02	HU10	HU16
	HU21		HU03	HU11	HU17
	HU22		HU04	HU12	HU18
	HU23		HU05	HU13	
	HU24		HU06	HU14	
	HU25		HU07		
	HU26		HU08		
	HU27				

### • HISTORIAS DE USUARIO DEL SPRINT

Tabla 39. Historia de Usuario 19

HISTORIA DE USUARIO		HU19
<b>Título:</b> Gestión de Usuarios Finales.		
<b>Descripción:</b> Leer, editar y eliminar los registros de usuarios finales en la aplicación web. Además, quiero poder buscar a los usuarios finales por cualquier parámetro.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 6	
<p><b>Criterios de aceptación:</b></p> <ol style="list-style-type: none"> <li>1. Debe haber una barra de búsqueda que permita buscar a los usuarios por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.</li> <li>2. Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.</li> <li>3. Debe haber paginación que permita visualizar y organizar los registros disponibles.</li> <li>4. En una tabla se deben visualizar los registros de los usuarios con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.</li> <li>5. Debe haber un botón para editar la información del usuario seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.</li> <li>6. Debe haber un botón para eliminar a un usuario específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.</li> </ol>		

Tabla 40. Historia de Usuario 20

<b>HISTORIA DE USUARIO</b>		<b>HU20</b>
<b>Título:</b> Gestión de Administradores.		
<b>Descripción:</b> Leer, editar y eliminar los registros de los demás administradores en la aplicación web. Además, quiero poder buscar a todos los administradores por cualquier parámetro.		
<b>Prioridad:</b> Alta		<b>Esfuerzo:</b> 6
<p><b>Criterios de aceptación:</b></p> <ol style="list-style-type: none"> <li>1. Debe haber una barra de búsqueda que permita buscar a los usuarios administradores por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.</li> <li>2. Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.</li> <li>3. Debe haber paginación que permita visualizar y organizar los registros disponibles.</li> <li>4. En una tabla se deben visualizar los registros de los usuarios administradores con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.</li> <li>5. Debe haber un botón para editar la información del usuario administrador seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.</li> <li>6. Debe haber un botón para eliminar a un usuario específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.</li> </ol>		

Tabla 41. Historia de Usuario 21

<b>HISTORIA DE USUARIO</b>		<b>HU21</b>
<b>Título:</b> Gestión de Suscriptores.		
<b>Descripción:</b> Leer, editar y eliminar los registros de usuarios suscriptores en la aplicación web.		
<b>Prioridad:</b> Media		<b>Esfuerzo:</b> 3

**Criterios de aceptación:**

1. Debe haber una barra de búsqueda que permita buscar a los usuarios suscriptores por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.
2. Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.
3. Debe haber paginación que permita visualizar y organizar los registros disponibles.
4. En una tabla se deben visualizar los registros de los usuarios suscriptores con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.
5. Debe haber un botón para editar el estado del registro seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal que permita el cambio de estado (Activo / Inactivo).
6. Debe haber un botón para eliminar a un usuario suscriptor específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.

Tabla 42. Historia de Usuario 22

HISTORIA DE USUARIO		HU22
<b>Título:</b> Gestión de Reportes Adopciones.		
<b>Descripción:</b> Leer, editar y eliminar los registros de reportes de casos de adopción en la aplicación web. Además, quiero poder buscar a dichos reportes por cualquier parámetro.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 6	
<b>Criterios de aceptación:</b> <ol style="list-style-type: none"> <li>1. Debe haber una barra de búsqueda que permita buscar reportes de adopción por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.</li> <li>2. Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.</li> <li>3. Debe haber paginación que permita visualizar y organizar los registros disponibles.</li> <li>4. En una tabla se deben visualizar los registros de reportes de adopción con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.</li> </ol>		

5. Debe haber un botón para editar la información de reporte seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.
6. Debe haber un botón para eliminar a un reporte de adopción, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del reporte.

Las Historias de Usuario 23, 24 y 25 son similares a la Historia de Usuario 22, tienen la misma prioridad, requieren el mismo esfuerzo, pero permiten visualizar al administrador distintos tipos de reportes. El contenido de las HU 23, 24 y 25 se encuentra detallado en el Anexo A.

Tabla 43. Historia de Usuario 26

HISTORIA DE USUARIO		HU26
<b>Título:</b> Visualización de Estadísticas Específicas.		
<b>Descripción:</b> Visualizar datos estadísticos básicos de los registros realizados en la aplicación web para los casos de animales en adopción, rescate, perdidos y encontrados.		
<b>Prioridad:</b> Alta	<b>Esfuerzo:</b> 8	
<b>Criterios de aceptación:</b> <ol style="list-style-type: none"> <li>1. Se debe visualizar en general datos estadísticos por medio de texto y gráficos para el año actual y 4 años atrás.</li> <li>2. En una tabla se debe tener detallado cada mes del año actual datos estadísticos básicos.</li> <li>3. En una tabla se debe tener detallado en un periodo desde el año actual y 4 años atrás, datos estadísticos básicos.</li> <li>4. Se debe tener un campo de fecha de inicio y fecha fin para generar datos estadísticos para periodos de tiempo específicos.</li> <li>5. Por medio de gráficos, texto y un mapa, se debe visualizar información estadística que incluya información por cada parroquia.</li> </ol>		

Tabla 44. Historia de Usuario 27

HISTORIA DE USUARIO		HU27
<b>Título:</b> Gestión información general de aplicación.		

<b>Descripción:</b> Gestionar la información sobre redes sociales, contacto y términos y condiciones que se visualiza en la aplicación web.	
<b>Prioridad:</b> Media	<b>Esfuerzo:</b> 3
<b>Criterios de aceptación:</b> 1. Debe haber un formulario por medio del cual pueda editar las redes sociales que se visualiza en el pie de página de la aplicación web. 2. Debe haber un formulario por medio del cual pueda editar la información de contacto que se visualiza en el pie de página de la aplicación web. 3. Debe haber un formulario por medio del cual pueda editar los términos y condiciones que se visualiza en el registro de usuario de la aplicación web.	

• **PUNTOS DE ESFUERZO PARA EL SPRINT 4**

Tabla 45. Puntos de esfuerzo Sprint 4

Nº HISTORIA DE USUARIO	PUNTOS DE ESFUERZO
HU19	6
HU20	6
HU21	3
HU22	6
HU23	6
HU24	6
HU25	6
HU26	8
HU27	3
Total Puntos de Esfuerzo	<b>50</b>

• **SPRINT BACKLOG**

Tabla 46. Product Backlog del Sprint 4

CÓDIGO	HISTORIA DE USUARIO	CRITERIO ACEPTACIÓN	DE	ACTIVIDADES
HU19	Gestión de Usuarios Finales.	Debe haber una barra de búsqueda que permita buscar a los usuarios por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.		<ul style="list-style-type: none"> <li>▪ Crear método para obtener registros de usuarios.</li> <li>▪ Crear controlador para filtrar información desde la Vista aplicando AngularJS.</li> </ul>
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.		<ul style="list-style-type: none"> <li>▪ Crear controlador para configuración de paginación dinámica.</li> </ul>



		Debe haber paginación que permita visualizar y organizar los registros disponibles.	<ul style="list-style-type: none"> <li>▪ Implementar controlador para paginar los registros que se visualizan.</li> </ul>
		En una tabla se deben visualizar los registros de los usuarios con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	<ul style="list-style-type: none"> <li>▪ Implementar método para obtener registros de usuarios.</li> <li>▪ Crear controlador para visualizar registros en una tabla.</li> <li>▪ Implementar filtros de AngularJS para permitir ordenar columnas.</li> </ul>
		Debe haber un botón para editar la información del usuario seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.	<ul style="list-style-type: none"> <li>▪ Crear controlador para obtener un registro específico.</li> <li>▪ Crear controlador para visualizar información en texto, imagen y mapa en un formulario.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>
		Debe haber un botón para eliminar a un usuario específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.	<ul style="list-style-type: none"> <li>▪ Crear método para eliminar un registro específico.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>
HU20	Gestión de Administradores.	Debe haber una barra de búsqueda que permita buscar a los usuarios administradores por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener registros de usuarios.</li> <li>▪ Crear controlador para filtrar información desde la Vista aplicando AngularJS.</li> </ul>
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	<ul style="list-style-type: none"> <li>▪ Crear controlador para configuración de paginación dinámica.</li> </ul>
		Debe haber paginación que permita visualizar y organizar los registros disponibles.	<ul style="list-style-type: none"> <li>▪ Implementar controlador para paginar los registros que se visualizan.</li> </ul>

		<p>En una tabla se deben visualizar los registros de los usuarios administradores con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.</p>	<ul style="list-style-type: none"> <li>▪ Implementar método para obtener registros de usuarios.</li> <li>▪ Crear controlador para visualizar registros en una tabla.</li> <li>▪ Implementar filtros de AngularJS para permitir ordenar columnas.</li> </ul>
		<p>Debe haber un botón para editar la información del usuario administrador seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.</p>	<ul style="list-style-type: none"> <li>▪ Crear controlador para obtener un registro específico.</li> <li>▪ Crear controlador para visualizar información en texto, imagen y mapa en un formulario.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>
		<p>Debe haber un botón para eliminar a un usuario específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.</p>	<ul style="list-style-type: none"> <li>▪ Crear método para eliminar un registro específico.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>
HU21	Gestión de Suscriptores.	<p>Debe haber una barra de búsqueda que permita buscar a los usuarios suscriptores por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.</p>	<ul style="list-style-type: none"> <li>▪ Crear método para obtener registros de usuarios.</li> <li>▪ Crear controlador para filtrar información desde la Vista aplicando AngularJS.</li> </ul>
		<p>Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.</p>	<ul style="list-style-type: none"> <li>▪ Crear controlador para configuración de paginación dinámica.</li> </ul>
		<p>Debe haber paginación que permita visualizar y organizar los registros disponibles.</p>	<ul style="list-style-type: none"> <li>▪ Implementar controlador para paginar los registros que se visualizan.</li> </ul>
		<p>En una tabla se deben visualizar los registros de los usuarios suscriptores con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.</p>	<ul style="list-style-type: none"> <li>▪ Implementar método para obtener registros de usuarios.</li> <li>▪ Crear controlador para visualizar registros en una tabla.</li> <li>▪ Implementar filtros de AngularJS para permitir ordenar columnas.</li> </ul>

		<p>Debe haber un botón para editar el estado del registro seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal que permita el cambio de estado (Activo / Inactivo).</p>	<ul style="list-style-type: none"> <li>▪ Crear controlador para obtener un registro específico.</li> <li>▪ Crear controlador para visualizar información en texto, imagen y mapa en un formulario.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>
		<p>Debe haber un botón para eliminar a un usuario suscriptor específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.</p>	<ul style="list-style-type: none"> <li>▪ Crear método para eliminar un registro específico.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>
HU22	Gestión de Reportes de Adopciones.	<p>Debe haber una barra de búsqueda que permita buscar reportes de adopción por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.</p>	<ul style="list-style-type: none"> <li>▪ Crear método para obtener registros de usuarios.</li> <li>▪ Crear controlador para filtrar información desde la Vista aplicando AngularJS.</li> </ul>
		<p>Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.</p>	<ul style="list-style-type: none"> <li>▪ Crear controlador para configuración de paginación dinámica.</li> </ul>
		<p>Debe haber paginación que permita visualizar y organizar los registros disponibles.</p>	<ul style="list-style-type: none"> <li>▪ Implementar controlador para paginar los registros que se visualizan.</li> </ul>
		<p>En una tabla se deben visualizar los registros de reportes de adopción con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.</p>	<ul style="list-style-type: none"> <li>▪ Implementar método para obtener registros de usuarios.</li> <li>▪ Crear controlador para visualizar registros en una tabla.</li> <li>▪ Implementar filtros de AngularJS para permitir ordenar columnas.</li> </ul>
		<p>Debe haber un botón para editar la información de reporte seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.</p>	<ul style="list-style-type: none"> <li>▪ Crear controlador para obtener un registro específico.</li> <li>▪ Crear controlador para visualizar información en texto, imagen y mapa en un formulario.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>

		Debe haber un botón para eliminar a un reporte de adopción, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del reporte.	<ul style="list-style-type: none"> <li>▪ Crear método para eliminar un registro específico.</li> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> </ul>
HU26	Visualización de Estadísticas Específicas	Se debe visualizar en general datos estadísticos por medio de texto y gráficos para el año actual y 4 años atrás.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener información de los registros sobre reportes un caso en específico.</li> <li>▪ Crear método para implementar por medio de API Highcharts gráficos.</li> <li>▪ Crear controlador para realizar cálculos estadísticos básicos.</li> </ul>
		En una tabla se debe tener detallado cada mes del año actual datos estadísticos básicos.	<ul style="list-style-type: none"> <li>▪ Crear controlador para separar registros de reportes por cada mes del año actual.</li> <li>▪ Crear controlador para visualizar registros filtrados por medio de gráficos.</li> </ul>
		En una tabla se debe tener detallado en un periodo desde el año actual y 4 años atrás, datos estadísticos básicos.	<ul style="list-style-type: none"> <li>▪ Crear controlador para separar registros de reportes por cada año hasta el año actual.</li> <li>▪ Crear controlador para visualizar registros filtrados por medio de gráficos.</li> </ul>
		Se debe tener un campo de fecha de inicio y fecha fin para generar datos estadísticos para periodos de tiempo específicos.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener registros de reportes en un periodo dado.</li> <li>▪ Crear controlador para visualizar registros por medio de texto, gráficos y mapa.</li> </ul>
		Por medio de gráficos, texto y un mapa, se debe visualizar información estadística que incluya información por cada parroquia.	<ul style="list-style-type: none"> <li>▪ Crear método para obtener información específica de una parroquia.</li> <li>▪ Crear controlador para visualizar información por medio de gráficos.</li> </ul>

HU27	Gestión información general de aplicación	Debe haber un formulario por medio del cual pueda editar las redes sociales que se visualiza en el pie de página de la aplicación web.	<ul style="list-style-type: none"> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> <li>▪ Crear método para actualizar información de redes sociales.</li> </ul>
		Debe haber un formulario por medio del cual pueda editar la información de contacto que se visualiza en el pie de página de la aplicación web.	<ul style="list-style-type: none"> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> <li>▪ Crear método para actualizar información de contacto.</li> </ul>
		Debe haber un formulario por medio del cual pueda editar los términos y condiciones que se visualiza en el registro de usuario de la aplicación web.	<ul style="list-style-type: none"> <li>▪ Implementar ventana modal por medio de Bootstrap.</li> <li>▪ Crear método para actualizar información de términos y condiciones.</li> </ul>

De igual manera que son similares las Historias de Usuario 23, 24 y 25 con la Historia de Usuario 22, los criterios de aceptación y actividades que se plantean para la implementación de las mismas son iguales. Por tanto, se detallan los criterios de aceptación y actividades solo para la Historia de Usuario 22.

#### Tablero Kanban Final Sprint 4

Todas las actividades planificadas para este Sprint se llevaron a cabo con éxito, por lo tanto, al finalizar el Sprint 4 pasan a estar en fase de prueba.

Tabla 47. Tablero Kanban final Sprint 4

PENDIENTE	EN DESARROLLO	EN PRUEBAS	FINALIZADO		
			HU01	HU09	HU15
		HU19	HU02	HU10	HU16
		HU20	HU03	HU11	HU17
		HU21	HU04	HU12	HU18
		HU22	HU05	HU13	
		HU23	HU06	HU14	
		HU24	HU07		
		HU25	HU08		
		HU26			
		HU27			

- PROTOTIPO



Figura 19. Prototipo de interface - Inicio perfil administrador

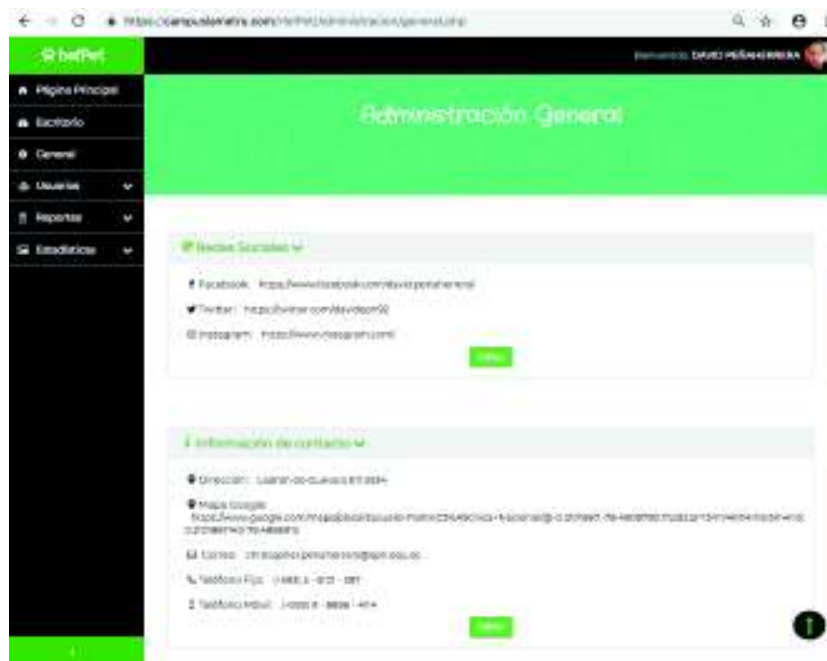


Figura 20. Prototipo de interface - Inicio configuraciones generales



Figura 21. Prototipo de interface - Gestión usuarios / administradores / suscriptores



Figura 22. Prototipo de interface - Gestión reportes adopciones / rescate / encontrados / perdidos



Figura 23. Prototipo de interface - Estadísticas adopciones / rescate / encontrados / perdidos

• SPRINT REVIEW

Tabla 48. Revisión de Criterios de Aceptación del Sprint 4

CÓDIGO	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	CUMPLIDO
HU19	Gestión Usuarios Finales.	de	
		Debe haber una barra de búsqueda que permita buscar a los usuarios por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	SI
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	SI
		Debe haber paginación que permita visualizar y organizar los registros disponibles.	SI
		En una tabla se deben visualizar los registros de los usuarios con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	SI



		Debe haber un botón para editar la información del usuario seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.	SI
		Debe haber un botón para eliminar a un usuario específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.	SI
HU20	Gestión de Administradores.	Debe haber una barra de búsqueda que permita buscar a los usuarios administradores por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	SI
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	SI
		Debe haber paginación que permita visualizar y organizar los registros disponibles.	SI
		En una tabla se deben visualizar los registros de los usuarios administradores con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	SI
		Debe haber un botón para editar la información del usuario administrador seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.	SI
		Debe haber un botón para eliminar a un usuario específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.	SI
HU21	Gestión de Suscriptores.	Debe haber una barra de búsqueda que permita buscar a los usuarios suscriptores por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	SI
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	SI

		Debe haber paginación que permita visualizar y organizar los registros disponibles.	SI
		En una tabla se deben visualizar los registros de los usuarios suscriptores con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	SI
		Debe haber un botón para editar el estado del registro seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal que permita el cambio de estado (Activo / Inactivo).	SI
		Debe haber un botón para eliminar a un usuario suscriptor específico, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del registro.	SI
HU22	Gestión de Reportes de Adopciones.	Debe haber una barra de búsqueda que permita buscar reportes de adopción por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	SI
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	SI
		Debe haber paginación que permita visualizar y organizar los registros disponibles.	SI
		En una tabla se deben visualizar los registros de reportes de adopción con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	SI
		Debe haber un botón para editar la información de reporte seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.	SI
		Debe haber un botón para eliminar a un reporte de adopción, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del reporte.	SI

HU23	Gestión de Reportes Rescate.	Debe haber una barra de búsqueda que permita buscar reportes de rescate por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	SI
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	SI
		Debe haber paginación que permita visualizar y organizar los registros disponibles.	SI
		En una tabla se deben visualizar los registros de reportes de rescate con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	SI
		Debe haber un botón para editar la información de reporte seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.	SI
		Debe haber un botón para eliminar a un reporte de rescate, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del reporte.	SI
HU24	Gestión de Reportes Perdidos.	Debe haber una barra de búsqueda que permita buscar reportes de animales perdidos por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	SI
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	SI
		Debe haber paginación que permita visualizar y organizar los registros disponibles.	SI

		En una tabla se deben visualizar los registros de reportes de animales perdidos con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	SI
		Debe haber un botón para editar la información de reporte seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.	SI
		Debe haber un botón para eliminar a un reporte de animales perdidos, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del reporte.	SI
HU25	Gestión de Reportes Encontrados.	Debe haber una barra de búsqueda que permita buscar reportes de animales encontrados por medio de cualquier parámetro de la información disponible. Al momento de buscar se debe filtrar la información automáticamente.	SI
		Debe haber una etiqueta semántica SELECT que permita seleccionar el número de registros que se van a visualizar por página.	SI
		Debe haber paginación que permita visualizar y organizar los registros disponibles.	SI
		En una tabla se deben visualizar los registros de reportes de animales encontrados con toda la información disponible. Cada columna de la tabla, debe permitir ordenar la información en forma ascendente y descendente.	SI
		Debe haber un botón para editar la información de reporte seleccionado, por medio del mismo al ser accionado se debe visualizar una ventana modal con la información dentro de un formulario.	SI
		Debe haber un botón para eliminar a un reporte de animales encontrados, al ser accionado se debe visualizar una ventana modal para confirmar o cancelar la eliminación del reporte.	SI

HU26	Visualización de Estadísticas Específicas	Se debe visualizar en general datos estadísticos por medio de texto y gráficos para el año actual y 4 años atrás.	SI
		En una tabla se debe tener detallado cada mes del año actual datos estadísticos básicos.	SI
		En una tabla se debe tener detallado en un periodo desde el año actual y 4 años atrás, datos estadísticos básicos.	SI
		Se debe tener un campo de fecha de inicio y fecha fin para generar datos estadísticos para periodos de tiempo específicos.	SI
		Por medio de gráficos, texto y un mapa, se debe visualizar información estadística que incluya información por cada parroquia.	SI
HU27	Gestión información general de aplicación	Debe haber un formulario por medio del cual pueda editar las redes sociales que se visualiza en el pie de página de la aplicación web.	SI
		Debe haber un formulario por medio del cual pueda editar la información de contacto que se visualiza en el pie de página de la aplicación web.	SI
		Debe haber un formulario por medio del cual pueda editar los términos y condiciones que se visualiza en el registro de usuario de la aplicación web.	SI

- **GRÁFICO BURNDOWN CHART SPRINT 4**

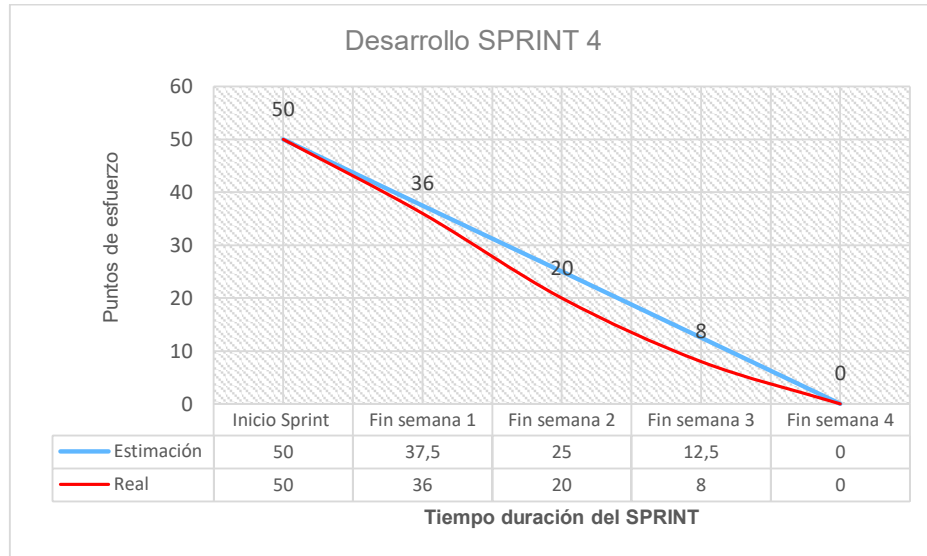


Figura 24. Burndown Chart del Sprint 4

- **SPRINT RETROSPECTIVE**

El desarrollo en el Sprint 4 se llevó a cabo dentro del tiempo establecido logrando cumplir con todos los criterios de aceptación planteados. Como se puede observar en el gráfico Burndown Chart, a pesar de tener varias historias de usuario en este Sprint, se avanzó rápidamente gracias a la reutilización de servicios.

## 6. APLICACIÓN CASOS DE PRUEBA

Concluido el desarrollo de los Sprints en su totalidad, se procede a realizar pruebas funcionales de integración en todos los módulos para validar las funcionalidades que se establecieron en los requerimientos.

### 6.1. Prueba en módulo 1 – Gestión de Usuarios

El módulo 1 corresponde a la Gestión de Usuarios, es decir, a las siguientes funcionalidades:

- Pantalla de inicio.
- Inicio de sesión.
- Registro de Usuario.
- Recuperación de cuentas.

En la siguiente tabla se puede visualizar que los resultados que se esperaban son los obtenidos.

Tabla 49. Casos de prueba Módulo 1 - Pantalla de inicio

<b>CASO DE PRUEBA EN MÓDULO 1</b>				
<b>SUBMÓDULO:</b> Pantalla de inicio				
<b>CÓDIGO</b>	<b>FUNCIONALIDAD EVALUADA</b>	<b>PRERREQUISITOS</b>	<b>RESULTADOS</b>	
			<b>ESPERADO</b>	<b>ALCANZADO</b>
CP01	Acceso página principal de la aplicación web.	Ninguno	La aplicación debe estar desplegada y funcional en un hosting web.	Satisfactorio
CP02	Acceso a páginas por medio del menú.	Ninguno	La aplicación contiene un menú que permite al usuario navegar por toda la aplicación web.	Satisfactorio
CP03	Visualización últimos casos reportados en página de inicio.	Ninguno	Se debe visualizar los 3 últimos casos reportados por cada categoría (Perdido, Adopción, Rescate y Encontrados)	Satisfactorio

Tabla 50. Casos de prueba Módulo 1 - Inicio de sesión

<b>CASO DE PRUEBA EN MÓDULO 1</b>				
<b>SUBMÓDULO:</b> Inicio de sesión				
<b>CÓDIGO</b>	<b>FUNCIONALIDAD EVALUADA</b>	<b>PRERREQUISITOS</b>	<b>RESULTADOS</b>	
			<b>ESPERADO</b>	<b>ALCANZADO</b>
CP04	Inicio de sesión con credenciales correctas	El usuario debe haber creado una cuenta de usuario anteriormente.	<ul style="list-style-type: none"> <li>▪ Validación de campos de formulario.</li> <li>▪ Visualización mensaje de inicio de sesión exitoso.</li> <li>▪ Redireccionamiento a la página de inicio de usuario o</li> </ul>	Satisfactorio

			<p>administrador (dependiendo de la cuenta).</p> <ul style="list-style-type: none"> <li>Visualización de menú correspondiente al usuario o administrador (dependiendo de la cuenta).</li> <li>Sección footer con nombre del usuario que inicio sesión.</li> <li>Creación de sesión activa con información de usuario.</li> </ul>	
CP05	Inicio de sesión de cuenta inactiva.	El usuario debe haber creado una cuenta de usuario anteriormente.	<ul style="list-style-type: none"> <li>Visualización mensaje de cuenta suspendida.</li> <li>No se redirecciona a ninguna página, permanece en Inicio de sesión.</li> </ul>	Satisfactorio
CP06	Inicio de sesión con credenciales incorrectas	Ninguno	<ul style="list-style-type: none"> <li>Visualización mensaje credenciales incorrectas.</li> <li>No se redirecciona a ninguna página, permanece en Inicio de sesión.</li> </ul>	Satisfactorio
CP07	Inicio de sesión con campos incorrectos	Ninguno	<ul style="list-style-type: none"> <li>Validación de campos de formulario.</li> <li>Visualización mensaje del campo de formulario incorrecto.</li> </ul>	Satisfactorio

Tabla 51. Casos de prueba Módulo 1 - Registro de usuario

<b>CASO DE PRUEBA EN MÓDULO 1</b>				
<b>SUBMÓDULO: Registro de usuario</b>				
<b>CÓDIGO</b>	<b>FUNCIONALIDAD EVALUADA</b>	<b>PRERREQUISITOS</b>	<b>RESULTADOS</b>	
			<b>ESPERADO</b>	<b>ALCANZADO</b>
CP08	Registro de usuario con campos completos	El usuario no debe tener una cuenta creada previamente.	<ul style="list-style-type: none"> <li>Validación de campos de formulario.</li> </ul>	Satisfactorio



			<ul style="list-style-type: none"> <li>▪ Visualización mensaje de registro de usuario exitoso.</li> <li>▪ Redireccionamiento a página Inicio de Sesión.</li> <li>▪ Envío de mensaje de bienvenida al correo de usuario registrado.</li> </ul>	
CP09	Registro de usuario con campos incompletos o incorrectos	Ninguno	<ul style="list-style-type: none"> <li>▪ Validación de campos del formulario.</li> <li>▪ Visualización mensaje especificando el campo de formulario en estado incorrecto o faltante.</li> <li>▪ No se redirecciona a otra página, permanece en registro de usuario.</li> </ul>	Satisfactorio
CP10	Registro de usuario con un correo ya registrado.	El usuario debe haber creado una cuenta de usuario previamente.	<ul style="list-style-type: none"> <li>▪ Validación de campos del formulario.</li> <li>▪ Visualización mensaje cuenta ya existente.</li> <li>▪ No se redirecciona a ninguna página, permanece en Inicio de sesión.</li> </ul>	Satisfactorio

Tabla 52. Casos de prueba Módulo 1 - Recuperación de cuentas

<b>CASO DE PRUEBA EN MÓDULO 1</b>				
<b>SUBMÓDULO: Recuperación de cuentas</b>				
<b>CÓDIGO</b>	<b>FUNCIONALIDAD EVALUADA</b>	<b>PRERREQUISITOS</b>	<b>RESULTADOS</b>	
			<b>ESPERADO</b>	<b>ALCANZADO</b>
CP11	Recuperación de cuenta de usuario.	El usuario debe tener una cuenta creada previamente.	<ul style="list-style-type: none"> <li>▪ Validación de campos de formulario.</li> <li>▪ Visualización mensaje de recuperación de cuenta exitoso.</li> <li>▪ Envío de mensaje con nueva</li> </ul>	Satisfactorio

			contraseña al correo de usuario especificado.	
CP12	Recuperación de cuenta de usuario existente de no	Ninguno	<ul style="list-style-type: none"> <li>▪ Validación de campos del formulario.</li> <li>▪ Visualización mensaje especificando que el correo electrónico no tiene una cuenta creada</li> </ul>	Satisfactorio

## 6.2. Prueba en módulo 2 – Reportes de Casos de Animales

El módulo 2 corresponde a los Reportes de Casos de Animales, es decir, a las siguientes funcionalidades:

- Reporte animal en adopción.
- Reporte rescate animal.
- Reporte animal perdido.
- Reporte animal encontrado.

El contenido de las pruebas funcionales para el módulo 2 se encuentran detalladas en el Anexo E Casos de prueba.

## 6.3. Prueba en módulo 3 – Perfil de Usuario / Galerías

El módulo 3 corresponde al Perfil de Usuario, es decir, a las siguientes funcionalidades:

- Perfil de Usuario
- Historial de Reportes.
- Casos Cercanos.
- Galerías animales (adopción, encontrados y perdidos).

El contenido de las pruebas funcionales para el módulo 3 se encuentran detalladas en el Anexo E Casos de prueba.

## 6.4. Prueba en módulo 4 – Estadísticas Casos Reportados

El módulo 4 corresponde a Estadísticas Casos Reportados, es decir, a las siguientes funcionalidades:

- Estadísticas animales en adopción.
- Estadísticas animales en rescate.
- Estadísticas animales encontrados.
- Estadísticas animales perdidos.

El contenido de las pruebas funcionales para el módulo 3 se encuentran detalladas en el Anexo E Casos de prueba.

## 6.5. Prueba en módulo 5 – Perfil Administrativo

El módulo 5 corresponde al Perfil Administrativo, es decir, a las siguientes funcionalidades:

- Gestión Usuarios Administradores, Usuarios Finales y Suscriptores.
- Gestión Reportes de Animales en Adopción, Rescate, Encontrados y Perdidos.
- Visualización Estadísticas Animales en Adopción, Rescate, Encontrados y Perdidos.
- Gestión Información General de la Aplicación.

El contenido de las pruebas funcionales para el módulo 3 se encuentran detalladas en el Anexo E Casos de prueba.

Al finalizar la ejecución de pruebas por cada módulo de la aplicación web, se ha concluido la aplicación con éxito y con resultados deseados, En el siguiente tablero Kanban se puede visualizar las historias de usuario propuestas totalmente finalizadas.

Tabla 53 Tablero Kanban final

PENDIENTE	EN DESARROLLO	EN PRUEBAS	FINALIZADO			
			HU01	HU09	HU15	HU19
			HU02	HU10	HU16	HU20
			HU03	HU11	HU17	HU21
			HU04	HU12	HU18	HU22
			HU05	HU13		HU23
			HU06	HU14		HU24
			HU07			HU25
			HU08			HU26
						HU27

## 7. EVALUACIÓN DE USO DE LA APLICACIÓN

Además de las pruebas de aceptación e integración que se aplicaron para la aplicación web, se realizó una encuesta adicional para poder obtener información en relación a la satisfacción y usabilidad que tiene el usuario final al momento de utiliza la aplicación.

### 7.1. Estructura de la encuesta

La encuesta constará de dos partes, la primera parte se enfoca en la eficiencia y desempeño de la aplicación web, y la segunda parte se enfoca en la experiencia y satisfacción que presenta el usuario al momento de hacer uso de la aplicación.

A continuación, se presenta la encuesta que se aplicó. La encuesta fue realizada por medio de Google Forms para facilidad de recolección y análisis de datos.

Para la calificación se utilizará los siguientes parámetros: 1: Malo, 2: Regular, 3: Bueno y 4: Muy bueno. Se adjunta en el Anexo F el formulario que se utilizó para la encuesta de usabilidad y satisfacción del usuario.

Tabla 54. Encuesta eficiencia y desempeño de la aplicación

<b>Encuesta eficiencia y desempeño de la aplicación</b>					
<b>N°</b>	<b>Preguntas</b>	<b>Puntuación</b>			
		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
1	Velocidad con la que ejecuta la aplicación las peticiones.				
2	Los servicios que ofrece la aplicación se ejecutan correctamente.				
3	El mecanismo de ubicación / localización del usuario en la aplicación funciona correctamente.				
4	Los formularios de la aplicación se envían de forma satisfactoria.				
5	El control de acceso a funcionalidades de la aplicación funciona correctamente.				
6	Rendimiento de la aplicación al ser utilizada en dispositivos móviles.				
7	Los mensajes de retroalimentación (éxito, error, advertencia) al usuario son claros y concisos.				

Tabla 55. Encuesta experiencia y satisfacción del usuario

<b>Encuesta experiencia y satisfacción del usuario</b>					
<b>N°</b>	<b>Preguntas</b>	<b>Puntuación</b>			
		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
8	Las interfaces, menú, imágenes, fuente son adecuadas.				
9	La navegación / uso de la aplicación es amigable e intuitiva.				
10	Utilizaría la aplicación con frecuencia.				
11	La aplicación es fácil de entender y utilizar.				
12	Las funciones / servicios que brinda la aplicación están bien integradas / distribuidas.				

13	La información solicitada se visualiza de forma adecuada.				
14	La búsqueda y filtros funcionan correctamente.				
15	Perjudica su privacidad la aplicación al solicitar su ubicación actual para brindar ciertos servicios.				
16	Recomendaría la aplicación web a otros usuarios.				

## 7.2. Análisis de los resultados de la encuesta

A continuación, se presentan un breve análisis de las encuestas aplicadas. Los resultados serán analizados por pregunta y de forma general como se detalla a continuación:

### 7.2.1. Resultados Parciales

Como se puede observar en la Figura 25, todas las preguntas reflejan un nivel de experiencia y desempeño de la aplicación alto, es decir, las funcionalidades que ofrece la aplicación se están ejecutando de una manera eficaz.

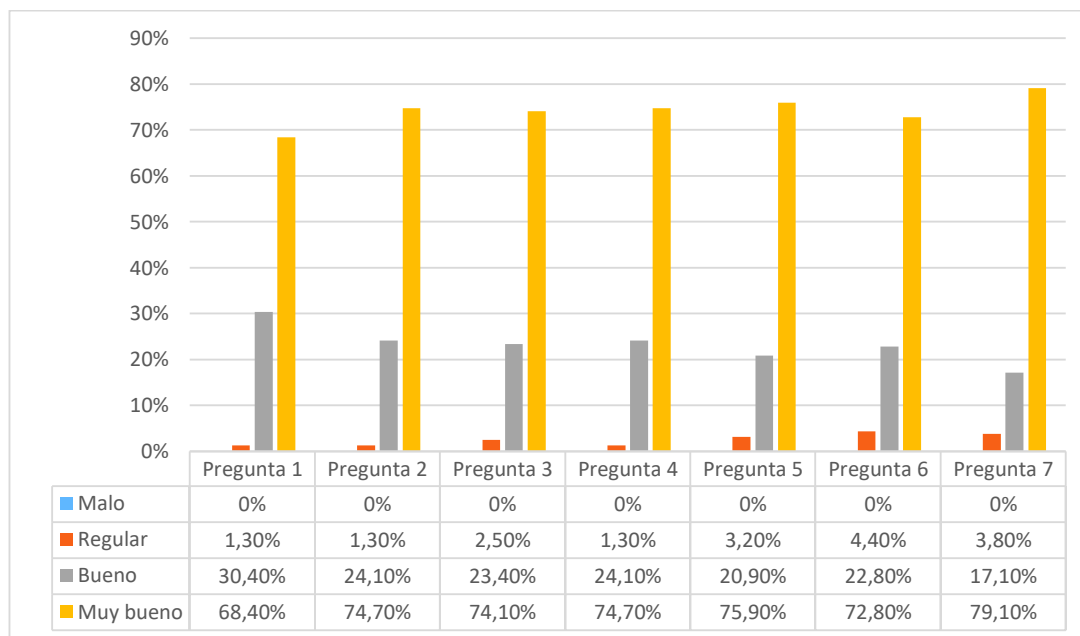


Figura 25. Resultados preguntas eficiencia y desempeño de la aplicación

La pregunta 7, Los mensajes de retroalimentación (éxito, error, advertencia) al usuario son claros y concisos, es la que mejor puntuación ha obtenido (79,10%). Esto debido a que en cada acción que realiza el usuario con la aplicación se puede visualizar un mensaje de retroalimentación. Por otra parte, la pregunta 1, Velocidad con la que ejecuta la aplicación las peticiones, es la que menor puntuación ha obtenido (68,40%).

Nota: la velocidad con la que se ejecuta la aplicación web, depende de ciertos factores, tales como la conexión a internet y la calidad de servicio que ofrece el servidor web hosting.

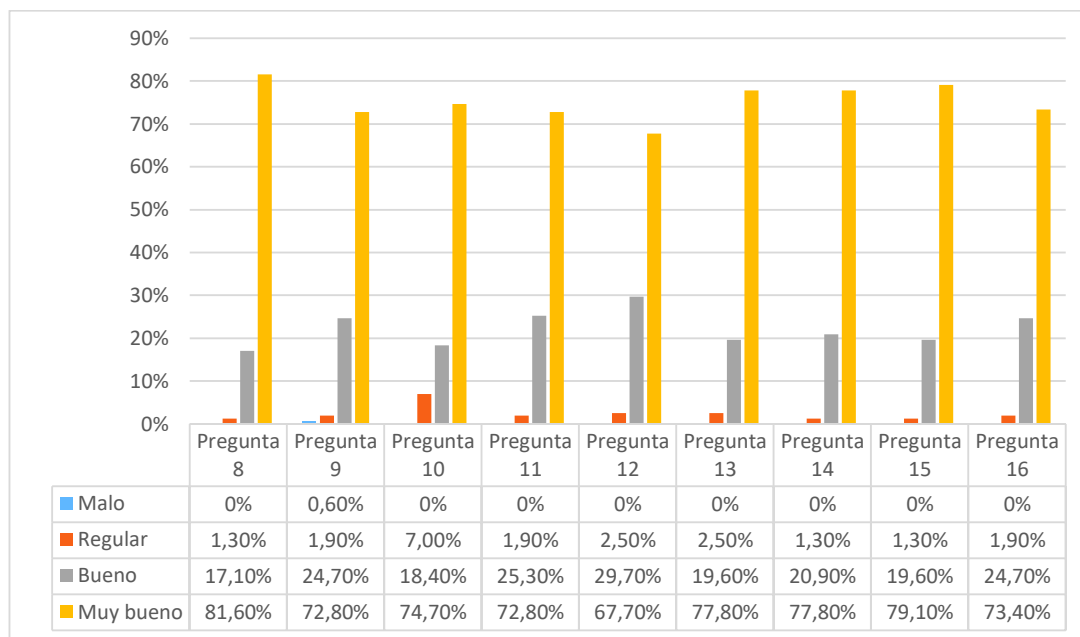


Figura 26. Resultados preguntas experiencia y satisfacción del usuario

En la Figura 26, se puede observar las respuestas a las preguntas enfocadas en la Experiencia y Satisfacción del usuario al momento de interactuar con la aplicación web.

La pregunta 1, Las interfaces, menú, imágenes, fuente son adecuadas, es la que mejor puntuación ha obtenido (81,60%). Esto debido a que el diseño de la aplicación web es adaptable a cualquier dispositivo manteniendo su funcionalidad. Sin embargo, la pregunta 12, Las funciones / servicios que brinda la aplicación están bien integradas / distribuidas, es la que menor puntuación ha obtenido (67,70%).

### 7.2.2. Resultado General

En general, la aplicación web presenta resultados positivos al momento de ser utilizada por usuarios finales, los resultados generales se presentan a continuación:

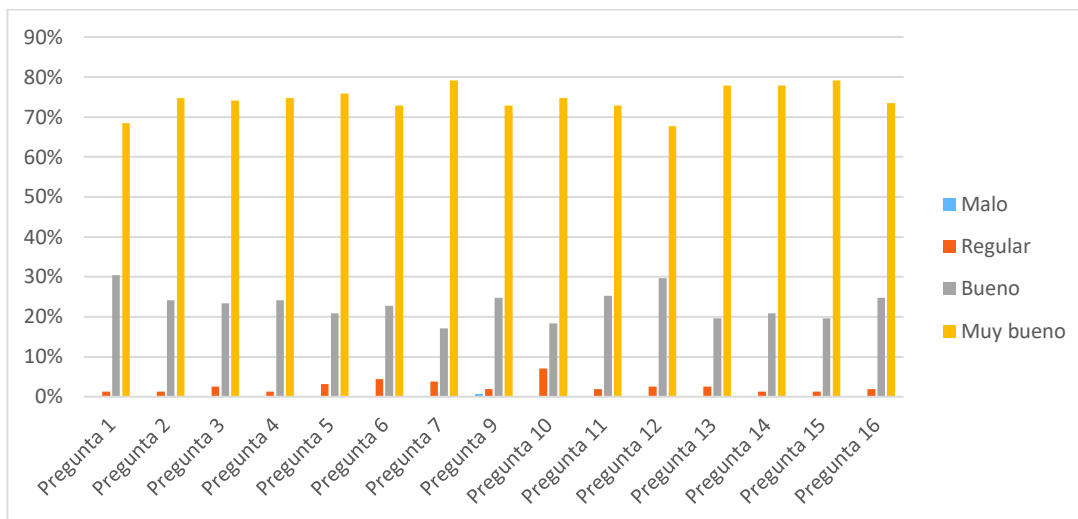


Figura 27. Resultados generales de la encuesta

En la Figura 27, se puede visualizar que, en todas las preguntas de la encuesta, la opción “Muy bueno” es la calificación que más porcentaje tiene. Con la encuesta se concluye que la aplicación web al momento de ser utilizada brinda una buena experiencia al usuario y sus funcionalidades se ejecutan correctamente.

## 8. EVALUACIÓN DE RENDIMIENTO DE LA APLICACIÓN

Para determinar el rendimiento de la aplicación web, se realizaron pruebas por medio de la herramienta JMeter en la versión 5.1.r1853635. Las pruebas que se aplicaron se hicieron sobre el servidor web Hostgator en el cual está la aplicación web.

Se aplicaron 4 pruebas con un Grupo de Hilos o usuarios de 100, 75, 50 y 25 respectivamente. Por medio de peticiones HTTP Request se obtuvieron resultados a través de los siguientes receptores:

- Tabla de Resultados.
- Gráfico de Resultados.
- Árbol de Resultados.
- Reporte de Resumen.
- 

Todas las gráficas correspondientes a los resultados se encuentran en el Anexo G. El método de petición fue GET aplicado a la dirección web de la aplicación, es decir, [campuslametro.com/HeIPet/index.php](http://campuslametro.com/HeIPet/index.php). En general los resultados obtenidos se muestran en la siguiente tabla:

Tabla 56. Resultados pruebas de rendimiento

<b>N° Muestras</b>	<b>Media (ms)</b>	<b>Min (ms)</b>	<b>Max (ms)</b>	<b>Error (%)</b>	<b>Rendimiento</b>	<b>Media Bytes</b>
25	650	580	724	0,00	14,8/sec	35751,0
50	747	241	1155	22,00	25,1/sec	28039,6
75	573	239	1296	52,00	40,1/sec	17336,0
100	473	239	1126	64,00	58,9/sec	13359,1
1000	17388	348	20249	98,00	49,3/sec	2917,0

Como se puede visualizar en la Tabla 56, el servidor web responde de manera exitosa con pocos usuarios concurrentes realizando peticiones, con los 25 usuarios tiene un error en las peticiones del 0.00% pero al duplicar la cantidad de usuarios, el porcentaje de error crece considerablemente. Al aplicar la prueba con 1000 usuarios concurrentes el error casi es total con un 98,00%

Tabla 57. Códigos de respuesta y latencias

<b>N° Muestras</b>	<b>Código de respuesta</b>	<b>Latencia Max (ms)</b>	<b>Latencia Min (ms)</b>
25	200 (OK)	872	451
50	429 (Too Many Request)	1130	241
75	503 (Service Temporarily Unavailable)	1127	239
100	503 (Service Temporarily Unavailable)	1160	239
1000	Non HTTP response code	1026	0

La latencia que se muestra en la Tabla 57, es variable debido a que al incrementar el número de usuarios o hilos concurrentes se tiene un gran porcentaje de error y la latencia llega a niveles de hasta 0 ms.



## 9. CONCLUSIONES

En Quito, actualmente existen varias instituciones y fundaciones que brindan ayuda a los animales callejeros domésticos, pero no logran atender todas las solicitudes de ayuda que reciben debido a la falta de recursos económicos y tiempo. Por otra parte, dichas organizaciones no logran saber que sucede en las calles en tiempo real con los animales y no cuentan con procesos establecidos enfocados al rescate animal, por medio de estudios realizados por otras entidades, llegan a determinar números aproximados de animales callejeros por cada sector de Quito.

Mediante el desarrollo del presente trabajo, se ha logrado implementar una aplicación web, la cual representa una fuente de información para que los usuarios y organizaciones en base a la información registrada, puedan brindar su ayuda en el rescate animal y procesos de adopción y pérdida de animales.

Gracias a la técnica del '*Geotargeting*' se puede personalizar el contenido que el usuario obtiene por medio de la aplicación web presentada, y por ende puede brindar su ayuda a los casos que se encuentren cercanos a su ubicación.

Por medio de una presentación sencilla y amigable de formularios web, se puede obtener información de lo que sucede en las calles con los animales, y por medio de un análisis previo de dicha información se puede determinar zonas concurrentes de maltrato, abandono, animales perdidos entre otras situaciones.

En cuanto a la metodología ágil aplicada en el presente proyecto, se ha logrado concluir de forma exitosa con el desarrollo de la aplicación web gracias a la estimación correcta de tiempos para la entrega de partes parciales utilizables y probadas. Con respecto a la utilización de patrones de diseño, cabe recalcar que se ha logrado optimizar el desarrollo web basándose en la reutilización de servicios y estructurando de manera efectiva la aplicación por medio de la separación de los datos y la lógica del negocio.

Finalmente, con la información sobre animales en casos de adopción, rescate, pérdida y encontrados que brinda la aplicación web, se puede llegar a determinar sectores con mayor número de animales callejeros en cada caso mencionado anteriormente.

## 10. REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Gabriela, «Población de perros callejeros en Quito sigue en aumento,» El Comercio, 09 octubre 2018. [En línea]. Available: <https://www.elcomercio.com/narices-frias/poblacion-perros-callejeros-quito-censo.html>. [Último acceso: 25 noviembre 2018].
- [2] G. Castellanos, «AMC ya no retirará animales del espacio público,» El Comercio, 03 agosto 2018. [En línea]. Available: <https://www.elcomercio.com/narices-frias/amc-noretiro-animales-quito-mascotas.html>. [Último acceso: 25 noviembre 2018].
- [3] D. Valentín, «Por cada 22 habitantes hay un perro abandonado en Quito, según un censo,» El Comercio, 05 octubre 2018. [En línea]. Available: <https://www.elcomercio.com/afull/perros-abandonados-quito-censo-parroquias.html>. [Último acceso: 25 noviembre 2018].
- [4] Infobae, «En Quito, hay un perro abandonado por cada 22 personas,» Infobae, 04 octubre 2018. [En línea]. Available: <https://www.infobae.com/america/america-latina/2018/10/04/en-quito-hay-un-perro-abandonado-por-cada-22-personas/>. [Último acceso: 25 noviembre 2018].
- [5] C. Gabriela, «Un mapeo virtual de animales callejeros se realiza en Quito,» El Comercio, 09 mayo 2018. [En línea]. Available: <https://www.elcomercio.com/narices-frias/mapeo-virtual-animales-callejeros-quito.html>. [Último acceso: 25 noviembre 2018].
- [6] C. Gabriela, «Los lugares donde se abandonan más perros en Quito,» El Comercio, 09 enero 2016. [En línea]. Available: <https://www.elcomercio.com/tendencias/perros-abandonos-quito-mascotas-animales.html>. [Último acceso: 25 noviembre 2018].
- [7] F. Affinity, «Las cifras del abandono de perros y gatos aún lejos de descender,» Fundación Affinity, [En línea]. Available: <https://www.fundacion-affinity.org/perros-gatos-y-personas/busco-una-mascota/las-cifras-del-abandono-de-perros-y-gatos-aun-lejos-de>. [Último acceso: 25 noviembre 2018].
- [8] C. Gabriela, «4 tipos de abandono de mascotas en Quito,» El Comercio, 11 enero 2016. [En línea]. Available: <https://www.elcomercio.com/tendencias/tipos-abandono-mascotas-quito.html>. [Último acceso: 25 noviembre 2018].
- [9] AnimalesDe, «Animales Domésticos: Lista, Características, Tipos y Ejemplos,» AnimalesDe, 07 diciembre 2016. [En línea]. Available: <https://animalesde.net/domesticos/#Animales-salvajes>. [Último acceso: 25 noviembre 2018].
- [10] O. Workshop, «The Health Benefits of Pets,» National Institutes of Health, 10 septiembre 2011. [En línea]. Available: <https://consensus.nih.gov/1987/1987HealthBenefitsPetsta003html.htm>. [Último acceso: 25 noviembre 2018].
- [11] American Veterinary Medical Association, «The Health Benefits of Pets,» American Veterinary Medical Association, 10 septiembre 2011. [En línea]. Available: <https://web.archive.org/web/20090819045212/http://www.avma.org/reference/marketstats/ownership.asp>. [Último acceso: 25 noviembre 2018].
- [12] L. Gersón, «La Geolocalización social,» Departamento de Geografía. Universitat de València, 14 julio 2015. [En línea]. Available: <http://revpubli.unileon.es/ojs/index.php/poligonos/article/view/3290/2503?fbclid=IwAR2jrTT->

T6S9A7pK\_UkAQq7S8nbaxpsifPX20YfsDY5cLMnEYyUmxhDH1is. [Último acceso: 25 noviembre 2018].

- [13] ISACA, «Qué es la geolocalización y cómo funciona,» Evaluando Software, 31 julio 2017. [En línea]. Available: <http://www.evaluandosoftware.com/la-geolocalizacion-funciona/>. [Último acceso: 25 noviembre 2018].
- [14] Merca2.0, «¿Qué es geotargeting y cuál es su porcentaje de inversión?,» Merca 2.0, 02 noviembre 2015. [En línea]. Available: <https://www.merca20.com/que-es-geotargeting-y-cual-es-su-porcentaje-de-inversion/>. [Último acceso: 25 noviembre 2018].
- [15] D. Guide, «Geotargeting: así funciona la optimización regional de los contenidos,» Digital Guide, 30 agosto 2017. [En línea]. Available: <https://www.ionos.es/digitalguide/online-marketing/analisis-web/que-es-el-geotargeting/>. [Último acceso: 25 noviembre 2018].
- [16] G. M. Platform, «Maps JavaScript API,» Google Developers, 13 noviembre 2018. [En línea]. Available: <https://developers.google.com/maps/documentation/javascript/tutorial>. [Último acceso: 25 noviembre 2018].
- [17] MicroSiervo, «Google Maps,» MicroSiervo, 27 octubre 2015. [En línea]. Available: <https://www.microsiervos.com/archivo/tecnologia/coordenadas-en-google-maps.html>. [Último acceso: 25 noviembre 2018].
- [18] C. Á. Caules, «Introducción a Servicios REST,» Arquitectura Java, 14 junio 2013. [En línea]. Available: <https://www.arquitecturajava.com/servicios-rest/>. [Último acceso: 25 noviembre 2018].
- [19] G. J. Carlos, «Principios REST,» Youtube, 11 abril 2016. [En línea]. Available: <https://www.youtube.com/watch?v=YYhttBcZWu8>. [Último acceso: 25 noviembre 2018].
- [20] Bibing, «SOAP Y WSDL,» Bibing, 10 octubre 2016. [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/11247/fichero/Memoria%252F7-SOAP+y+WSDL.pdf>. [Último acceso: 25 noviembre 2018].
- [21] W3C Architecture Domain, «Simple Object Access Protocol,» W3C Architecture Domain, 10 julio 2009. [En línea]. Available: <https://www.w3.org/2000/xp/Group/>. [Último acceso: 25 noviembre 2018].
- [22] I. Matehuala, «SOAP (Simple Object Access Protocol),» Instituto Tecnológico de Matehuala, 28 noviembre 2014. [En línea]. Available: <https://programacionwebisc.wordpress.com/5-2-1-soap-simple-object-access-protocol/>. [Último acceso: 25 noviembre 2018].
- [23] Culturacion, «¿Qué es AJAX?,» Culturacion, 12 junio 2017. [En línea]. Available: <http://culturacion.com/que-es-ajax/>. [Último acceso: 25 noviembre 2018].
- [24] C. Paloma, «Procesos Ágiles para el Desarrollo de Aplicaciones Web,» Universidad Rey Juan Carlos, 12 agosto 2015. [En línea]. Available: <http://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf>. [Último acceso: 25 noviembre 2018].
- [25] C. José, «Métodologías Ágiles en el Desarrollo de Software,» Universidad Politécnica de Valencia, 12 junio 2015. [En línea]. Available: <http://roa.ult.edu.cu/bitstream/123456789/476/1/TodoAgil.pdf>. [Último acceso: 25 noviembre 2018].

- [26] T. Oscar, «Criterios de selección de metodologías de desarrollo de software,» Revista de la Facultad de Ingeniería Industrial, 14 diciembre 2010. [En línea]. Available: <http://www.redalyc.org/html/816/81619984009/> . [Último acceso: 25 noviembre 2018].
- [27] S. Ken, «La Guía Definitiva de Scrum: Las Reglas del Juego,» Scrum Guides Org, 01 noviembre 2017. [En línea]. Available: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf#zoom=100>. [Último acceso: 25 noviembre 2018].
- [28] Scrum Manager, «TRONCAL I: Scrum Master,» Scrum Manager, 01 agosto 2016. [En línea]. Available: <https://www.scrummanager.net/bok/index.php?title=Epic>. [Último acceso: 25 noviembre 2018].
- [29] L. Elviar, «Patrones de diseño en Java: MVC, DAO y DTO,» Programación Full Stack, 20 noviembre 2016. [En línea]. Available: <https://www.ecodeup.com/patrones-de-diseno-en-java-mvc-dao-y-dto/>. [Último acceso: 25 noviembre 2018].
- [30] Platzi, «¿Cómo funciona la metodología Scrum?,» Platzi, 10 enero 2015. [En línea]. Available: <https://platzi.com/blog/metodologia-scrum-fases/> . [Último acceso: 25 noviembre 2018].

## **11. ANEXOS**

### **Anexo A: Pila del producto**

Documento adjunto:

- Pila del producto – Lista de Historias de Usuario (formato xlsx).

### **Anexo B: Base de Datos (Modelo Físico)**

Documentos adjuntos:

- Base de datos - Modelo Físico (formato png).
- Script para implementación de estructura Base de Datos (formato sql).

### **Anexo C: Prototipos de interfaces (Perfil Usuario - Administrador)**

Documentos adjuntos:

- Carpeta imágenes prototipos interfaces usuario final (formato png).
- Carpeta imágenes prototipos interfaces usuario administrador (formato png).

### **Anexo D: Tablas y gráficos Burndown Charts**

Documento adjunto:

- Tablas y gráficos Burndown Sprint 1 - 4 (formato xlsx).

## **Anexo E: Casos de prueba**

Documento adjunto:

- Tablas Casos de pruebas Sprint 1 - 4 (formato xlsx).

## **Anexo F: Encuesta de Usabilidad y Satisfacción del Usuario**

Documentos adjuntos:

- Encuesta de Usabilidad y Satisfacción del Usuario (formato HTML).
- Enlace Web Google Forms (formato txt).

## **Anexo G: Pruebas de Rendimiento con JMeter**

Documentos adjuntos:

- Resultados Pruebas de Rendimiento (formato png).