

Desarrollo de una solución tecnológica basada en dispositivos móviles para gestionar procesos de bienestar en animales domésticos con condiciones de vulnerabilidad (PetsClub)

Jhon Freddy Gómez Franco & Feyber Mauricio García Duque

Marzo 2019.

Institución Universitaria Antonio José Camacho.

Facultad de Ingeniería en sistemas.

Proyecto de grado.

Tabla de contenido

1. Resumen	8
2. Introducción.....	9
3. Generalidades de la investigación	10
3.1 Planteamiento del problema.....	10
3.2 Formulación del problema	11
3.3 Preguntas de sistematización.....	11
4. Justificación.....	12
5. Objetivos.....	13
5.1 Objetivo general	13
5.2 Objetivos específicos.....	13
6. Metodología.....	13
6.1 Fases de la Metodología Scrum para móviles.....	16
6.2 Marco teórico	18
6.3 Comunidades animalistas.....	18
6.3.1 Características de las comunidades animalistas.....	19
6.4 Tecnologías habilitadoras para el desarrollo del proyecto	21
6.4.1 Desarrollo de aplicaciones móviles híbridas.....	21
6.4.2 Framework Ionic para desarrollo híbrido	22
6.4.3 Servicios de Geolocalización en aplicaciones móviles híbridas.....	24
6.4.4 Notificaciones push en aplicaciones móviles	25
6.4.5 Base de datos en tiempo real.....	27
6.4.6 Servicios Web basados en REST.....	29
6.4.7 Diseño de arquitectura de software basado el modelo 4+1 Vistas de Kruchten.....	31
7. Antecedentes.....	31
8. Caracterización y definición de requerimientos de la App para el mercado de las mascotas y protección animal.....	33
8.1 Definición de los requerimientos de software.....	38
8.1.1 Requerimientos de Software.....	39
8.1.2 Requerimientos funcionales.....	39

8.1.3	Roles de usuarios	40
8.1.4	Requerimientos no funcionales.....	40
9.	Diseño de la solución tecnológica orientada a servicios que integre los requerimientos del mercado de las mascotas y protección animal	41
9.1	Vista de escenarios de la arquitectura de software.....	41
9.2	Arquitectura propuesta para el despliegue de servicios animalistas	45
9.2.1	Criterios de diseño de la arquitectura para el despliegue de servicios animalistas en las aplicaciones móviles híbrida.....	46
9.2.2	Vista de despliegue de la arquitectura	47
9.2.3	Vista lógica de la arquitectura de software propuesta	50
9.2.4	Vista de procesos de la arquitectura.....	54
10.	Implementación de la solución tecnológica con relación a la usabilidad de sus interfaces y tiempos de respuesta.	55
10.1	Pruebas de calidad.....	68
11.	Conclusiones.....	73
12.	Referencias	74
13.	Anexos.....	74

Lista de tablas

Tabla 1 Matriz comparativa de metodologías.....	15
---	----

Lista de figuras

Figura 1. Planteamiento del problema	11
Figura 2. Diagrama Scrum.....	17
Figura 3. Arquitectura de componentes Ionic.....	24
Figura 4. Geolocalización	25
Figura 5. Notificaciones Push.....	27
Figura 6. Vistas de la arquitectura de Kruchten.....	31
Figura 7. Encuesta falta de canales de comunicación.....	34
Figura 8. Encuesta Medio de comunicación	35
Figura 9. Encuesta necesidad de una App	36
Figura 10. Encuesta Perfil de mascota en una App	36
Figura 11. Encuesta espacio de adopciones en la App	37
Figura 12. Encuesta canalización de entidades.....	38
Figura 13. Casos de uso: Usuario Animalista.....	42
Figura 14. Gestionar usuario.....	43
Figura 15. Registrar usuario.....	44
Figura 16. Mensajes.....	45
Figura 17. Diagrama de nodos y componentes.....	48
Figura 18. Diagrama de secuencia eventos.....	51
Figura 19. Diagrama de secuencia adopción	52
Figura 20. Diagrama de secuencia servicios sociales	53
Figura 21. Diagrama de Comunicación	54
Figura 22. Vista del proceso de la arquitectura.....	55

Figura 23. Diagrama inicio de sesión	56
Figura 24. Inicio sesión HTML	56
Figura 25. Configuración desde Firebase	57
Figura 26. Visualización usuarios autenticados.....	58
Figura 27. Interfaz inicio sesión.....	58
Figura 28. Diagrama publicación eventos	59
Figura 29. Etiquetas HTML publicación eventos	60
Figura 30. Método geolocalización.....	60
Figura 31. Estructura BD Firebase eventos	61
Figura 32. Interfaz de eventos.....	62
Figura 33. Diagrama servicios sociales	62
Figura 34. Imprimir servicios	63
Figura 35. Método para agregar datos	64
Figura 36. Estructura BD Firebase servicios sociales.....	64
Figura 37. Interfaz servicios sociales.....	65
Figura 38. Diagrama adopciones	66
Figura 39. Modulo opciones HTM	66
Figura 40. Crear servicio.....	67
Figura 41. Interfaz de adopciones.....	68
Figura 42. Archivo de sonar-project.properties	68
Figura 43. Análisis de prueba	69
Figura 44. Resultado primer análisis	69
Figura 45. Resultado primer análisis descripción	70

Figura 46. Errores para corregir.....	70
Figura 47. Resultado segundo análisis descripción	71
Figura 48. Resultado prueba final.....	72
Figura 49. Comportamiento de las correcciones.....	72
Figura 50. Datos inicio sesión animalista	75
Figura 51. Escenario inicio sesión animalista.....	76
Figura 52. Datos gestión de eventos	77
Figura 53. Escenario gestionar evento.....	78
Figura 54. Datos gestionar servicios.....	79
Figura 55. Escenario servicio social	80
Figura 56. Casos de uso	81
Figura 57. Compromisos y cantidad de Sprints	82
Figura 58. Sprints.....	82

1. Resumen

Se estima que el 64 por ciento de los colombianos tiene mascotas porque creen que son una buena compañía. La presencia de un perro, gato u otro animal doméstico enriquece la vida de una persona o familia, pues mejora su estado anímico y contribuyen en su salud emocional. En Colombia hay 9 millones de animales de compañía y se estima que uno de cada tres hogares del país tiene una mascota. No obstante, cerca de 2 millones de perros y gatos deambulan por las calles de cuatro de las principales ciudades del país (Bogotá, Cartagena, Medellín y Cali), sufriendo hambre e inclemencias del clima, según datos del Departamento Nacional de Planeación (DNP). Teniendo en cuenta lo anterior, el DNP adelanta la elaboración de una política de protección de animales domésticos, que se convertirá en un documento Conpes en el segundo semestre de este año el cual busca dar lineamientos en materia de conceptos y competencias institucionales, relacionados con la tenencia, reproducción, adopción, distribución y comercialización, entre otros temas (Tiempo, 2017).

El presente proyecto tiene como finalidad desarrollar una solución tecnológica móvil (App) dirigida a las comunidades que preservan el cuidado y la protección animal de mascotas, con el fin de gestionar procesos de bienestar en animales domésticos con condiciones de vulnerabilidad, campañas sociales y eventos para la comunidad animalista de Cali.

Inicialmente se hicieron encuestas en eventos de tipo animaltón (encuentros culturales de las comunidades animalistas), en la cual se evidenció la necesidad de contar con un desarrollo tecnológico para mejorar las tareas rutinarias que realizan los animalistas, enfocadas en tener canales de comunicación efectivos que les permita tener un acercamiento a las personas que desean adoptar una mascota o gestionar y publicar eventos relacionados con actividades de este tipo de comunidades.

Con la caracterización de las necesidades de la comunidad de animalistas, se establecieron los requerimientos tecnológicos de la app (Funcionales y No Funcionales) y se diseñó la solución utilizando el modelo 4+1 de Kruchten, con el fin de establecer las vistas estáticas y dinámicas de la arquitectura de software de la solución a desarrollar. Finalmente se implementaron los módulos de la arquitectura utilizando herramientas de desarrollo móvil híbrido y bases de datos en tiempo real, para culminar con algunas pruebas de rendimiento y usabilidad.

2. Introducción

Actualmente las políticas de protección animal han tenido un impacto favorable en la sociedad colombiana, considerando que existen leyes (Ley 1774 de 2016) que promueve el cuidado y amparo de animales domésticos, que se cuentan con comunidades legalmente constituidas como fundaciones sin ánimo de lucro que establecen estrategias misionales enfocadas al cuidado, la adopción y todas aquellas prácticas que beneficien la supervisión y la calidad de animales de uso doméstico.

Las tecnologías de la información y la comunicación han contribuido en la divulgación de las actividades y buenas prácticas que ofrecen las entidades sin ánimo de lucro a las comunidades animalistas. Por medio de las redes sociales, sitios web y herramientas de trabajo colaborativo, se comparten servicios culturales, sociales y de adopción, con el fin de dinamizar el envío de información, que permita establecer tiempos de respuestas adecuadas a las necesidades de este tipo de organizaciones. Sin embargo el uso de estas herramientas no es suficiente para desplegar efectivamente la información que interesa a este tipo de comunidades, debido que no hay un tratamiento responsable de los contenidos que se publican por estos medios, generando noticias falsas y por ende desconfianza en los servicios sociales que promueven los animalistas.

Teniendo en cuenta lo anterior, en este proyecto se pretende desarrollar una solución tecnológica basada en dispositivos móviles que permita la gestión de los procesos para el bienestar de animales domésticos. Con el auge las políticas nacionales de los derechos de protección animal, se espera contribuir por medio de la tecnología en contar con una comunidad virtual centralizada que integre las necesidades de fundaciones, empresas y animalistas ofreciéndoles una app confiable con información clara y objetiva de veterinarias, tiendas, spa, fundaciones, animalistas y demás empresas relacionadas.

Para el presente proyecto se definió como técnica de recolección de información la encuesta, la cual fue aplicada en fundaciones, veterinarias, guarderías de mascotas, personas interesadas en los procesos de bienestar animal y en eventos animalistas. Con la información obtenida en las encuestas, se realizó una propuesta de requerimientos de software, que fue validada en las tres principales fundaciones de protección animal de la ciudad. Finalmente se diseñó (Utilizando el Modelo 4+1 de Kruchten) y desarrolló una solución móvil híbrida que permite desplegar en dispositivos móviles con sistemas operativos IOS, Android y Windows Phone

servicios relacionados con los procesos de gestión de novedades, eventos y servicios sociales para la comunidad animalista.

3. Generalidades de la investigación

3.1 Planteamiento del problema

Actualmente los derechos de los animales han tomado fuerza en nuestra sociedad, por tal motivo se han ido creando fundaciones que velan por el bienestar de diversos animales. En el año 2014 en Cali se encuentran alrededor de 10.700 perros y 1.500 gatos callejeros (Ospina, 2014), siendo este el motivo principal de la recolección de insumos y presupuesto para su óptimo funcionamiento, lamentablemente el apoyo no es suficiente para sostenerse, las causas de falta de recursos se deben a dos factores principales:

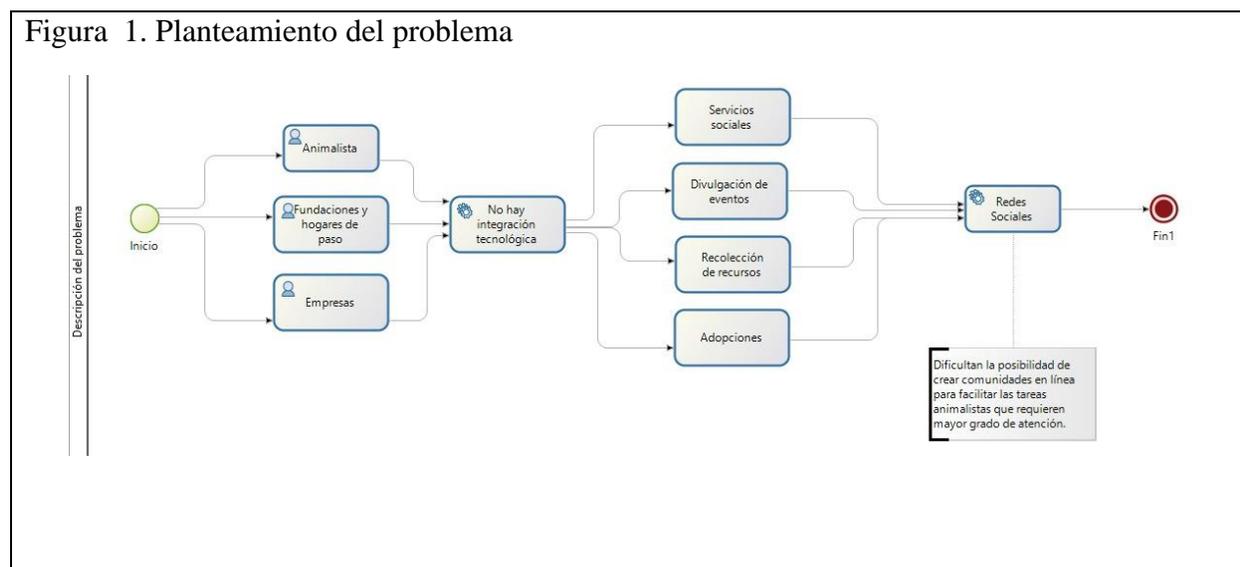
- No son reconocidas por las entidades o empresas dispuestas a realizar donaciones en las diferentes ciudades.
- Las personas del común se muestran escépticas a realizar donaciones por las múltiples estafas que se han presentado en las redes sociales. La desintegración de la comunidad animalista conlleva a crear vacíos utilizados por delincuentes para su lucro.

De acuerdo con datos de la Cámara de Comercio, en Cali hay registradas 272 clínicas veterinarias, en esta cifra se encuentra incluidas entidades como guarderías y spa, ya que solo se cuenta con un código único para todo el comercio que se enmarca con los animales, algunos de estos establecimientos han creado alianzas y convenios con fundaciones y animalistas con el fin de ayudar aquellos animales que se encuentran desprotegidos y por lo general sufren maltrato o se encuentran en las calles en condiciones precarias. Estos esfuerzos se han visto afectados puesto que se encuentran fundaciones actualmente sin registrar y esta informalidad contribuye a diferentes estafas debido a que no hay control de las autoridades en este tema.

Actualmente los animalistas se enfrentan a dificultades como falta de medios para la divulgación de eventos, recolección de alimentos, utensilios y otras necesidades básicas de los hogares de paso y fundaciones, por otra parte, se ven afectados a la hora de realizar adopciones y los servicios sociales, ya que actualmente las realizan por medio de panfletos y redes sociales,

siendo estos medios de divulgación no muy eficientes. Todas estas dificultades obedecen a la falta de servicios tecnológicos que permitan agilizar y optimizar los procesos.

La falta de aplicativos tecnológicos que integren los procesos cotidianos que realizan los animalistas, veterinarias, guarderías para mascotas y de más empresas y/o personas que intervienen en las labores de protección y servicio a los animales, dificultan la posibilidad de crear comunidades en línea para facilitar las tareas animalistas que requieren mayor grado de atención, como lo son las adopciones, los eventos, servicios sociales y demás servicios prestados.



3.2 Formulación del problema

- ¿Cómo desarrollar una aplicación móvil que articule la labor social de las fundaciones, animalistas y empresas que intervienen en el beneficio de los animales?

3.3 Preguntas de sistematización

- ¿Cómo caracterizar el mercado de las mascotas y protección animal con el fin de establecer los requerimientos tecnológicos de la solución a implementar?
- ¿Cómo diseñar una solución tecnológica orientada a servicios que integre los requerimientos del mercado de las mascotas y protección animal?

- ¿Cómo implementar los módulos de la solución tecnológica relacionados con los servicios de gestión de mascotas, adopción, publicidad, donaciones, eventos y reportes de la frecuencia de interacción de los usuarios?
- ¿Cómo evaluar la solución tecnológica con relación a la usabilidad de sus interfaces y tiempos de respuesta?

4. Justificación

La comunidad animalista preocupada por las desapariciones de mascotas y la falta de herramientas digitales para difusión masiva de eventos, servicios sociales y promociones en centros médicos veterinarios, han promovido la implementación de estrategias de divulgación masiva por medio de las redes sociales. Este método de comunicación ha demostrado ser efectivo en algunos casos ya que tiene alcance a una gran cantidad de personas, sin embargo no siempre es eficiente en los tiempos de entrega de información, ni en la población objetivo. Por otra parte, existen aplicaciones móviles independientes que buscan facilitar la gestión de eventos y servicios sociales, siendo estas plataformas especializadas en la temática, con operatividad muy similar sin la integración de servicios de interés general para la comunidad animalista relacionadas con eventos, servicios sociales y ofertas de mascotas en adopción.

Este proyecto busca aportar en la construcción de una herramienta tecnológica social que integre los procesos que realizan los animalistas en busca del bienestar de las mascotas. El desarrollo tecnológico propuesto es centralizado y brinda información de una red de apoyo para gestionar procesos relacionados con este sector. Se espera impactar la comunidad animalista con el uso de una aplicación móvil híbrida que les permita manejar de manera responsable servicios relacionados con la pérdida, adopción de mascotas y eventos culturales.

Por otro lado, el mercado de las mascotas ha crecido exponencialmente en Colombia, convirtiéndose en tendencia y generando diversas opciones de negocio. A pesar de ser un segmento grande y llamativo no se ha explotado completamente, por esta razón se encuentra aquí una gran oportunidad para emprender, ya que el mercadeo de productos relacionados con perros y gatos ha crecido 44% en Latinoamérica durante los últimos cinco años y se proyecta que crezca por lo menos 15% en los próximos años. Colombia no ha sido ni será ajena a esta tendencia (Forero, 2016) y es muy común encontrar en el mercado oferentes de servicios para mascotas que buscan

reconocimiento y mayor rentabilidad. Todo el crecimiento de este sector conlleva a una competencia en el mercado en la cual se busca vías de comunicación para llegar al cliente final y que mejor que un desarrollo tecnológico que integre las entidades productivas y los consumidores.

5. Objetivos

5.1 Objetivo general

- Desarrollar una solución tecnológica basada en dispositivos móviles para gestionar procesos de bienestar en animales domésticos con condiciones de vulnerabilidad.

5.2 Objetivos específicos

- Caracterizar el mercado de las mascotas y protección animal con el fin de establecer los requerimientos tecnológico de la solución a implementar.
- Diseñar una solución tecnológica orientada a servicios que integre los requerimientos del mercado de las mascotas y protección animal.
- Evaluar la solución tecnológica con relación a la usabilidad de sus interfaces y tiempos de respuesta.

6. Metodología

Las metodologías de desarrollo de software son decisivas en el éxito o fracaso de un proyecto. En general las metodologías ponen en práctica una serie de procesos comunes, que son buenas prácticas para lograr los objetivos de negocio, costes, funcionalidad, sencillez, etc. La elección de una metodología inadecuada o su mala aplicación pueden conducir a que el proyecto no llegue a su fin.

Actualmente existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones

que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad.

Cuando los proyectos se enfocan en metodologías de desarrollo de software tradicionales aparecen variantes como el Proceso Unificado Racional (RUP) que es un proceso metodológico de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Esta metodología se basa en 5 principios de desarrollo: Adaptabilidad en el proceso, Balance de las prioridades, Demostrar valor iterativamente, Elevar el nivel de abstracción, Enfocarse en la calidad.

Otra alternativa para trabajar con metodologías tradicionales es Microsoft Solution Framework (MSF) que es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere, más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información. Todo proyecto que aplique esta metodología se estructura en cinco fases, las cuales son: Visión y Alcances, Planificación, Desarrollo, Estabilización e Implantación.

En cuanto a metodologías de desarrollo de software ágiles, tenemos la Programación Extrema (XP) que está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Consta de 4 fases: Planeación, Diseño, Codificación y Pruebas.

Otra variante de las metodologías de desarrollo de software ágil es SCRUM que fue desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle y define un marco para la gestión

de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas Sprint, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

En la tabla 1, se propone una comparación entre las metodologías tradicionales RUP y MSF y las metodologías de desarrollo de software ágiles XP y SCRUM.

Tabla 1 Matriz comparativa de metodologías.

Característica	RUP	MSF	XP	SCRUM
Más enfocado a los procesos	X	X		
Resultados rápidos			X	X
Heredan Modelos	X	X		
Documentación estricta	X	X		
Refactorización del código			X	

Nota: Tabla elaborada a partir de <http://biblioteca.uniminuto.edu/ojs/index.php/Inventum/article/download/9/9>.

Teniendo en cuenta el mercado al que va dirigido el desarrollo tecnológico y el tiempo que se tiene para dicho desarrollo, es necesario tener un proceso definido con una definición formal de requerimientos, análisis de estos, diseño e implementación de la arquitectura con enfoque UML. Por otro lado, es indispensable tener una metodología en la que se valore la importancia de los individuos y las iteraciones, el funcionamiento de las aplicaciones, la colaboración del cliente y la respuesta al cambio. Dado que los métodos ágiles hacen más explícita la importancia en el manejo del equipo y personas, se pueden pensar cómo un complemento para las metodologías tradicionales que están más inclinadas a los procesos y la documentación.

Dicho lo anterior, y con el fin de desarrollar una solución tecnológica basada en dispositivos móviles para gestionar procesos de bienestar en animales domésticos con condiciones de vulnerabilidad, se propone la metodología de Software Scrum, teniendo en cuenta que es una metodología de desarrollo simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Se decidió esta metodología porque:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil, incremental basada en iteraciones y revisiones.
- Se aceptan requisitos cambiantes.

6.1 Fases de la Metodología Scrum para móviles

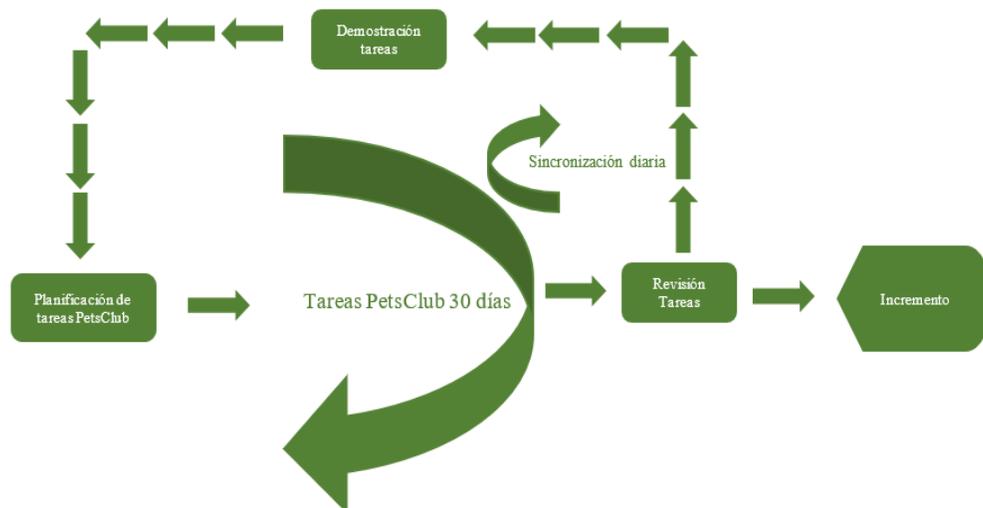
Scrum no posee las mismas fases que otras metodologías, se caracteriza por la realización de Sprint con duración de dos o tres semanas. En este proyecto tendremos las siguientes etapas:

- **Planificación Inicial:** Para el desarrollo de la solución tecnológica se realizó una reunión sólo al inicio del proyecto, donde se reunió el grupo de trabajo con el representante del cliente. Se determinaron los requisitos iniciales y la visión del producto desde el punto de vista del negocio. Se consolida esta información en el Product Backlog.
- **Product Backlog:** Para el desarrollo de la solución tecnológica se realizó una lista de todas las tareas que se llevaran a cabo durante el desarrollo del proyecto.
- **Planificación del Sprint:** Para el desarrollo de la solución tecnológica se realizó la planificación de los sprint el cual se lleva a cabo reuniones semanales, con el fin de revisar los avances que se han tenido.

Cada sprint, a excepción del primero, genera un producto probado y funcional de interés para el cliente. Se auto asignan las tareas a un integrante del equipo y se estiman los recursos necesarios para su desarrollo.

- **Sprint:** Para el desarrollo de la solución tecnológica se lleva a cabo la realización de cada uno de los sprint que se ha planificado, cada sprint duró entre 3 y 4 semanas.
- **Sprint Backlog:** Para el desarrollo de la solución tecnológica se realiza el Sprint Backlog el cual incluye un listado de las funcionalidades que se desarrollan durante la siguiente iteración.
- **Desarrollo del Sprint:** Para el desarrollo de la solución tecnológica se tendrá en cuenta que una vez realizado el Sprint Backlog el sprint puede iniciar su desarrollo. Su duración no deberá exceder los 30 días.
- **Revisión del Sprint:** Para el desarrollo de la solución tecnológica se tuvo en cuenta al final de cada sprint la realización de una reunión en la que el equipo de trabajo presenta el producto y demás involucrados el incremento construido durante el sprint. Su duración es de máximo 4 horas.
- **Retrospectiva del Sprint:** Para el desarrollo de la solución tecnológica se tuvo en cuenta la retrospectiva del Sprint, es una reunión facilitada por el Scrum máster en la cual los miembros del equipo de Scrum discuten el Sprint que acaba de finalizar y determina qué podría cambiarse en el próximo Sprint para que sea más productivo y mejor.

Figura 2. Diagrama Scrum



Marco de referencias

6.2 Marco teórico

6.3 Comunidades animalistas

El Animalismo es una nueva tendencia sociocultural, Sus orígenes están en los llamados nuevos movimientos sociales, como el ecologismo y el pacifismo surgidos a finales de la década de los sesenta, años después de ser aprobada la Declaración Universal de los Derechos del Animal, al mismo tiempo que autores célebres comenzarían a otorgar un papel más relevante a los animales. La ecología le aporta la teoría del biocentrismo que propende por el respeto a la vida en todas sus manifestaciones y propone que “la actividad humana cause el menor impacto posible sobre otras especies y sobre el planeta en sí”.

El Animalismo es un notable avance de la civilización al asumir frente a los animales otras actitudes como la compasión y la solidaridad. Y más allá, al reconocerlos como seres que sienten y como sujetos de derechos.

El enfoque animalista o movimiento de liberación animal surge con una visión fundación hacia los animales; dado que todo animal cuenta con un sistema nervioso semejante al humano, es decir, sienten dolor, alegría o tristeza, es necesario un marco jurídico para protegerlos.

Para este movimiento es importante que el ser humano se aleje de la concepción material hacia el animal, es decir, considerarlo como un bien mueble semoviente y le faculte para otorgarle la naturaleza de sujeto de derecho; de esta manera también serían titulares de derechos fundamentales y su marco de protección también sería mucho más amplio.

Colombia es un país que intenta encaminarse por las vías de la convivencia y del respeto, como salida a la situación de violencia, injusticia e indiferencia. Resulta imprescindible que este reconocimiento se dé, no sólo hacia los congéneres, sino hacia los seres de otras especies. Esto, en el entendido que los animales son seres sintientes, es decir, también experimentan placer, dolor y angustia; así mismo comparten espacios con el ser humano y son seres que poseen comportamientos naturales e intereses ajenos a las costumbres o esquemas humanos.

Es importante establecer relaciones de sana convivencia y respeto por los animales y por esta razón desde hace algunas décadas, en distintas regiones colombianas, organizaciones compuestas por la sociedad civil, vienen trabajando para que Colombia reconozca la existencia de los derechos de los animales y el Estado asuma su protección ya que actualmente esas funciones las están afrontando sectores ciudadanos, sin recursos y sin suficiente amparo legal.

6.3.1 Características de las comunidades animalistas

Podemos considerarlo como una fuerza política cada vez más valorada en una sociedad injusta e insolidaria. Sus posturas se acercan a las corrientes ecologistas y al pensamiento de la izquierda verde, aunque para el enfoque animalista o movimiento de liberación animal, la protección hacia los animales llega al extremo de proclamar una alimentación 100% vegetal, descartándose el consumo de animales o productos elaborados con materia de origen animal.

Las Sociedades Fundación de Animales funcionan en Colombia desde hace medio siglo. Pero, en los últimos años, el movimiento ha adquirido una gran fuerza y cada vez son más los jóvenes que se interesan y participan activamente reclamando protección del estado hacia los animales, abolición de corridas de toros u otro tipo de participación en fiestas populares; mayor control en la comercialización animal, velando por evitar el sufrimiento que se produce en la manipulación de estos: mataderos, mercado de pieles, animales de laboratorios, caza, entre otros.

Cuando se habla de animalismo las ideas que generalmente llegan a nuestra mente imágenes de activistas y protestantes boicoteando industrias o impidiendo el desarrollo de espectáculos donde se usan animales, pero en los últimos años las redes sociales han contribuido a la creación de redes de apoyo y grupos juveniles que asume el respeto y responsabilidad con los animales como su forma de vida, activistas unidos por las tecnologías de la información que aman lo que hacen y convencen con el ejemplo.

Las corridas de toros en Colombia son cada vez menos populares y es muestra de ello la cancelación de este tipo de eventos en la plaza de toros la Santamaría desde el año 2012 y la fuerte reacción que tuvo la población capitalina cuando la corte constitucional fallo a favor de la tauromaquia como una forma de expresión cultural y artística. Miles de personas participaron de una movilización pidiendo la consulta popular antitaurina.

Los jóvenes son cada vez más reacios a asistir a este tipo de espectáculos, lo que indica que el animalismo está despertando nuevas conciencias y generando grandes cambios culturales, si consideramos la tauromaquia como un espectáculo tradicional en las ferias y fiestas de las principales ciudades del país.

El movimiento animalista se caracteriza por la realización de manifestaciones o protestas pacíficas, en donde a través de simbolismo educa a la sociedad acerca de la igualdad que se reclama para los animales, como seres vivos con quien compartimos el planeta.

Un gran logro para los animalistas fue en el año 2013, cuando el senado de Colombia aprobó en plenaria la prohibición del uso de animales silvestres, sean nativos o exóticos, en circos fijos e itinerantes en todo el territorio nacional.

Después de 10 años de lucha, esta ley reafirma que nuestro país está comprometido con protección de la fauna y da pie para cambiar la mentalidad de nuestros niños acerca de ver los animales como fuente de diversión o entretenimiento, contrario a esto crear una cultura de protección y preocupación por conservación de las especies.

Otra actividad conocida como tradición en Colombia son las populares cabalgatas, las cuales también han sido diezmadas o al menos las autoridades han reglamentado y ejercido más control sobre esta práctica. El antropólogo Fabián Sanabria, director del Instituto Colombiano de Antropología e Historia (Icanh), explicó las cabalgatas engalanan al mundo rural, el ganado y las extensiones de tierra para ganado, al mismo tiempo, es una manera de mostrar el poderío de un gran patrón.

Sobre el creciente movimiento animalista que se opone a las cabalgatas, señaló además, que para las nuevas generaciones las prácticas que se realizan en el modelo rural no son para seguirlas, porque están cambiando sus formas de ver el mundo. El rechazo a estas actividades, según el experto, demuestra que “hay una conciencia hacia las otras especies que tocan mucho la sensibilidad de los jóvenes. Muestra la transición generacional”.

Actualmente en la primera comisión de la cámara de representantes fue aprobado un proyecto de ley con el objetivo de crear un tipo penal que se llame maltrato animal. Esto permitiría emular experiencias de 8 países en latinoamericanos donde se ha penalizado este tipo de maltratos y, porque no, soñar con llegar a un nivel como el alcanzado en Alemania donde el reconocimiento de los animales es constitucional.

El proyecto de ley como está redactado permitiría pasar de un código de policía en el que se penaliza a quien maltrate a un animal con 200 mil pesos a una multa de 60 salarios mínimos legales vigentes y poder imponer penas cercanas a los 36 meses de cárcel.

Una de las grandes fortalezas con las que cuenta actualmente el movimiento animalista en Colombia es la conformación de grupos y asociaciones fundación de animales, comunidades virtuales con la misión de velar por la protección de los animales que encuentran en condición de abandono o maltrato, publican casos que requieren acción inmediata las cuales son resueltas a través de todos los colaboradores de la red con: donaciones, colectas, hogares de paso, jornadas de esterilización, adopciones, entre otras.

Este tipo de organizaciones ha se ha encargado de crear una conciencia colectiva de responsabilidad social hacia los animales, y cada vez son más los adeptos que ofrecen su ayuda. estos jóvenes desde la cotidianidad de sus días hacen que el animalismo se conforme como cultura social en nuestro país, ya que está cambiando las ideas desde sus comunidades (Londoño, 2015).

6.4 Tecnologías habilitadoras para el desarrollo del proyecto

6.4.1 Desarrollo de aplicaciones móviles híbridas

En la actual era de la tecnología digital, donde cada vez dependemos de grandes desarrollos tecnológicos para llevar a cabo labores de la cotidianidad, cómo, por ejemplo, enviar emails y comunicarnos con nuestros seres queridos por mensajerías instantáneas; diseñar aplicaciones para ser instaladas únicamente en un solo sistema operativo, es una inversión poco beneficiosa. ¿Por qué razón? Imagínate que necesitas responder un correo desde tu teléfono, pero la app de tu Web mail no es compatible con el sistema operativo de tu móvil, razón por la que nunca podrás enviar el mensaje. ¡Vaya sorpresa! Esto no ocurriría si este servicio se creara con aplicaciones híbridas.

Las aplicaciones híbridas son aplicaciones móviles diseñadas en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un framework que permite adaptar la vista web a cualquier vista de un dispositivo móvil. En otras palabras, no son más que una aplicación construida para ser utilizada o implementada en distintos sistemas operativos móviles, tales como, iOS, Android o Windows Phone, evitándonos la tarea de crear una aplicación para cada sistema operativo. De esta manera, una aplicación híbrida puede ser adaptada a múltiples

plataformas móviles sin crear nuevos códigos, pero ajustándose a algunos cambios operacionales para cada uno de ellos.

Sin embargo, a pesar de que el desarrollo de aplicaciones híbridas y nativas requiere de una construcción totalmente distinta, la forma de utilizarlas es igual. Para ello, solo debes dirigirte hasta la tienda de aplicaciones de tu dispositivo móvil, buscar la App que quieres instalar y descargarla. Aunque ambas son iguales en su forma de usabilidad, el rendimiento de una aplicación híbrida comparada con una nativa es mucho menor, debido a que estas últimas aprovechan de forma óptima los recursos de hardware del dispositivo, por ejemplo, la cámara, el GPS, los sensores en el interior del dispositivo, entre otros. Por el contrario, las aplicaciones híbridas también pueden utilizar estos recursos de hardware, pero no al mismo nivel en comparación con las nativas.

No obstante, en la mayoría de los casos es difícil diferenciar una aplicación nativa de una híbrida. Una de las formas de detectarlo, es comparando el diseño visual en varios dispositivos con sistemas operativos distintos; si notamos que visualmente el comportamiento y estructura es igual podríamos decir que es una App híbrida, pero si los elementos visuales están distribuidos en posiciones desiguales con características diferentes, se deduce que la App es nativa. Cabe aclarar que, aunque lo expuesto anteriormente es una de las razones para diferenciarlas, es posible desarrollar aplicaciones nativas visualmente idénticas para distintos sistemas operativos.

Entonces, ¿por qué desarrollar un proyecto en una aplicación móvil híbrida? Aquí 3 razones:

- Su creación es mucho más sencilla y económica.
- El código base con el que se crea la app puede utilizarse en múltiples plataformas.
- No necesitas de permisos externos para publicarla en las tiendas de aplicaciones («Aplicaciones híbridas», 2017).

6.4.2 Framework Ionic para desarrollo híbrido

Ionic es un framework propiedad de la empresa Drifty. Es un marco de desarrollo en HTML5 dirigida a la creación de aplicaciones móviles híbridos, estas son esencialmente pequeños sitios web que se ejecutan en un navegador que tienen acceso a la capa de plataforma nativa. Las

aplicaciones híbridas tienen muchas ventajas sobre las aplicaciones nativas puras, específicamente en términos de soporte de la plataforma, la velocidad de desarrollo, y el acceso al código.

Además, Ionic nos va a aportar un ecosistema de herramientas que en la actualidad son completamente gratuitas y que nos van a facilitar en gran medida las labores de desarrollo. Entre ellas encontramos:

- **Ionic.io Platform:** Conjunto de herramientas para gestionar nuestra aplicación, entre ellas las que nos serían más útiles para nuestro proyecto encontramos:
- **Ionic Creator:** Herramienta de desarrollo gráfica que nos permite realizar las primeras fases de implementación de la aplicación de una forma gráfica, arrastrando y soltando componentes y creando páginas de forma automática.
- **Ionic Deploy:** Ha aparecido recientemente y es la solución back-end de Ionic que nos va a permitir realizar actualizaciones en nuestra aplicación. o **Ionic Analytics:** Servicio de analíticas para nuestra aplicación.
- **Ionic View App:** Aplicación móvil para realizar pruebas en terminales.
- **Ionic Lab:** Herramienta para testear nuestra aplicación en múltiples tamaños de pantalla y sistemas operativos.

Ionic funciona sobre Angular JS. Angular JS es un framework de JavaScript de código abierto propiedad de Google que proporciona a los desarrolladores web la posibilidad de escribir aplicaciones completas de forma rápida y proporciona una buena estructura de la aplicación.

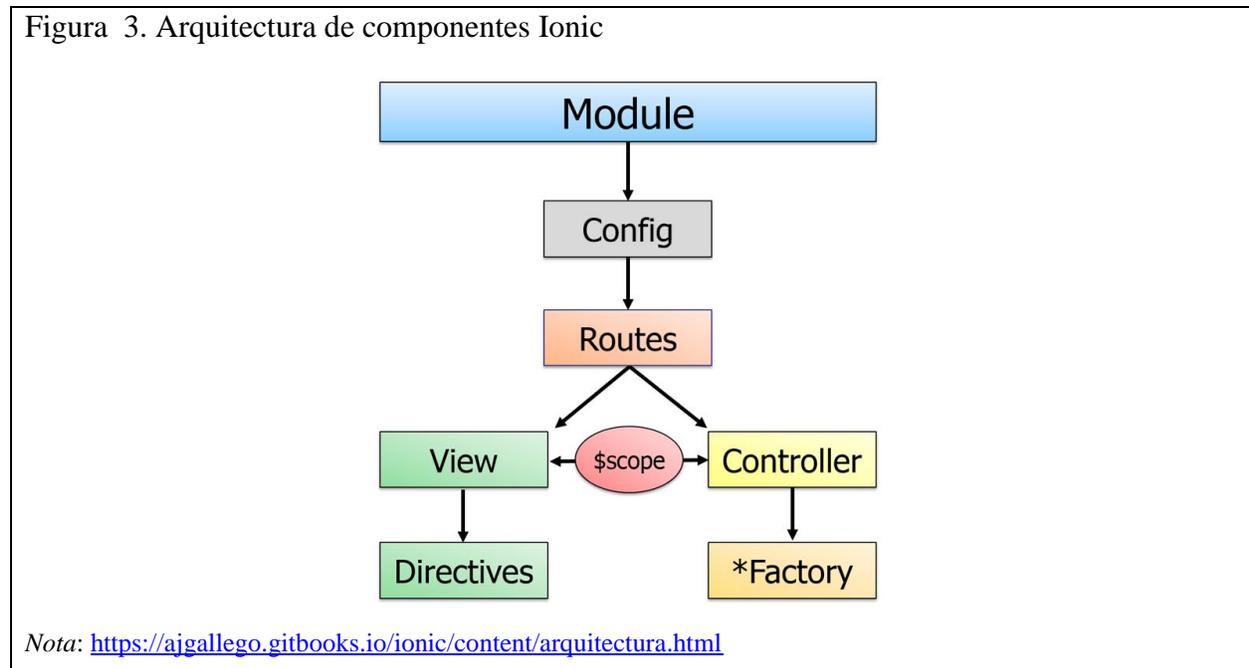
En la arquitectura de una aplicación intervienen muchos tipos de componentes además de las Vistas y Controladores, iremos viendo cada uno de ellos poco a poco, pero principalmente los que nos interesan son las Vistas, Controladores, Servicios o Factorías, y la Configuración y Rutas.

Intuitivamente, la tarea de cada uno de estos componentes en una aplicación con Ionic es la siguiente:

- Los controladores obtienen los datos de uno o varios Servicios o Factorías y lo envían a una vista o template a través de la variable `$scope`.
- Las vistas o templates contienen la descripción visual de una pantalla (o de una parte de una pantalla) y obtienen los datos a mostrar de la variable `$scope`.

- La configuración y las rutas de la aplicación permiten enlazar los controladores con las vistas o templates correspondientes.
- Las directivas permiten crear y usar componentes con aspecto y comportamiento personalizado.

En la figura 3, se muestra la arquitectura de Ionic.



6.4.3 Servicios de Geolocalización en aplicaciones móviles híbridas

Muchos de los inventos que hoy son de uso común en el diario vivir de las personas fueron desarrollados por la milicia como herramientas de supervivencia o de inteligencia. De la amplia lista que existe se pueden destacar algunos artículos tan básicos como el reloj pulsera, la comida enlatada, las gafas de sol y los vehículos todo terreno.

Sin embargo, existe otra larga lista de artículos tecnológicos que realmente les han cambiado la vida a los seres humanos transformando radicalmente la manera en la que se hacen las cosas como son las computadoras, de internet y los geolocalizadores o GPS, (Global Positioning Systems). Este último se ha convertido en un gran aliado de identificar la ubicación exacta de una cosa o una persona, se ha traducido en una inmensidad de posibilidad y usos.

Con la asistencia de los geocalizadores podemos saber cómo llegar a un destino que nunca hemos visitado, le permite conocer al conductor de Uber nuestra ubicación exacta para que pueda recogerlos, a poder encontrar las llaves perdidas, el celular, una mascota o inclusive monitorear y saber dónde está una persona de interés. Los GPS además se han convertido en una herramienta indispensable para medir las millas recorridas cuando hacemos destino («Tecnología GPS», 2017).

Figura 4. Geocalización



Nota: <https://www.elnuevodia.com/negocios/consumo/nota/latecnologiadegeocalizacionavanza-2345240/>

6.4.4 Notificaciones push en aplicaciones móviles

Si bien las Apps continúan dominando el mercado de dispositivos móviles, muchos responsables de marketing buscan alternativas creativas a este contexto. Existen diferentes caminos que pueden tomar. Sin embargo, las notificaciones push son las más eficaces.

Más complejo que el mensaje de texto, el push representa una moderna forma de comunicación. Además, es una de las estrategias Mobile que mayor ROI tiene. Al igual que los textos, las notificaciones push se entregan directamente a los teléfonos de los usuarios cuando han instalado una aplicación. Pero, a diferencia de los textos planos, tienen una amplia gama de opciones de personalización que pueden aumentar la participación e interacción del usuario.

Las notificaciones push existen desde hace varios años, aunque han ganado fuerza con la llegada de los Smartphone. Lo que se debe tener claro si quieres saber qué son las notificaciones

push es que no solamente existen en los dispositivos móviles. A continuación, podrás entender qué son las notificaciones push y por qué deberías aprovecharlas.

Según Wikipedia, “la tecnología push es una forma de comunicación a través de Internet en la que la petición de envío tiene origen en el servidor, por oposición a la tecnología pull, en la que la petición tiene origen en el cliente”.

Partiendo de esta premisa las notificaciones push son mensajes instantáneos que recibes en tus dispositivos. Los mensajes de WhatsApp, por ejemplo, son mensajes push. También lo son los SMS de promociones, las notificaciones que recibes en tu navegador web o las notificaciones de aviso de un nuevo email.

Es importante tener claro que pese a que este tipo de notificaciones sean más frecuentes en los dispositivos móviles también cada vez se utilizan más en Internet. A través de un sistema de suscripción se habilita a las marcas para que te envíen notificaciones push.

Está claro qué son las notificaciones push. Por eso, no se deben mezclar con los mensajes de texto. Los textos y las notificaciones push son tan similares que hay veces que se habla como si fuesen una misma cosa. Ambos llegan al dispositivo móvil de un usuario y deben ajustarse dentro de los límites estrictos de caracteres.

Sin embargo, pese a tener características parecidas, sí que existen unas diferencias marcadas. Entonces, ¿por qué se dice que las notificaciones push son mejores? Pero ¿qué son las notificaciones push y en qué se diferencian con los mensajes de texto? Es importante aclarar la diferencia que existe entre estos dos tipos de acciones.

A menudo son consideradas la misma técnica de marketing y eso es un error. De ahí la negativa de responsables de marketing y usuarios a recibir este tipo de notificaciones. Sin embargo, las notificaciones push son mensajes cero intrusivos ya que el propio usuario decide si quiere recibirlas y cómo.

El coste es una de las mayores diferencias. Muchos usuarios no tienen tarifas de mensajes de texto ilimitados y se les cobra por cada mensaje de texto que envíe como marca. Esto puede enfadar a los usuarios y crear una mala imagen de marca. Las notificaciones push, en cambio, son ilimitadas y totalmente gratuitas.

La dinámica opt-in/opt-out también favorece las notificaciones push, cuando se trata de la actitud del usuario. Las personas tienen control total sobre qué aplicaciones les envían notificaciones e, incluso, pueden especificar qué tipos de notificaciones quieren de cada aplicación.

Los mensajes de texto, sin embargo, a menudo no son solicitados. De hecho, debido a la venta de listas de números de teléfono entre compañías, algunos usuarios pueden recibir mensajes de texto no deseados. De ahí parte la gran fama negativa de las notificaciones push. Ya que, a menudo, los usuarios confunden los mensajes de texto con este tipo de notificaciones. Por ello, si tienes claro qué son las notificaciones push sabrás que no son nada intrusivas. Las notificaciones push también aumentan el engagement y, además, mejoran las tasas de retención de usuarios.

Por lo tanto, no solo pueden ayudar a impulsar cierta actividad concreta con el mensaje, sino que también ayudan a aumentar el ROI. Las notificaciones push cumplen una doble función porque, además de ofrecer una serie de beneficios, la marca consigue mejorar su engagement.

Las aplicaciones a través de las que se envían las notificaciones push tienen la posibilidad de ser personalizadas. De esta manera, conseguirás atraer a tus clientes o dar señales de que tu marca sigue viva.

Por ejemplo, puedes enviar un mensaje de notificación push fácilmente diseñado en segundos y ofreciendo a los usuarios un 10% de descuento en alguna acción que propongas. Esto no solo ayuda a atraer nuevos clientes, sino que también, puede ayudar a aumentar la tasa de fidelización de tus clientes («Qué son las notificaciones push | Guía efectiva», 2018).

Figura 5. Notificaciones Push



Nota: <https://www.mdirector.com/sms-marketing/que-son-las-notificaciones-push.html>

6.4.5 Base de datos en tiempo real

La base de datos en tiempo real fue la primera característica que lanzamos para Firebase, hace más de cuatro años. Las bases de datos de JSON alojadas en la nube eran bastante nuevas en ese momento. Por ello, se distinguió de inmediato. Lo que hizo realmente única a la base de datos

de Firebase fue su capacidad para sincronizar datos entre los distintos clientes al instante con una sola API fácil de usar.

A través de una interfaz de programación uniforme, la base de datos en tiempo real hace que desaparezca la complejidad a la hora de crear aplicaciones. Esto permitió crear fácilmente aplicaciones colaborativas en tiempo real.

La base de datos en tiempo real en verdad es simplemente un objeto JSON de gran magnitud que puedes administrar en tiempo real. Esto significa que solo es un árbol de valores.

Esto permite modelar datos de manera simple y flexible. La base de datos no tiene esquema. Esto significa que no debes adoptar una estructura fija desde el comienzo. Sin embargo, si te preocupa la validación de datos, la base de datos en tiempo real incorpora un lenguaje de reglas impuestas por servidor que te permite validar la estructura de datos de cada acción de escritura en la base de datos.

El aspecto realmente único de la base de datos de Firebase es su sincronización en tiempo real. Con la mayoría de las bases de datos tradicionales, se debe trabajar aplicando un modelo de solicitud y respuesta. Se escribe una solicitud y luego se solicitan a la base de datos los resultados de dicha solicitud. Si se necesitan nuevamente los resultados, o se intenta buscar actualizaciones, se ejecuta nuevamente la solicitud.

Nuestra base de datos es diferente. Nuestra base de datos notifica los cambios que se producen. Se le indica: “si algo sucede en esta ubicación del árbol, notificame”. Esto facilita el seguimiento de los cambios y el mantenimiento de la sincronización para todos los usuarios («Base de datos en tiempo real», 2017).

6.4.5.1 Eventos en tiempo real

Con una única API, la base de datos de Firebase proporciona a tu aplicación el valor actual y las actualizaciones de los datos.

Los datos se muestran en la interfaz de usuario de tu aplicación y el usuario puede manipularlos. Es así de simple.

¿Qué sucede cuando no hay conexión? ¿Alguna vez usaste una aplicación que dejó de responder en el momento en que se perdió la conexión de red? La mayoría de los desarrolladores desean que sus aplicaciones funcionen sin importar las condiciones de la red. Por ello, la base de

datos de Firebase conserva un caché interno con todos los datos que se muestran en la aplicación. Cuando se pierde temporalmente la conexión de red, la aplicación continúa funcionando a través del caché. Durante estas condiciones de conexión, el cliente de datos de Firebase deja en cola operaciones de escritura locales. Tu aplicación continúa funcionando y responde. Todo ello, sin necesidad de escribir código adicional.

El poder de la plataforma Firebase yace en la integración de todas nuestras funciones. Agregaremos más integraciones, pero a continuación te mostramos algunos ejemplos de las que ya puedes usar: podrás permitir que tus usuarios inicien sesión en tu aplicación y protejan el acceso a sus datos según su identidad. Tendrás la posibilidad de almacenar los archivos de los usuarios en Firebase Storage y luego usar la base de datos en tiempo real a fin de sincronizar el estado y la disponibilidad para todos los demás usuarios de la aplicación. También podrás hacer un seguimiento de los archivos más populares con Firebase Analytics e invitar a otros usuarios a colaborar con esos archivos. Las posibilidades son casi ilimitadas y crear una aplicación nunca fue tan sencillo como con Firebase («Base de datos en tiempo real», 2017).

6.4.6 Servicios Web basados en REST

El protocolo REST (Representational State Transfer), plantea un estilo de arquitectura cliente-servidor en la cual un servicio es visto como un recurso que se identifica a través de una dirección URL, mediante la cual puede ser accedido o consumido. Para acceder a estos servicios web, se hace uso de mensajes en formato simple, los cuales se intercambian entre cliente y servidor.

En realidad, REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen como los recursos son definidos. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional. REST se sustenta sobre los estándares de HTTP y URI (Uniform Resource Identifier). Un concepto importante en REST es la existencia de recursos; Para manipular estos recursos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de estos recursos. Para considerar un sistema como REST se debe apoyar en los siguientes principios básicos:

- **Escalabilidad de la interacción con los componentes:** La Web ha crecido exponencialmente sin degradar su rendimiento. Una prueba de ellos es la variedad de clientes que pueden acceder a través de la Web como las estaciones de trabajo, sistemas industriales, dispositivos móviles, entre otros.
- **Generalidad de interfaces:** Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial. Esto no es del todo cierto para otras alternativas, como SOAP para los Servicios Web.
- **Puesta en funcionamiento independiente:** Este hecho es una realidad que debe tratarse cuando se trabaja en Internet. Los clientes y servidores pueden ser puestos en funcionamiento durante años. Por lo tanto, los servidores antiguos deben ser capaces de entenderse con clientes actuales y viceversa. Diseñar un protocolo que permita este tipo de características resulta muy complicado. HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.
- **Compatibilidad con componentes intermedios:** Los más populares intermediarios son varios tipos de proxys para Web. Algunos de ellos, las caches que se utilizan para mejorar el rendimiento. Otros permiten reforzar las políticas de seguridad: firewalls, por último, otro tipo importante de intermediarios tipo Gateway.
- **Interfaz uniforme:** La principal característica que distingue a REST del resto de estilos de arquitecturas de red es el énfasis de usar una interfaz uniforme entre los componentes. Las operaciones disponibles sobre los recursos son siempre las mismas y su semántica es conocida por todos los clientes y servicios. En REST sobre HTTP, lo más normal es tener estas operaciones como:
 - **POST:** Crea nuevos recursos. Como retorno, se ofrece el ID automáticamente creado.
 - **GET:** Lista un recurso.
 - **PUT:** Reemplaza un recurso con otro (Útil para el UPDATE).
 - **DELETE:** Elimina un recurso.

6.4.7 Diseño de arquitectura de software basado el modelo 4+1 Vistas de Kruchten

La importancia de tener un modelo de desarrollo, que sea eficiente y eficaz, hace que se tomen estándares necesarios para suplir todas las necesidades en el proceso de desarrollo de software. Por otro lado, los diseñadores construyen la arquitectura usando varios elementos de diseños elegidos apropiadamente. Estos elementos satisfacen la mayor parte de los requisitos de funcionalidad y desempeño del sistema, así como también otros requisitos no funcionales tales como confiabilidad, escalabilidad, portabilidad y disponibilidad del sistema.

Una de las estrategias más importantes para modelar arquitecturas de software es la que propone Philippe B. Kruchten, en la cual se define que un desarrollo de software se debe de documentar y mostrar (por medio de diagramas UML) con 4 vistas bien diferenciadas (vista lógica, vista de procesos, vista de despliegue y vista física), las cuales se deben relacionar entre sí con una vista más (vista +1). En la figura 6, se muestra la distribución de las 4 vistas propuestas por Kruchten.

Figura 6. Vistas de la arquitectura de Kruchten



Nota: Moya. Modelo "4+1" vista de Kruchten [en línea]. [Consultado el 12 de junio de 2015]. Disponible en internet: <http://jarroba.com/modelo-41-vistas-de-kruchten-paradummies/>

7. Antecedentes

Hoy en día con las nuevas tecnologías móviles como son los dispositivos Android, se ha conseguido que la sociedad tenga a su disposición en cualquier lugar y momento la posibilidad de tomar todo tipo de información utilizando una gran diversidad de hardware como son las cámaras

de fotos, los grabadores de audio o las pantallas táctiles, y transmitirla fácilmente a cualquier lugar del mundo. De ahí surge la idea de utilizar estas tecnologías móviles para solucionar el problema antes mencionado y poner a disposición de la comunidad de animalistas aplicativos móviles que hagan uso de estos recursos para gestionar servicios sociales y culturales que contribuyan a un manejo responsable de la información de las situaciones relacionadas con los animales domésticos. En el contexto internacional encontramos proyectos referentes al trabajo de grado que se basan en soluciones web y móviles que promueven las buenas prácticas de protección animal y el despliegue de servicios sociales:

- **“Adopta un Gos”**: Consiste en una aplicación web desarrollada en México que permite buscar y filtrar datos de perros perdidos y/o abandonados de una base de datos centralizada de webs protectoras de animales. Esta aplicación no solo permite visualizar la descripción e imagen del perro perdido, sino también un mapa con la última localización antes de la pérdida, promoviendo la difusión de información veraz a los demás usuarios (García y Valveny, 2016).
- **Miwuki**: Esta app es conocida como el Tinder para perros y gatos. Pero no se trata de una aplicación para buscar pareja a nuestras mascotas. En realidad, se trata de dar una segunda oportunidad a perros y gatos abandonados. Casi 300 protectoras de animales colaboran con esta app para que la gente pueda elegir mascota y dar una nueva vida a un animal.
- **Social animals**: Es una red social creada en Noruega para los apasionados de las mascotas, ya que todo se centra a su alrededor. Los usuarios se crean un perfil conjunto con su mascota, y con esa base, se inician las relaciones sociales. La aplicación también tiene eventos para los dueños y los animales, búsqueda por proximidad, por tipo de mascota.

En el contexto nacional, las aplicaciones tecnológicas que impactan la comunidad de animales domésticos han centrado su modelo de negocio en la prestación de servicios complementarios a la cuida de las mascotas. Algunas de las aplicaciones que sirven de referente para analizar el desarrollo tecnológico que se plantean en estas tesis son:

- **GoDog**: Es un emprendimiento digital desarrollado en Medellín y se basa en una app que busca formalizar el negocio de los paseadores de perros. Primero nació GoPets, la primera empresa de paseadores de perros del país. El aplicativo GoDog carga con la información

que se hace antes de que el perro sea aceptado para los paseos, “se hace una encuesta larga donde se deja toda la información del perro, fechas de vacunas, etc y una cantidad de datos del comportamiento y con eso se define si el perro es paseable o no y con qué tipo de mascotas se puede pasear para unirla a determinados grupos, el promedio es de 4 perros por paseo pero máximo se pasean 6. La información de la encuesta se lleva a una base de datos y al perro se le entrega un carné con un microchip. Antes de iniciar la caminata se pasa el carné y en la app puede hacer el seguimiento desde la oficina o desde donde esté”.

- **Distrito App animal:** Esta aplicación móvil desarrollada por la alcaldía de Bogotá y disponible para todos los sistemas operativos móviles cuenta con una pestaña de aplicaciones en la que se encuentra una **galería de fotos de todos los animales** que están disponibles para encontrar un hogar, de inmediato los ciudadanos accederán al formulario de solicitud para iniciar el proceso de adopción de un perro o gato.

Entre tanto si un animal esta extraviado, en esta aplicación los ciudadanos podrán subir la **foto de su gato o perro**, indicando nombre del animal, raza, edad y lugar en el que se perdió.

Los reportes de investigación consultados son trabajos recientes, con menos de cinco años de realización, en ellos se citan datos bibliográficos correspondientes al tema en cuestión.

8. Caracterización y definición de requerimientos de la App para el mercado de las mascotas y protección animal

Se realizó un acercamiento con el sector animalista, permitiendo conocer las funciones y procesos que realizan de una manera más profunda, se interactuó con miembros de la fundación animal SAFE, es un refugio canino que alberga alrededor de 900 perros, es una entidad privada por lo cual no reciben ningún tipo de ayuda por parte del gobierno, trabajan con las donaciones que realizan las personas voluntarias, por otra parte obtiene ayudas por medio de apadrinamiento el cual consiste en ir hasta el refugio donde eligen la mascota que quieren apadrinar, una vez escogida, el padrino debe de dar los datos personales como nombre, cédula, teléfono, correo electrónico, esto con el fin de tener un contacto directo con la persona a cargo y así saber las condiciones en que se encuentra, las cuotas para la manutención de la mascota son voluntarias,

aunque el costo mensual es aproximadamente 50.000 mil pesos. Por otra parte, las personas que quieran adoptar también lo pueden hacer, el cual debe de realizar unas preguntas para evaluar si el adoptante es idóneo, posteriormente la persona elige el canino que desea llevar y una vez elegido deberá de firmar un documento donde se compromete a no maltratar y mantenerla en perfectas condiciones, la fundación realiza visitas domiciliarias para constatar que su estado es favorable.

También se interactuó con otras fundaciones como lo son albergue patitas de amor y huellitas milagrosas, éstas se dedican a brindar los cuidados necesarios a las mascotas, además realizan adopciones y cuentan con apadrinamiento o donaciones.

Por otra parte, se entrevistó un cementerio dedicado solo para las mascotas el cual se llama “Parque cementerio tierra de mascotas”, este cementerio realiza todo el proceso funerario, también se entrevistaron veterinarias (súper mascotas, petto, entre otras).

A partir de dicha participación con la comunidad animalista donde se realizaron entrevistas y/o encuestas a 71 personas, entre ellas: organizadores y participantes de los eventos, tanto fundaciones como veterinarias y personas que velan por el bienestar animal, de ahí se analizaron los siguientes puntos:

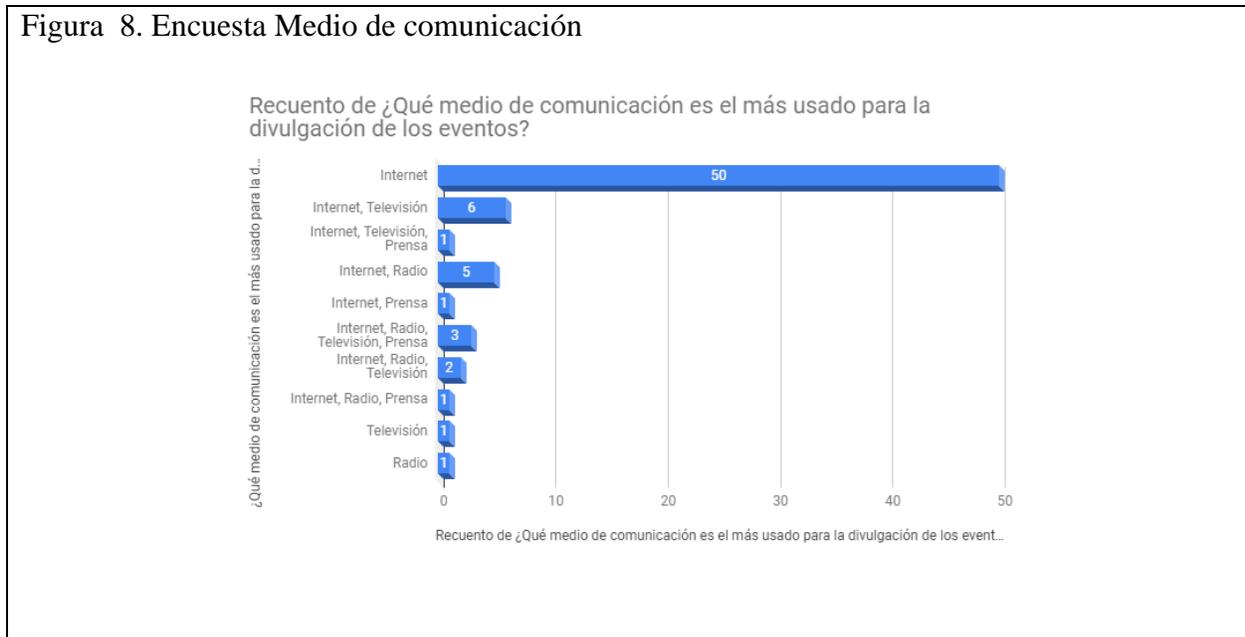
Figura 7. Encuesta falta de canales de comunicación



Se les preguntó a los encuestados si es necesario un canal que les permita dar a conocer las diferentes necesidades que ellos tienen, como se observa en la gráfica el 100 % de los encuestados dijeron que si es necesario dicha herramienta, los cuales manifestaron verbalmente que actualmente no tienen un medio tecnológico y les dificulta la divulgación, como conclusión

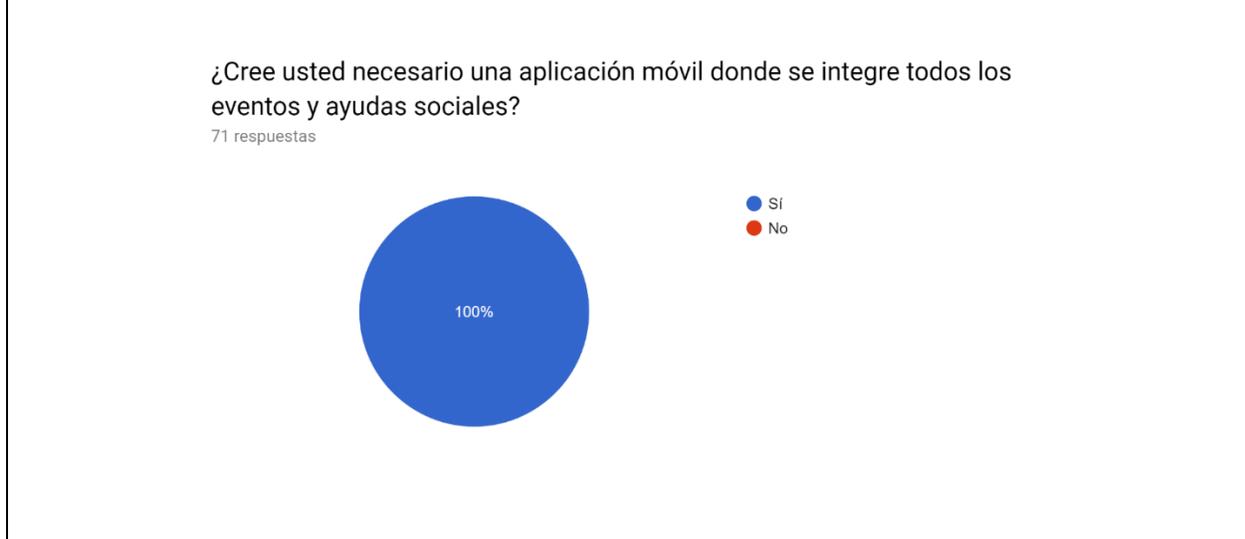
podemos decir que la creación de un medio para difundir las necesidades de los animalistas es aceptada por los protectores de animales.

Figura 8. Encuesta Medio de comunicación



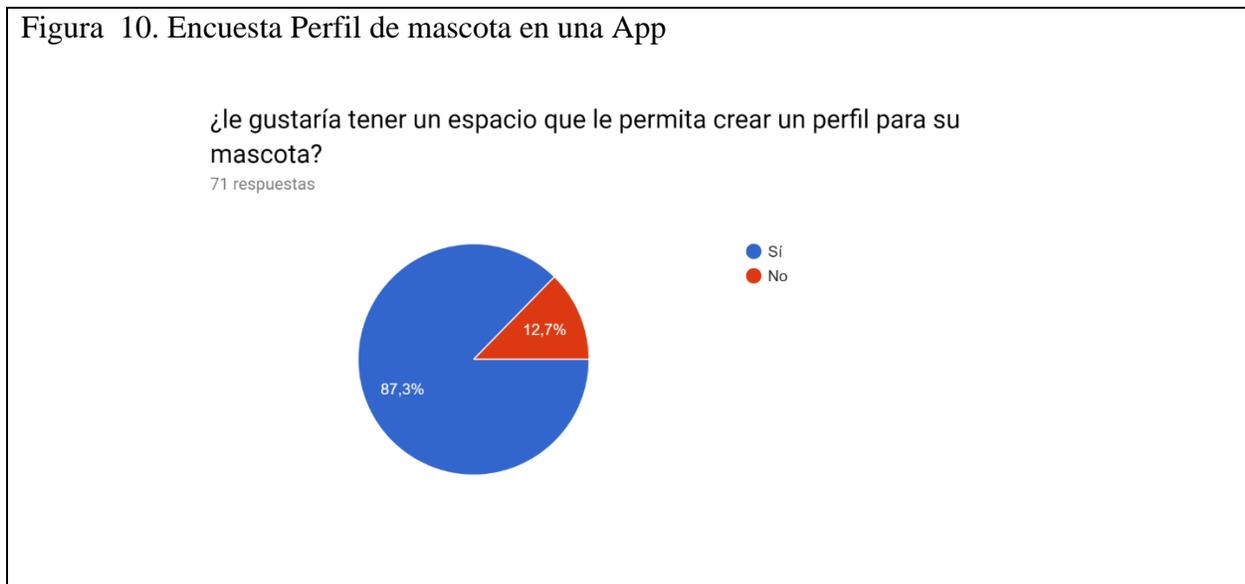
Se preguntó a los encuestados que cual era el medio de comunicación más utilizado para la divulgación de eventos y campañas, tales como animaltón, recolección de alimento, vacunación, esterilización, entre otros, la cual se puede observar en la gráfica que 50 personas de los encuestados utilizan internet, esto equivale a un 70.4% aproximadamente, pero también hay otros que utiliza internet y televisión con 6 personas de los encuestados, esto equivale al 8.5%, seguido por internet y radio con 5 personas de los encuestados, esto equivale al 7 %, como conclusión se evidencia en la gráfica que el internet es el medio de comunicación más utilizado para publicar los eventos, ya que si publican por otro medio lo combinan con el internet, de la población encuestado el 97% utilizan el internet.

Figura 9. Encuesta necesidad de una App



Se les preguntó a las personas encuestadas que si ellos veían necesario o útil una aplicación móvil que integre todos los eventos y ayudas sociales, como se puede observar en la gráfica que el 100% de las personas encuestadas ven la necesidad de tener una aplicación para que les facilite las labores que ellos realizan.

Figura 10. Encuesta Perfil de mascota en una App



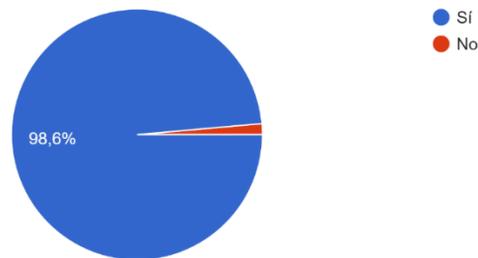
Se le preguntó a las personas encuestadas que si ellos teniendo una aplicación móvil que integre todos los eventos y ayudas sociales, les gustaría tener un espacio para crearle un perfil a las mascotas, donde el 87,3% de los encuestados dijeron que sí, pero el 12,7% dijeron que no, entre

los encuestados y los entrevistados que tenían mascota les parecía que era una buena idea, pero los que no tenían mascota o pertenecen a una protectora de animales decían que no porque no tenían mascota o eran varios y no era lógico registrarse varias veces para crear un perfil para cada animal, como conclusión se puede decir que la gran mayoría de personas están de acuerdo, ya son personas naturales que tienen una sola mascota.

Figura 11. Encuesta espacio de adopciones en la App

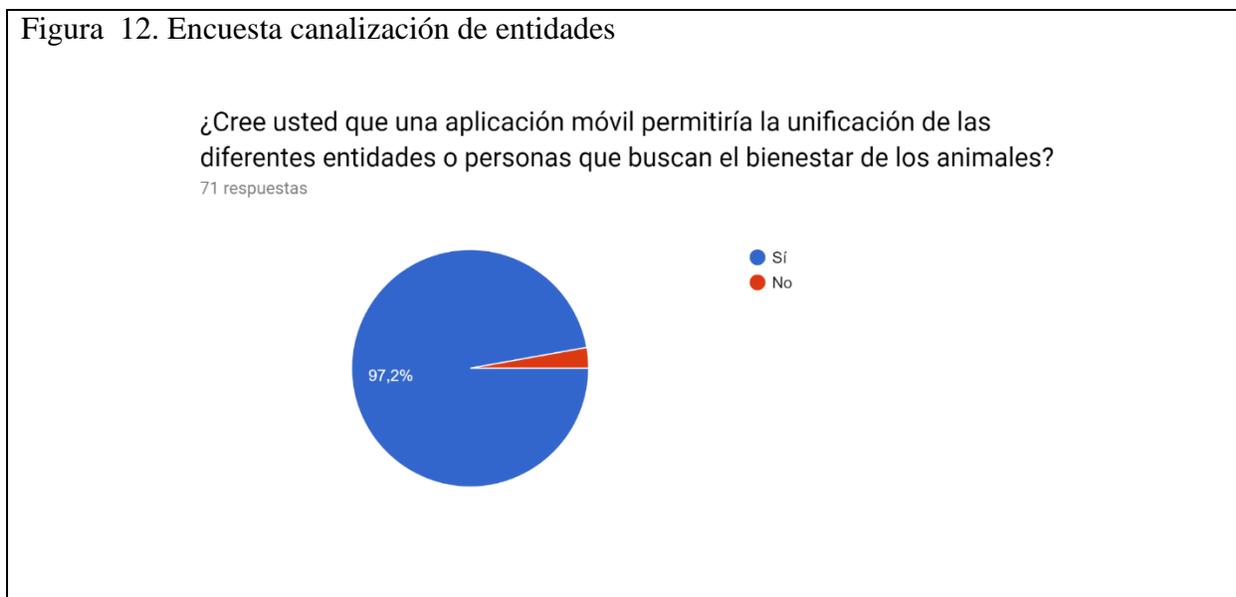
¿Usted cree que sería útil tener un espacio en la aplicación la cual le permita gestionar las adopciones?

71 respuestas



Se le preguntó a las personas encuestadas que si ellos teniendo una aplicación móvil que integre todos los eventos y ayudas sociales de las mascotas, le sería útil tener una opción donde puedan publicar mascotas que estén dando en adopción y a través de esa opción poder obtener el contacto de la persona que está dando en adopción, como se puede observar en la gráfica que 70 de los encuestados están de acuerdo, esto equivale a un 98,6% y una persona no está de acuerdo lo cual equivale al 1,4%, como conclusión de los encuestados y entrevistados les pareció muy buena la opción de tener un medio por el cual se pueda contactar o dar a conocer las adopciones.

Figura 12. Encuesta canalización de entidades



Se les preguntó a las personas encuestadas si una aplicación móvil permitiría integrar o canalizar todas las entidades y/o personas que velan por el bienestar de los animales, de las cuales el 97,2% de las personas encuestadas manifiestan que sí, ya que en la actualidad se encuentran muy dispersos y no existen App móvil que contribuyan con ello, aunque el 2,8% de los encuestados dicen que no.

Una vez analizado los datos que arrojaron las encuestas aplicadas, se puede concluir que es viable desarrollar una aplicación móvil que permita gestionar y canalizar las labores que realizan tanto las personas naturales como los que tienen fundaciones para los animales, ya que a la gran mayoría de los encuestados les pareció interesante la idea de tener un medio tecnológico que unifique cada proceso que realizan los animalistas, por otra parte estas personas manifestaron la falta de herramientas tecnológicas que permitan facilitar las labores que ellos realizan día a día, también se evidencia y lo manifestaron la falta de medios que permitan dar a conocer las fundaciones y sus diferentes necesidades, es por esto que les parece muy interesante y útil la idea.

8.1 Definición de los requerimientos de software

De acuerdo con la caracterización de los procesos animalistas relacionados con los servicios sociales, culturales y comerciales y su posterior validación con las entrevistas y encuestas realizadas en el primer y segundo animaltón “dona de corazón”, se hace la especificación de los

requerimientos funcionales y no funcionales para diseñar los módulos de la arquitectura. Inicialmente a través de la vista de los casos de uso se modela una vista de escenarios para realizar una definición del alcance funcional del producto software en cada uno de los subsistemas funcionales que los constituyen, luego se plantea una arquitectura inicial (basada en la descripción de los casos de uso) y se especifican algunas.

8.1.1 Requerimientos de Software

Para implementar una arquitectura de software enfocada en el desarrollar una solución tecnológica basada en dispositivos móviles para gestionar procesos de bienestar en animales domésticos con condiciones de vulnerabilidad es necesario definir algunas características como:

- El administrador a través de una página web se encargará de monitorear todos los procesos que realicen los animalistas.
- Inicio de sesión para los animalistas.
- Los animalistas tendrán un espacio para la publicación de los eventos, los servicios sociales, adopciones y apareamiento de las mascotas.
- Se debe permitir la visualización de eventos, servicios sociales, adopciones y apareamiento de las mascotas por parte de los animalistas. Por otra parte, los animalistas podrán utilizar geolocalización y códigos QR para realizar consultas de los eventos y adopciones.

Con base en estas características se define los requerimientos funcionales y no funcionales para el diseño de la arquitectura de software para el desarrollo de la solución tecnológica que permita la gestión de procesos en los animales domésticos.

8.1.2 Requerimientos funcionales

Teniendo en cuenta que los animalistas las necesidades que tiene hoy en día, se realiza una descripción de los aspectos más importantes que permitan tener información de las mascotas, se propone el diseño de la arquitectura de software (obedeciendo a las especificaciones de dicha metodología) que permita ejecutar las siguientes funcionalidades:

- Registrar el usuario animalista.
- Iniciar sesión.

- Publicar, visualizar, modificar y eliminar los eventos.
- Publicar, visualizar, modificar y eliminar los servicios sociales.
- Publicar, visualizar, modificar, y cambiar de estado de las adopciones.
- Gestionar gestación.
- Visualizar las publicaciones de otros animalistas, como los eventos, los servicios sociales, adopciones y gestaciones.
- Mostrar los sitios donde se realizarán los eventos por medio de la geolocalización.
- Obtener información de los eventos por medio de códigos QR.

8.1.3 Roles de usuarios

El sistema permite gestionar procesos de bienestar en animales domésticos con condiciones de vulnerabilidad, el cual tiene el siguiente rol:

- **Animalista:** Usuario que se encarga de interactuar con la aplicación móvil.

8.1.4 Requerimientos no funcionales

Las Características que se tendrán en cuenta en el proceso de desarrollo son:

- **Adaptabilidad:** El acceso a los servicios sociales, culturales y comerciales por parte de los animalistas se debe hacer por medio de un dispositivo móvil con sistema operativo IOS y/o Android con conexión a Internet.
- **Experiencia interactiva:** Los animalistas que utilicen los servicios de la app móvil podrán disfrutar de una experiencia interactiva, ya que podrán acceder a espacios asincrónicos como muros, chat y formulario de contacto en los cuales tendrán comentarios de las publicaciones realizadas.
- **Experiencia multiusuario:** El sistema debe permitir que varios usuarios accedan a los servicios definidos, y que a su vez puedan interactuar con los recursos que allí se encuentran.

- **Seguridad:** La arquitectura enfocada a los servicios de la App debe garantizar y validar que solamente el usuario animalista por medio de su nombre de usuario y contraseña sea quien pueda realizar cambios en su perfil.
- **Usable:** Los servicios definidos en la aplicación móvil deben ser de fácil uso para los usuarios animalistas que van a interactuar con los diferentes contenidos sociales, culturales y comerciales.

9. Diseño de la solución tecnológica orientada a servicios que integre los requerimientos del mercado de las mascotas y protección animal

Con base en la caracterización de los procesos animalistas relacionados con los servicios sociales, de adopción, culturales y comerciales, se hizo la especificación de los requerimientos funcionales y no funcionales para diseñar los módulos de la arquitectura propuesta. Inicialmente a través de la vista de los casos de uso se modela una vista de escenarios para realizar una definición del alcance funcional del producto software en cada uno de los subsistemas funcionales que los constituyen, luego se plantea una arquitectura inicial (basada en la descripción de los casos de uso) y se especifican algunas decisiones de diseño para la implementación de la arquitectura final del sistema que está conformada por las cuatro vistas de Kruchten (descritas en 5.4.7), que son:

- **Vista de despliegue:** Compuesta por un diagrama de Nodos y Componentes que explica en detalle cada uno de los servidores (Nodos) que conforman la arquitectura y sus aplicaciones a nivel de software (Componentes).
- **Vista lógica:** Describe tres escenarios para modelar la interacción en el tiempo de los componentes de software por medio de diagramas de secuencia.
- **Vista de procesos:** Describe un diagrama de actividad en el cual se especifica la interacción del usuario animalista con los procesos del software enfocado a los contenidos de la app y los servidores.

9.1 Vista de escenarios de la arquitectura de software

Al contar con un rol definidos en el Sistema (Animalista) y los requerimientos funcionales, se propone en la vista de escenarios de la arquitectura, un diagrama de casos de uso, con su descripción general y la definición de los escenarios para verificar la interacción con la App móvil PetsClub.

Casos de Uso: Animalista

CU_01: CU Gestionar usuario.

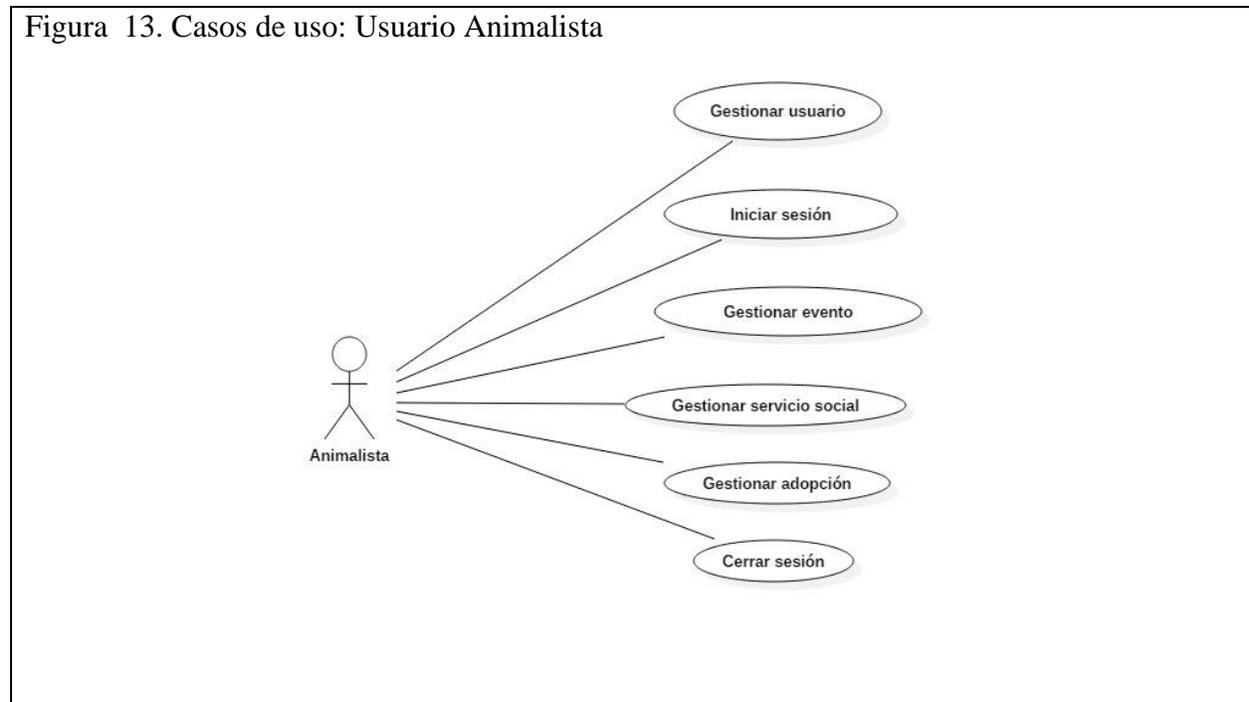
CU_02: CU Iniciar sesión.

CU_03: CU Gestionar evento.

CU_4: CU Gestionar servicio social.

CU_5: CU Gestionar adopción.

CU_6: CU Cerrar sesión.



Caso de uso: Gestionar usuario.

Figura 14. Gestionar usuario

ID:	CU_01
Nombre del Caso de Uso:	CU Gestionar usuario.
Propósito:	Ir a la página principal de app (PetsClub) y registrar un usuario.
Descripción:	Se requiere que la App (PetsClub), permita registrar los datos del Animalista, para poder iniciar sesión.
Actores:	Animalista.
Precondiciones:	.
Postcondiciones de éxito:	Registrar el Animalista correctamente en la App (PetsClub).
Postcondiciones de fracaso:	No registrar el Animalista correctamente en la App (PetsClub).
Casos de uso invocados:	.
Casos de uso invocantes:	.

Escenario: Registrar usuario.

Figura 15. Registrar usuario

Escenario Principal	Escenario de Extensión
1. El Animalista ingresa el correo electrónico en el campo de texto de la interfaz de registro del Animalista.	1.1 El Sistema valido que la información no es correcta MSJ_03.
2. El Animalista ingresa la contraseña en el campo de texto de la interfaz de registro del Animalista.	2.1 El Sistema valido que la información no es correcta MSJ_04.
3. El Animalista da clic en el botón registrar de la interfaz de registro del Animalista.	3.1. El Sistema valido que la información no es correcta MSJ_05.
4. El Sistema valida que la información del Animalista sea la correcta y lo direcciona a otro formulario para completar el registro.	.
5. El Animalista da clic en cambiar imagen para ponerle la foto deseada al perfil.	.
6. El Animalista ingresa el o los nombres en el campo de texto de la interfaz de registro del Animalista.	.
7. El Animalista ingresa el o los apellidos en el campo de texto de la interfaz de registro del Animalista.	.
8. El Animalista ingresa el nombre de usuario en el campo de texto de la interfaz de registro del Animalista.	.
9. El Animalista ingresa la fecha de nacimiento en el campo de texto de la interfaz de registro del Animalista.	.
10. El sistema valida que todos los campos estén diligenciados, de no ser así	.

Figura 16. Mensajes

ID MSJ	Tipo	Mensaje
MSJ_03	Validador	Introduzca una dirección de correo electrónico válido.
MSJ_04	Validador	La contraseña debe tener al menos cinco caracteres.
MSJ_05	Validador	La dirección de correo electrónica ya está en uso por otra cuenta.
MSJ_06	Validador	No hay registro de usuario correspondiente a este identificador. El usuario puede haber sido eliminado.

La descripción de los casos de uso restantes para el animalista se encuentra en el anexo A

9.2 Arquitectura propuesta para el despliegue de servicios animalistas

La arquitectura del sistema para la oferta de servicios sociales, culturales y comerciales de los procesos de bienestar en animales domesticos consta de una vista de despliegue, compuesta de un diagrama de nodos y componentes, en la que se detalla cada uno de los módulos de la arquitectura, sus componentes y funciones. Se especifica también tres escenarios que permiten modelar una vista lógica de la arquitectura por medio de diagramas de secuencia para los roles administrador y animalista, y un diagrama de comunicación (enfocada en la aplicación móvil híbrida para los procesos que ejecuta el animalista), en los que se analiza el comportamiento dinámico del sistema. Finalmente se propone una vista de procesos utilizando un diagrama de actividad, con el fin de analizar el comportamiento de los componentes de software de la aplicación móvil, desde la perspectiva del usuario y los servidores que conforman el sistema.

9.2.1 Criterios de diseño de la arquitectura para el despliegue de servicios animalistas en las aplicaciones móviles híbrida

Los criterios de diseño de la arquitectura propuesta para el despliegue de servicios relacionados con los procesos de bienestar en animales domésticos se determinan a partir de la especificación de requerimientos y el análisis de los casos de uso, para garantizar soporte de los procesos administrativos, de despliegue de contenidos sociales, culturales y comerciales, es necesario definir los siguientes aspectos:

- **Escalabilidad del sistema:** El aumento constante e impredecible del tráfico (carga) a los servicios y negocios disponibles a través de Internet es cada vez mayor. Para asegurar un funcionamiento continuo y adecuado no basta con adquirir componentes de hardware cada vez más grandes y caros, sino que se requiere de una arquitectura particular. Para que un sistema logre ser escalable se requiere pensar cuidadosamente su diseño desde el principio de su desarrollo. Una arquitectura escalable es aquella que posibilita incrementar el rendimiento sin necesidad de rediseñarla, aprovechando recursos que se le añadan.
La representación de la arquitectura propuesta para el despliegue de servicios relacionados con los procesos animalistas y sus componentes deben ser concebidos para un sistema escalable y susceptible a modificaciones en el tiempo, sin que esto afecte el funcionamiento de dicho sistema. Teniendo en cuenta la complejidad que implicaría más adelante servicios de uso masivo y con interacción en tiempo real para este tipo de comunidades.
- **Pertinencia entre Nodos de la arquitectura y Tecnologías de apache cordova:** La comunicación entre la aplicación web y la aplicación nativa, normalmente ocurre mediante JavaScript APIs que nos dan la posibilidad de leer datos adicionales del móvil, y lanzar acciones como la vibración, el acelerómetro o acceder a ficheros de un modo estándar a través de diferentes plataformas, por tal motivo es necesario integrar la tecnología apache cordova para garantizar acceso a los diferentes recursos de hardware del dispositivo móvil,
- **Enfoque de la arquitectura orientada a servicio:** La arquitectura debe tener un enfoque orientado a servicios (Teniendo en cuenta que para el acceso de los animalistas se deben consumir los contenidos de un Web Service), para eso se debe hacer uso de la tecnología REST, por lo que cada servicio de los procesos animalistas se conciba como un recurso

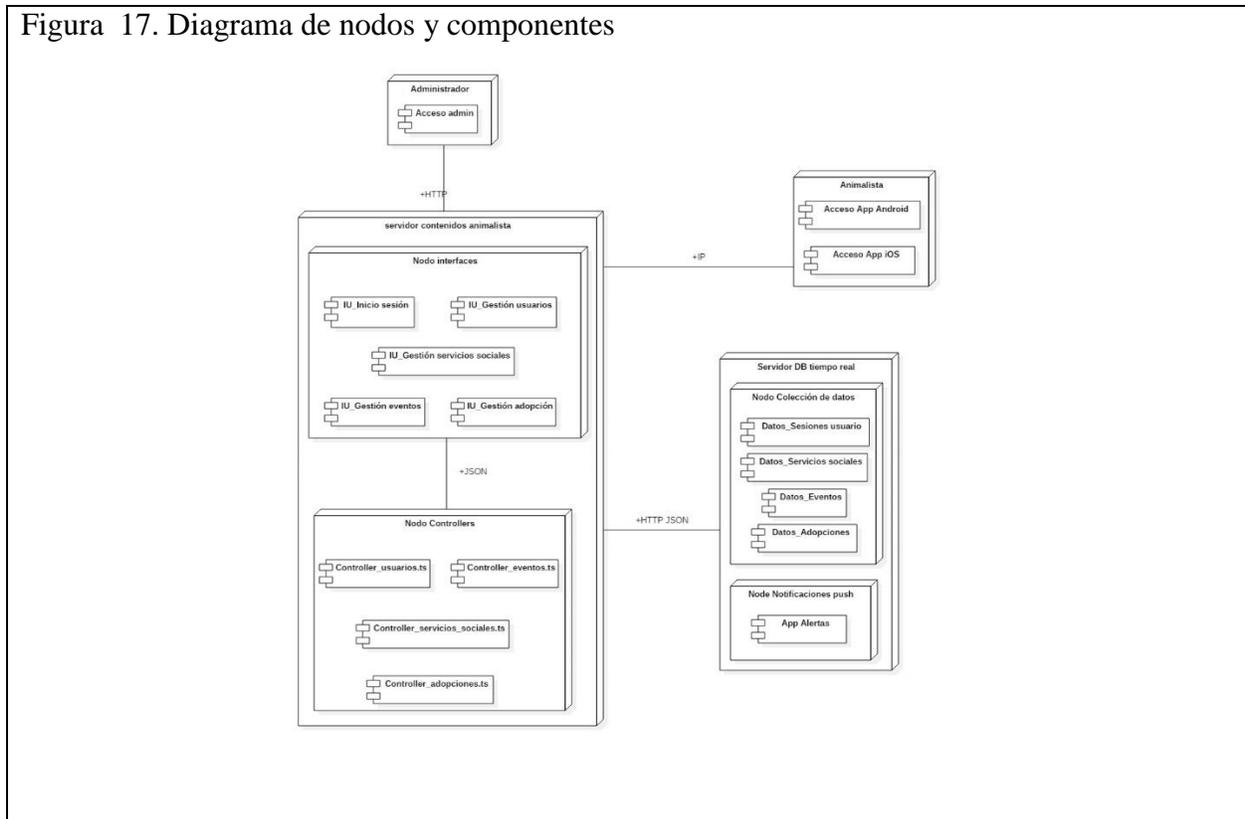
que puede ser consumido por medio de peticiones GET y POST. Las ventajas que aportan el uso de esta tecnología en el diseño de la arquitectura son:

- **Separación Cliente-servidor:** Una interfaz uniforme separa los clientes de los servidores, facilitando la portabilidad del código del cliente y simplificando los servidores, haciéndolos más escalables y reemplazables, manteniendo siempre la interfaz.
- **Ausencia de estado:** Los servicios sin estado son mucho más simples de diseñar, escribir y distribuir a través de múltiples servidores. Un servicio sin estado no solo funciona mejor, sino que además mueve la responsabilidad de mantener el estado al cliente de la aplicación. En un servicio web REST, el servidor es el responsable de generar las respuestas y proveer una interfaz que le permita al cliente mantener el estado de la aplicación por su cuenta.
- **Sistema de capas:** Un cliente no sabe, normalmente, si está conectado directamente al servidor final o a un intermediario. Los servidores intermedios pueden contribuir a la escalabilidad de la aplicación proporcionando balanceo de carga o cachés compartidos.
- **Interfaz uniforme:** Permite simplificar y desarrollar la infraestructura de forma independiente.
- **Escalabilidad de la interacción con los componentes:** La Web ha crecido exponencialmente sin degradar su rendimiento. Una prueba de ellos es la variedad de clientes que pueden acceder a través de la Web como las estaciones de trabajo, sistemas industriales, dispositivos móviles, etc.

9.2.2 Vista de despliegue de la arquitectura

En la figura 17, se muestra un diagrama de nodos y componentes de la arquitectura para la oferta de servicios para el sector animalista. Cada componente es un módulo de software ejecutable.

Figura 17. Diagrama de nodos y componentes



Las tareas que componen cada bloque funcional son:

- **Servidor de contenidos Animalista**

En este servidor se especifican las aplicaciones que contienen los recursos relacionados con los procesos de bienestar para animales domésticos y que serán consumidos por medio de servicios Web, con el fin de lograr una interacción de los entornos de la App móvil híbrida con los usuarios. Lo conforma un nodo de interfaces y un nodo de controllers.

- **Nodo de interfaces:** Contiene las interfaces graficas de usuario para el envío y recepción de contenidos relacionados con los procesos animalistas. Los datos definidos en las interfaces son recibidos por los diferentes controllers para su procesamiento con las bases de datos en tiempo real. Está conformada por los siguientes componentes de software HTML5 para iniciar sesión, gestionar usuarios, adopciones, servicios sociales y eventos.

- **Nodo Controllers:** Contiene los módulos que permiten interactuar con las interfaces de usuario. Los controllers nos permiten mediante programación implementar la lógica de la presentación en la solución móvil híbrida. En ellos podemos mantener el código necesario para inicializar una aplicación, gestionar los eventos, etc. Podemos decir que gestionan el flujo de la parte del cliente, lo que sería programación para implementar la funcionalidad asociada a la presentación. Está conformada por componentes de software TypeScript para iniciar sesión, gestionar usuarios, adopciones, servicios sociales y eventos.
- **Servidor de Bases de datos en tiempo real**

En este servidor se estructura el modelo de datos relacionado con las colecciones de datos para procesos de gestión de usuarios, eventos, adopciones y servicios sociales. Dichas colecciones serán consumidas en tiempo real por los usuarios desde las aplicaciones móviles híbridas. Los conforma un nodo de colecciones de datos y un nodo de notificaciones PUSH.

 - **Nodo colecciones de datos:** Consta de unos documentos que se almacenan en colecciones. Los documentos contienen subcolecciones y objetos anidados, y ambos incluyen campos primitivos como Strings y tipos de objetos complejos como listas. Los datos de las colecciones se leen por medio de los controladores y son visualizados en tiempo real por los usuarios animalistas que interactúan con la aplicación móvil híbrida. Está conformada por componentes como sesiones de usuario, servicios sociales, adopciones y eventos.
 - **Nodo notificaciones PUSH:** Este nodo consta de una solución de mensajería multiplataforma que permite enviar mensajes de forma segura y gratuita. Con FCM, permite notificar a una app cliente que un correo electrónico nuevo o que otros datos están disponibles para la sincronización. Este nodo para poder desplegar mensajes a los usuarios finales por medio de la interfaz de la app híbrida, se sincroniza por medio de una Api Rest que se ejecutan desde los diferentes controllers.
- **Usuario administrador:** Se encarga de interactuar con el servidor de contenidos animalista para gestionar procesos relacionados con usuarios, eventos, adopciones y

servicios sociales. El escenario de navegación del administrador será por medio de una aplicación web.

- **Usuario animalista:** Se encarga de interactuar con el servidor de contenidos animalista para interactuar con eventos, adopciones y servicios sociales, además de visualizar notificaciones PUSH de situaciones relevantes que se presenten en este tipo de comunidad. El escenario de navegación del administrador será por medio de una aplicación web.

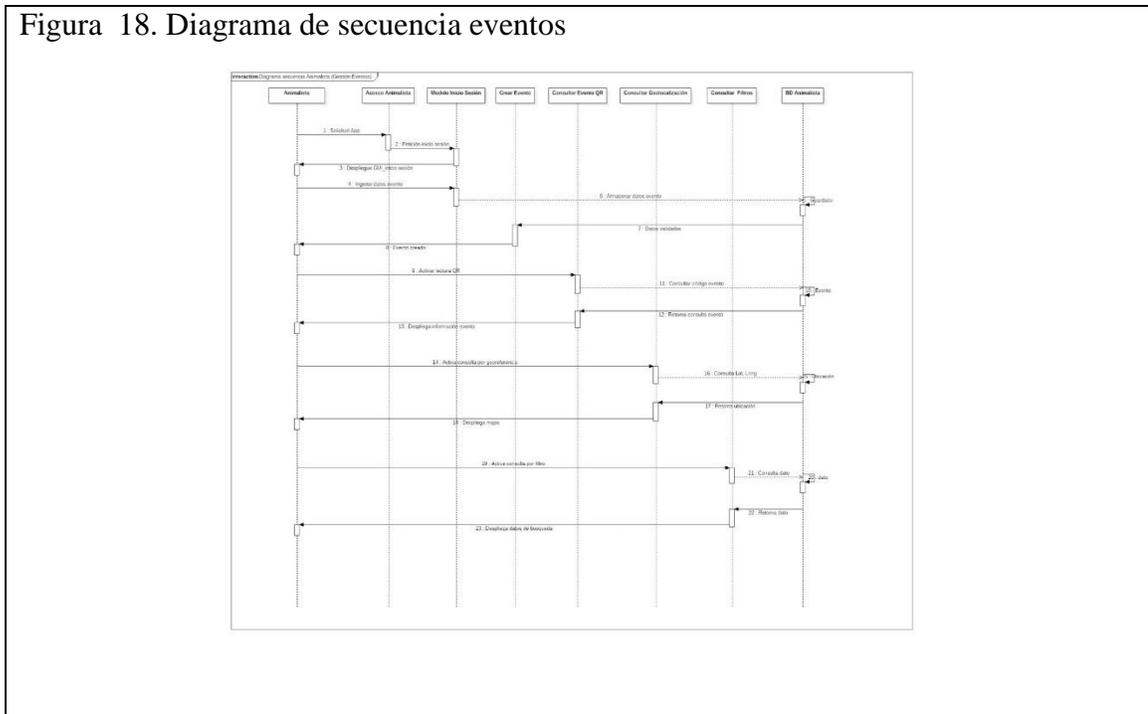
9.2.3 Vista lógica de la arquitectura de software propuesta

La vista lógica de la arquitectura del sistema se describe a partir de los escenarios de interacción del administrador y animalista con los componentes descritos en la vista de despliegue (Diagrama de nodos y componentes). Se proponen para esta vista, tres diagramas de secuencia (para un análisis dinámico de los objetos que se asocian a las aplicaciones más relevantes de los animalistas) y un diagrama de comunicación para la aplicación de los servicios animalistas con los que va a interactuar este tipo de usuario.

9.2.3.1 Diagrama de secuencia eventos

El animalista tiene una interacción directa con el servidor contenido. Inicialmente realiza el proceso de autenticación por medio del inicio de sesión, se hace la respectiva validación y se ejecutan las acciones de software relacionadas con la creación de los eventos y las consultas por QR, por geolocalización y por filtros. La información es verificada en las colecciones de datos de la base de datos en tiempo real y retornadas al usuario por medio de la interfaz móvil. En la figura 18, se puede apreciar el diagrama de secuencia para los eventos del animalista.

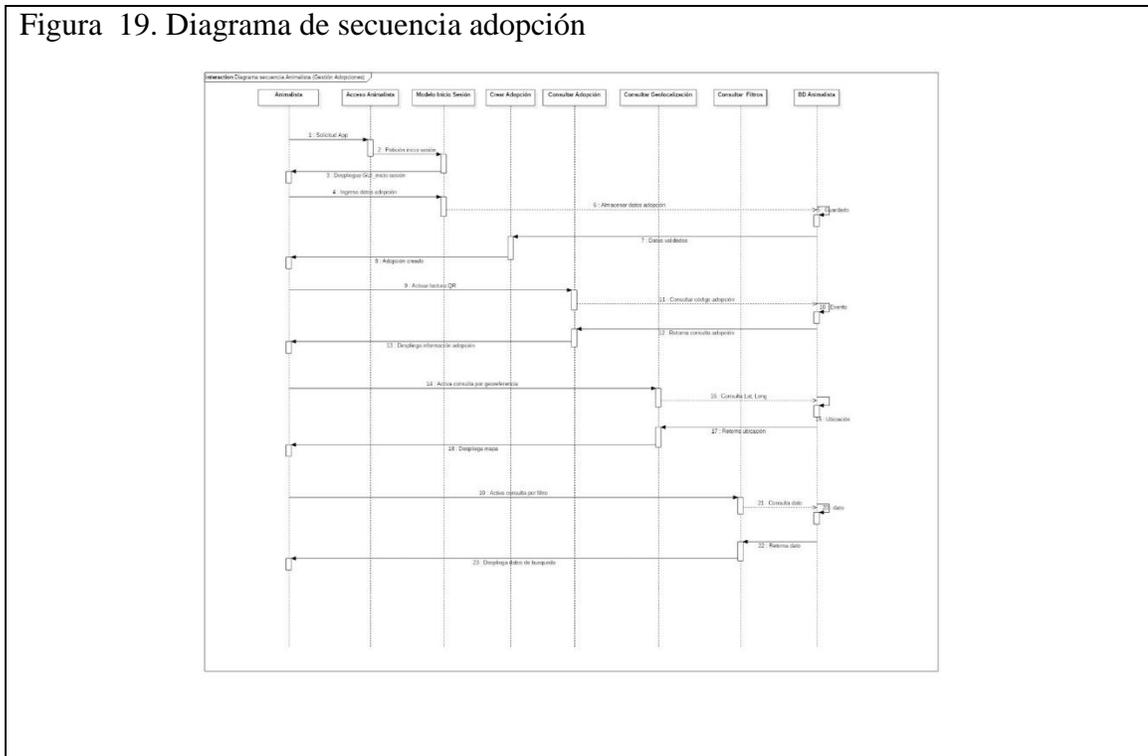
Figura 18. Diagrama de secuencia eventos



9.2.3.2 Diagrama de secuencia procesos de adopción

El animalista tiene una interacción directa con el servidor contenido. Inicialmente realiza el proceso de autenticación por medio del inicio de sesión, se hace la respectiva validación y se ejecutan las acciones de software relacionadas con la creación de las adopciones y las consultas por QR, por geolocalización y por filtros. La información es verificada en las colecciones de datos de la base de datos en tiempo real y retornadas al usuario por medio de la interfaz móvil. En la figura 19, se puede apreciar el diagrama de secuencia para las adopciones del animalista.

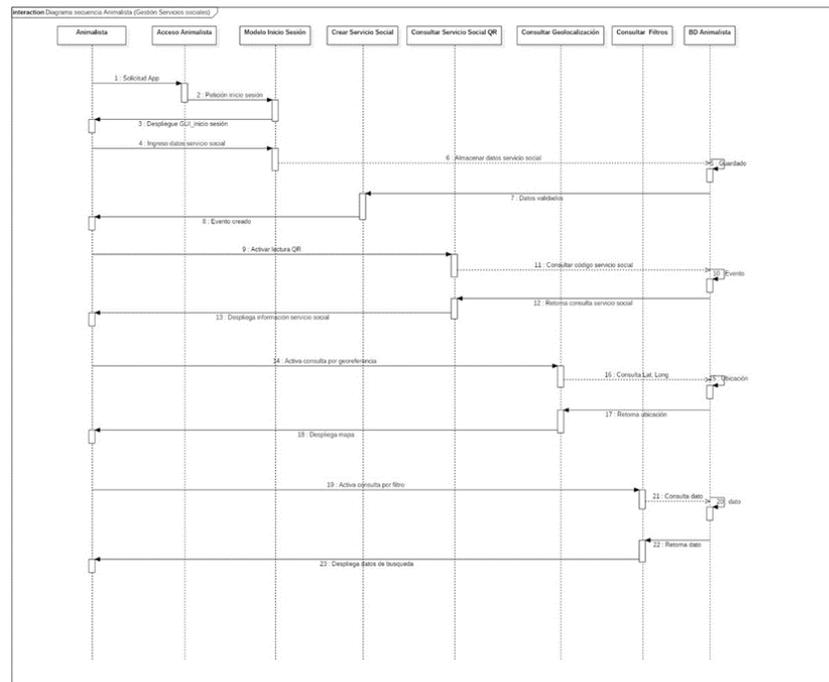
Figura 19. Diagrama de secuencia adopción



9.2.3.3 Diagrama de secuencia de servicios sociales

El animalista tiene una interacción directa con el servidor contenido. Inicialmente realiza el proceso de autenticación por medio del inicio de sesión, se hace la respectiva validación y se ejecutan las acciones de software relacionadas con la creación de servicios sociales y las consultas por QR, por geolocalización y por filtros. La información es verificada en las colecciones de datos de la base de datos en tiempo real y retornadas al usuario por medio de la interfaz móvil. En la figura 20, se puede apreciar el diagrama de secuencia para los servicios sociales.

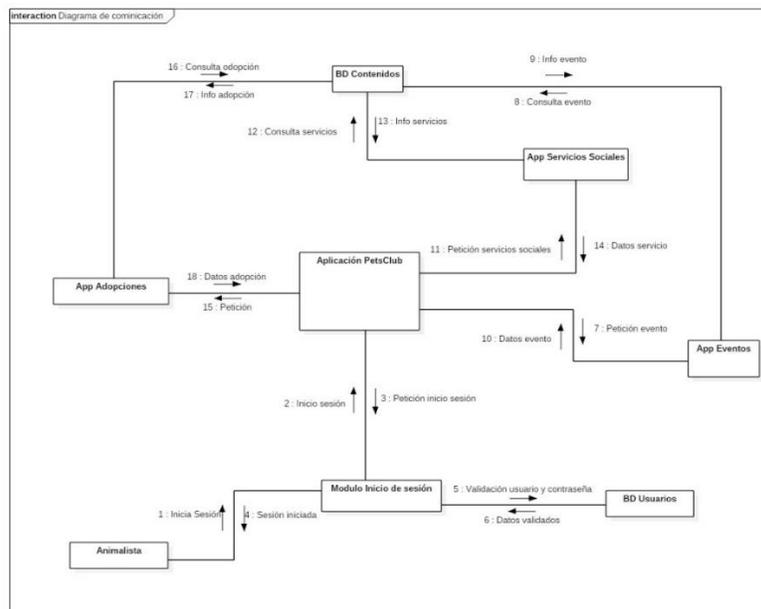
Figura 20. Diagrama de secuencia servicios sociales



9.2.3.4 Diagrama de comunicación para la aplicación móvil del animalista

Para el diagrama de comunicación se define el mismo escenario de la aplicación para el rol del animalista, con el fin de modelar la interacción entre los diferentes objetos (PetsClub) que se producen mediante mensajes en secuencia, es decir, que se pueden observar como los mensajes se pasan los objetos entre sí y en qué orden. Con este diagrama se obtiene una representación espacial de los componentes de software de la arquitectura con relación a la vista lógica. En la figura 21, se muestra el diagrama de comunicación para la aplicación del animalista.

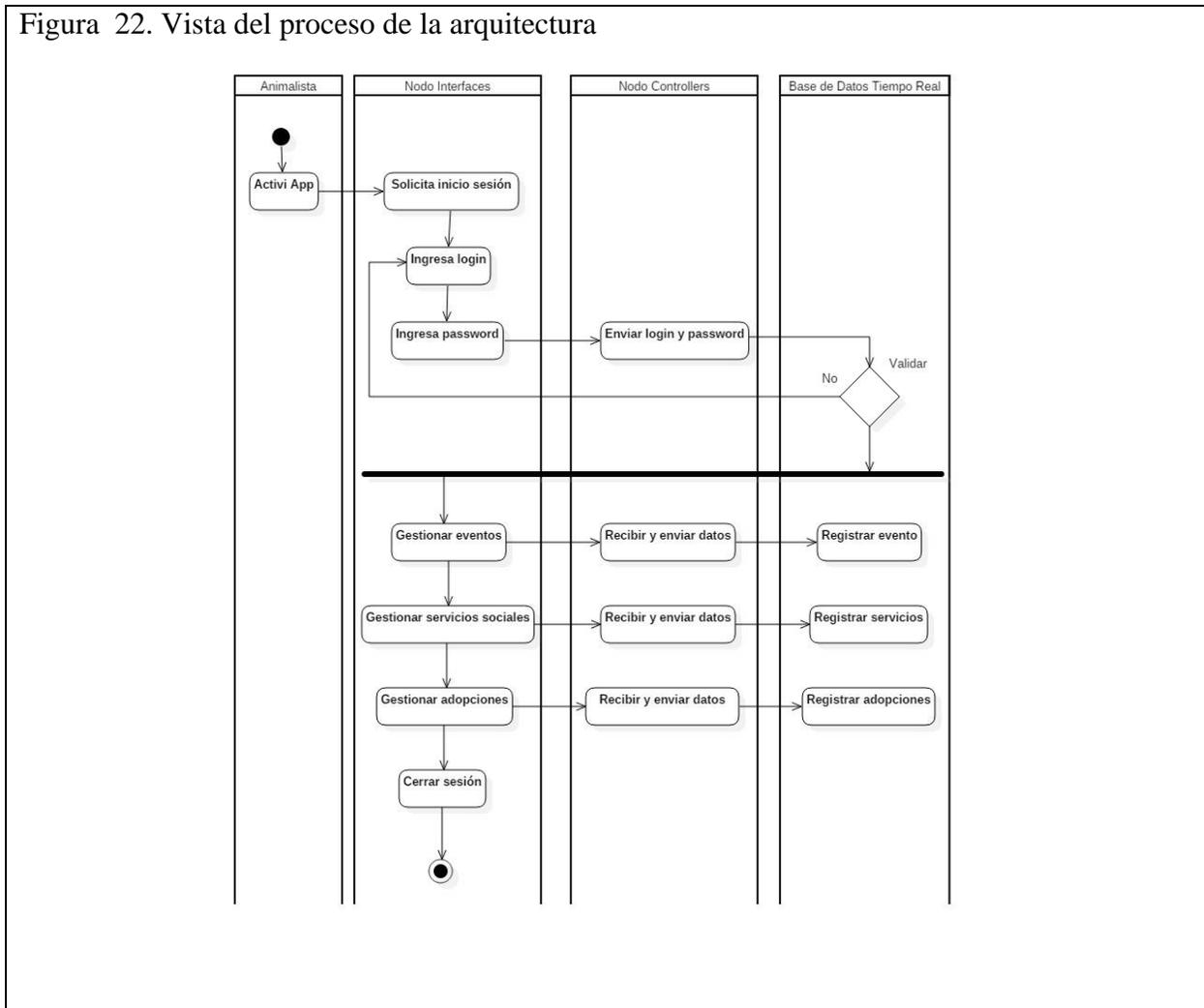
Figura 21. Diagrama de Comunicación



9.2.4 Vista de procesos de la arquitectura

La vista de procesos de la arquitectura se describe por medio de un diagrama de actividad que permite representar la App móvil híbrida como un flujo de trabajo por medio de una serie de acciones de software ejecutadas por el usuario animalista y soportadas por los componentes de dicha aplicación, que se ejecutan desde los servidores de contenidos y de bases de datos en tiempo real, que fueron descritos en el diagrama de Nodos y componentes. En la figura 22, se muestra el diagrama de actividad para la aplicación del rol animalista.

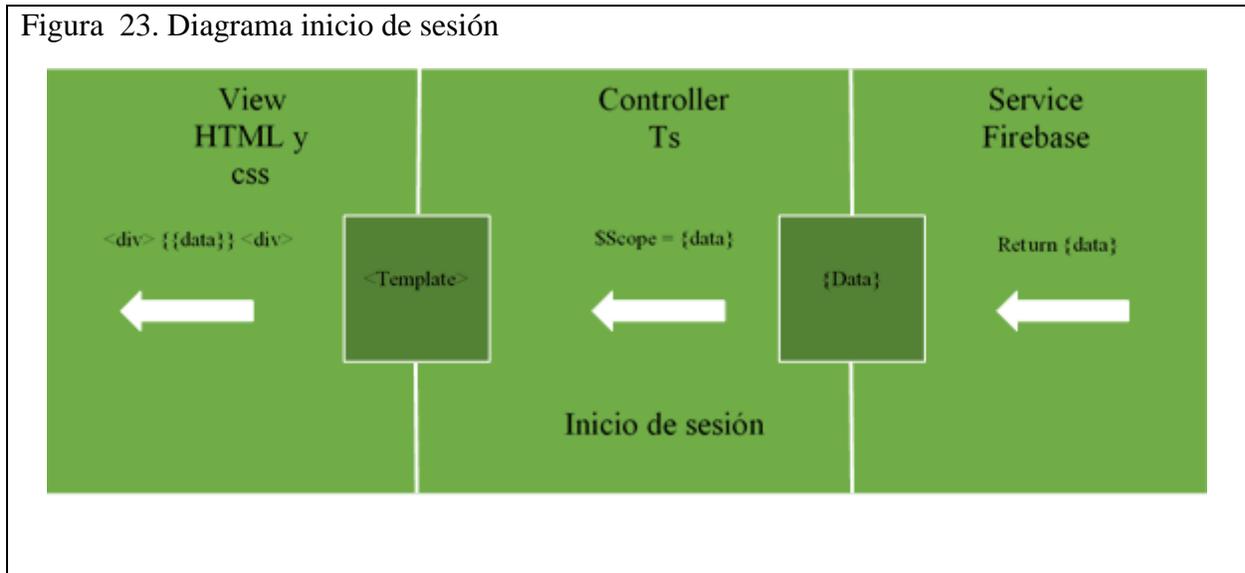
Figura 22. Vista del proceso de la arquitectura



10. Implementación de la solución tecnológica con relación a la usabilidad de sus interfaces y tiempos de respuesta.

La solución tecnológica está enfocada en una aplicación móvil desarrollada con el Framework de Ionic, el cual permite crear aplicaciones híbridas en un solo desarrollo para sistemas operativos móviles como Android, iOS y web. Este Framework está basado en apache cordova, TypeScript, HTML y css para la parte visual. En la Figura 23, se detallan los aspectos de desarrollo relacionados con la vista que define la interfaz de la aplicación, el controlador que se encarga de comunicarse con la colección de datos y el servicio que tiene el almacenamiento de la información relacionada con los diferentes procesos de la aplicación.

Figura 23. Diagrama inicio de sesión



En diagrama anterior se puede ver que en la vista de la aplicación esta Ionic con HTML y css, en el controlador esta los métodos que permiten interactuar con la base de datos y en la base de datos esta la colección en Firestore.

El módulo inicio de sesión fue construido con una aplicación móvil híbrida en Ionic, utilizando en la vista etiquetas HTML como ion-content, ion-container, ion-button, ion-row, ion-grid y CSS para mejorar la parte visual, en la figura 24, se puede apreciar las etiquetas utilizadas para el inicio de sesión.

Figura 24. Inicio sesión HTML

```
<ion-item class="item-login" >
  <ion-label floating color="secondary">Correo electrónico</ion-label>
  <ion-input type="text" formControlName="email" color="secondary"></ion-input>
</ion-item>
  Caja de texto para ingresar el correo electrónico
<ion-item class="item-login">
  <ion-label floating color="secondary">Contraseña</ion-label>
  <ion-input type="password" formControlName="password" class="form-controll" required></ion-input>
</ion-item>
  Caja de texto para ingresar la contraseña, de tipo password
```

En el controlador se utilizó TypeScript embebido dentro de Ionic, se importaron las librerías nativas de Firebase, y se creó la configuración que permite interactuar con la base de datos, en la figura 25 se puede apreciar la configuración que permite la comunicación con Firebase.

Figura 25. Configuración desde Firebase

```
export const environment = {
  production: false,
  firebase: {
    apiKey: "AIzaSyBaTiH-RiQu1L2ttN9c3P_Jeygle6jX2QY",
    authDomain: "petsclub-2e596.firebaseio.com",
    databaseURL: "https://petsclub-2e596.firebaseio.com",
    projectId: "petsclub-2e596",
    storageBucket: "petsclub-2e596.appspot.com",
    messagingSenderId: "106688202660"
  }
};
```

Código obtenido desde el Firebase

Para la colección de datos se habilitó el inicio de sesión desde el Firebase, para esta base de datos no existe tablas ni filas, por lo cual se creó los datos en documentos que se organizan en colecciones, en la figura 26 se puede apreciar cómo se visualiza los usuarios registrados en el Firebase.

Figura 26. Visualización usuarios autenticados

Usuarios autenticados en Firebase

Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
admin@petsclub.com	✉	17 ene. 2019	17 ene. 2019	9FAwwLzKuAfoOELoRseJJqXSI1o1
alexander.agredo.franco@g...	✉	27 ene. 2019	28 ene. 2019	JCY9bg5cDaf2tx59xEAjlxGMzSY2
jhonfregom@gmail.com	✉	30 ene. 2019	30 ene. 2019	NMDCvkv5mvQoQaxIm6WOXTKzr...
daniela@gmail.com	✉	28 ene. 2019	28 ene. 2019	evBxrGEIZUTUCaDKKmbLm5ZyyFr2
m@gmail.com	✉	27 ene. 2019	29 ene. 2019	hMZqmjNyUxeL2xbDc0DVQs6Ohx...
j@gmail.com	✉	26 ene. 2019	30 ene. 2019	oBCenTWhQuWkJA0n6JB3QMdB...
mahecha_17@hotmail.com	✉	27 ene. 2019	30 ene. 2019	vUJBpu6vqwMrDukrwygTus59Np32

Filas por página: 50 1-7 de 7 < >

La aplicación en ejecución para el inicio de sesión se puede observar de la siguiente manera, ver figura 27.

Figura 27. Interfaz inicio sesión

Ingresar

Pets Club

Correo electrónico

Contraseña

INGRESAR

ó

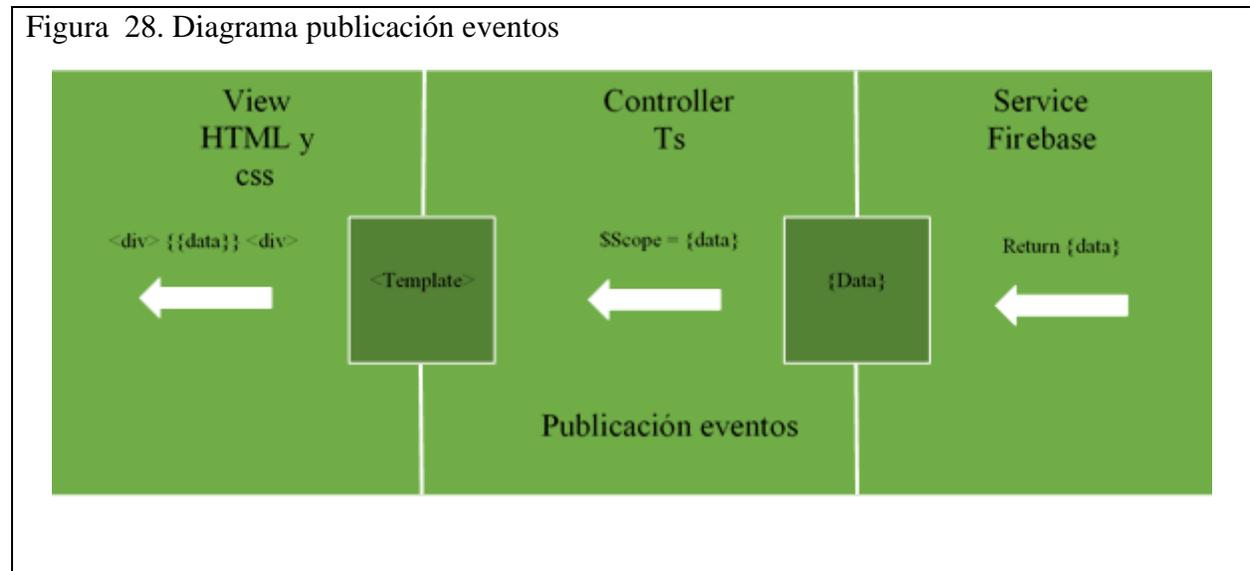
Iniciar sesión con

FACEBOOK GOOGLE

No tienes cuenta? [Regístrate aquí.](#)

Olvidaste tu contraseña? [Restaurar aquí.](#)

En el módulo de publicación de eventos maneja una vista en Ionic basado en HTML y css el cual se conecta por medio de un JSON al controlador, el controlador que está desarrollado en TypeScript, se conecta con el Firebase por medio de Http, en la figura 28 se puede apreciar el bloque de componentes, del módulo de publicación de eventos.



El módulo de publicación de eventos se desarrolló en Ionic el cual es una aplicación híbrida, utilizando en la vista etiquetas HTML como `ion-col`, `ion-thumbnail`, `img`, `ion-textarea` y `css` para mejorar la experiencia visual, esta vista recibe respuesta JSON la cual permite visualizar los datos almacenado en la colección, en la figura 29 se puede apreciar algunas de las etiquetas utilizadas para la vista de los eventos.

Figura 29. Etiquetas HTML publicación eventos

```

<ion-row no-padding>
  <ion-col no-padding col-6 offset-3>
    <ion-thumbnail item-start>
      
    </ion-thumbnail>
  </ion-col>
</ion-row>
<ion-col>
  <button class="take-picture" ion-button round outline block default (click)="openImagePicker()">Subir Foto</button>
</ion-col>
</div>
<form class="new-user-form" [formGroup]="validations form" (ngSubmit)="onSubmit(validations form.value)">
  <ion-item>
    <ion-label floating color="secondary">Evento</ion-label>
    <ion-input type="text" formControlName="nameevent" class="form-control" required</ion-input>
  </ion-item>
  <ion-item>
    <ion-label floating color="secondary">Fecha evento</ion-label>
    <ion-input type="date" formControlName="date" class="form-control" required</ion-input>
  </ion-item>

```

Imprimir imagen obtenida

Botón para agregar imagen

Caja de texto para el nombre del evento

Caja de texto para el nombre del evento

En el controlador de los eventos se utilizó TypeScript embebido en Ionic, se creó un método llamado `getImageFile` el cual permite cargar las imágenes que se encuentran en el dispositivo y así poderlas utilizar a la hora de publicar los eventos, por otra parte se instalaron y se importaron librerías de la api de Google, para implementar la geolocalización, la cual se creó un método llamado `initMap` donde permite poner el marcador en la ubicación deseada, esto lo hace a través de unas funciones que se encuentran dentro del método, en la figura 30 se puede apreciar el método que permite inicializar el mapa y colocar un marcador.

Figura 30. Método geolocalización

```

@ViewChild('map') mapElement: ElementRef;

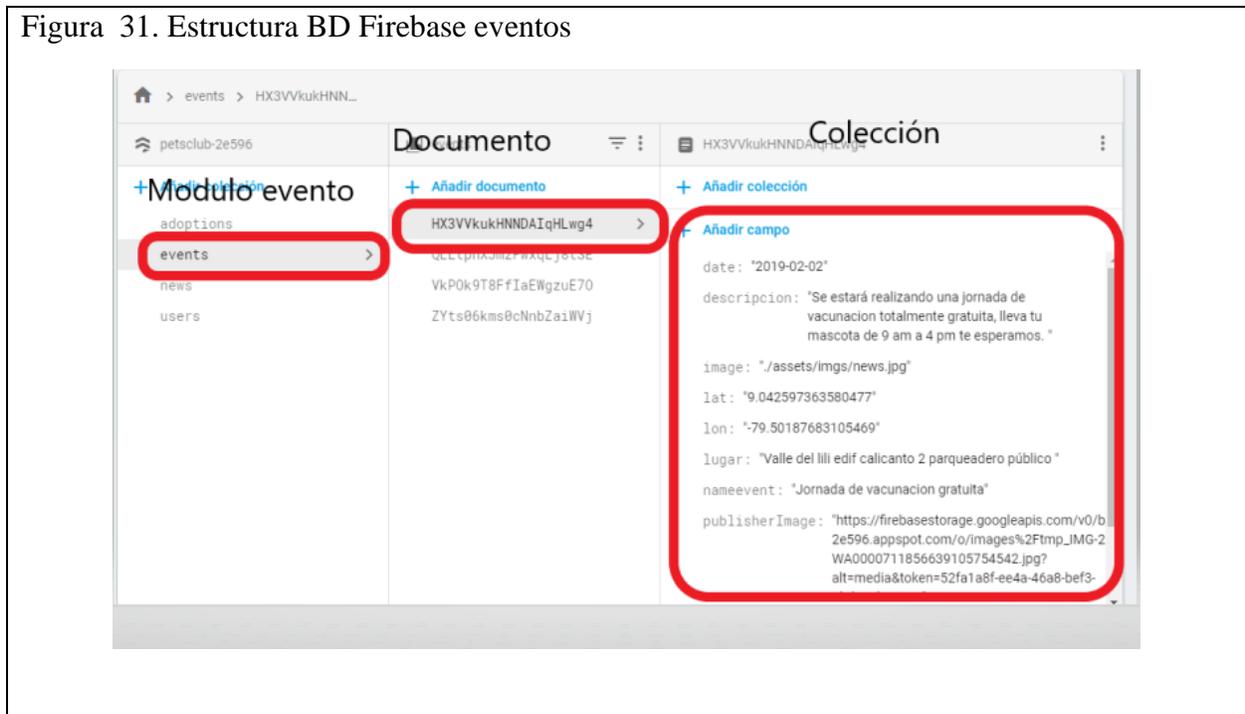
initMap() { | Método para iniciar el mapa
  var markers = []; | Arreglo de marcadores
  localStorage.setItem("lat", ""); | Almacena latitud
  localStorage.setItem("lng", ""); | Almacena longitud

  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function showPosition(pos) {
      var x = new google.maps.Map(document.getElementById('map'), {
        center: { lat: pos.coords.latitude, lng: pos.coords.longitude },
        zoom: 15
      });
      // Obtiene la posición actual del dispositivo
      // Colocar el Marcador
      x.addListener('click', function (e) { ... Pinta el marcador
    });
  });
}

```

Para la colección de datos de los eventos se creó un documento, realizó una estructura de los datos empezando por la referencia del usuario que está haciendo la publicación, posteriormente se crea un documento para los eventos y este a su vez está contenido por una colección, la cual contiene todos los campos requeridos para realizar la publicación, en la figura 31 se puede apreciar la estructura que tiene base de datos de los eventos.

Figura 31. Estructura BD Firebase eventos



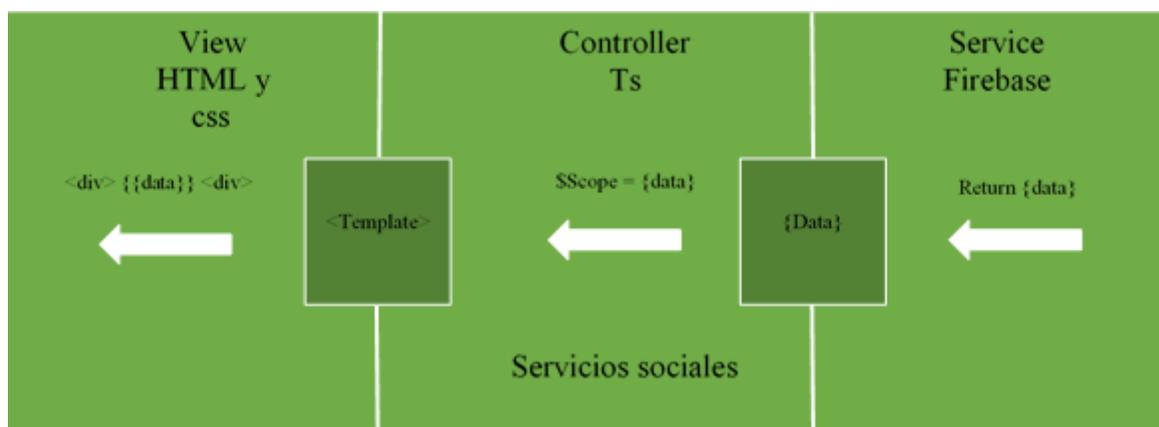
Una vez este la aplicación en funcionamiento, se verá la página de los eventos de la siguiente manera, ver figura 32.

Figura 32. Interfaz de eventos



En el módulo de servicios sociales maneja una vista en Ionic basado en HTML y css el cual se conecta por medio de un JSON al controlador, el controlador que está desarrollado en TypeScript, se conecta con el Firebase por medio de Http, en la figura 33 se puede apreciar el bloque de componentes, del módulo de publicación de eventos.

Figura 33. Diagrama servicios sociales



El módulo de servicios sociales se desarrolló en Ionic, la cual es una aplicación híbrida, para la implementación de esta interfaz se utilizó en la vista etiquetas HTML como ion-card, ion-avatar, img, ion-note, *ngFor para obtener los datos y css para mejorar la experiencia visual, esta vista recibe respuesta JSON la cual permite visualizar los datos almacenado en la colección, en la figura 34 se puede apreciar algunas de las etiquetas utilizadas para la vista de los servicios sociales.

Figura 34. Imprimir servicios

```
<ion-list *ngSwitchCase="'views'">
  <ion-list *ngFor="let newa of news | async; trackBy: trackByFn">
    <ion-card >
      <ion-item (click)="viewProfile(newa)">
        <ion-avatar item-start>
          <img [src]="newa.publisherImage">
        </ion-avatar>
        <p>{{newa.username}}</p>
        <h2> {{newa.titlenew}}</h2>
      </ion-item>
      <img [src]="newa.image">
      <ion-card-content>
        <p>{{newa.news}}</p>
      </ion-card-content>
    </ion-card >
  </ion-list >
</ion-list >
```

Se recorre los datos que viene desde el Firebase y se almacena en un arreglo

Imprime los datos

En el controlador de los servicios sociales se utilizó TypeScript embebido en Ionic, se crearon unos providers para definir la lógica de negocio, con el objetivo de que sea reutilizable e independizar las vistas, en estos providers se encuentra definido el método que permite crear y visualizar los servicios sociales, dentro del método se encuentran las funciones que permiten capturar los datos solicitados para poder crear el método, en la figura 35 se puede apreciar el método.

Figura 35. Método para agregar datos

```

createNews(value) {
  return new Promise<any>((resolve, reject) => {
    let currentUser = firebase.auth().currentUser;
    this.afs.collection('news').add({
      titlenew: value.titlenew,
      news: value.news,
      time: new Date().getTime(),
      image: value.image,
      uid: currentUser.uid,
      username : localStorage.getItem("AuthUSR"),
      publisherImage : localStorage.getItem("AuthUSRImage"),
      newsid : localStorage.getItem("AuthNEWS"),
    })
    .then(
      res => resolve(res),
      err => reject(err)
    )
  })
}

```

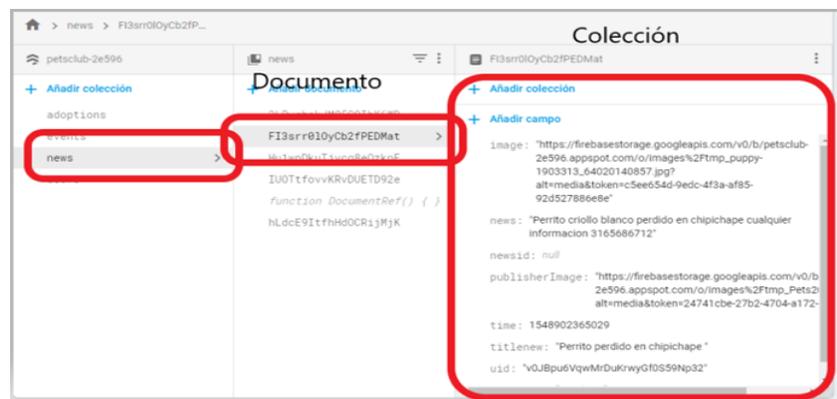
Obtener el usuario que inicio sesión

Agregar datos a la base de datos

Datos agregar

Para la colección de datos de los servicios sociales maneja una estructura igual a la de los eventos, la cual consiste en crear un documento de los servicios sociales y este a su vez está contenido por una colección, la cual contiene todos los campos requeridos para realizar la publicación, en la figura 36 se puede apreciar la estructura que tiene base de datos de los servicios sociales.

Figura 36. Estructura BD Firebase servicios sociales



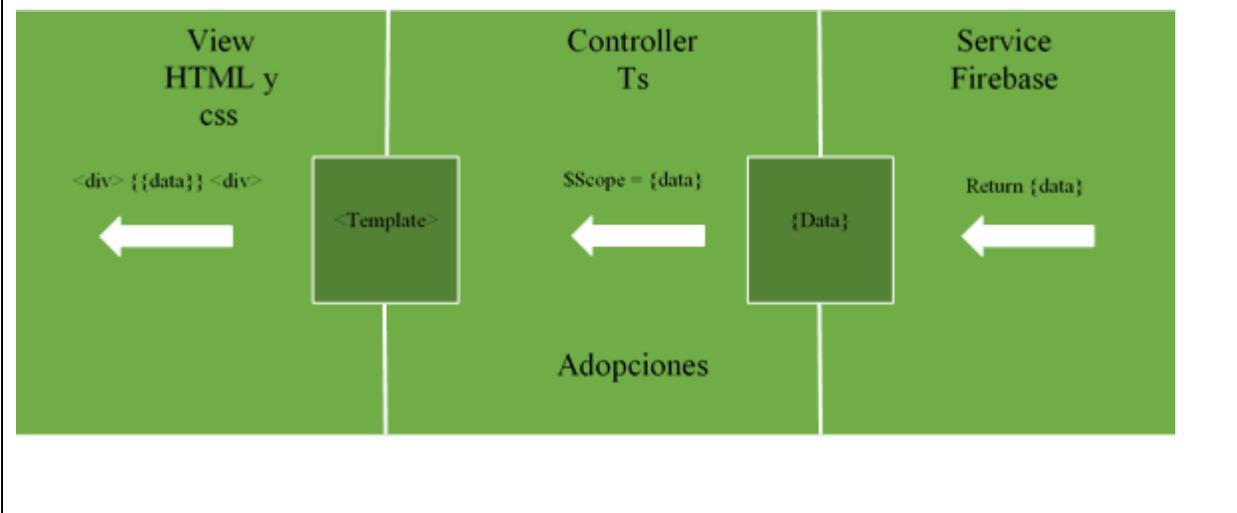
Una vez este la aplicación en ejecución, los servicios sociales se verán de la siguiente forma, ver figura 37.

Figura 37. Interfaz servicios sociales



En el módulo de adopciones maneja una vista en Ionic basado en HTML y css el cual se conecta por medio de un JSON al controlador, el controlador que está desarrollado en TypeScript, se conecta con el Firebase por medio de Http, en la figura 38 se puede apreciar el bloque de componentes, del módulo de adopciones.

Figura 38. Diagrama adopciones



El módulo de adopciones se desarrolló en Ionic, la cual es una aplicación híbrida, para la implementación de esta interfaz se utilizó en la vista etiquetas HTML como ion-card, ion-avatar, img, ion-note, *ngFor para obtener los datos y css para mejorar la experiencia visual, esta vista recibe respuesta JSON la cual permite visualizar los datos almacenado en la colección, en la figura 39 se puede apreciar algunas de las etiquetas utilizadas para la vista de las adopciones.

Figura 39. Modulo opciones HTM

```

<form class="new-user-form" [formGroup]="validations_form" (ngSubmit)="onSubmit(validations_form.value)">
  <ion-item>
    <ion-label floating color="secondary">Especie</ion-label>
    <ion-input type="text" formControlName="typeanimal" class="form-control" required</ion-input>
  </ion-item>
  <ion-item>
    <ion-label floating color="secondary">Nombre del animal</ion-label>
    <ion-input type="text" formControlName="nameanimal" class="form-control" required</ion-input>
  </ion-item>
  <ion-item>
    <ion-label floating color="secondary">Fecha de nacimiento</ion-label>
    <ion-input type="date" formControlName="age" class="form-control" ></ion-input>
  </ion-item>
  <ion-item>
    <ion-label>Sexo</ion-label>
    <ion-select formControlName="gender" class="form-control">
      <ion-option value="Hembra">Hembra</ion-option>
      <ion-option value="Macho">Macho</ion-option>
    </ion-select>
  </ion-item>
</form>

```

Caja de texto para ingresar datos del módulo adopción

Caja de texto para ingresar datos del módulo adopción

Caja de tipo date para ingresar fechas

Lista desplegable para elegir una opción

En el controlador de las adopciones se utilizó TypeScript embebido en Ionic. Para la creación del controlador de adopciones se importaron las clases y se inyectó las clases para proporcionar un servicio, estos cambios se realizan en la carpeta appmodule.ts, en la figura 40 se puede apreciar cómo se encuentran inyectadas las clases.

Figura 40. Crear servicio

```
import { Geolocation } from '@ionic-native/geolocation';
import { GeolocationPage } from '../pages/geolocation/geolocation';
import { NewAdoptionPage } from '../pages/new-adoption/new-adoption';
import { MyadoptionsPage } from '../pages/myadoptions/myadoptions';
import { EditAdoptionsPage } from '../pages/edit-adoptions/edit-adoptions';
import { DetailsAdoptionsPage } from '../pages/details-adoptions/details-adoptions';

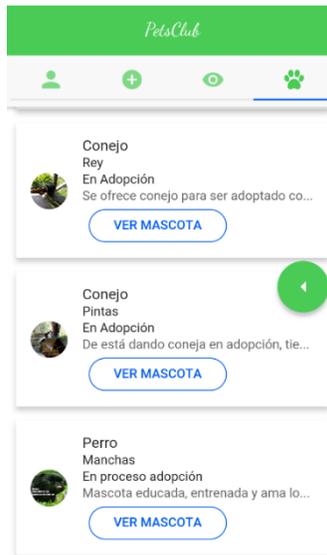
@NgModule({
  declarations: [
    MyApp,
    HomePage,
    RegisterPage,
    MenuPage,
    ListPage,
    EditPerfilPage,
    NewsNotesPage,
    EditNewsPage,
    EventsPage,
    CreateperfilPage,
    EditEventsPage,
    ReactionsPage,
```

Importación del
modulo para utilizar
el servicio

Declarar el servicio

Una vez se encuentre en ejecución la aplicación se puede observar de la siguiente manera, ver figura 41.

Figura 41. Interfaz de adopciones



10.1 Pruebas de calidad

Las pruebas de calidad se realizaron a través de SonarCloud, se hace instalación y configuración para enlazar el proyecto que se encuentra en el GitHub.

Se anexa el archivo sonar-project.properties en el proyecto petsclub, el cual contiene la configuración para el análisis del código.

Figura 42. Archivo de sonar-project.properties

```
html  menu.html  sonar-project.properties  details-adoptions.html  home.html  TS menus  TS news-notes.ts  profi  ...
1  sonar-scanner.bat -D"sonar.projectKey=petsclub" -D"sonar.organization=petsclub-github" -D"sonar.sources="
2  -D"sonar.host.url=https://sonarcloud.io" -D"sonar.login=029e350fc4e968fc3317c77b1c4c53dbb233963e"
```

Figura 43. Análisis de prueba

```
C:\sonar-scanner\bin>sonar-scanner.bat
INFO: Scanner configuration file: C:\sonar-scanner\bin\..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarQube Scanner 3.3.0.1492
INFO: Java 1.8.0_121 Oracle Corporation (64-bit)
INFO: Windows 10 10.0 amd64
INFO: User cache: C:\Users\JhonFreGom\.sonar\cache
INFO: SonarQube server 7.7.0
INFO: Default locale: "es_MX", source code encoding: "windows-1252" (analysis is platform dependent)
INFO: Load global settings
INFO: Load global settings (done) | time=952ms
INFO: Server id: BD367519-AWHW8ct9-T_TB3XqouNu
INFO: User cache: C:\Users\JhonFreGom\.sonar\cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=223ms
INFO: Load/download plugins (done) | time=468ms
INFO: Loaded core extensions: developer-scanner
INFO: Process project properties
INFO: Execute project builders
INFO: Execute project builders (done) | time=10ms
INFO: Project key: petsclub
INFO: Base dir: D:\PROYECTO\pcb\petsclub
INFO: Working dir: D:\PROYECTO\pcb\petsclub\.scannerwork
INFO: Load project settings
INFO: Load project settings (done) | time=198ms
```

Se realiza el análisis del código estático del proyecto con las herramientas y se obtiene los siguientes resultados:

- Se obtiene una cantidad de 84 Bugs, la calificación de confiabilidad es D cuando al menos hay un error crítico, se procede a realizar depuración en el código.
- Se obtiene una cantidad de 29 Code Smells, aun así, la calificación es A se procede a realizar depuración.

Figura 44. Resultado primer análisis

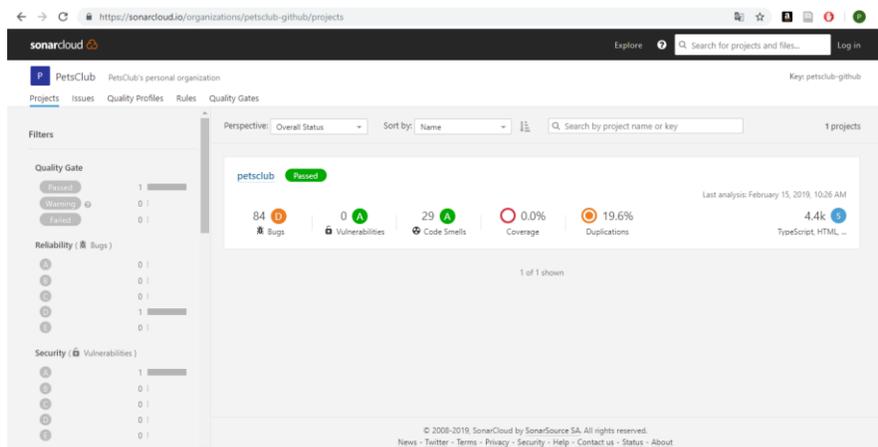


Figura 45. Resultado primer análisis descripción

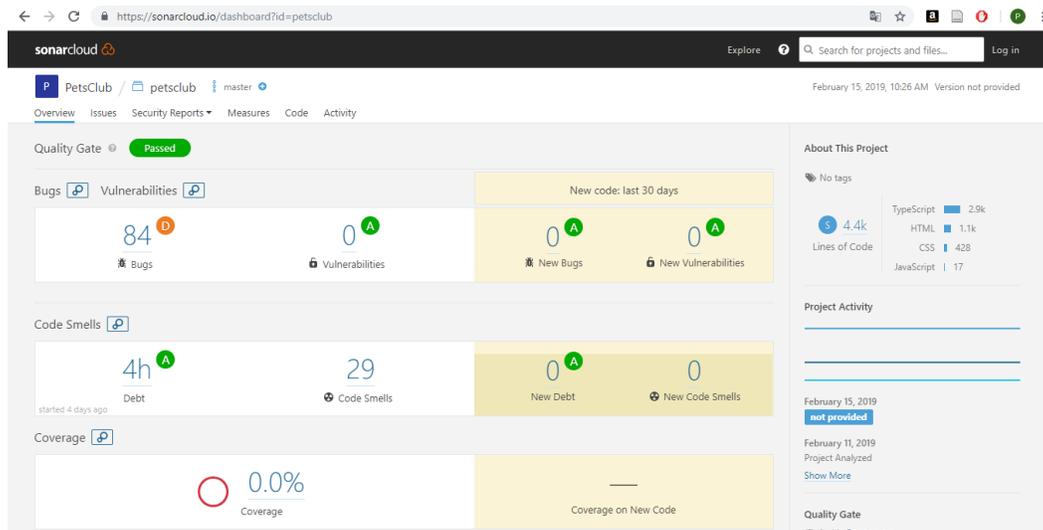
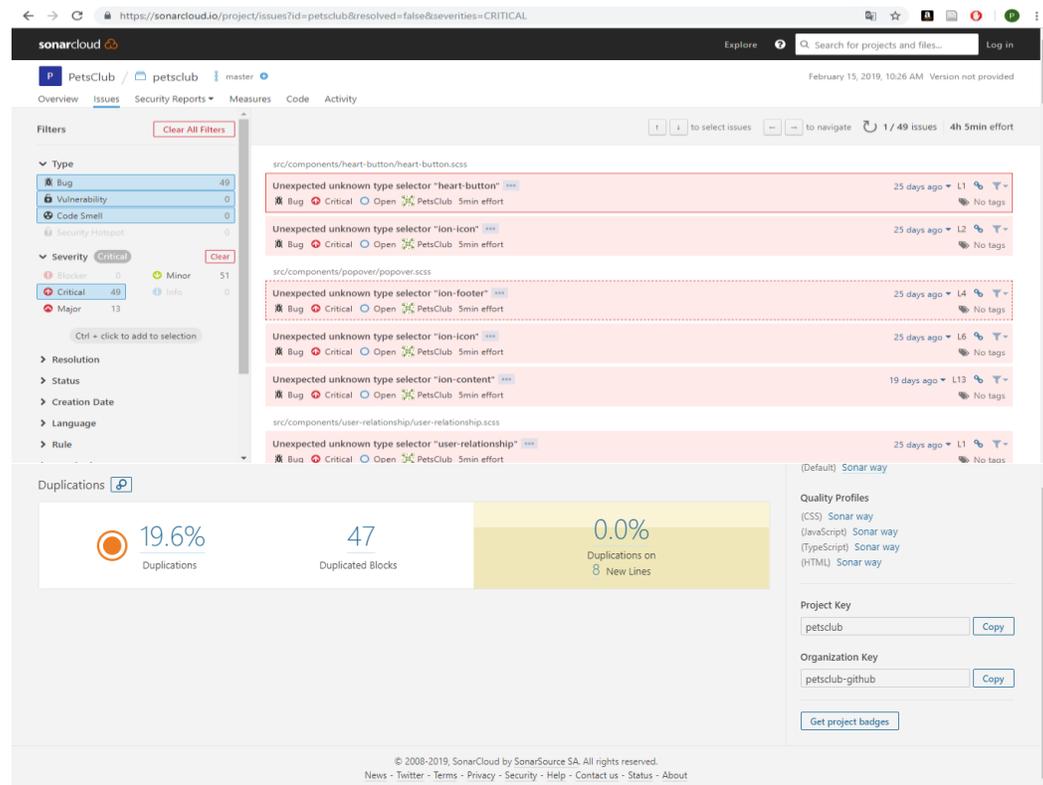
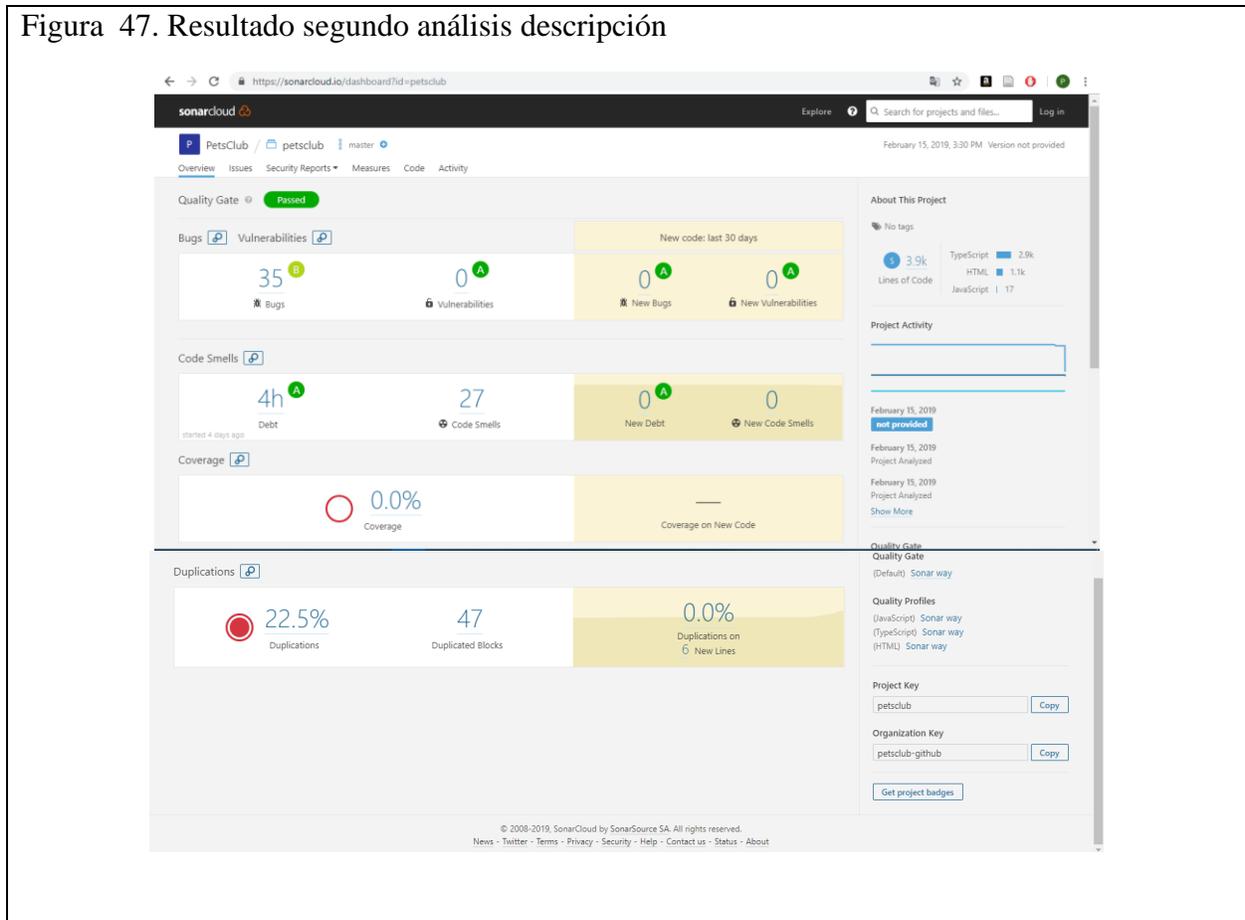


Figura 46. Errores para corregir



Se realiza de nuevo el análisis de calidad y el resultado es de 35 Bugs y en estado B que se produce cuando al menos hay un error.

Figura 47. Resultado segundo análisis descripción



Al realizar la depuración del código se obtiene la puerta de calidad en estado Passed, esto nos indica que el proyecto está listo para producción con una buena puntuación en Confiabilidad, Seguridad, Mantenibilidad, Cobertura, Complejidad.

Figura 48. Resultado prueba final

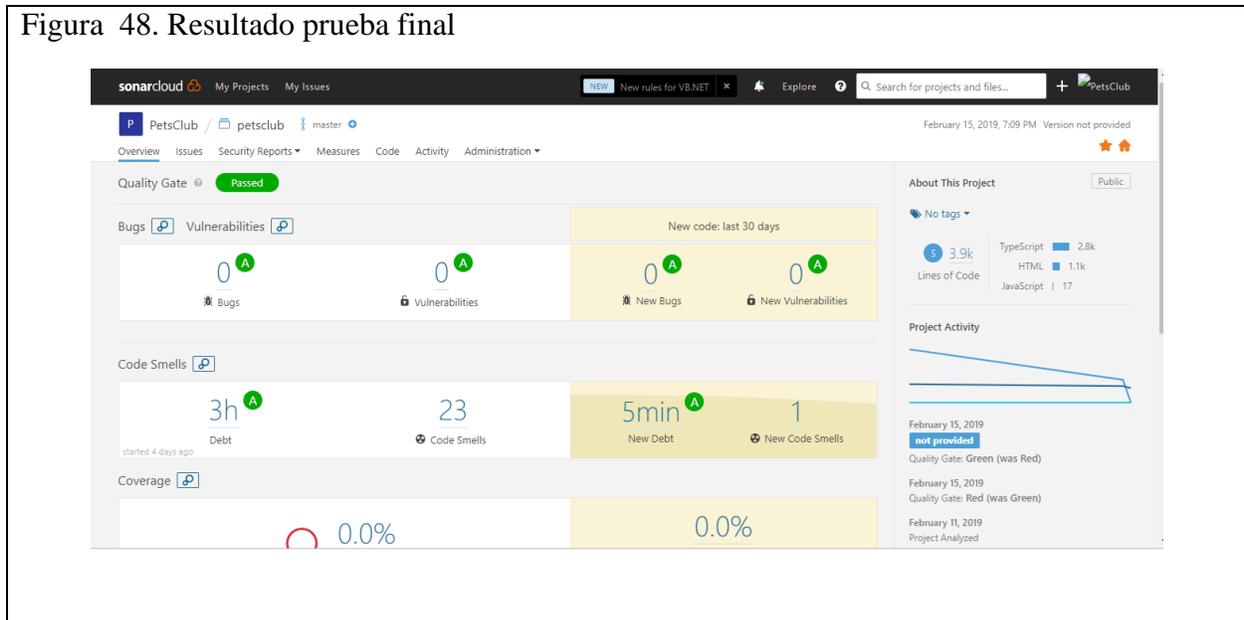


Figura 49. Comportamiento de las correcciones



De acuerdo con la gráfica 49, se evidencia que la curva disminuye conforme se realiza el análisis de calidad, los Bugs y Vulnerabilities se fueron depurando hasta llegar a 0, por otra parte, los Code Smells se obtiene 23 cuestiones donde se subsanan y se refactoriza para un mejor desempeño.

Las pruebas de calidad se realizaron con la herramienta SonarCloud con ayuda guiada del personal de **SmartCampus**, estas nos permitieron reforzar y mejorar las prácticas de desarrollo, identificar posibles fallos y así garantizar un buen funcionamiento de la aplicación con calidad en el producto ya que se pueden presentar errores y de esta forma reducir los riesgos en la implementación.

11. Conclusiones

El esquema de diseño de la arquitectura y la vista de despliegue presentada permite la integración de servicios móviles con las colecciones de datos con el fin de facilitar la gestión de usuarios animalistas, el acceso a contenidos y servicios sociales y demás para desplegar aplicaciones móviles híbridas a los usuarios finales. Este esquema de arquitectura puede ser tomado como referencia para futuros desarrollos que también necesiten integrar servicios móviles en otros contextos.

El diseño e implementación de aplicaciones basados en REST-JSON son independientes de la presentación de los datos, lo cual es acorde con la tendencia actual en la web de permitir la interoperabilidad entre aplicaciones de software independientes. También al utilizar este protocolo se facilitan el consumo de servicios a través de internet debido a su sencillez y flexibilidad.

Con la implementación del protocolo de pruebas para el despliegue de servicios móviles híbridos, se pudo comprobar que el diseño de una arquitectura orientada a servicios bajo el esquema REST – JSON fue una solución adecuada ya que los tiempos de respuesta fueron adecuados , obtenidos al hacer las pruebas de rendimiento de los servicios web de Moodle fueron buenos, garantizando con esto el cumplimiento de los principios del protocolo REST que establece la posibilidad de realizar múltiples peticiones simultaneas, para dichas pruebas se utilizó SonarCloud.

El desarrollo tecnológico presentado permite proyectar a los usuarios del sector animalista en el contexto de servicios interactivos para el apoyo de sus procesos colaborativos en temas de cuidado y protección animal, lo cual permitirá dar solución a las necesidades con las que cuenta este sector de contar con soluciones tecnológicas móviles híbridas para sistematizar y centralizar algunos de sus procedimientos más importantes con la ventaja de poder desplegar contenidos en sistemas operativos IOS y Android.

12. Referencias

- 4Patas. (2018). ABC de la nueva Ley Animal. *Recuperado 11 de diciembre de 2018, de <https://www.4patas.com.co/perros/historias/articulo/ley-animal-en-colombia/2417>*
- APLICACIONES HÍBRIDAS: ¿QUÉ SON Y CÓMO USARLAS? (2017, marzo 9). *Recuperado 11 de diciembre de 2018, de <https://www.nextu.com/blog/aplicaciones-híbridas-que-son-y-como-usarlas/>*
- ¿Conoces la base de datos en tiempo real? (2017, mayo 3). *Recuperado 15 de diciembre de 2018, de <http://developers-latam.googleblog.com/2017/05/conoces-la-base-de-datos-en-tiempo-real.html>*
- La tecnología de geolocalización avanza. (2017, agosto 1). *Recuperado 11 de diciembre de 2018, de <https://www.elnuevodia.com/negocios/consumo/nota/latecnologiadegeolocalizacionavanza-2345240/>*
- Londoño, A. (2015, mayo 26). movimiento animalista: Movimiento Animalista En Colombia. *Recuperado 11 de diciembre de 2018, de <http://culturaanimalista.blogspot.com/2015/05/movimiento-animalista-en-colombia.html>*
- Qué son las notificaciones push | Guía efectiva. (2018, enero 23). *Recuperado 11 de diciembre de 2018, de <https://www.mdirector.com/sms-marketing/que-son-las-notificaciones-push.html>*
- Tiempo, C. E. E. (2017, mayo 5). Mascotas tendrán una política de protección. *Recuperado 10 de diciembre de 2018, de <https://www.eltiempo.com/vida/medio-ambiente/politica-de-proteccion-a-las-mascotas-en-colombia-84940>*
- Yefferson Ospina. (2014, noviembre 24). El drama que viven los animales callejeros en Cali [Noticia]. *Recuperado 5 de abril de 2018, de <http://www.elpais.com.co/cal/el-drama-que-viven-los-animales-callejeros-en.html>*

13. Anexos

Anexo A. Descripción de los casos de uso Animalista

Caso de uso: Iniciar sesión Animalista.

Figura 50. Datos inicio sesión animalista

ID:	CU_02
Nombre del Caso de Uso:	CU Iniciar Sesión Animalista.
Propósito:	Ingresar a la página principal de la app (PetsClub).
Descripción:	Se requiere que la app (PetsClub), permita iniciar sesión con el usuario ya registrado.
Actores:	Animalista.
Precondiciones:	El Animalista debe estar registrado en el Sistema.
Postcondiciones de éxito:	Ingresar a la página principal de la App (PetsClub).
Postcondiciones de fracaso:	No ingresar a la página principal de la App (PetsClub).
Casos de uso invocados:	.
Casos de uso invocantes:	.

Escenario: Iniciar sesión animalista.

Figura 51. Escenario inicio sesión animalista

Escenario Principal	Escenario de Extensión
1. El Animalista ingresa el correo electrónico en el campo de texto de la interfaz inicio Animalista.	1.1. El Sistema validó que la información no es correcta MSJ_03.
2. El Animalista ingresa su contraseña en el campo de texto de la interfaz inicio Animalista.	2.1. El Sistema validó que la información no es correcta MSJ_04.
3. El administrador da clic en el botón ingresar de la interfaz inicio administrador.	3.1. El Sistema validó que la información no es correcta MSJ_06.
4. El Sistema valida la información del Animalista.	.
5. El Animalista ingresa a la app (PetsClub).	.
6. Fin CU Inicio de Sesión Animalista.	.

Caso de uso: Gestionar evento.

Figura 52. Datos gestión de eventos

ID:	CU_3
Nombre del Caso de Uso:	CU Gestionar evento.
Propósito:	Ingresar al módulo de gestión de eventos de la App (PetsClub).
Descripción:	Se requiere que la App (PetsClub), permita ingresar al módulo de gestión de eventos, para realizar las publicaciones y visualizaciones de dichos eventos.
Actores:	Animalista.
Precondiciones:	El animalista debe haber iniciado sesión en el Sistema.
Postcondiciones de éxito:	Ingresar al módulo de gestión de eventos de la App (PetsClub).
Postcondiciones de fracaso:	No ingresar al módulo de gestión de eventos de la App (PetsClub).
Casos de uso invocados:	.
Casos de uso invocantes:	.

Escenario: Gestionar evento.

Figura 53. Escenario gestionar evento

Escenario Principal	Escenario de Extensión
1. El Animalista ingresa a la página principal.	.
2. El Animalista visualiza los eventos que han publicado otros usuarios, en la opción de visualización de eventos.	.
3. El Animalista da clic en el icono de eventos.	.
4. Animalista da clic en el botón cargar imagen.	.
5. El Animalista ingresa el nombre del evento en el campo de texto de la interfaz de publicación de eventos.	.
6. El Animalista ingresa la descripción del evento en el campo de texto de la interfaz de publicación de eventos.	.
7. El Animalista ingresa el lugar del evento en el campo de texto de la interfaz de publicación de eventos.	.
8. El Animalista da clic en el botón crear.	.
9. Fin CU Gestión evento.	.

Caso de uso: Gestionar servicios sociales.

Figura 54. Datos gestionar servicios

ID:	CU_4
Nombre del Caso de Uso:	CU Gestionar servicios sociales.
Propósito:	Ingresar al módulo de gestión de servicios sociales de la App (PetsClub).
Descripción:	Se requiere que la App (PetsClub), permita ingresar al módulo de gestión de servicios sociales, para realizar las publicaciones y visualizaciones de dichos servicios.
Actores:	Animalista.
Precondiciones:	El animalista debe haber iniciado sesión en el Sistema.
Postcondiciones de éxito:	Ingresar al módulo de gestión de eventos de la App (PetsClub).
Postcondiciones de fracaso:	No ingresar al módulo de gestión de eventos de la App (PetsClub).
Casos de uso invocados:	.
Casos de uso invocantes:	.

Escenario: Gestionar servicios sociales.

Figura 55. Escenario servicio social

Escenario Principal	Escenario de Extensión
1. El Animalista ingresa a la página principal.	.
2. El Animalista visualiza los servicios sociales que han publicado otros usuarios, en la opción de visualización de servicios sociales.	.
3. El Animalista da clic en el icono de servicios sociales.	.
4. Animalista da clic en el botón cargar imagen.	.
5. El Animalista ingresa el nombre del servicio social en el campo de texto de la interfaz de publicación de servicios sociales.	.
6. El Animalista ingresa la descripción del servicio social en el campo de texto de la interfaz de publicación de servicios sociales.	.
7. El Animalista ingresa el lugar del servicio social en el campo de texto de la interfaz de publicación de servicios sociales.	.
8. El Animalista da clic en el botón crear.	.
9. Fin CU Gestión servicio social.	.

Figura 56. Casos de uso

		Historias de Usuario y Criterios de Aceptación		Versión: 1		Código: F-GA-20	PESO	PRIORIDAD
Id de la Historia	Rol	Enunciado de la Historia		Criterio de Aceptación / Título	Ver Detalle			
		Característica / Funcionalidad	Razón / Resultado					
HU - PETSCLUB 001	Yo, como [Usuario]	Necesito [Ingresar en el sistema]	Para [acceder a las funcionalidades del aplicativo PETSCLUB]	1. La pantalla deberá contar con un encabezado (Bienvenido). 2. La autenticación se realizará contra la base de datos. 3. La pantalla deberá contar con un campo para ingresar el correo electrónico, no puede ser nulo. 3.1 En caso de que sea nulo, el sistema debe generar el siguiente mensaje (El correo electrónico es requerido). 4. La pantalla deberá contar con un campo para Ingresar la Contraseña con mínimo 6 caracteres, no puede ser nulo y deberá ser visible con carácter (*). 4.1 En caso de que sea nulo, el sistema debe generar el siguiente mensaje (Campo de Contraseña requerido). 5. Si el correo electrónico no se ha registrado deberá salir un "usuario no registrado". 6. Si la contraseña está mal digitada deberá salir un "contraseña o usuario incorrecto". 7. La pantalla deberá contar con un link para restablecer contraseña. 8. La pantalla deberá contar con un link que nos lleve a un formulario que permita registrar nuevos usuarios. 9. La pantalla deberá contar con el logo de la aplicación PETSCLUB y fondo de ambiente. 10. La pantalla deberá contar con un botón color azul centrado. (INGRESAR), una vez el usuario presione este botón, deberá ingresar al perfil.	Hoja 1	5	S	
HU - PETSCLUB 002	Yo, como [Usuario]	Necesito [Registrarme]	Para [ingresar al aplicativo PETSCLUB]	1. La pantalla deberá contar con un encabezado (Regístrate). 2. El Formulario de registro deberá contener los siguientes campos: correo electrónico, contraseña (la contraseña deberá contener mínimo 6 caracteres). Los campos no pueden ser nulos y deberá ser visible con carácter (*). 2.1 En caso de que sea nulo el sistema debe generar el siguiente mensaje (Campos requeridos). 3. La pantalla deberá contar con un botón color verde centrado. (REGISTRAR), una vez el usuario presione este botón, deberá enviar a creación del perfil. 4. La pantalla deberá contar con un link ¿ya tienes cuenta? ingresar aquí.		3	S	
HU - PETSCLUB 003	Yo, como [Usuario] registrado	Necesito [Crear perfil de usuario]	Para [ingresar al perfil PETSCLUB]	1. La pantalla deberá contar con encabezado (Crear perfil). 2. El Formulario de registro deberá contener los siguientes campos: nombre, apellido, tipo de usuario, fecha de nacimiento (genera un calendario), dirección, teléfono y descripción del usuario. 2.1 En caso de que sea nulo el sistema no activa el botón de color verde (CREAR USUARIO). (Dirección a página principal) 3. La pantalla deberá contar con un botón (SUBIR FOTO). 3.1 La pantalla deberá contar con un espacio donde se muestre la foto escogida por el usuario.		3	S	
HU - PETSCLUB 004	Yo, como [Usuario] registrado	Necesito [Diligenciar un formulario]	Con la finalidad de [Compartir los servicios sociales a demás usuarios]	1. La pantalla deberá contar con un encabezado (Compartir). 2. El formulario deberá contener los siguientes campos: título, descripción. 3. La pantalla deberá contar con un botón (SUBIR FOTO). 3.1 La pantalla deberá contar con un espacio donde se muestre la foto escogida por el usuario. 4. La pantalla deberá contar con un botón (TOMAR FOTO). 4.1 En caso de que sea nulo el sistema no activa el botón de color verde (COMPARTIR SERVICIO SOCIAL). (Dirección a página principal).		20	M	
HU - PETSCLUB 005	Yo, como [Usuario] registrado	Necesito [Diligenciar un formulario]	Con la finalidad de [Compartir los eventos a demás usuarios]	1. La pantalla deberá contar con un encabezado (Crear evento). 2. El formulario deberá contener los siguientes campos: título evento, fecha evento (genera un calendario), lugar, descripción del evento. 3. La pantalla deberá contar con un botón (SUBIR FOTO). 3.1 La pantalla deberá contar con un espacio donde se muestre la foto escogida por el usuario. 4. En caso de que sea nulo el sistema no activa el botón de color verde (COMPARTIR EVENTO). (Dirección a página principal). 5. La pantalla deberá contar con un mapa, si no se selecciona un lugar saldrá una alerta ("Selecciona en el mapa, el lugar donde se realizará el evento") (El marker debe ser un dog).		8	M	
HU - PETSCLUB 006	Yo, como [Usuario] registrado	Necesito [Diligenciar un formulario]	Con la finalidad de [Crear adopciones para los demás usuarios]	1. La pantalla deberá contar con un encabezado (Crear adopción). 2. El formulario deberá contener los siguientes campos: especie, nombre del animal, fecha de nacimiento (genera un calendario), sexo (desplegable: hembra, macho), peso, teléfono, estado adopción (desplegable: En adopción, En proceso de adopción, Adoptado), Descripción. 3. La pantalla deberá contar con un botón (SUBIR FOTO). 3.1 La pantalla deberá contar con un espacio donde se muestre la foto escogida por el usuario. 4. La pantalla deberá contar con un botón (TOMAR FOTO). 4.1 En caso de que sea nulo el sistema no activa el botón de color verde (CREAR ADOCIÓN). (Dirección a página principal).		8	M	
HU - PETSCLUB 007	Yo, como [Usuario] registrado	Necesito [Diligenciar un formulario]	Con la finalidad de [Generar códigos QR]	1. La pantalla deberá contar con un encabezado (TU código QR). 2. La pantalla deberá tener un label con el siguiente enunciado: "En este espacio podrás crear tus códigos QR con información de tu empresa, url, ayudas sociales." 3. El formulario deberá contener el siguiente campo: Descripción QR. 4. La pantalla deberá contar con un botón azul (CREAR CÓDIGO QR). 4.1 La pantalla deberá contar con un campo donde se genere el código QR. 5. La pantalla deberá contar con un botón azul (SCAN CÓDIGO QR) (Deberá activar la cámara).		20	M	
HU - PETSCLUB 008	Yo, como [Usuario] registrado	Necesito [Ingresar a la pantalla principal]	Con la Finalidad de [acceder a las funcionalidades del aplicativo Pets Club]	1. La pantalla deberá contar con un encabezado (PetsClub). 2. La pantalla deberá tener segmentos: Ver mi Perfil, Ver Eventos, Ver servicios sociales, Ver Adopciones. 2.1 Segmento Mi Perfil deberá ser un icono "person". Deberá contener un espacio centrado de visualización de foto, visualización de datos: Nombre de usuario, teléfono, Dirección, Descripción. Deberá contener tres botones verdes en la parte inferior "MIS EVENTOS", "MIS SERVICIOS", "ADOPCIONES" (Deberá contener un icon "List" que me envíe a gestión de perfil HU-004). 2.2 Segmento Ver eventos deberá ser un icono "add-circle". Deberá contener "cards" visualizando todos los eventos publicados de los demás usuarios, deberá mostrar: Usuario que publica evento (Dirección a visualizar perfil HU-003), Foto, Fecha de evento, fecha de publicación, Lugar, Descripción y opción de Ver Mapa (Debe desplegar mapa con ubicación del evento con un marker tipo dog). 2.3 Segmento Ver servicios sociales deberá ser un icono "eye". Deberá contener "cards" visualizando todos los servicios sociales publicados de los demás usuarios, deberá mostrar: Usuario que publica servicio (Dirección a visualizar perfil HU-003), Foto, Fecha de publicación, Descripción. 2.4 Segmento Ver adopciones deberá ser un icono "paw". Deberá contener "cards" visualizando todas las adopciones publicadas de los demás usuarios, deberá mostrar: Foto de mascota, Especie, Nombre mascota, Estado de adopción, Descripción. Deberá contener un botón azul (VER MASCOTA) donde despliega detalles de la adopción. 3. La pantalla deberá contar con un FAB centrado hacia la derecha el cual se debe desplegar hacia abajo y se visualiza: "add-circle", "eye", "paw" y deberá abrir cada uno de los formularios. (Compartir servicio, Compartir eventos, Crear adopciones). 4. La pantalla deberá contar con un enlace para cerrar sesión. (Redirección a la hoja de ingreso). 4.1 El sistema deberá destruir la sesión y redireccionar al usuario a la pantalla de Inicio de Sesión (Login).		20	M	
HU - PETSCLUB 009	Yo, como [Usuario] registrado	Necesito [visualizar mis servicios sociales]	Con la finalidad de [Gestionar mis publicaciones]	1. La pantalla deberá contar con un encabezado (Mis servicios sociales). 2. Deberá contener "cards" visualizando todos los servicios sociales publicados del mismo usuario, deberá mostrar: Foto, Fecha de publicación, Descripción. 3. En las publicaciones cada "cards" deberá tener un icono "List" (Envía a pantalla de gestión de publicación) 3.1 La pantalla de gestión de publicación deberá permitir modificar o eliminar cada una.		13	M	
HU - PETSCLUB 010	Yo, como [Usuario] registrado	Necesito [visualizar mis eventos]	Con la finalidad de [Gestionar mis publicaciones]	1. La pantalla deberá contar con un encabezado (Mis eventos). 2. Deberá contener "cards" visualizando todos los eventos publicados del mismo usuario, deberá mostrar: Foto, Fecha de evento, fecha de publicación, Lugar, Descripción y opción de Ver Mapa (Debe desplegar mapa con ubicación del evento con un marker tipo dog). 3. En las publicaciones cada "cards" deberá tener un icono "List" (Envía a pantalla de gestión de publicación) 3.1 La pantalla de gestión de publicación deberá permitir modificar o eliminar cada una.		5	S	
HU - PETSCLUB 011	Yo, como [Usuario] registrado	Necesito [visualizar mis adopciones creadas]	Con la finalidad de [Gestionar mis publicaciones]	1. La pantalla deberá contar con un encabezado (Mis adopciones). 2. Deberá contener "cards" visualizando todos los eventos publicados del mismo usuario, deberá mostrar: Estado adopción (Tamaño de letra un poco más grande), Nombre animal, Foto, Especie, Fecha de nacimiento, Sexo, Peso, Contacto, Descripción, fecha de publicación. 3. Deberá tener un botón "List" (Envía a pantalla de gestión de publicación)		8	M	
HU - PETSCLUB 012	Yo, como [Usuario] registrado	Necesito [una pantalla de perfil]	Con la finalidad de [visualizar el perfil de usuario]	1. La pantalla deberá contar con un encabezado (Perfil). 2. Deberá contener un espacio centrado de visualización de foto, visualización de datos: Nombre de usuario, teléfono, Dirección, Descripción. 3. Deberá tener un botón hacia atrás para regresar a la pantalla anterior.		8	S	
HU - PETSCLUB 013	Yo, como [Usuario] registrado	Necesito [un botón List]	Para [editar el perfil de usuario]	1. Tendrá opción de actualizar: Foto, Nombre, Apellido, Usuario, Fecha de nacimiento (despliega calendario), Dirección, Teléfono, Descripción. 2. Deberá tener un botón hacia atrás para regresar a la pantalla anterior. 3. Deberá tener un botón verde, centrado (Actualizar usuario).		5	S	
HU - PETSCLUB 014	Yo, como [Usuario] registrado	Necesito [Cerrar Sesión]	Con la Finalidad de [salir correctamente del sistema]	1. La aplicación deberá contar con un botón Cerrar Sesión. 1.1 Cerrar Sesión (Redirección a la página de Ingreso de usuario)		1	S	
HU - PETSCLUB 015	Yo, como [Usuario]	Necesito [Notificación de registro]	Para [Confirmar el registro]	1. La aplicación deberá mostrar un mensaje de confirmación de registro y enviar correo electrónico para confirmar.		1	S	
						128		

Figura 57. Compromisos y cantidad de Sprints

Compromisos	
Jhon	10
Mauricio	10
Total:	20

de Sprints = Peso de historias / compromiso 6.4

de Sprints = 6

NUMERO	FECHA SPRINT
0	04 junio al 18 junio 2018
1	19 junio al 09 julio 2018
2	10 julio al 06 agosto 2018
3	07 agosto al 03 septiembre 2018
4	04 septiembre al 01 octubre 2018
5	02 octubre al 29 octubre 2018
6	30 octubre al 26 noviembre 2018

Figura 58. Sprints

SPRINT 1			SPRINT 5		
HU - PETSCLUB - 005	8	M	HU - PETSCLUB - 010	5	S
HU - PETSCLUB - 004	20	M	HU - PETSCLUB - 012	8	S
	28			13	
SPRINT 2			SPRINT 6		
HU - PETSCLUB - 008	20	M	HU - PETSCLUB - 013	5	S
HU - PETSCLUB - 002	3	S	HU - PETSCLUB - 014	1	S
	23		HU - PETSCLUB - 015	1	S
			HU - PETSCLUB - 006	8	M
				15	
SPRINT 3					
HU - PETSCLUB - 001	5	S			
HU - PETSCLUB - 003	3	S			
HU - PETSCLUB - 007	20	M			
	28				
SPRINT 4					
HU - PETSCLUB - 009	13	M			
HU - PETSCLUB - 011	8	M			
	21				