



UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS
RECURSOS NATURALES NO RENOVABLES**

CARRERA DE INGENIERÍA EN SISTEMAS

**DESARROLLO DE UNA APLICACIÓN
WEB DE ADOPCIÓN DE MASCOTAS
EN LA CIUDAD DE LOJA**

**“Tesis de Grado previo a la obtención del
Título de Ingeniero en Sistemas”**

Autora: Nadia Susana Paucar Cabrera

Director: Ing. Alfredo Vinicio Zúñiga Tinizaray Mg. Sc.

LOJA – ECUADOR

2017

Certificación

Ing. Alfredo Vinicio Zúñiga Tinizaray Mg. Sc.

DIRECTOR DE TESIS

CERTIFICA:

Que el presente proyecto de fin de carrera elaborado previa la obtención del Título de Ingeniería en Sistemas, titulado: **“DESARROLLO DE UNA APLICACIÓN WEB DE ADOPCIÓN DE MASCOTAS EN LA CIUDAD DE LOJA”**, realizada por la postulante Nadia Susana Paucar Cabrera, cumple con los requisitos establecidos por las normas generales para la graduación en la Universidad Nacional de Loja, tanto en aspecto de forma como de contenido.

Por lo tanto, autorizo proseguir los trámites legales pertinentes para su presentación y defensa.

Loja, 12 de Septiembre del 2017



.....
Ing. Alfredo Vinicio Zúñiga Tinizaray Mg. Sc.

DIRECTOR DE TESIS

Autoría

Yo **NADIA SUSANA PAUCAR CABRERA**, declaro ser la autora del presente trabajo de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

Firma:



Cédula: 1104650856

Fecha: 12 de septiembre del 2017

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LA AUTORA, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO

Yo, **NADIA SUSANA PAUCAR CABRERA**, declaro ser la autora de la tesis titulada: **DESARROLLO DE UNA APLICACIÓN WEB DE ADOPCIÓN DE MASCOTAS EN LA CIUDAD DE LOJA**, como requisito para optar el grado de **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja a los 11 días del mes de septiembre del dos mil diecisiete.

Firma:

Autor: Nadia Susana Paucar Cabrera

Cédula: 1104650856

Dirección: Loja, (Américo Vespucio y Thomas Alva Edison).

Correo Electrónico: nspaucarc@unl.edu.ec

Teléfono: 072545307 **Celular:** 0985116900

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Alfredo Vinicio Zúñiga Tinizaray, Mg. Sc.

Tribunal de Grado: Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.

Ing. Milton Leonardo Labanda Jaramillo, Mg. Sc.

Ing. Boris Marcel Díaz Pauta, Mg. Sc.

Dedicatoria

Dedico todo el esfuerzo de este trabajo principalmente a Dios y mi familia; en especial a mi madre, quien con amor, paciencia y sacrificio ha logrado que llegue hasta este momento tan importante en mi vida. A mi abuelita, que con cariño supo darme aliento en los momentos difíciles de mi vida estudiantil. A mis amigos, quienes han compartido todos los momentos buenos y malos de este proceso universitario.

La Autora

Agradecimiento

Agradezco a Dios por ser quien me da las fuerzas día a día para cumplir todas mis metas, en especial la de culminar con éxito este ciclo académico.

A mi madre, abuelita y familia; quienes son mi apoyo fundamental y me inspiran todos los días a seguir adelante, me llenan de fuerza y valor para no decaer ante los problemas, brindándome apoyo y comprensión incondicional.

A todos los docentes de la carrera de Sistemas, quienes con espíritu de enseñanza supieron brindarme sus conocimientos a lo largo de mi vida universitaria. A mi director de tesis por su guía, preocupación y sus importantes consejos que me ayudaron a culminar exitosamente este proyecto.

A mis compañeros de aula, quienes fueron partícipes de todo el esfuerzo realizado, con quienes compartí gratas experiencias y días de estudio.

Nadia Susana Paucar Cabrera.

Tabla de Contenidos

Índice General

Certificación	II
Autoría	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LA AUTORA.....	IV
Dedicatoria	V
Agradecimiento.....	VI
Tabla de Contenidos.....	VII
Índice General.....	VII
Índice de Figuras.....	X
Índice de Tablas	XII
1. Título	1
2. Resumen	2
Abstract.....	3
3. Introducción	4
4. Revisión de literatura	6
4.1. Situación actual de la ciudad de Loja.....	6
4.1.1. Responsabilidad del gobierno local en el control y protección animal.....	6
4.2. Tecnologías en la adopción de mascotas	7
4.3. Herramientas de desarrollo.....	11
4.3.1. Lenguaje de Programación Python.....	11
4.3.2. Frameworks para Python.....	11
4.3.3. Materialize Framework	15
4.3.4. PostgreSQL.....	15
4.3.5. Django Rest Framework	16
4.3.6. Servidor Nginx	17

5.	Materiales y Métodos.....	18
5.1.	Técnicas de recolección de información	18
5.2.	Métodos.....	18
5.2.1.	Método Deductivo.....	18
5.2.2.	Método Inductivo	18
5.2.3.	Método de Revisión Sistemática de Literatura	18
5.3.	Metodología de desarrollo de software	26
6.	Resultados.....	27
6.1.	Fase de concepción.....	27
6.1.1.	Definición de los requisitos	27
6.1.2.	Determinación de Requerimientos.....	27
6.1.3.	Diagrama de casos de uso	29
6.1.4.	Especificación de casos de uso.....	30
6.1.5.	Glosario de términos.....	35
6.2.	Fase de Elaboración.....	36
6.2.1.	Arquitectura	36
6.2.2.	Prototipo de la aplicación.....	37
6.2.3.	Diagrama de clases.....	40
6.2.4.	Diagrama de paquetes	43
6.2.5.	Diagrama de despliegue.....	44
6.3.	Fase de construcción.....	44
6.3.1.	Diagramas de secuencia	44
6.3.2.	Implementación	47
6.4.	Fase de transición	56
6.4.1.	Test de la aplicación	56
6.4.2.	Pruebas de funcionalidad	67
7.	Discusión.....	71
7.1.	Desarrollo de la propuesta alternativa.....	71

7.2. Valoración técnica económica ambiental	73
8. Conclusiones	76
9. Recomendaciones	77
10. Bibliografía	78
11. Anexos	81
Anexo 1: Estudios seleccionados en la revisión sistemática de literatura	81
Anexo 2: Entrevista a la Fundación “Ángeles con Patas”	82
Anexo 3: Entrevista a la Fundación Conciencia Animal	85
Anexo 4: Documento de Especificación de Requerimientos	87
Anexo 5. Plantilla para pruebas funcionales	97
Anexo 6. Pruebas funcionales de PetHome	98
Anexo 7: Pruebas de accesibilidad	110

Índice de Figuras

Figura 1. Arquitectura de Django Framework.....	13
Figura 2. Estructura de archivos Django	13
Figura 3. Diagrama de casos de uso	30
Figura 4. Arquitectura del sistema.....	36
Figura 5. Prototipo de la página de inicio	38
Figura 6. Prototipo de inicio de sesión	38
Figura 7. Prototipo de página de inicio de usuario	39
Figura 8. Prototipo para mostrar mascotas	40
Figura 9. Diagrama de clases	41
Figura 10. Diagrama de paquetes.....	43
Figura 11. Diagrama de despliegue	44
Figura 12. Diagrama de secuencia del CU001	45
Figura 13. Diagrama de secuencia CU002	45
Figura 14. Diagrama de secuencia del CU003.....	45
Figura 15. Diagrama de secuencia del CU004.....	46
Figura 16 Diagrama de secuencia del CU005.....	46
Figura 17. Diagrama de secuencia del CU006.....	46
Figura 18. Aplicaciones instaladas en "PetHome".....	47
Figura 19. Estructura del proyecto "PetHome"	48
Figura 20. Pantalla de inicio de "PetHome".....	49
Figura 21. Pantalla de inicio de sesión en "PetHome".....	50
Figura 22. Sitio de administración de "PetHome"	50
Figura 23. Código para la administración del sitio.....	51
Figura 24. Página de inicio de usuario_normal	51
Figura 25. Gestión de mascotas en "PetHome"	52
Figura 26. Formulario de registro de mascota.....	52
Figura 27. Mascotas en adopción de "PetHome"	53
Figura 28. Formulario de solicitud de adopción.....	54
Figura 29. Página de inicio de usuario fundación.....	55
Figura 30. Historial de solicitudes de adopciones aceptadas	55
Figura 31. Pruebas unitarias	56
Figura 32. Resultado del test unitario.....	57
Figura 33. Prueba unitaria corregida.....	57

Figura 34. Resultado del test unitario corregido	57
Figura 35. Configuración de número de peticiones en Apache JMeter	58
Figura 36. Prueba de carga (Página de inicio)	58
Figura 37. Resultados de la prueba de carga (Página de inicio)	59
Figura 38. Resumen de la prueba de carga (Página de inicio)	59
Figura 39. Prueba de carga (Página de inicio de sesión)	59
Figura 40. Resultados de la prueba de carga (Página de inicio de sesión)	60
Figura 41. Resumen de la prueba de carga (Página de inicio de sesión)	60
Figura 42. Resumen de la prueba de estrés con 200 peticiones	61
Figura 43. Resumen de la prueba de estrés con 300 peticiones	61
Figura 44. Resumen de la prueba de estrés con 400 peticiones	61
Figura 45. Resultado de la prueba de estrés con 500 peticiones	62
Figura 46. Resumen de la prueba de estrés con 500 peticiones	62
Figura 47. Prueba de accesibilidad incorrecta de página de inicio	63
Figura 48. Commits de corrección de código de página de inicio	64
Figura 49. Prueba de accesibilidad correcta de página de inicio	64
Figura 50. Prueba de accesibilidad incorrecta de página de registro de usuario	65
Figura 51. Commits de corrección de código de página de registro de usuario	65
Figura 52. Prueba de accesibilidad correcta de página de registro de usuario	66
Figura 53. Prueba de accesibilidad incorrecta página de registro de fundación	110
Figura 54. Prueba de accesibilidad correcta de registro de fundación	111
Figura 55. Prueba de accesibilidad incorrecta página de inicio de sesión	111
Figura 56. Prueba de accesibilidad correcta de página de inicio de sesión	112
Figura 57. Prueba de accesibilidad incorrecta página de recuperar contraseña	112
Figura 58. Prueba de accesibilidad correcta de página de recuperar contraseña	113
Figura 59. Prueba de accesibilidad incorrecta página de mascotas en adopción	113
Figura 60. Prueba de accesibilidad correcta de página de mascotas en adopción	114

Índice de Tablas

TABLA I. ESTUDIOS SELECCIONADOS.....	8
TABLA II. RESULTADOS DEL ESTUDIO ES1[3]	8
TABLA III. RESULTADO DEL ESTUDIO ES2[4].....	9
TABLA IV. RESULTADOS DEL ESTUDIO ES3[5].....	9
TABLA V. RESULTADOS DEL ESTUDIO ES4[6].....	10
TABLA VI. RESULTADO DEL ESTUDIO ES5[7]	10
TABLA VII. COMPARATIVA DE FRAMEWORKS PYTHON	14
TABLA VIII. VALORES DE ALMACENAMIENTO DE POSTGRESQL	16
TABLA IX. ESTUDIOS PRELIMINARES Y TÉRMINOS.....	19
TABLA X. CADENAS DE BÚSQUEDA	21
TABLA XI. ESTUDIOS SELECCIONADOS	22
TABLA XII. TECNOLOGÍAS OBTENIDAS DEL SLR	24
TABLA XIII. REQUERIMIENTOS FUNCIONALES.....	27
TABLA XIV. REQUISITOS NO FUNCIONALES	29
TABLA XV. DESCRIPCIÓN C.U. GESTIONAR USUARIOS	30
TABLA XVI. DESCRIPCIÓN C.U. GESTIONAR MASCOTAS EN ADOPCIÓN.....	31
TABLA XVII. DESCRIPCIÓN C.U. REALIZAR ADOPCIÓN	32
TABLA XVIII. DESCRIPCIÓN C.U. GESTIONAR MASCOTA EXTRAVIADA.....	33
TABLA XIX. DESCRIPCIÓN C.U. GESTIONAR EVENTO.....	34
TABLA XX. DESCRIPCIÓN C.U. CONSULTA MASCOTA EXTRAVIADA	35
TABLA XXI. GLOSARIO DE TÉRMINOS.....	35
TABLA XXII. COMPONENTES DE LA ARQUITECTURA SERVIDOR	37
TABLA XXIII. COMPONENTES DE LA ARQUITECTURA CLIENTE	37
TABLA XXIV. RESULTADO DE LAS PRUEBAS DE ACCESIBILIDAD A LAS PÁGINAS DE "PETHOME"	66
TABLA XXV. RESULTADO DE LAS PRUEBAS FUNCIONALES DEL CU. GESTIONAR USUARIOS.....	67
TABLA XXVI. RESULTADOS PRUEBA FUNCIONAL CU. GESTIONAR MASCOTA EN ADOPCIÓN	68
TABLA XXVII. RESULTADOS PRUEBA FUNCIONAL CU. SOLICITAR ADOPCIÓN ..	69
TABLA XXVIII. RESULTADO PRUEBAS FUNCIONALES CU. GESTIONAR MASCOTA EXTRAVIADA.....	69

TABLA XXIX. RESULTADOS PRUEBA FUNCIONAL CU. GESTIONAR EVENTO	70
TABLA XXX. RESULTADOS PRUEBAS FUNCIONALES CU. PROMOCIONAR MASCOTA EXTRAVIADA.....	70
TABLA XXXI. VALORACIÓN ECONÓMICA DE RECURSOS HUMANOS	73
TABLA XXXII. VALORACIÓN ECONÓMICA DE RECURSOS MATERIALES	73
TABLA XXXIII. VALORACIÓN ECONÓMICA DE RECURSOS DE HARDWARE	74
TABLA XXXIV. VALORACIÓN ECONÓMICA DE RECURSOS DE SOFTWARE	74
TABLA XXXV. VALORACIÓN ECONÓMICA DE RECURSOS DE COMUNICACIONES	75
TABLA XXXVI. RESUMEN DEL PRESUPUESTO.....	75
TABLA XXXVII. RESPONSABLE DEL DESARROLLO DEL PROYECTO	88
TABLA XXXVIII – DIRECTOR DEL PROYECTO	89
TABLA XXXIX. PLANTILLA DE PRUEBAS FUNCIONALES	97
TABLA XL. PRUEBAS FUNCIONALES DE CREAR DE USUARIO.....	98
TABLA XLI. PRUEBA FUNCIONAL DE ACTUALIZAR USUARIO	99
TABLA XLII. PRUEBA FUNCIONAL DE REGISTRAR MASCOTA EN ADOPCIÓN..	101
TABLA XLIII. PRUEBAS FUNCIONALES DE ACTUALIZAR MASCOTA EN ADOPCIÓN	102
TABLA XLIV. PRUEBAS FUNCIONALES DE SOLICITAR ADOPCIÓN	104
TABLA XLV. PRUEBA FUNCIONAL DE CREAR MASCOTA EXTRAVIADA.....	105
TABLA XLVI. PRUEBA FUNCIONAL DE CREAR EVENTO	107
TABLA XLVII. PRUEBA FUNCIONAL DE ACTUALIZAR EVENTO	108
TABLA XLVIII. PRUEBA FUNCIONAL DE PROMOCIONAR MASCOTA EXTRAVIADA	109

1. Título

**“DESARROLLO DE UNA APLICACIÓN WEB DE
ADOPCIÓN DE MASCOTAS EN LA CIUDAD DE LOJA”**

2. Resumen

El aumento de la natalidad de mascotas como perros y gatos a nivel mundial se encuentra en niveles problemáticos, provocando una disminución de su calidad de vida y dañando el entorno de las ciudades. Esta realidad no es ajena a Ecuador y específicamente a la ciudad de Loja, en donde los casos de mascotas extraviadas y abandonadas son un problema que diariamente afectan a la comunidad local.

El presente trabajo de titulación describe como propósito el desarrollo de una aplicación web usando el framework Django para la adopción de mascotas en Loja (PetHome), que permita mejorar los procesos en favor de las mascotas como: adopción, búsqueda de mascotas extraviadas y eventos.

El proceso de desarrollo se realizó mediante una serie de pasos como la recolección de información, utilizando técnicas como la entrevista y la encuesta. Para el análisis de los datos recolectados se utilizó los métodos: inductivo, deductivo y la revisión sistemática de literatura.

El sistema PetHome utiliza herramientas de programación como Python, el framework Django y como gestor de base de datos PostgreSQL. Estos elementos permiten gran estabilidad y seguridad con la información que se maneja.

La metodología aplicada en el desarrollo de aplicación web fue la AUP (Proceso Unificado Ágil), el cual es una adaptación simplificada del Proceso Unificado Racional, el cual utiliza técnicas ágiles y concepciones utilizados en RUP. La arquitectura implementada para el sistema es la dada por Django MVT (Model-View-Template), que permite separar la parte lógica, interfaz de usuario y el modelo de datos.

Para concluir se aplicaron pruebas de carga y estrés al servidor, para determinar el soporte de conexiones que posee el servidor. Además, se desarrollaron pruebas de usabilidad que permiten comprobar que el diseño y los requerimientos se cumplan según lo establecido en el documento de la especificación de requerimientos.

Abstract

The increase in the birth rate of pets such as dogs and cats worldwide is at problematic levels, causing a decrease in their quality of life and damaging the cities environment. This reality is not stranger to Ecuador and specifically of Loja city, where cases of abandoned pets are a daily problem that affect the local community.

The present research work describes as purpose the development of a web application using the framework Django for the adoption of pets in Loja (PetHome), which allows to improve the processes in favor of the pets like: adoption, search of pets and events.

The development process was performed through a series of steps such as the collection of information, using techniques like an interview and survey.

The analysis of the collected data we used the methods: inductive, deductive and systematic literature review.

The PetHome system uses programming tools like: Python, the Django framework and as a PostgreSQL data base manager. These elements allow great stability and security with the information handled.

The methodology applied in web application development was the AUP (Agile Unified Process), which is a simplified adaptation of the Rational Unified Process, which uses agile techniques and concepts used in RUP. The architecture implemented for the system is given by Django MVT (Model-View-Template), which allows to separate the logical part, user interface and data model.

In conclusion, load and stress tests were applied the server to determine the connection support that the server has. In addition, we developed usability tests that allow to verify the design and requirements are met as established in the document of requirements specification.

3. Introducción

La tecnología web y su evolución ha ganado un gran protagonismo en la sociedad, ya que ha constituido como un instrumento en varias áreas como en la educación, seguridad, accesibilidad y en el desarrollo de aplicaciones web.

La sobrepoblación de mascotas como perros y gatos a nivel mundial crece cada vez más, esta situación ha llevado a que las fundaciones a nivel mundial emprendan campañas a favor de los animales, siendo una problemática aún no solucionada.

Este proyecto presenta una aplicación web utilizando el framework Django, con funciones para la adopción y búsqueda de mascotas extraviadas, además, gestión de eventos para fundaciones colectivos que lo requieran. La aplicación se encuentra disponible en la web en la dirección <https://www.pethomeloja.tk> y posee el nombre de PetHome.

El presente trabajo de titulación está conformado por secciones y anexos que permiten la comprensión de la aplicación web realizada.

En el apartado de *Revisión de Literatura*, se especifica la información teórica correspondiente a las Tecnologías aplicadas a la adopción de mascotas. Esta se realizó utilizando el método de la revisión sistemática de literatura. Además, también se hace referencia a las herramientas de desarrollo utilizadas para realizar este proyecto, donde se describe el framework Django, la base de datos PostgreSQL y el servidor Nginx.

En la sección "*Materiales y Métodos*" se expone las técnicas de recolección de información utilizadas como la entrevista y encuesta. Además, se explica el método deductivo, inductivo y el método de revisión sistemática de literatura.

La sección *Resultados* se hace referencia las fases seguidas utilizando la metodología AUP para realizar este proyecto. En esta se describe los requerimientos funcionales y no funcionales, los diagramas de caso de uso, diagrama de clases, diagrama de secuencia, diagrama de paquetes y componentes.

En la sección "*Discusión*" se expone la propuesta alternativa, en donde se describe los pasos seguidos para dar el cumplimiento de los objetivos planteados, así como la valoración técnica, económica y ambiental.

Finalmente, se redacta las conclusiones obtenidas al finalizar el presente trabajo y las recomendaciones generadas como sugerencia para mejoras en el servicio para la

adopción de mascotas. La Bibliografía es el sustento de la investigación del trabajo realizado y por último los Anexos que sirven de información adicional para la completa justificación y sustento para el proyecto realizado.

4. Revisión de literatura

En esta sección se realiza una introducción a la situación actual en la ciudad de Loja con respecto a la tenencia de mascotas en los hogares, así como la ordenanza vigente implantada por el Municipio de Loja. Luego se realiza una revisión bibliográfica de literatura para identificar los tipos de tecnologías que se ha utilizado para contribuir en el proceso de adopción y búsqueda de mascotas, de manera que pueda apoyar como base en la elección de herramientas para el desarrollo del trabajo de titulación.

El uso de la RSL (Revisión Sistemática de Literatura) permite analizar el equilibrio entre la revisión preliminar de estudios y los resultados, para dar respuesta a la pregunta establecida, en base a los criterios de inclusión y exclusión determinadas.

4.1. Situación actual de la ciudad de Loja

4.1.1. Responsabilidad del gobierno local en el control y protección animal

El artículo 54 del COOTAD en su literal r) expresa: “es función del gobierno municipal, crear las condiciones materiales para la aplicación de políticas integrales y participativas en torno a la regulación del manejo responsable de la fauna urbana”. Por esta razón el Municipio de Loja ha publicado la ordenanza número 39-2011 vigente desde noviembre del 2011 con respecto al control y protección animal, esta puede ser consultada en la dirección <http://www.loja.gob.ec/documentos/ordenanza-municipal-de-control-y-proteccion-animal> .

En esta expresa:

- Que, es deber del Estado y los gobiernos autónomos descentralizados, adoptar políticas y medidas oportunas que garanticen la conservación de la biodiversidad y sus componentes flora y fauna; así como el respeto a los derechos colectivos de la naturaleza;
- Que, cada especie animal es digna de consideración, admiración y conservación, por lo tanto, debemos tratarlos con respeto y evitar situaciones que pongan en riesgo su integridad;
- Que, el incremento de la población canina, asociado al desarrollo urbano, ha derivado en la existencia de animales en la vía y espacios públicos, constituyéndose en riesgo para la salud del colectivo; y

- Que, en gran parte de los seres humanos existe una débil cultura y educación sobre la tenencia y manejo responsable de perros y animales, misma que afecta el factor urbanístico y al desarrollo ambiental local. [1]

Entre los artículos más relevantes en esta ordenanza indican:

Art. 17.- Albergue Público.- Créase por parte de la Municipalidad del cantón Loja, un albergue público como lugar de hospedaje o resguardo temporal de los perros callejeros; perdidos; rescatados; y, requisados, para brindarles atención y seguridad, resguardando su salud y el de la población.[1]

Art. 21.- De la Adopción.- Todo perro que ingresare al albergue público municipal que no fuere retirado en el plazo máximo de treinta días contados desde que hubiere sido recogido, previo criterio médico, entrará en proceso de adopción. El interesado en adoptar deberá llenar un formulario previamente elaborado, mismo que se agregará al expediente levantado desde que el perro ingresó en el albergue público. Si a criterio del Jefe Municipal de Salud, en conjunto con un delegado de la Jefatura Provincial de Salud y un representante de la institución protectora de animales, se autorizará la adopción. El nuevo dueño del perro, durante el primer año, deberá llevarlo trimestralmente hasta el albergue para su valoración, sin perjuicio de que la Jefatura Municipal de Salud, en caso de detectar alguna de las prohibiciones establecidas en la presente ordenanza, ordene su retiro.[1]

El albergue municipal entró en funcionamiento desde el 9 de enero del 2017. Cuenta con capacidad de albergar hasta 100 canes. Para la adopción de mascotas han establecido los días martes y sábados en los parques de la ciudad para realizar esta actividad. Toda la publicación de las mascotas disponibles se las realiza en las redes sociales de la Alcaldía de Loja.

4.2. Tecnologías en la adopción de mascotas

La sobrepoblación de mascotas como perros y gatos abandonados o extraviados a nivel mundial crece cada vez más, por lo que millones entran a los refugios de Estados Unidos, de los cuales, según la Sociedad Americana para la Prevención de la Crueldad contra Animales (ASPCA), alrededor de 4 millones son perros. [2]

Principales hallazgos

En todos los estudios seleccionados (Ver tabla I), se presenta una solución para la adopción y recuperación de mascotas, utilizando cada uno la tecnología como medio para agilizar los procesos.

TABLA I. ESTUDIOS SELECCIONADOS

#	Artículos
ES1	Leveraging Mobile Technology to Increase the Permanent Adoption of Shelter Dogs
ES2	Mobile-Based Distributed System for Managing Abandoned or Lost Pets
ES3	LFD: Lost and Found Dog Application on Mobile
ES4	Speed of Dog Adoption: Impact of Online Photo Traits
ES5	An Evaluation of the Role the Internet Site Petfinder Plays in Cat Adoptions

Los resultados obtenidos de los estudios seleccionados se muestran a continuación:

TABLA II. RESULTADOS DEL ESTUDIO ES1[3]

Problemática	Metodología	Tecnologías, herramientas utilizadas y/o desarrolladas	Conclusiones relevantes
Hay un estimado de 4 millones de perros abandonados en los refugios de Estados Unidos y casi la mitad son sacrificados	Estudio piloto de 8 semanas con la Sociedad Protectora de Animales de Silicon Valley con tecnología proporcionada por Silbato Labs, Inc.	Monitor de actividad	Al ser capaz de ver de forma remota la actividad de su perro, los adoptantes fueron capaces de sentirse más conectados con el perro, lo que contribuyó a la menor tasa promedio de renuncia de adopción.
El problema de los refugios se agrava a casusa del regreso de los animales recién adoptados a los refugios en un 18.8%	Un total de 41 de los 55 perros fueron adoptados durante la duración del estudio. Todos los perros en el grupo experimental llevaban el Whistle Activity Monitor en su cuello.	Humane Society Silicon Valley website Tecnología móvil Widget html Css Aplicación de Smartphone IOS y Android	

TABLA III. RESULTADO DEL ESTUDIO ES2[4]

Problemática	Metodología	Tecnologías, herramientas utilizadas y/o desarrolladas	Conclusiones relevantes
Sociedades Protectoras de Animales (APS) carecen de información comunes, bases de datos y canales para recibir avisos de animales y notificar noticias.	El Sistema posee dos tipos de usuario: administrador y anunciantes Los anunciantes pueden enviar dos tipos de información: datos acerca de las características de la mascota y datos sobre la localización, fecha, hora y contacto del usuario. Un web service recibe los datos y los envía a una base de datos. El sistema procesa los datos del animal para determinar si existe en la base de datos otro similar o no. En el caso de existir, el usuario administrador es notificado de la localización de la mascota para su rescate.	Web service Framework Sencha Touch PHP MySQL Java Android SDK	Consiste en lograr un nivel de intercalación que permite aumentar la localización de los animales perdidos, delimitando la búsqueda a los lugares recientes en los que el animal ha estado. Dispositivos móviles inteligentes, computación en la nube y optimización de almacenamiento de datos pueden proporcionar apoyo tecnológico para APS

TABLA IV. RESULTADOS DEL ESTUDIO ES3[5]

Problemática	Metodología	Tecnologías, herramientas utilizadas y/o desarrolladas	Conclusiones relevantes
El método más general que las personas utilizan cuando perdieron su perro es tratar de hacer el anuncio a través de muchas redes sociales o lugares públicos con la esperanza de que alguien va a ver a su perro, pero casi no sucede.	Se realizó una búsqueda de aplicaciones similares que sirvan como guía para el desarrollo de la aplicación.	Android Activity Lifecycle SQLite Java Plataforma Android	La aplicación ha desarrollado una base de datos sistemática para recopilar información de perros perdidos. Esta base de datos permite que la gente tenga una fuente de información común útil.

TABLA V. RESULTADOS DEL ESTUDIO ES4[6]

Problemática	Metodología	Tecnologías, herramientas utilizadas y/o desarrolladas	Conclusiones relevantes
<p>2 a 6 millones de perros y gatos son sacrificados cada año en los Estados Unidos a causa de la sobrepoblación en los refugios.</p> <p>Investigaciones han demostrado que ciertos rasgos de perros afectan el adoptabilidad o la velocidad de adopción incluyendo raza, color, edad, y el comportamiento</p>	<p>El estudio se realizó en mezclas de Labrador Retriever de edades entre jóvenes y adultos de Estados Unidos.</p> <p>Los datos fueron extraídos del sitio web Rescuegroups.org en el periodo 1 de enero del 2011 al 15 de junio del 2012.</p> <p>Únicamente perros que hayan sido adoptados en 5 meses fueron incluidos en el estudio</p>	Fotografías	<p>Este estudio demostró que los rasgos y calidad de fotos en línea pueden afectar a la velocidad en la adopción de perros Labrador Retriever, tanto adultos como jóvenes.</p>

TABLA VI. RESULTADO DEL ESTUDIO ES5[7]

Problemática	Metodología	Tecnologías, herramientas utilizadas y/o desarrolladas	Conclusiones relevantes
<p>Millones de gatos entran en los refugios de animales no humanos cada año, y el 71% de los gatos que entran en estos refugios son sacrificados (Asociación Protectora de Animales de América, 2013).</p> <p>Estudios han demostrado que la apariencia y el comportamiento de un animal son fuertes predictores de las decisiones de adopción</p>	<p>Se analizaron los expedientes de adopción de gatos que entraron en el refugio Petpoint de Nueva York del 15 de octubre del 2012 al 6 de octubre del 2013.</p> <p>Petfinder envió informes semanales al refugio en los que incluía los nombres de los gatos y el número de clics que haya recibido antes de la adopción.</p>	<p>Internet</p> <p>Fotografías</p> <p>Sitio web Petfinder.com</p>	<p>Los gatos que fueron populares en Petfinder, como lo demuestra el número de clics, fueron los que estuvieron disponibles para la adopción por poco tiempo. Esto se vio influenciado por el tipo de foto y color del gato.</p>

4.3. Herramientas de desarrollo

4.3.1. Lenguaje de Programación Python

Python es un lenguaje de programación orientado a objetos de alto nivel creado por Guido van Rossum. Su sintaxis es simple y fácil de usar, por lo que su curva de aprendizaje es más veloz. [8]

4.3.1.1. Características

- Su sintaxis es simple y elegante por lo tanto la lectura y escritura de programas es mucho más sencillo en comparación con otros lenguajes de programación como C++, Java o C#.
- Su distribución es libre tanto para proyectos personales como comerciales. Además, se puede contribuir en la comunidad realizando cambios en el código fuente de Python.
- Python se puede ejecutar en casi cualquier plataforma sin ningún cambio, incluyendo Windows, Mac OS X y Linux.
- Se puede combinar bloques de código de otros lenguajes en Python mejorando así la capacidad de rendimiento del programa.
- El código en Python no necesita ser compilado para su ejecución, ya que se convierte automáticamente en código para el idioma que entiende el ordenador.
- Existe una gran cantidad de bibliotecas con documentación que facilitan la programación en Python. Estas bibliotecas son probadas y utilizadas por cientos de personas, así que su implementación es fiable.
- En Python todo es un objeto, por esta razón es capaz de dividir problemas complejos en conjuntos más pequeños mediante la creación de objetos.

4.3.2. Frameworks para Python

4.3.2.1. Framework TurboGears

TurboGears es un framework de aplicaciones web de Python, que se compone de varios módulos. Está diseñado en torno a la arquitectura MVC, que son similares a Ruby on Rails o puntales. TurboGears están diseñados para hacer que el desarrollo rápido de aplicaciones web en Python más fácil y más soportable. TurboGears es un framework de aplicaciones web escrito en Python. [9]

- **Modelo:** Es el responsable de gestionar los datos de la aplicación.
- **Vista:** Se encarga de mostrar los datos a los usuarios.
- **Controlador:** código de software que controla las interacciones entre el modelo y la vista. [10]

TurboGears se constituye por bibliotecas y herramientas dependiendo de la versión, como son:

- SQLAlchemy: es un kit SQL para ORM en código Python.
- Genshi: motor de plantillas para el front-end en turbogears.
- ToscaWidgets: es una librería para la generación de formularios HTML.

4.3.2.2. Framework Pyramid

Es un framework Python para facilitar la escritura de aplicaciones web. Funciona en todas las versiones compatibles con Python. Se basa en el patrón Modelo-Vista-Controlador (MVC). [11]

4.3.2.3. Framework Django

Django es un framework web de Python de alto nivel que fomenta el rápido desarrollo y diseño de páginas web. Es de código abierto y libre. [12]

Django es software libre, por lo que su código puede ser modificado según las necesidades del desarrollador. Cuenta con una comunidad de usuarios activa, lo que permite mantener un soporte permanente ante la presencia de cualquier error.

Características

- Django busca una programación rápida de páginas y aplicaciones web.
- Se basa en la filosofía DRY (Don't repeat yourself).
- Django se encuentra basado en el paradigma Modelo-Vista-Plantilla.
- Posee una base de datos embebida con SQLite. También soporta otras bases de datos como MySQL, PostgreSQL, MS-SQL.
- Cuenta con un sistema ORM para la conexión a base de datos.
- Django es compatible con Python 3.

Arquitectura de Django

Django propone una arquitectura MVT (model-view-template), donde:

- **Model:** hace referencia a la capa de acceso a la base de datos.

- **Views:** contiene la lógica de programación y el medio entre la base de datos y la plantilla.
- **Template:** esta capa de presentación puede ser un html, xml, css, javascript, cvs, etc.

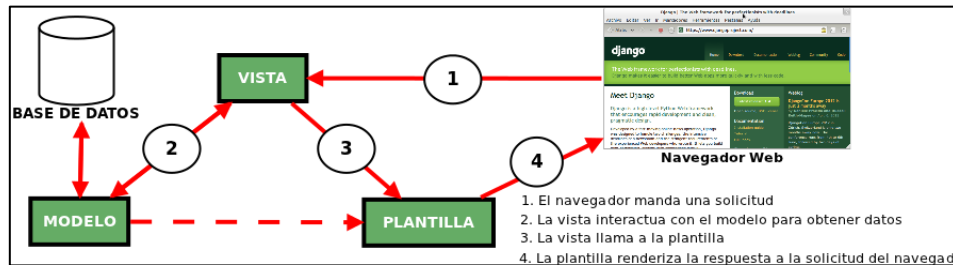


Figura 1. Arquitectura de Django Framework

Estructura de Django

Cuando se inicia un nuevo proyecto en Django se crea automáticamente una estructura de archivos predeterminados, estos son:

```
mysite/
  manage.py
  mysite/
    __init__.py
    settings.py
    urls.py
    wsgi.py
```

Figura 2. Estructura de archivos Django

- **mysite/:** es el directorio raíz que contiene todo el proyecto. Puede tener cualquier nombre.
- **manage.py:** archivo de ejecución que permite interactuar con el proyecto de Django.
- **mysite/:** es el directorio real del proyecto Django. Este es el nombre con el que se debe importar paquetes Python.
- **mysite/__init__.py:** es un archivo que le dice a Python que debe considerar el directorio como paquete Python.
- **mysite/settings.py:** contiene las configuraciones del proyecto Django.
- **mysite/urls.py:** posee una tabla de contenidos con las URL del proyecto Django.

- **mysite/wsgi.py:** es una entrada para los servidores compatibles con wsgi para servir al proyecto Django. [13]

Administración de Django

Una de las características más importantes es el sitio de administración automática. En el mismo se leen los metadatos de los modelos y proporciona una interfaz para la gestión del contenido.

El administrador viene habilitado por defecto. Para hacer uso del mismo se necesita lo siguiente:

- Añadir 'django.contrib.admin' en la sección INSTALLED_APPS del archivo settings.py.
- Determinar los modelos que se desean editar mediante el sitio de administración. Estos se los especifica en el archivo admin.py.
- Añadir el sitio de administración al archivo urls.py.

4.3.2.4. Comparativa de Frameworks basados en Python

Para la siguiente comparativa, se tomó en consideración los frameworks Django, TurboGears, Pyramid, los cuales son los más potentes y aceptados por la comunidad de desarrolladores web.

TABLA VII. COMPARATIVA DE FRAMEWORKS PYTHON

Características	FRAMEWORK		
	Django	TurboGears	Pyramid
Compatible con Python 3.5	X		X
Posee ORM	X	X	X
Base de datos embebida	X	X	X
Múltiples BD	X	X	X
Motor de Plantillas	X	X	X
Desarrollo full stack	X	X	
Administración incorporada	X	X	
Soporte multilenguaje	X		X
Comunidad	X		
Documentación oficial	X	X	X
Documentación en español	X		

Como muestra la TABLA VII, el framework que mayores características posee es Django, en relación con los demás. Este framework fue el elegido para el desarrollo del presente proyecto.

4.3.3. Materialize Framework

Creado y diseñado por Google, Material Design es un lenguaje de diseño que combina los principios clásicos del diseño exitoso junto con la innovación y la tecnología [14]. Ofrece recursos CSS, javascript y soporte para personalización mediante Sass.

Características

- Permite crear interfaces responsivas.
- Menús desplegados en función de la resolución.
- Filosofía Material Design
- Posee Parallax, pushpin y scrollspy.

4.3.4. PostgreSQL

Es un sistema de gestión de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. [15]

Características

- Cumplimiento del estándar SQL, siguiendo el último SQL:2011
- Arquitectura cliente-servidor con un amplio rango de drivers y clientes
- Diseño de alta concurrencia donde los lectores y escritores no se bloquean
- Altamente configurable y extensible para muchos tipos de aplicación
- Excelente escalabilidad y rendimiento con características de ajustes extensas
- Optimizador de consultas sofisticado, adecuado para inteligencia de negocios
- Soporta totalmente el acceso y procedimientos de base de datos en Java, Python, perl, PHP y muchos más.
- Altamente confiable con características extensas para durabilidad y alta disponibilidad.
- Tipos de datos avanzados como Información Geográfica, búsqueda completa de texto y más
- Características de internacionalización, codificaciones multibyte y colaciones.
- Excelente soporte. [16]

Aunque PostgreSQL tiene alta escalabilidad, posee algunos límites que se detallan a continuación:

TABLA VIII. VALORES DE ALMACENAMIENTO DE POSTGRESQL

Límite	Valor
Tamaño máximo de base de datos	Ilimitado
Tamaño máximo de tabla	32 TB
Tamaño máximo de registro	1.6 TB
Tamaño máximo de campo	1 GB
Máximos de registros por tabla	Ilimitado
Máximo de columnas por tabla	250 – 1600 dependiendo del tipo de columnas
Máximo de índices por tabla	Ilimitado

4.3.5. Django Rest Framework

Django Rest Framework es una herramienta potente y flexible para la construcción de API Web. [17]

En la actualidad los servicios Rest son esenciales en proyectos o aplicaciones para el consumo de datos y así obtener una separación entre cliente y servidor, para una escalabilidad del producto sin excesivos problemas.

Entre las principales características se encuentran:

- Existen políticas de autenticación, incluyendo paquetes para OAuth1 y OAuth2.
- Serialización, ya que soporta fuentes de datos ORM y sin ORM.
- Documentación extensa y una gran comunidad de apoyo.

Posee gran cantidad de código reutilizable y una interfaz de administración donde es posible realizar pruebas con operaciones como GET y POST.

Los métodos disponibles son:

- GET: consulta y lee recursos.
- POST: crear recursos.
- PUT: edita recursos.
- DELETE: elimina recursos.

- PATCH: editar partes concretas de recursos.

Requisitos

Django Rest requiere lo siguiente: [17]

- Python (2.7, 3.2, 3.3, 3.4, 3.5)
- Django (1.8, 1.9, 1.10)

4.3.6. Servidor Nginx

Nginx es uno de los servidores web más populares del mundo y es responsable de alojar algunos de los sitios más grandes y de mayor tráfico en Internet. Es más fácil de usar que Apache en la mayoría de los casos y puede usarse como un servidor web o un proxy inverso. [18]

Es de código abierto de alto rendimiento, incluye servicios de correo electrónico. Además, NGINX está listo para ser utilizado como un proxy inverso. En este modo, NGINX se utiliza para equilibrar la carga entre los servidores back-end, o para proporcionar almacenamiento en caché para un servidor back-end lento. [19]

Características

- Tiene la capacidad de manejar más de 10000 conexiones simultáneas con consumo bajo de memoria.
- Balanceo de carga.
- Alta tolerancia a fallos.
- Posee soporte para TLS, FastCGI, SSL, SCGI o uWSGI, etc.
- Compatible con IPv6.
- Reescritura de urls para la ayuda al posicionamiento web, modificándolas desde su configuración.
- Es posible limitar el número de conexiones.
- Geolocalización basada en direcciones IP.

5. Materiales y Métodos

5.1. Técnicas de recolección de información

En el desarrollo del trabajo de titulación se realizó la recolección de información aplicando diferentes tipos de técnicas a los actores que interactuaran con la aplicación web y en especial a la Fundación “Ángeles con Patas”.

Entrevista

La técnica de la entrevista se aplicó a los miembros de la Fundación “Ángeles con patas” para lograr recolectar los requisitos, que fueron indispensables en el desarrollo de la aplicación web. (Ver Anexo VIII)

Encuesta

Se usó la técnica de la encuesta para conseguir información sobre los problemas que tiene la comunidad en el trato con los animales, aplicándolos a una parte de la población de la ciudad de Loja, además, se la utilizó para realizar las pruebas de usabilidad a los actores que interactúan con la aplicación web. (Ver Anexo IX)

5.2. Métodos

5.2.1. Método Deductivo

Con el método deductivo se partió de aspectos generales para llegar a los específicos, con el cual fue de ayuda para la recopilación de la información general para poder plantear el problema del proyecto utilizando técnicas como la encuesta y la entrevista.

5.2.2. Método Inductivo

El método inductivo propone obtener conclusiones generales a partir de premisas particulares, con el cual se comprobó que la aplicación desarrollada cumpla con los requerimientos planteados en la fase de planeación, permitiendo corregir errores en la etapa de pruebas.

5.2.3. Método de Revisión Sistemática de Literatura

En este apartado, se hace referencia a la revisión sistemática realizada, con la finalidad de recabar información relevante para el estudio de las tecnologías aplicadas a favor de los animales. La revisión sistemática de literatura permite identificar, evaluar e

interpretar las investigaciones que se encuentran valederas para satisfacer una pregunta específica planteada de interés.

La estructura a seguir viene planteada por la propuesta de Kitchenham[20] y Biolchini [21], en el cual define las siguientes etapas: planificación de la SLR, ejecución de la SLR, extracción de información y resultados.

5.2.3.1. Planificación de la SLR

Para comenzar con la SLR, se realiza una planificación, en el cual se especifica el objetivo de la investigación.

Objetivo SLR

El objetivo planteado para esta investigación es:

Conocer los casos de éxito que hayan contribuido al proceso de adopción de mascotas

Pregunta de investigación

Se ha trazado una pregunta de investigación a partir de un estudio preliminar:

¿Qué tipo de tecnologías han favorecido en la elección de perfiles con respecto a la adopción de una mascota?

Términos

La identificación de las palabras claves que servirán de ayuda en la construcción de las cadenas de búsqueda viene dadas a partir de una revisión preliminar de información en artículos, libros y ponencias.

TABLA IX. ESTUDIOS PRELIMINARES Y TÉRMINOS

#	Estudio	Términos
E1	Leveraging Mobile Technology to Increase the Permanent Adoption of Shelter Dogs.	Animal-computer; interaction; quantimetrics; adoption
E2	LFD: Lost and Found Dog Application on Mobile	Mobile application; database; faster and effective searching; lost dogs; tool
E3	Desarrollo de una aplicación móvil como estrategia de difusión digital para el proceso	Aplicativo Móvil; adopción; abandono; responsabilidad social; animal; Guayaquil

	de adopción de perros y gatos en la ciudad de Guayaquil.	
E4	Speed of Dog Adoption: Impact of Online Photo Traits.	Internet; adoption; dog; photos; rehoming
E5	Buscador de adopción de perros haciendo uso de inteligencia artificial, obteniendo afinidad del futuro dueño	Diseño; interfaz; adopción y perros
E6	Estudio de la problemática actual de adopción de mascotas en la ciudad de Guayaquil y desarrollo de un aplicativo web para agilizar este proceso	Adopción de Mascotas, Mascotas; Fauna Urbana; Rescate de Mascotas; Refugio; Acogimiento de Animales; Aplicativo de Adopción Online

Una vez revisado cada estudio, se identificó las similitudes en los términos para así construir las cadenas de búsqueda.

Selección de fuentes y estrategias de búsqueda

Fuentes

- ACM: <http://dl.acm.org/>
- IEEE Xplorer: <http://ieeexplore.ieee.org/>
- Scopus: <https://www.scopus.com/>

Estrategias de búsqueda

A partir de la pregunta de investigación, se plantearon las palabras claves para las búsquedas específicas:

Adoption, pets, animal, lost, application, mobile, dogs, cats, Android, web, lost dog, internet, design, photo, animal shelter, canine, protection.

Utilizando los operadores AND, OR y las palabras claves seleccionadas, se realizaron diferentes combinaciones que permitan recuperar información adicional

Las búsquedas elaboradas fueron las siguientes:

TABLA X. CADENAS DE BÚSQUEDA

Biblioteca digital IEEE Xplorer: http://ieeexplore.ieee.org/
CD1: ((dogs OR pets) AND (web OR android) AND (information OR characteristics OR adoption))
Biblioteca digital SCOPUS: https://www.scopus.com/
CD2: PUBYEAR > 2010 AND ALL ("dog") AND TITLE-ABS-KEY (("mobile application" OR "design" OR "internet" OR "photos" OR "application" OR "web" OR "database") AND ("adoption" OR "lost dog" OR "protection" OR "lost") AND ("animal" OR "dog" OR "pets" OR "cats" OR "canine" OR "petfinder") AND ("animal shelter" OR "rehoming" OR "shelter"))
Biblioteca digital ACM: http://dl.acm.org/
CD3: (("mobile application" OR "design" OR "internet" OR "photos" OR "application" OR "web" OR "database") AND ("adoption" OR "lost dog" OR "protection" OR "lost") AND ("animal" OR "dog" OR "pets" OR "cats" OR "canine" OR "petfinder") AND ("animal shelter" OR "rehoming" OR "shelter"))

Criterios de inclusión

Para el desarrollo de este trabajo, se tuvo en consideración los siguientes criterios, a fin de obtener los mejores resultados para cumplir el objetivo planteado. Se ha considerado lo siguiente:

- **Idioma:** español e inglés
- **Motores de búsqueda:** ACM, Scopus, IEEE Xplorer
- **Fecha de publicación:** estudios publicados después del 2010
- **Tipos de producción científica:** artículos, libros y ponencias

Criterios de exclusión

Los resultados de la búsqueda que no han sido relevantes con el objetivo planteado han sido descartados tomando en cuenta los siguientes puntos:

- Aquellos artículos que no definan un tipo de tecnología o herramientas de diseño y almacenamiento.
- Artículos que no tengan relación con el área de la computación, ingeniería o veterinaria.
- Temas irrelevantes como:

- Estudios en medicina para animales
- Estudios que hacen referencia al comportamiento de animales

5.2.3.2. Ejecución de la SLR

Se desarrolló la búsqueda utilizando las cadenas como muestra en la Tabla 2. Una vez realizadas las búsquedas, se tomó los siguientes criterios a cumplir por parte de los estudios:

Presenten herramientas de diseño o almacenamiento de datos.

Presenten tipos de tecnologías utilizadas en el ámbito animal.

Exhiban resultados favorables en ayuda a asociaciones de protección animal.

Extracción de la información

Para la extracción de información, se tuvo en cuenta los criterios de selección para determinar los estudios que sean relevantes para el presente trabajo. Para cada estudio se extrajo información de al menos uno de estos campos:

- Problemática
- Metodología
- Tecnología
- Herramientas utilizadas y/o desarrolladas
- Conclusiones relevantes

Estudios incluidos y excluidos

Aplicadas las cadenas de búsqueda en las diferentes bases científicas se obtuvo un total de 59 estudios, de los cuales 1 se repitió, aplicando los criterios de inclusión, exclusión y de selección se han escogido 6 artículos que ayudan a cumplir el objetivo planteado.

TABLA XI. ESTUDIOS SELECCIONADOS

#	Artículos
ES1	Leveraging Mobile Technology to Increase the Permanent Adoption of Shelter Dogs
ES2	Mobile-Based Distributed System for Managing Abandoned or Lost Pets
ES3	LFD: Lost and Found Dog Application on Mobile

ES4	Speed of Dog Adoption: Impact of Online Photo Traits
ES5	Distributed system for intelligent management of lost or abandoned pets
ES6	An Evaluation of the Role the Internet Site Petfinder Plays in Cat Adoptions

5.2.3.3. Principales hallazgos

En todos los estudios se presenta una solución para la adopción y recuperación de mascotas, utilizando cada uno la tecnología como medio para agilizar los procesos.

- En los trabajos [S01], [S02] y [S04] se plantea la utilización de la tecnología móvil mediante aplicaciones tanto para IO's y Android, como solución para aumentar el número de adopciones y disminuir la tasa de devolución de mascotas a los refugios.
- [S02] y [S03] toma en cuenta como problemática la falta de comunicación y la carencia de información en común entre las sociedades protectoras de animales. Plantean la solución mediante la creación de un Servicio Web, el mismo que se encuentra como intermediario entre las fundaciones y una base de datos central construida en MySQL. La interfaz móvil ha sido desarrollada con el framework Sencha Touch, además se utiliza la API de voz de Google para que sea más fácil el ingreso de datos por parte del usuario.
- Otra solución que se ha implantado es el efecto de las fotografías en las preferencias de adopción, este tema se habla en el estudio [S05], donde demuestra que la calidad y tipo de fotografía que se encuentra en las Webs de las protectoras de animales influyen en la decisión de adopción.
- El estudio [S06] hace una evaluación del sitio web Petfinder y la influencia del internet y fotografías en la selección de mascotas, por lo que el uso de estas dos tecnologías es de principal apoyo en el éxito de encontrar un hogar para los animales.

TABLA XII. TECNOLOGÍAS OBTENIDAS DEL SLR

Tecnologías	Características	Costo
Monitor de actividad Silbato Labs	<ul style="list-style-type: none"> • Monitor de ritmo cardiaco • Controlador de actividad física • Control de descanso • Monitoreo de las tendencias de salud 	\$ 143.30
Sitio web	<ul style="list-style-type: none"> • Diseño basado en estándares • Maquetación por capas • Usabilidad y portabilidad • Accesibilidad web 	Dependiendo de las herramientas
Widget html	<ul style="list-style-type: none"> • Aplicaciones externas fácil de agregar a una página web 	
Web service	<ul style="list-style-type: none"> • Aportan interoperabilidad entre aplicaciones de software, independientemente de la plataforma • Estándares y protocolos • Compartir datos entre diferentes compañías o software • Independencia de lenguaje de programación [22] 	Gratuito
Framework Sencha Touch	<ul style="list-style-type: none"> • Multiplataforma • Diseñado para dispositivos táctiles • Integración de pruebas funcionales y unitarias 	Estándar: \$695 Premium: \$1195
Python	<ul style="list-style-type: none"> • Distribución libre • Sintaxis simple • Código no necesita ser compilado 	Gratuito
PHP	<ul style="list-style-type: none"> • Programación orientada a objetos • Soporte a variedad de bases de datos • Integración con bibliotecas externas 	Gratuito
MySQL	<ul style="list-style-type: none"> • Multiplataforma 	Gratuito

	<ul style="list-style-type: none"> • Disponible para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl. • Escalabilidad • Registros de longitud fija y longitud variable. 	
SQLite	<ul style="list-style-type: none"> • La base de datos completa se encuentra en un solo archivo • Puede funcionar enteramente en memoria • Dispone de librerías de acceso para muchos lenguajes de programación. 	Gratuito
Java	<ul style="list-style-type: none"> • Orientado a objetos • Portable • Lenguaje compilado • Lenguaje seguro 	Gratuito
Fotografías	<ul style="list-style-type: none"> • Estimulación visual • Transmite mensajes o ideas 	Gratuito
Sitio web petfinder.com	<ul style="list-style-type: none"> • Usa la tecnología del internet para permitir la adopción de mascotas • Brinda widgets gratuitos para páginas web externas 	Gratuito

Como se puede observar en la tabla, existen diferentes tipos de tecnologías utilizadas para el apoyo de animales. Cada una posee características y valores diferentes que se acoplan a las necesidades y recursos de cada proyecto.

Este estudio sirvió como base para la selección de algunas tecnologías para el desarrollo del trabajo de titulación. Entre ellas se tomó como referencia el sitio web petfinder.com y el lenguaje de programación Python.

5.3. Metodología de desarrollo de software

La metodología utilizada para el desarrollo del proyecto de fin de carrera es la AUP (proceso unificado ágil), la cual se divide en cuatro fases:

- Fase de concepción
- Fase de elaboración
- Fase de construcción
- Fase de transición

Fase de concepción

Esta fase consistió en la recolección de información, obteniéndola de la representante de la fundación “Ángeles con patas”, a través de las técnicas descritas anteriormente. De esta forma se pudo conocer los problemas y necesidades que poseen las fundaciones y colectivos en la ciudad de Loja. Estas necesidades se plasmaron haciendo uso del estándar IEEE 830 y así se obtuvo una especificación de requerimientos, identificación de actores y elaboración de la especificación de casos de uso, que son base fundamental para el desarrollo de la aplicación web.

Fase de elaboración

El objetivo principal de esta fase es profundizar en la comprensión de los requerimientos y validar la arquitectura del sistema que sirvió de punto de partida para la fase de elaboración. La arquitectura del sistema detalla los elementos con la que está diseñada y su relación entre ellos. Seguidamente se realizó el prototipado de la aplicación para poder visualizar el diseño y la distribución de elementos en la web. Además, se definió el diagrama de clases, que representa las clases y sus relaciones implementadas en el sistema “PetHome”. Así mismo el diagrama de paquetes que representa la forma en que está organizados y agrupados los paquetes con sus elementos proporcionando una mejor visualización del sistema y el diagrama de despliegue que muestra como está implementado de la forma cliente-servidor.

Fase de construcción

Esta fase comprende la implementación del sistema tomando en cuenta la arquitectura conseguida de la fase anterior, obteniendo así la versión ejecutable del producto cumpliendo con los requisitos y necesidades planteadas. Su construcción fue realizada con la ayuda del framework Django en su versión 1.9, basado en el lenguaje de programación Python.

Fase de transición

Comprende la fase final del proyecto, en el cual se lleva a cabo el despliegue de la aplicación en función del usuario final, así como las pruebas de validación en el cual se corrobora que los requerimientos sean cumplidos en su totalidad y que el rendimiento y usabilidad sea óptimo.

6. Resultados

La sección de resultados detalla el proceso seguido en cada fase para la elaboración de la aplicación web “PetHome”. La metodología AUP (proceso unificado ágil), proporcionó la guía de desarrollo mediante las fases de concepción, elaboración, construcción y transición.

6.1. Fase de concepción

6.1.1. Definición de los requisitos

Esta fase consistió en la recolección de información, obteniéndola de la representante de la fundación “Ángeles con patas”, a través de las técnicas de recolección descritas anteriormente. De esta forma se pudo conocer los problemas y necesidades que poseen las fundaciones y colectivos en la ciudad de Loja.

Con la aplicación de estas técnicas, se obtuvo una visión más clara de los problemas y así determinar las posibles soluciones.

6.1.2. Determinación de Requerimientos

Luego de haberse realizado el análisis previo, se logró determinar los requerimientos del sistema con el objetivo de mejorar el proceso de adopciones de mascotas en la ciudad de Loja. Estos requerimientos se clasifican en dos grupos: funcionales y no funcionales, y se los describe a continuación:

6.1.2.1. Requerimientos funcionales

TABLA XIII. REQUERIMIENTOS FUNCIONALES

El sistema permitirá:		
CÓDIGO	FUNCIÓN	CATEGORÍA
RF01	Registro de usuarios	Evidente
RF02	Autenticación de usuarios	Evidente

RF03	Validar el rol del usuario que inicia sesión	Evidente
El sistema permitirá al administrador:		
CÓDIGO	FUNCIÓN	CATEGORÍA
RF04	Crear, editar o eliminar usuarios	Evidente
RF05	Crear, editar o eliminar mascotas	Evidente
RF06	Crear, editar, o eliminar razas	Evidente
El sistema permitirá al usuario		
CÓDIGO	FUNCIÓN	CATEGORÍA
RF07	Crear, editar o eliminar mascotas	Evidente
RF08	Reportar mascotas extraviadas	Evidente
RF09	Solicitar adopción de mascotas	Evidente
RF10	Visualizar eventos de fundaciones	Evidente
RF11	Reportar mascotas encontradas	Evidente
RF12	Modificar información del perfil	Evidente
RF13	Modificar contraseña del usuario	Evidente
El sistema permitirá a la fundación		
CÓDIGO	FUNCIÓN	CATEGORÍA
RF14	Crear, editar o eliminar mascota en adopción	Evidente
RF15	Visualizar historial de adopciones	Evidente
RF16	Recibir solicitudes de adopción por parte de los usuarios	Evidente
RF17	Cambiar estado de la solicitud de adopción	Evidente
RF18	Enviar correo electrónico al solicitante de la adopción	Evidente
RF19	Generar reporte de la solicitud de adopción	Evidente
RF20	Visualizar mascotas extraviadas	Evidente
RF21	Crear, editar o eliminar eventos	Evidente
RF22	Modificar información del perfil	Evidente
RF23	Modificar contraseña del usuario	Evidente

6.1.2.2. Requerimientos no funcionales

TABLA XIV. REQUISITOS NO FUNCIONALES

RNF01	Requisitos de rendimiento
<ul style="list-style-type: none">• EL sistema deberá permitir un acceso simultáneo de hasta 100 conexiones sin afectar su rendimiento.• El sistema debe tener un tiempo de respuesta aceptable.	
RNF02	Requisitos de seguridad
<ul style="list-style-type: none">• El sistema utilizará https para la navegación de los usuarios.	
RNF03	Requisitos de fiabilidad
<ul style="list-style-type: none">• Todo tipo de operación o solicitud de información debe realizarse sin ningún tipo de conflicto y sin incidentes.	
RNF04	Requisitos de disponibilidad
<ul style="list-style-type: none">• El sistema debe estar disponible las 24 horas del día. En caso de algún fallo, el sistema debe recuperarse en el menor tiempo posible.	
RNF05	Requisitos de mantenibilidad
<ul style="list-style-type: none">• Una vez terminado la aplicación web, las tareas de administración y mantenimiento serán realizados por la desarrolladora.• El sistema podrá ampliarse con nuevas funcionalidades según como se requiera.	
RNF06	Requisitos de portabilidad
<ul style="list-style-type: none">• El sistema será multiplataforma ya que se podrá ejecutar desde cualquier navegador web.	

6.1.3. Diagrama de casos de uso

Los casos de uso representan la manera en que los actores interactúan con el sistema, además, captan las funciones visibles para el usuario.

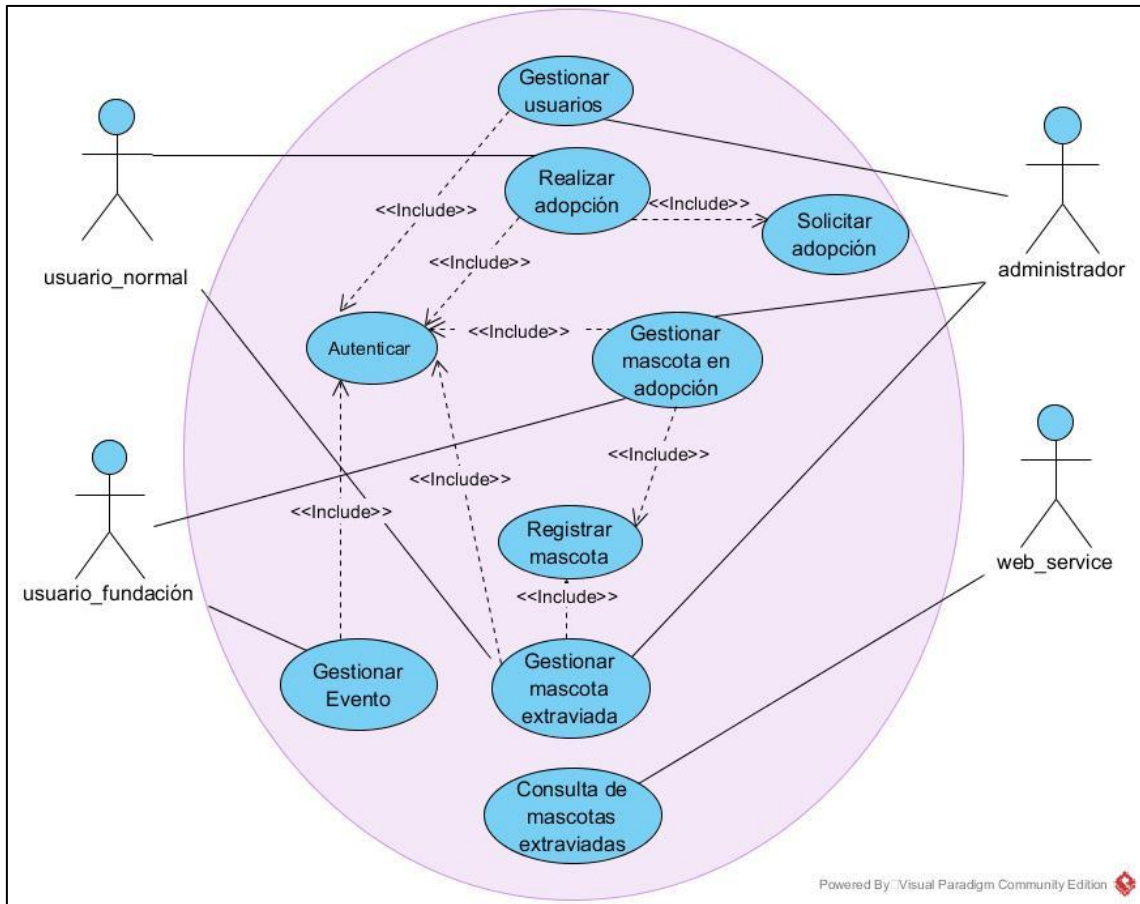


Figura 3. Diagrama de casos de uso

6.1.4. Especificación de casos de uso

TABLA XV. DESCRIPCIÓN C.U. GESTIONAR USUARIOS

Caso de uso:	Gestionar usuario	Actor:	Administrador
Código:	CU001	Tipo:	Aplicación web
Ref. Req:			
Resumen:	El sistema permitirá crear, modificar y eliminar usuarios.		
Objetivos:	<ul style="list-style-type: none"> • Crear usuario • Modificar usuario • Eliminar usuario 		
Precondiciones:	<ul style="list-style-type: none"> • Ingresar al sistema 		
Postcondiciones:	<ul style="list-style-type: none"> • Usuario creado • Usuario modificado 		

	<ul style="list-style-type: none"> • Usuario eliminado
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. Iniciar sesión como usuario-administrador. 2. Ingresar a la administración del sistema web. 3. Seleccionar el campo “usuarios”. 4. Seleccionar “agregar usuario”. 5. Llenar los campos obligatorios (username, nombres, apellidos, correo electrónico, contraseña, grupo de usuario). 6. El sistema verifica que los campos sean correctos y que el usuario no exista anteriormente. 7. El sistema guarda los datos del nuevo usuario. 	
Flujo alterno de eventos	
A. Modificar usuario.	
A.1 El administrador selecciona el usuario que desea modificar.	
A.2 Modifica los campos que desee.	
A.3 El sistema guarda los cambios y actualiza la información	
B. Eliminar usuario.	
B.1 El administrador selecciona el usuario que desea eliminar.	
B.2 Selecciona la opción de eliminar.	
B.3 El sistema muestra un mensaje de confirmación.	
B.4 El administrador selecciona la opción “sí, estoy seguro”.	
B.5 El sistema elimina el usuario.	

TABLA XVI. DESCRIPCIÓN C.U. GESTIONAR MASCOTAS EN ADOPCIÓN

Caso de uso:	Gestionar mascotas en adopción	Actor:	Usuario-fundación
Código:	CU002	Tipo:	Aplicación web
Ref. Req:			
Resumen:	El sistema permitirá crear, modificar y eliminar mascotas en adopción.		
Objetivos:	<ul style="list-style-type: none"> • Crear mascota en adopción • Modificar mascota en adopción • Eliminar mascota en adopción 		
Precondiciones:	<ul style="list-style-type: none"> • Ingresar al sistema 		

Postcondiciones:	<ul style="list-style-type: none"> • Mascota en adopción creada • Mascota en adopción modificada • Mascota en adopción eliminada
Flujo normal de eventos	
1. Iniciar sesión como usuario-fundación. 2. Seleccionar “mis mascotas”. 3. Seleccionar “nueva adopción”. 4. Elegir el tipo de mascota a registrar (perro, gato). 5. Llenar el formulario de registro de mascota en adopción. 6. El sistema verifica que los campos sean correctos. 7. El sistema guarda los datos de la nueva mascota en adopción.	
Flujo alterno de eventos	
A. Modificar mascota en adopción.	
A.1 El actor selecciona la mascota que desea modificar.	
A.2 Modifica los campos que desee.	
A.3 El sistema guarda los cambios y actualiza la información	
B. Eliminar mascota en adopción.	
B.1 El actor selecciona la mascota que desea eliminar.	
B.2 Selecciona la opción de eliminar.	
B.3 El sistema muestra un mensaje de confirmación.	
B.4 El actor selecciona la opción “aceptar”.	
B.5 El sistema elimina la mascota en adopción.	

TABLA XVII. DESCRIPCIÓN C.U. REALIZAR ADOPCIÓN

Caso de uso:	Realizar adopción	Actor:	Usuario-general
Código:	CU003	Tipo:	Aplicación web
Ref. Req:			
Resumen:	El sistema permitirá solicitar una mascota en adopción.		
Objetivos:	<ul style="list-style-type: none">• Enviar solicitud de adopción de una mascota		
Precondiciones:	<ul style="list-style-type: none">• Ingresar al sistema• Existencia de mascota en adopción		
Postcondiciones:	<ul style="list-style-type: none">• Solicitud de adopción enviada		
Flujo normal de eventos			
1. Iniciar sesión como usuario-general.			

2. Seleccionar “mascotas en adopción”.
3. Escoger la mascota deseada para adoptar.
4. Seleccionar “solicitar adopción”.
5. Llenar el formulario de solicitud de adopción.
6. El sistema verifica que los campos sean correctos.
7. El sistema guarda los datos de la nueva solicitud de adopción y envía la solicitud a la fundación correspondiente.
Flujo alternativo de eventos
A. Campos vacíos
A.1 El sistema muestra mensaje de campos requeridos

TABLA XVIII. DESCRIPCIÓN C.U. GESTIONAR MASCOTA EXTRAVIADA

Caso de uso:	Gestionar mascota extraviada	Actor:	Usuario-general
Código:	CU004	Tipo:	Aplicación web
Ref. Req:			
Resumen:	El sistema permitirá crear y eliminar mascotas extraviadas.		
Objetivos:	<ul style="list-style-type: none">• Crear mascota extraviada• Eliminar mascota extraviada		
Precondiciones:	<ul style="list-style-type: none">• Ingresar al sistema• Poseer mascota registrada previamente		
Postcondiciones:	<ul style="list-style-type: none">• Mascota extraviada creada• Mascota extraviada eliminada		
Flujo normal de eventos			
<div>1. Iniciar sesión como usuario-general.</div> <div>2. Seleccionar “mascotas extraviadas”.</div> <div>3. Seleccionar “nueva mascota extraviada”.</div> <div>4. Seleccionar la mascota extraviada.</div> <div>5. Llenar el formulario para reportar la mascota extraviada.</div> <div>6. El sistema verifica que los campos sean correctos.</div> <div>7. El sistema guarda los datos de la nueva mascota extraviada y publica en la sección de inicio de los perfiles de usuarios.</div>			
Flujo alterno de eventos			
A. Campos vacíos			
A.1 El sistema muestra mensaje de campos requeridos			

TABLA XIX. DESCRIPCIÓN C.U. GESTIONAR EVENTO

Caso de uso:	Gestionar evento	Actor:	Usuario-fundación
Código:	CU005	Tipo:	Aplicación web
Ref. Req:			
Resumen:	El sistema permitirá crear, modificar y eliminar eventos.		
Objetivos:	<ul style="list-style-type: none">• Crear evento.• Modificar evento.• Eliminar evento.		
Precondiciones:	<ul style="list-style-type: none">• Ingresar al sistema		
Postcondiciones:	<ul style="list-style-type: none">• Evento creado.• Evento modificado.• Evento eliminado.		
Flujo normal de eventos			
<div>1. Iniciar sesión como usuario-fundación.</div> <div>2. Seleccionar “mis eventos”.</div> <div>3. Seleccionar “nueva evento”.</div> <div>4. Llenar el formulario para evento.</div> <div>5. El sistema verifica que los campos sean correctos.</div> <div>6. El sistema guarda los datos del nuevo evento y publica en la sección de eventos de los perfiles de usuarios.</div>			
Flujo alterno de eventos			
A. Modificar evento			
<div>A.1 Seleccionar evento.</div> <div>A.2 Seleccionar ícono de “modificar”.</div> <div>A.3 Modificar los campos deseados.</div> <div>A.4 El sistema guarda los datos modificados.</div>			
B. Eliminar evento			
<div>B.1 El actor selecciona el evento a eliminar.</div> <div>B.2 Seleccionar ícono de eliminar.</div> <div>B.3 El sistema muestra mensaje de confirmación.</div> <div>B.4 El sistema elimina el evento.</div>			
C. Campos vacíos			
<div>C.1 El sistema muestra mensaje de campos requeridos</div>			

TABLA XX. DESCRIPCIÓN C.U. CONSULTA MASCOTA EXTRAVIADA

Caso de uso:	Consulta mascota extraviada	Actor:	WebService
Código:	CU006	Tipo:	Aplicación web
Ref. Req:			
Resumen:	El sistema proporciona servicio para consumir información de mascotas extraviadas.		
Objetivos:	<ul style="list-style-type: none">• API REST de mascotas extraviadas.		
Precondiciones:	<ul style="list-style-type: none">• Existir mascotas extraviadas• Aplicación externa para consumir servicio.		
Postcondiciones:	<ul style="list-style-type: none">• Visualizar mascotas extraviadas.		
Flujo normal de eventos			
1. Consultar servicio REST mediante la url proporcionada por el sistema			
Flujo alterno de eventos			
A. No existen mascotas en adopción			
A.1 El sistema no mostrará datos en el servicio REST.			

6.1.5. Glosario de términos

TABLA XXI. GLOSARIO DE TÉRMINOS

N°	Nombre	Definición
1	Administrador	Usuario encargado de la administración del sitio web
2	Fundación/colectivo	Persona jurídica sin ánimo de lucro, que cumple la voluntad de quien la fundó
3	Adopción	Proceso de tomar responsabilidad de un animal que un dueño previamente ha abandonado o dejado en un refugio de animales.
4	Web Service	Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones
5	Mascota	Animal doméstico
6	Evento	Acontecimiento o suceso

6.2. Fase de Elaboración

En esta fase, el objetivo principal es profundizar en la comprensión de los requerimientos y validar la arquitectura del sistema.

6.2.1. Arquitectura

La arquitectura de la aplicación permite visualizar los elementos con los que está diseñada y la relación que existe entre ellos. Se ha elegido una arquitectura cliente/servidor, en el cual el cliente realiza peticiones y el servidor envía una respuesta.

La arquitectura física se compone de los siguientes componentes: Servidor en la nube en Ubuntu Server, base de datos en Postgres, API REST, Servidor de google maps y GPS.

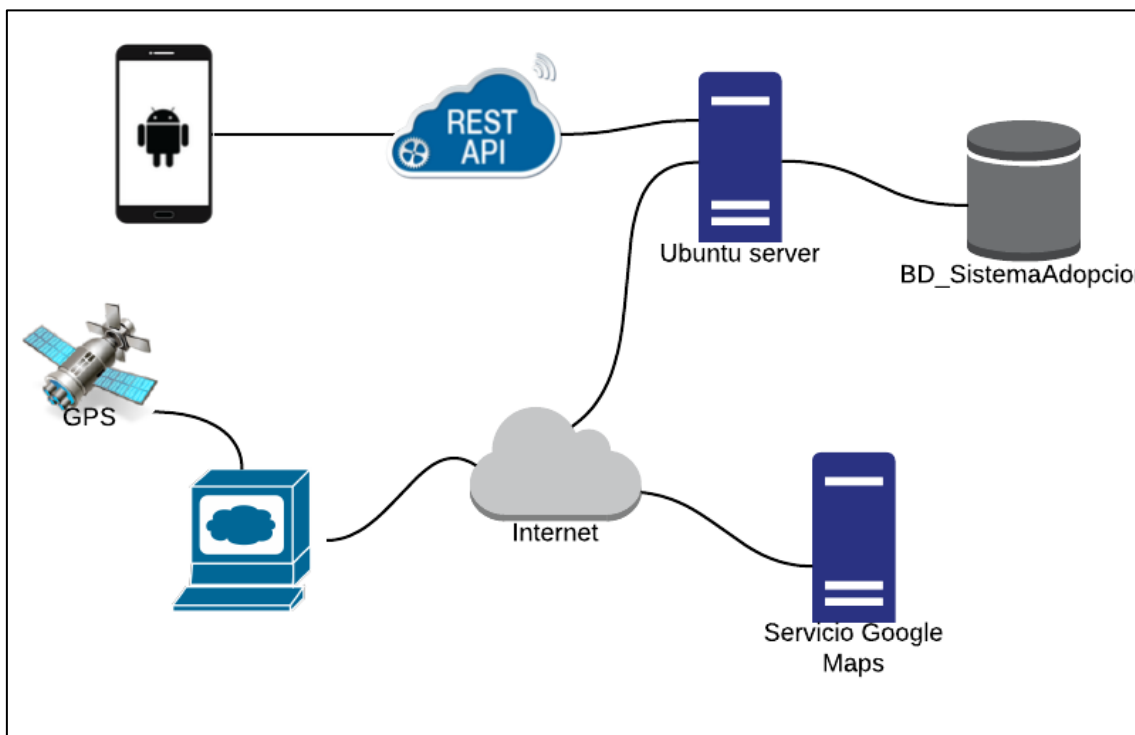


Figura 4. Arquitectura del sistema

6.2.1.1. Descripción de componentes

La arquitectura que se implementó lo conforma varios elementos que se detallan a continuación:

- **Servidor:** es el encargado de dar respuesta a las peticiones por parte del cliente.

TABLA XXII. COMPONENTES DE LA ARQUITECTURA SERVIDOR

Componente	Descripción
BD_SistemaAdopción	La base de datos fue desarrollada en Postgres para el almacenamiento de la información que se utilizará en la aplicación web.
Servidor Ubuntu_server	Se utiliza un servidor en la nube brindado por DigitalOcean con sistema operativo Ubuntu Server. La aplicación es alojada mediante el servidor NGINX.
Servicio Google Maps	Este servicio es utilizado para guardar la geolocalización y visualizar en el mapa.
API REST	Brinda servicio para ser usada por los dispositivos móviles.

- **Ciente:** es el responsable de realizar peticiones al servidor.

TABLA XXIII. COMPONENTES DE LA ARQUITECTURA CLIENTE

Componente	Descripción
Capa Presentación	Se compone de los templates utilizados en la aplicación encargados de interactuar con el usuario.

6.2.2. Prototipo de la aplicación

Para el desarrollo del prototipo se utilizó la herramienta Pencil, la misma que es software libre y multiplataforma para el desarrollo de prototipos web y móvil. Con la ayuda de los prototipos y las plantillas proporcionadas por el framework MaterializeCSS, permitió visualizar el diseño y posible distribución de los elementos en la página web.

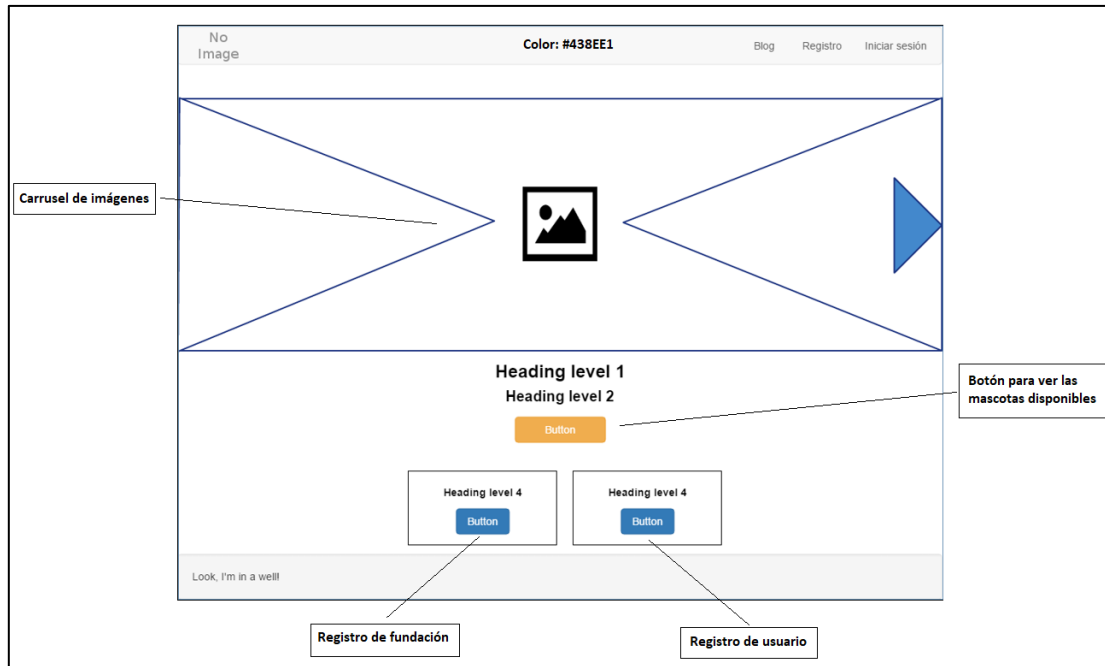


Figura 5. Prototipo de la página de inicio

La Figura 5, muestra la distribución de elementos en la página de inicio de la aplicación PetHome, la misma que contiene un header con opciones de registro e inicio de sesión, un carrusel de imágenes, sección para texto informativo y un footer para información adicional.

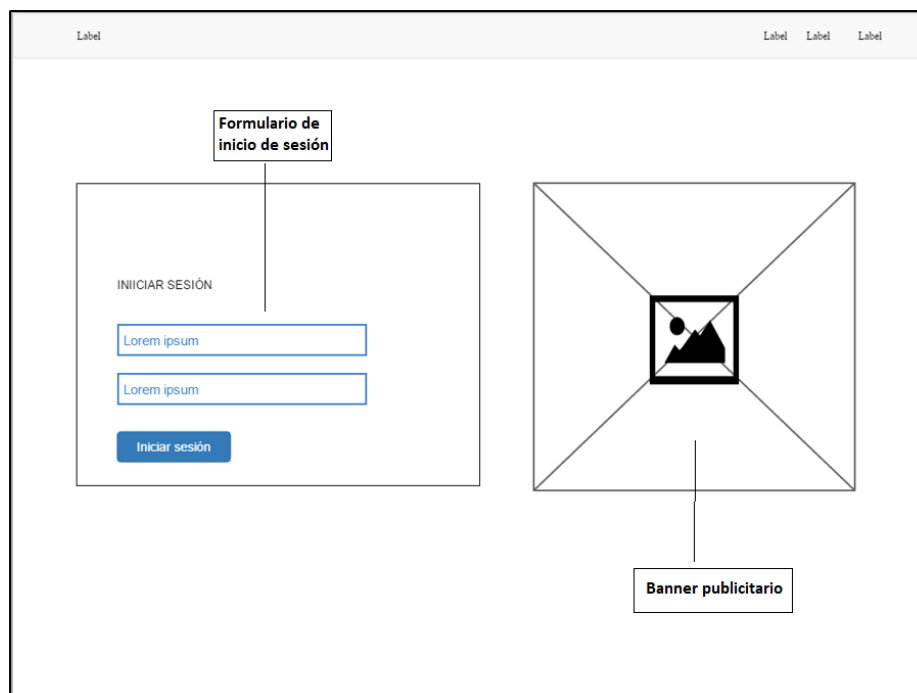


Figura 6. Prototipo de inicio de sesión

La Figura 6, representa la página de inicio de sesión de usuario, el cual contiene dos inputs que debe validar el nombre de usuario y contraseña para poder acceder al sistema.

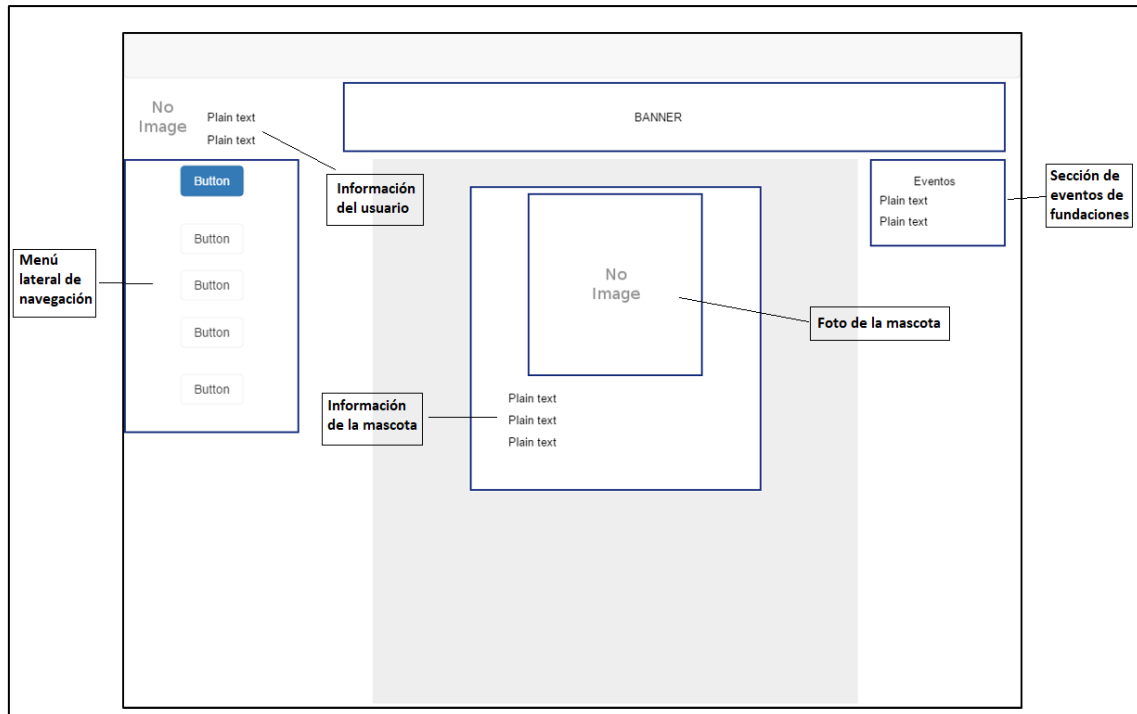


Figura 7. Prototipo de página de inicio de usuario

La figura 7, muestra la página de inicio de un usuario logueado en el sistema. Contiene un banner para publicidad, un navbar en la parte izquierda de la página para las diferentes opciones.

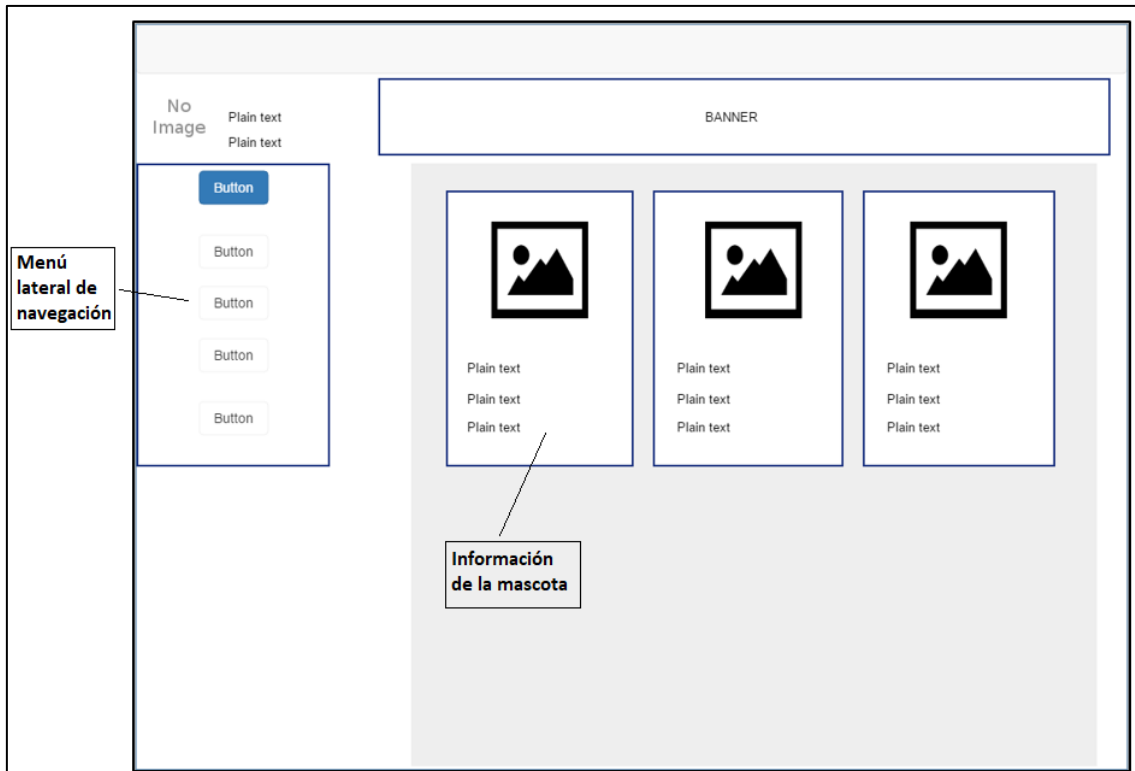


Figura 8. Prototipo para mostrar mascotas

La figura 8 muestra el prototipo para la presentación de mascotas. Está formado por 3 columnas, la cuales contiene la imagen de la mascota y texto descriptivo.

6.2.3. Diagrama de clases

Luego de haber obtenido los requerimientos e información necesaria, en la Figura 9 se representa las clases y sus relaciones implementadas en el sistema “PetHome”.

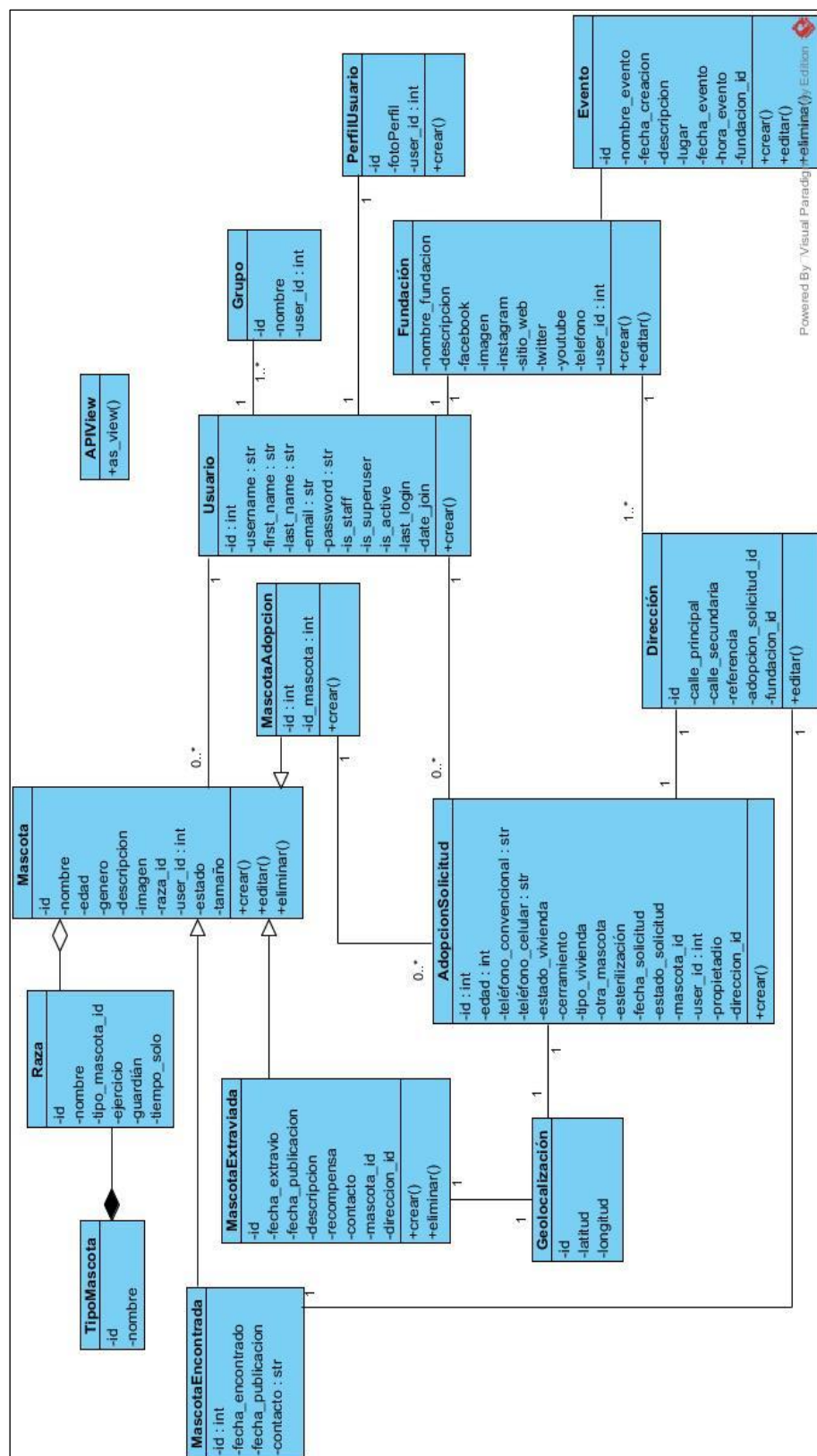


Figura 9. Diagrama de clases

6.2.3.1. Diccionario de clases

Para una mejor comprensión, se realizó un diccionario de clases, en el cual se muestran los elementos que participan en un flujo de datos de la aplicación.

Usuarios

- Clase Usuario: clase generada por el framework Django que contiene la información principal de registro de usuarios.
- Clase Grupo: clase generada por el framework Django para la creación de diferentes tipos de usuario.
- Clase PerfilUsuario: clase que contiene información extra del usuario.
- Clase Fundación: clase que contiene información extra del usuario fundación.

Mascotas en adopción

- Clase Mascota: clase que contiene información de mascota. Esta también es usada como padre para la herencia para la clase MascotaEncontrada, MascotaExtraviada y MascotaAdopcion.
- Clase Raza: clase utilizada para la gestión de razas de mascota. Es composición de la clase Mascota.
- Clase TipoMascota: clase utilizada para la gestión de tipos de mascota.
- Clase AdopciónSolicitud: clase que contiene la información necesaria para realizar la adopción de una mascota.

Fundaciones

- Clase Evento: clase utilizada para la gestión de eventos creados por el usuario fundación.

MascotasExtraviadas

- Clase Geolocalización: clase que permite obtener la geolocalización del usuario para una mejor referencia de la dirección donde se extravió la mascota.

MascotasEncontradas

- Clase Dirección: clase utilizada para obtener datos de la dirección del usuario o mascota. Esta es utilizada por las clases AdopcionSolicitud, MascotaEncontrada y Fundación.

6.2.4. Diagrama de paquetes

El diagrama de paquetes representa la forma en que están organizados y agrupados los paquetes con sus elementos proporcionando una mejor visualización del sistema.

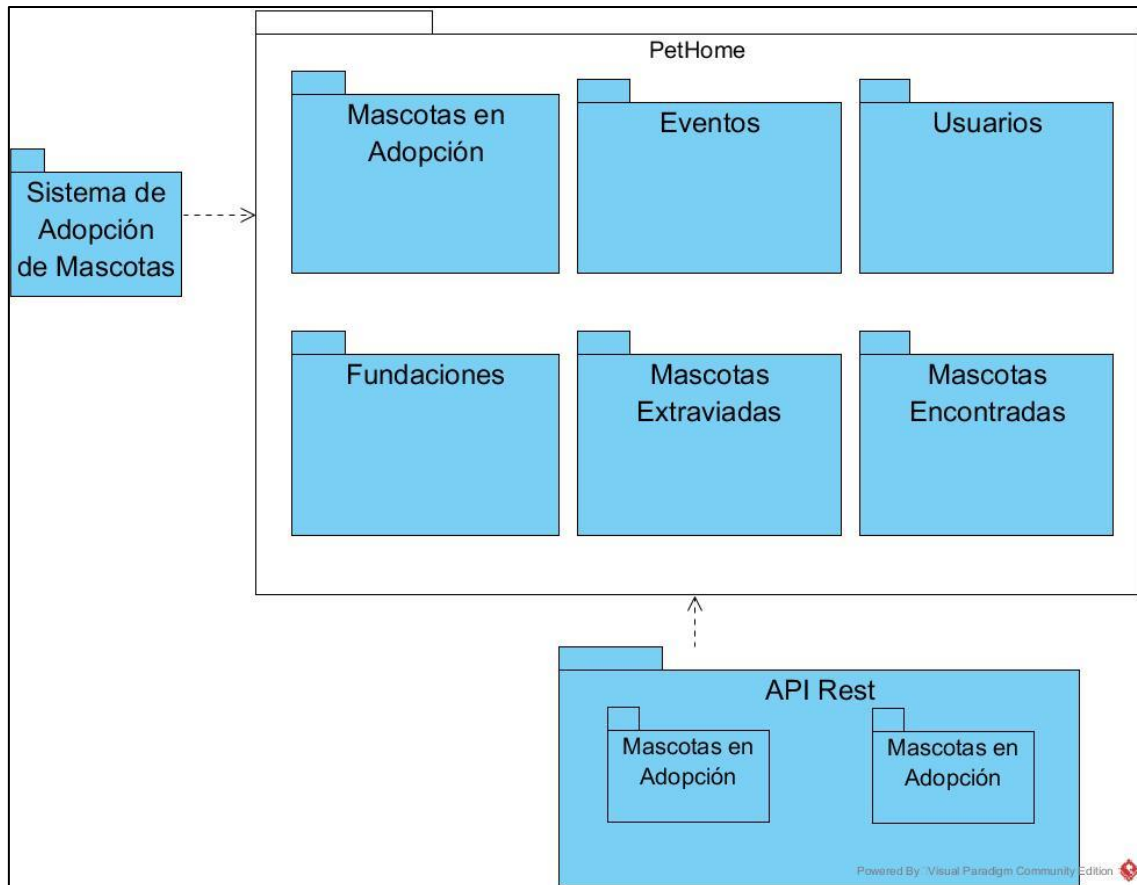


Figura 10. Diagrama de paquetes

La aplicación “PetHome” está distribuida en varios paquetes que son: mascotas en adopción, eventos, usuarios, fundaciones, mascotas extraviadas y mascotas encontradas. Cada paquete posee la estructura modelo-vista-plantilla dada por el framework Django. El conjunto de estos con el paquete de el API REST constituyen el sistema de adopción “PetHome”.

6.2.5. Diagrama de despliegue

La Figura 11, muestra el diagrama de despliegue implementado en la aplicación PetHome como cliente-servidor, en el cual el cliente mediante cualquier dispositivo con navegador web realiza la petición al servidor mediante una conexión HTTPS con certificado SSL dado por Let's Encrypt. El servidor Nginx procesa la petición y se comunica con la aplicación PetHome por medio de Gunicorn, utilizado para servir la aplicación en Django. De ser necesario consulta a la base de datos PostgreSQL por medio del conector Psycopg2.

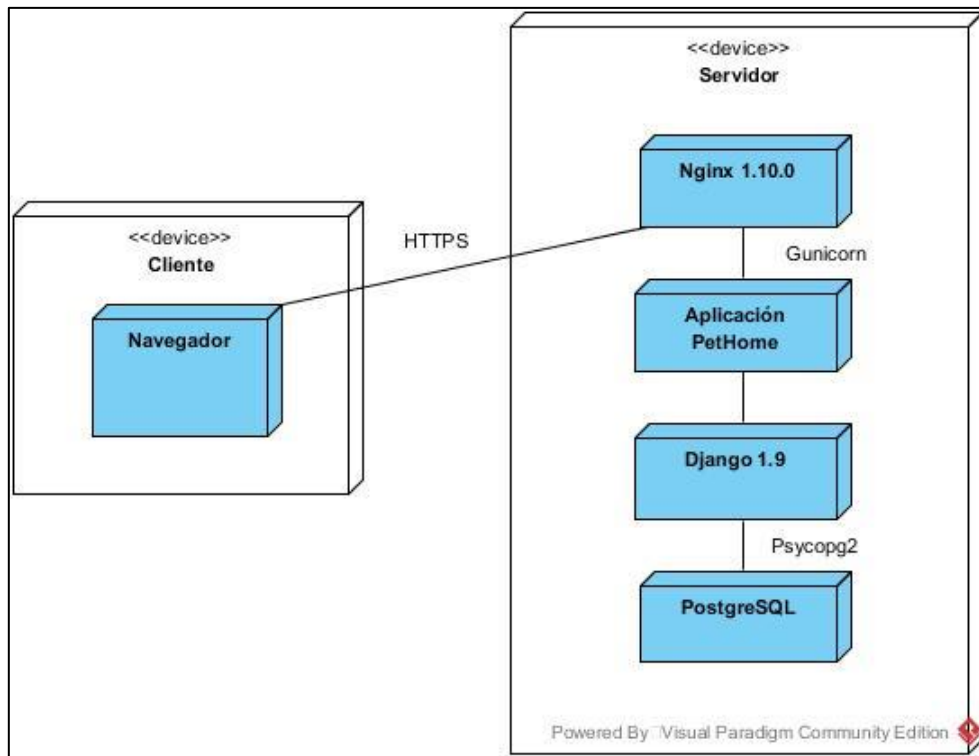


Figura 11. Diagrama de despliegue

6.3. Fase de construcción

Esta fase comprende la implementación del sistema tomando en cuenta la arquitectura conseguida de la fase anterior, obteniendo así la versión ejecutable del producto cumpliendo con los requisitos y necesidades planteadas.

6.3.1. Diagramas de secuencia

El siguiente diagrama de secuencia muestra la interacción por medio de mensajes secuenciales entre componentes y actores del sistema. A continuación, se muestra los diagramas de secuencia del flujo normal para cada caso de uso.

Diagrama de secuencia: Gestionar usuarios

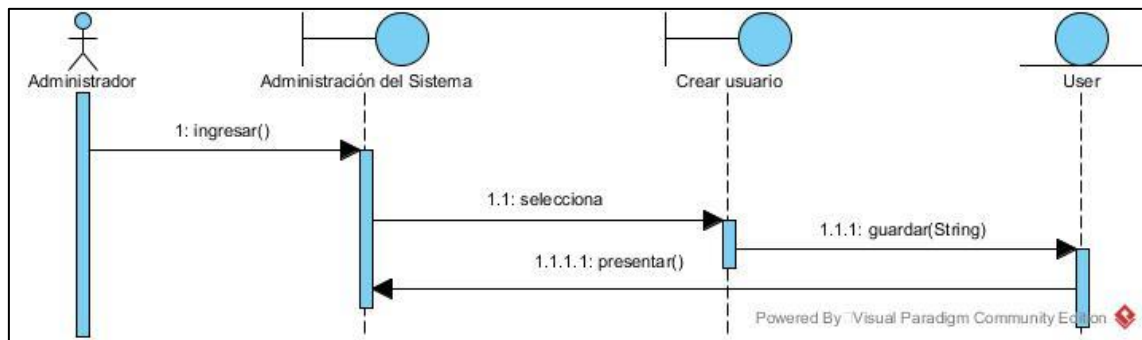


Figura 12. Diagrama de secuencia del CU001

Diagrama de secuencia: Gestionar mascota en adopción

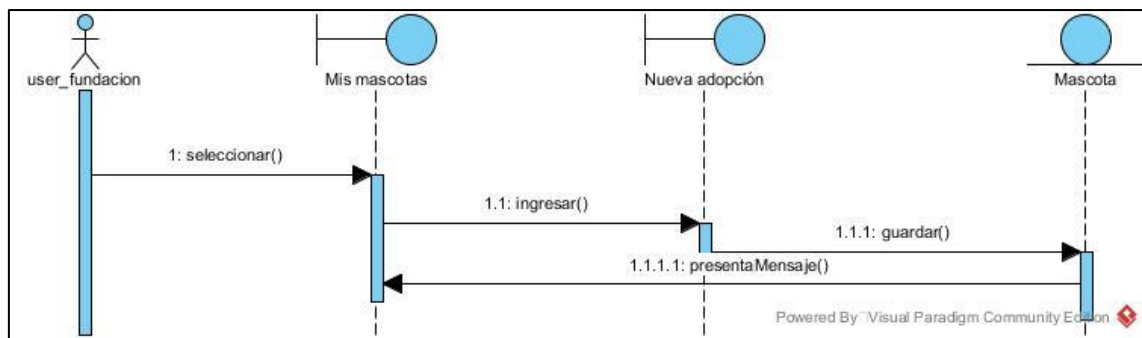


Figura 13. Diagrama de secuencia CU002

Diagrama de secuencia: Realizar adopción

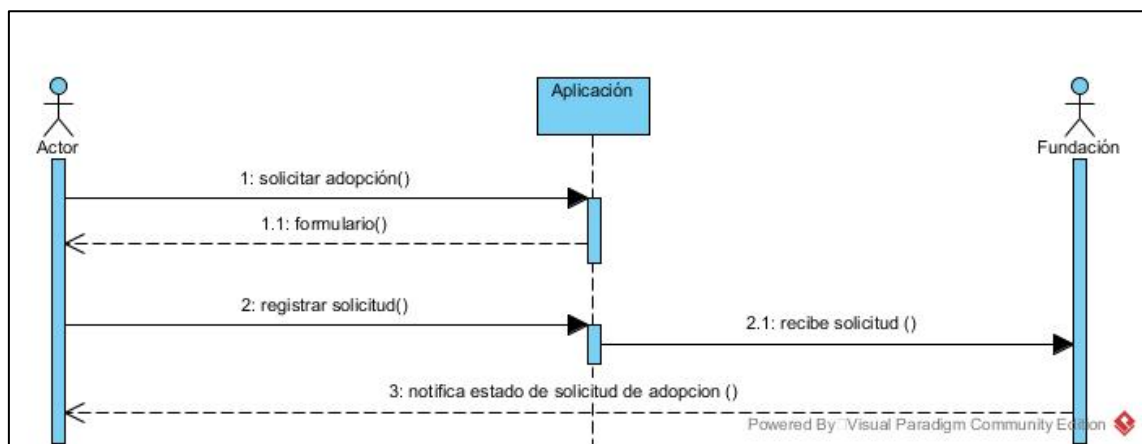


Figura 14. Diagrama de secuencia del CU003

Diagrama de secuencia: Gestionar mascota extraviada

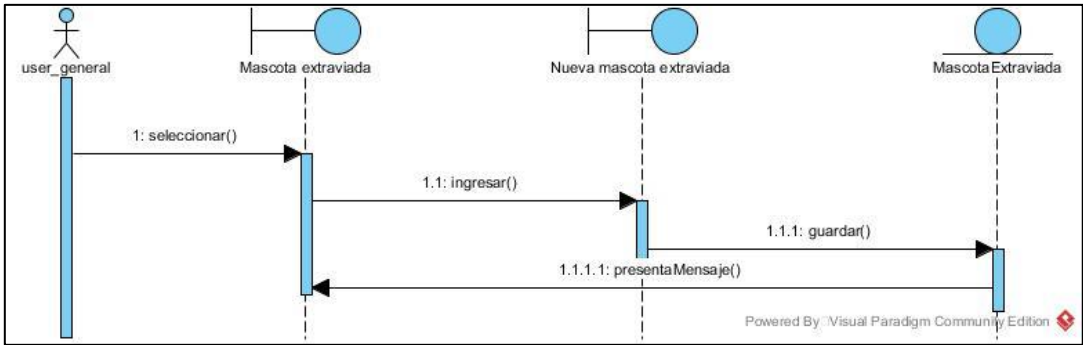


Figura 15. Diagrama de secuencia del CU004

Diagrama de secuencia: Gestionar evento

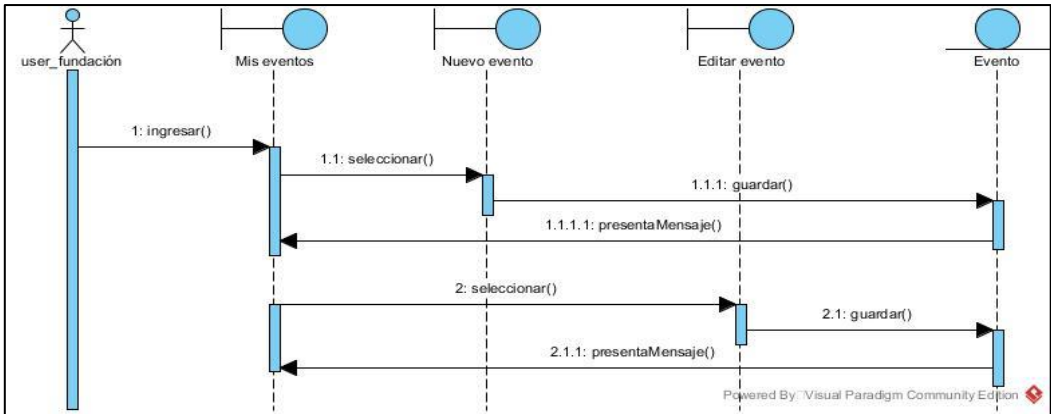


Figura 16 Diagrama de secuencia del CU005.

Diagrama de secuencia: Consulta de mascotas extraviadas

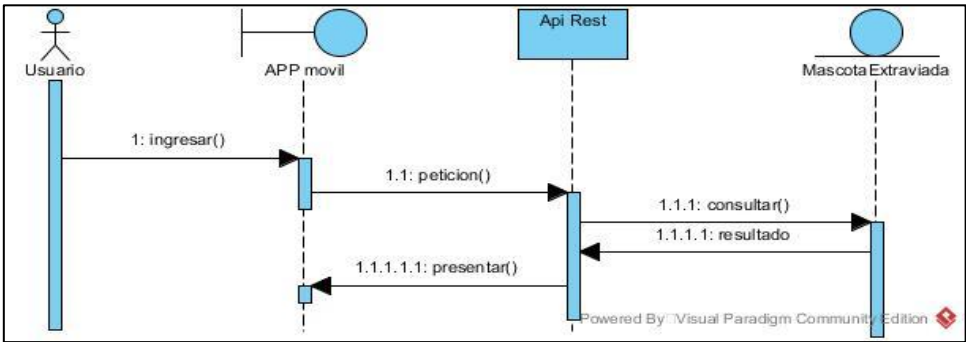


Figura 17. Diagrama de secuencia del CU006

6.3.2. Implementación

Luego de haber realizado el diseño de la aplicación, se desarrolló la implementación de la aplicación web. El código se encuentra disponible para su descarga en el repositorio de GitHub: <https://github.com/nadiapc128/SistemaAdopcion>. El resultado de la aplicación "PetHome" se encuentra en la dirección <https://www.pethomeloja.tk>

6.3.2.1. Componentes

Para la realización de la aplicación "PetHome" es necesario instalar algunos requerimientos:

- Tener un ambiente creado con Virtualenv para tener separado los recursos como librerías y entorno.
- Tener instalado Python 3.5, Django 1.9, PostgreSQL y librerías como gunicorn y pycppg2 que funciona como adaptador para la conexión con la base de datos.

Configuración de Django

La configuración del proyecto "PetHome" se encuentra en el archivo *settings.py*. Las principales son:

Base de datos: Para realizar la configuración de la conexión de django con la base de datos en PostgreSQL se debe editar el archivo *settings.py* en la sección DATABASES. Ahí se debe especificar el nombre de la base de datos, información de usuario y contraseña y especificar el puerto que se utiliza.

Aplicaciones instaladas: Django utiliza aplicaciones para su funcionamiento. Estas se activan cuando se las especifica en la sección de INSTALLED_APPS.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'apps.usuario',  
    'apps.mascota',  
    'apps.adopcion',  
    'apps.fundacion',  
    'rest_framework',  
    'chartit',  
    'corsheaders',  
]
```

Figura 18. Aplicaciones instaladas en "PetHome"

- **Django.contrib.admin:** activa la interfaz de administración proporcionada por django.
- **Django.contrib.auth:** activa los componentes de autenticación de django.
- **Django.contrib.contenttype:** permite realizar seguimiento de los modelos instalados en el proyecto.
- **Django.contrib.sessions:** soporte para sesiones por medio de cookies.
- **Django.contrib.messages:** mensajería de solicitudes basada en cookies y sesiones.
- **Django.contrib.staticfiles:** recopila archivos estáticos de las aplicaciones en una sola ubicación. [12]
- **Rest_framework:** para la utilización de django rest framework.

Organización del proyecto

El proyecto realizado en django está organizado por medio de módulos de usuario, mascota, adopción y fundación. Cada una contiene la estructura modelo-vista-platilla.

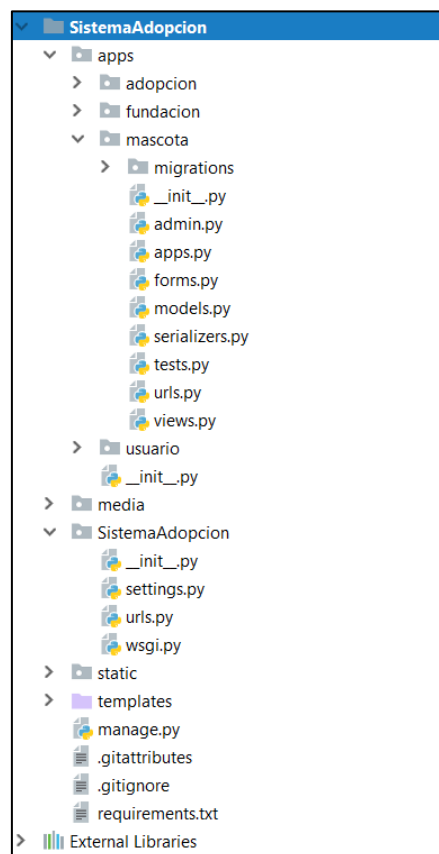


Figura 19. Estructura del proyecto "PetHome"

- **Apps:** se encuentran los módulos de la aplicación de usuarios, mascotas, adopción y fundación, los mismos que contienen el código del proyecto en los archivos views.py, models.py, forms.py, urls.py, etc.
- **Media:** directorio donde se almacenan las imágenes subidas por los usuarios.
- **SistemaAdopcion:** ubicación donde se encuentra el archivo settings.py para configuración del proyecto django.
- **Static:** directorio donde se encuentran los archivos estáticos como *.css, *.js e imágenes para el funcionamiento de la aplicación.
- **Templates:** directorio para los archivos html de la aplicación.
- **External libraries:** ubicación donde se encuentran librerías externas instaladas.

6.3.2.2. Vistas de la aplicación

Página de inicio

Al ingresar a la dirección <https://www.pethomeloja.tk>, se muestra la página de inicio que contiene información de la aplicación y acceso a registro e inicio de sesión.

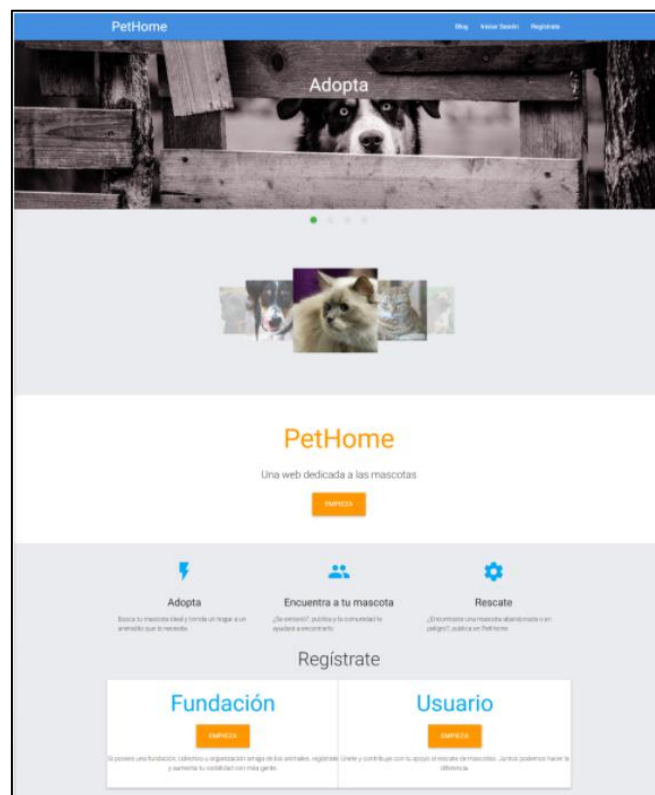


Figura 20. Pantalla de inicio de "PetHome"

Inicio de Sesión

Para el ingreso al sistema, el usuario debe estar registrado en la aplicación como usuario normal o usuario fundación, las cuales, dependiendo del tipo de usuario ingresará a la sección correspondiente de la aplicación.

Figura 21. Pantalla de inicio de sesión en "PetHome"

Panel de administración

La administración de la aplicación se realiza mediante el sitio proporcionado por Django, en el cual se tiene manejo directo de los modelos creados con los permisos de superusuario.

Administración de Django		
Administración del sitio		
ADOPCION		
Adopcion solicitudes	+ Agregar	✎ Modificar
AUTHENTICATION AND AUTHORIZATION		
Grupos	+ Agregar	✎ Modificar
Usuarios	+ Agregar	✎ Modificar
FUNDACION		
Eventos	+ Agregar	✎ Modificar
MASCOTA		
Mascota encontradas	+ Agregar	✎ Modificar
Mascota extraviadas	+ Agregar	✎ Modificar
Mascotas	+ Agregar	✎ Modificar
Razas	+ Agregar	✎ Modificar
Tests	+ Agregar	✎ Modificar
Tipo mascotas	+ Agregar	✎ Modificar

Figura 22. Sitio de administración de "PetHome"

El código donde se especifica los modelos que serán gestionados en el sitio de administración se encuentra en el archivo *admin.py* y se lo representa en la siguiente figura:

```
from django.contrib import admin
from .models import Mascota, MascotaExtraviada, \
    Raza, TipoMascota, MascotaEncontrada, Test

admin.site.register(Mascota)
admin.site.register(MascotaExtraviada)
admin.site.register(Raza)
admin.site.register(TipoMascota)
admin.site.register(MascotaEncontrada)
admin.site.register(Test)
```

Figura 23. Código para la administración del sitio

Página de inicio de usuario normal

Una vez iniciado sesión, el usuario visualizará la página de inicio en donde se visualiza las mascotas que han sido puestas en adopción, mascotas extraviadas y mascotas encontradas. Además, posee un menú de perfil de usuario para modificar información o contraseña.

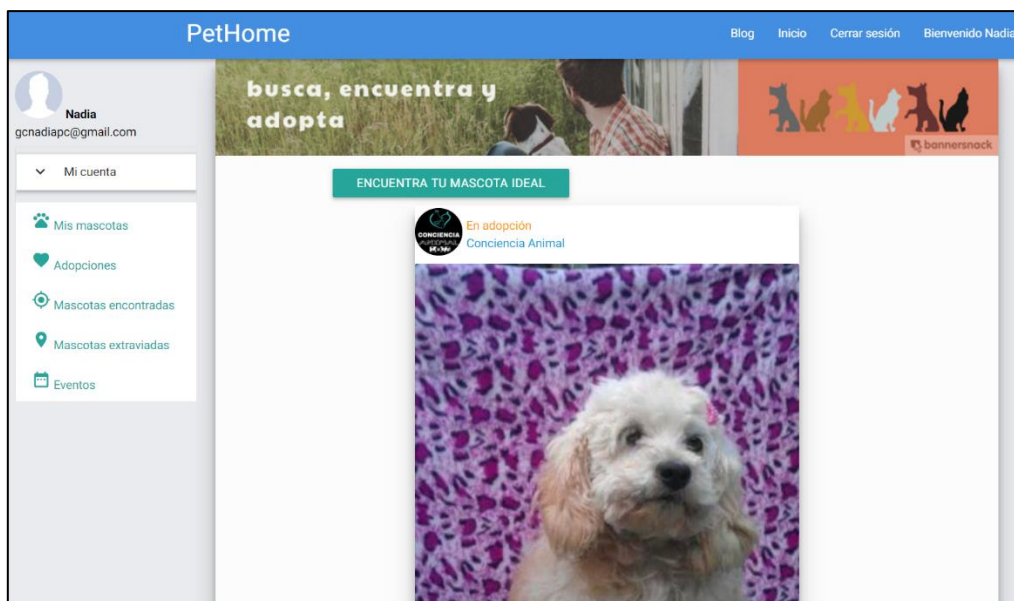


Figura 24. Página de inicio de usuario_normal

Gestión de mascotas

El usuario puede realizar la gestión de usuarios ingresando a la sección de Mis mascotas, en ella se muestra las opciones para crear, modificar y eliminar mascotas.

Mis Mascotas

+

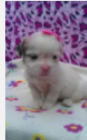








#	Nombre	Raza	Edad	Foto	Estado		
1	rolly	Esquimal Canadiense	1 Meses		Mi mascota		
2	ninfa	Galgo	3 Meses		Mi mascota		
3	Loky	Chow Chow (de pelo duro)	5 Años		Extraviado		

Figura 25. Gestión de mascotas en "PetHome"

Formulario de registro de mascota

Para crear una nueva mascota se debe ingresar los datos del formulario que se presentan en la siguiente imagen.

Registro de Mascota

Nombre

Tamaño

Raza

Affenpinscher

Género

Edad

Descripción

IMAGEN

GUARDAR >

Figura 26. Formulario de registro de mascota

Página de mascotas en adopción

La visualización de todas las mascotas que se encuentran en estado de adopción se la realiza ingresando a la sección de adopciones, dando la opción de solicitar adopción.

Mascotas en adopción



Cielo

Conciencia Animal

Raza: Somalí

Género: macho

SOLICITAR ADOPCIÓN



tigresa

Conciencia Animal

Raza: Bombay

Género: hembra

SOLICITAR ADOPCIÓN



pelusa

Conciencia Animal

Raza: Maine coon

Género: macho

SOLICITAR ADOPCIÓN

Figura 27. Mascotas en adopción de "PetHome"

Formulario de solicitud de adopción

Para realizar la solicitud de una adopción se debe llenar correctamente el siguiente formulario.

Solicitar adopción

Calle Principal	Calle Secundaria
<input type="text"/>	<input type="text"/>
Referencia	
<input type="text"/>	
Edad	Teléfono convencional
18 ▼	<input type="text"/>
Teléfono celular	
<input type="text"/>	
Tipo de vivienda	Cerramiento
Propia ▼	Si ▼
Usted vive en:	
Departamento ▼	
<input type="text"/>	
¿Posee otra mascota?	¿Está de acuerdo con la esterilización
Si ▼	Si ▼
<input type="text"/>	<input type="text"/>

Mostrar Mapa

REGISTRAR

Figura 28. Formulario de solicitud de adopción

Página de inicio de usuario fundación

Una vez iniciado sesión, el usuario fundación visualizará la página de inicio en donde se visualiza los eventos próximos y un gráfico estadístico en relación de la edad de los solicitantes de adopción y la raza de la mascota. Además, posee un menú de perfil de usuario para modificar información o contraseña.

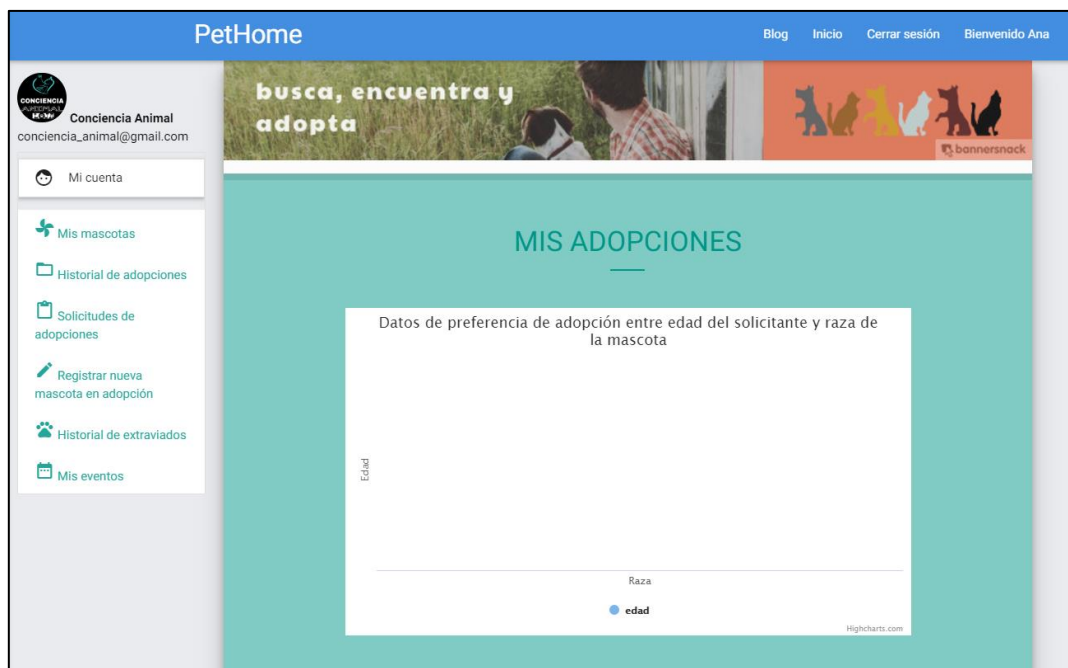


Figura 29. Página de inicio de usuario fundación

Historial de adopciones

En esta sección se muestra todas las solicitudes de adopciones que han sido aceptadas por parte del usuario y su detalle, con la opción para ser impresas.


Estado	Mascota	Tipo	Solicitante	Foto	PDF
aceptado	rolly	Perro	Nadia Paucar Cabrera		
aceptado	ninfa	Perro	Nadia Paucar Cabrera		

Figura 30. Historial de solicitudes de adopciones aceptadas

6.4. Fase de transición

Comprende la fase final del proyecto, en el cual se lleva a cabo el despliegue de la aplicación en función del usuario final, así como las pruebas de validación en el cual se corrobora que los requerimientos sean cumplidos en su totalidad y que el rendimiento y usabilidad sea óptimo.

6.4.1. Test de la aplicación

6.4.1.1. Pruebas unitarias

Este tipo de prueba permite comprobar si un módulo o sección de código funciona correctamente de acuerdo a los datos de entrada y a la salida definida.

Las pruebas unitarias fueron realizadas al culminar una funcionalidad por parte de la desarrolladora del proyecto utilizando la herramienta brindada por el IDE PyCharm 2016.3 para test.

La clase utilizada para realizar las pruebas es TestCase, la misma que engloba diferentes tipos de pruebas que pueden ser realizadas. Un lugar convencional para las pruebas de aplicación es en el archivo test.py. El sistema de pruebas encontrará automáticamente cualquier archivo que empiece con test.

Como ejemplo se utiliza el archivo test.py del paquete usuario para la funcionalidad de *index* y *registro de usuario*.

```
class IndexTest(TestCase):
    def test_index(self):

        response = self.client.get(reverse('index'))
        self.assertEqual(response.status_code, 200)

class RegistroUsuarioTest(TestCase):
    def test_formulario_valido(self):

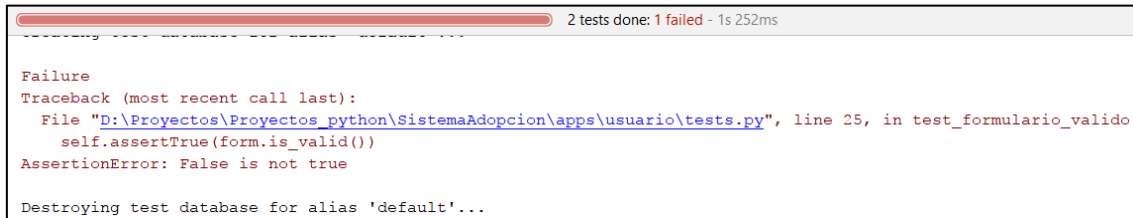
        form_data = {
            'username': 'nadiapc',
            'first_name': 'nadia',
            'last_name': 'cabrera',
            'password1': 'nadia1234',
            'email': 'gcnadiapc@gmail.com',
            'password2': '123456'
        }
        form = formu.RegistroForm(data=form_data)
        self.assertTrue(form.is_valid())
```

Figura 31. Pruebas unitarias

La figura 36 muestra el código para realizar el test. IndexTest realiza la prueba para verificar que el redireccionamiento sea correcto.

La clase RegistroUsuarioTest comprueba la validación del formulario de registro cuando los datos de entrada son los correctos. Para realizar el test se utiliza el comando **manage.py test**.

El resultado de la prueba se muestra a continuación:



```
2 tests done: 1 failed - 1s 252ms

Failure
Traceback (most recent call last):
  File "D:\Proyectos\Proyectos_python\SistemaAdopcion\apps\usuario\tests.py", line 25, in test_formulario_valido
    self.assertTrue(form.is_valid())
AssertionError: False is not true

Destroying test database for alias 'default'...
```

Figura 32. Resultado del test unitario

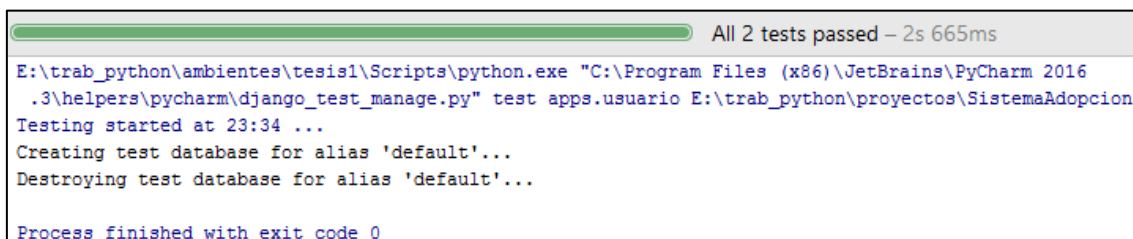
La ejecución del test unitario es fallida, ya que muestra un error en la validación del formulario del registro de usuario. Se procede a corregir los datos ingresados.



```
class RegistroUsuarioTest(TestCase):
    def test_formulario_valido(self):
        form_data = {
            'username': 'nadiapc',
            'first_name': 'nadia',
            'last_name': 'cabrera',
            'password1': 'nadia1234',
            'email': 'gcnadiapc@gmail.com',
            'password2': 'nadia1234'
        }
        form = formu.RegistroForm(data=form_data)
        self.assertTrue(form.is_valid())
```

Figura 33. Prueba unitaria corregida

El resultado de la prueba se muestra a continuación:



```
All 2 tests passed - 2s 665ms

E:\trab_python\ambientes\tesis1\Scripts\python.exe "C:\Program Files (x86)\JetBrains\PyCharm 2016
.3\helpers\pycharm\django_test_manage.py" test apps.usuario E:\trab_python\proyectos\SistemaAdopcion
Testing started at 23:34 ...
Creating test database for alias 'default'...
Destroying test database for alias 'default'...

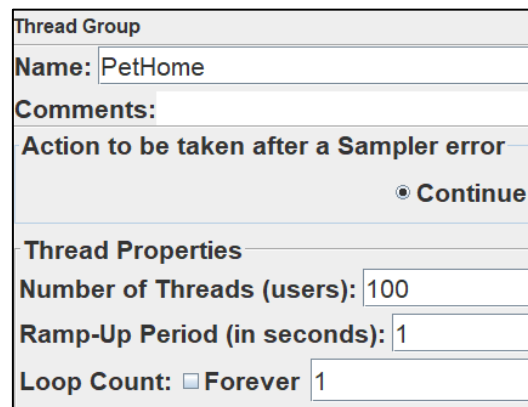
Process finished with exit code 0
```

Figura 34. Resultado del test unitario corregido

En resultado del test es válido como se observa en la figura 35.

6.4.1.2. Pruebas de carga

Con la herramienta Apache JMeter 3.2, se simuló el uso de la aplicación web con 100 peticiones a la vez como se muestra en la figura 35 y así comprobar el correcto funcionamiento del servidor. Esta herramienta permite realizar testeos de servidores Apache o cualquier otro de una manera sencilla.



The screenshot shows the 'Thread Group' configuration window in Apache JMeter. The 'Name' field is set to 'PetHome'. The 'Action to be taken after a Sampler error' is set to 'Continue'. Under the 'Thread Properties' section, 'Number of Threads (users)' is set to 100, 'Ramp-Up Period (in seconds)' is set to 1, and 'Loop Count' is set to 1 with the 'Forever' checkbox unchecked.

Figura 35. Configuración de número de peticiones en Apache JMeter

Página de inicio



The screenshot shows the 'HTTP Request' configuration window in Apache JMeter. The 'Name' field is set to 'Inicio'. The 'Basic' tab is selected. Under the 'Web Server' section, 'Protocol [http:]' is set to 'http' and 'Server Name or IP' is set to 'www.pethomeloja.tk'. Under the 'HTTP Request' section, 'Method' is set to 'GET' and 'Path' is empty. At the bottom, the 'Follow Redirects' and 'Use KeepAlive' checkboxes are checked.

Figura 36. Prueba de carga (Página de inicio)

Se aplicó la prueba de carga a la página de inicio de la aplicación PetHome, obteniendo los siguientes resultados:

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
60	01:09:38.913	Thread Group 1-100	Index	12395	✓	9169	244	2210	843
61	01:09:38.577	Thread Group -97	Index	12731	✓	10169	244	2548	1179
62	01:09:38.679	Thread Group -77	Index	12635	✓	10169	244	2796	1075
63	01:09:38.730	Thread Group -92	Index	12586	✓	10169	244	2397	1026
64	01:09:38.009	Thread Group -1	Index	12754	✓	10169	244	2353	1151
65	01:09:38.957	Thread Group -73	Index	12699	✓	10169	244	2885	1259
66	01:09:38.506	Thread Group -60	Index	13222	✓	10169	244	3192	1568
67	01:09:38.198	Thread Group -30	Index	13344	✓	10169	244	3319	1690
68	01:09:38.106	Thread Group -21	Index	13195	✓	10169	244	3138	1477
69	01:09:38.658	Thread Group -75	Index	13303	✓	10169	244	3035	1415
70	01:09:38.280	Thread Group -38	Index	13553	✓	10169	244	3008	1640
71	01:09:38.350	Thread Group -45	Index	13543	✓	10169	244	2966	1513
72	01:09:38.116	Thread Group -22	Index	13624	✓	10169	244	3382	1625
73	01:09:38.197	Thread Group -24	Index	13690	✓	10169	244	3121	1753
74	01:09:38.003	Thread Group -11	Index	13235	✓	10169	244	2934	1272
75	01:09:38.003	Thread Group -8	Index	13235	✓	10169	244	2834	1469
76	01:09:38.485	Thread Group -58	Index	13651	✓	10169	244	2958	1490
77	01:09:38.629	Thread Group -39	Index	14290	✓	10169	244	3475	1750
78	01:09:38.677	Thread Group -71	Index	13918	✓	10169	244	3005	1382
79	01:09:38.003	Thread Group -4	Index	14361	✓	10169	244	3257	1509
80	01:09:38.383	Thread Group -48	Index	14750	✓	10169	244	2818	1227
81	01:09:38.218	Thread Group -32	Index	15588	✓	10169	244	2759	1040
82	01:09:38.538	Thread Group -63	Index	15585	✓	10169	244	2789	1125
83	01:09:38.718	Thread Group -81	Index	15841	✓	10169	244	2668	1008
84	01:09:38.629	Thread Group -72	Index	16218	✓	10169	244	2712	1341
85	01:09:38.749	Thread Group -69	Index	16218	✓	10169	244	2908	1201
86	01:09:38.419	Thread Group -51	Index	16977	✓	10169	244	3044	1677
87	01:09:38.557	Thread Group -65	Index	169170	✓	10169	244	2690	1057
88	01:09:38.079	Thread Group -18	Index	16557	✓	10169	244	2955	1367
89	01:09:38.708	Thread Group -60	Index	16957	✓	10169	244	2866	946
90	01:09:38.403	Thread Group -50	Index	21445	✓	10169	244	18786	1755
91	01:09:38.819	Thread Group -91	Index	21699	✓	10169	244	2677	1309
92	01:09:38.003	Thread Group -3	Index	21699	✓	10169	244	2526	1159
93	01:09:38.447	Thread Group -54	Index	22398	✓	10169	244	3304	1639
94	01:09:38.086	Thread Group -19	Index	22855	✓	10169	244	3120	1751
95	01:09:38.003	Thread Group -10	Index	22346	✓	10169	244	2949	1242
96	01:09:38.516	Thread Group -61	Index	22858	✓	10169	244	3374	1751
97	01:09:38.014	Thread Group -12	Index	22559	✓	10169	244	3121	1752
98	01:09:38.003	Thread Group -2	Index						

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 22859 Average 12090 Deviation 4620

Figura 37. Resultados de la prueba de carga (Página de inicio)

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Index	100	12090	5649	22859	4620.27	0.00%	4.4/sec	39.17	1.04	9169.0
TOTAL	100	12090	5649	22859	4620.27	0.00%	4.4/sec	39.17	1.04	9169.0

Figura 38. Resumen de la prueba de carga (Página de inicio)

La figura 37 y 38 muestra que las 100 peticiones aplicadas a la página de inicio de PetHome fueron procesadas sin ningún inconveniente y el margen de error es de 0%.

Página de inicio de sesión

HTTP Request

Name: Login

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: Port Number:

HTTP Request

Method: GET Path: Content encoding:

Figura 39. Prueba de carga (Página de inicio de sesión)

Se aplicó la prueba a la página de inicio de sesión de la aplicación PetHome, obteniendo los siguientes resultados:

View Results in Table										
Name: View Results in Table										
Comments:										
Write results to file / Read from file										
Filename										
Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>										
Sample #	Start Time	Thread Name	Label	Sample Ti...	Status	Bytes	Sent Bytes	Latency	Connect Ti...	
59	15:00:29.391	PetHome 1-46	Login	4766	✓	5250	276	701	216	▲
60	15:00:29.603	PetHome 1-64	Login	4570	✓	5250	276	773	196	
61	15:00:29.530	PetHome 1-72	Login	4674	✓	5250	276	704	202	
62	15:00:29.613	PetHome 1-43	Login	4614	✓	5250	276	841	208	
63	15:00:29.896	PetHome 1-79	Login	4346	✓	5250	276	866	204	
64	15:00:29.946	PetHome 1-48	Login	4305	✓	5250	276	869	190	
65	15:00:29.953	PetHome 1-45	Login	4334	✓	5250	276	880	209	
66	15:00:29.608	PetHome 1-76	Login	4767	✓	5250	276	800	208	
67	15:00:30.070	PetHome 1-17	Login	4342	✓	5250	276	1029	208	
68	15:00:29.167	PetHome 1-21	Login	5312	✓	5250	276	729	204	
69	15:00:29.769	PetHome 1-82	Login	4770	✓	5250	276	831	205	
70	15:00:30.013	PetHome 1-22	Login	4537	✓	5250	276	914	213	
71	15:00:30.022	PetHome 1-20	Login	4539	✓	5250	276	916	204	
72	15:00:30.130	PetHome 1-83	Login	4439	✓	5250	276	1133	199	
73	15:00:30.167	PetHome 1-70	Login	4513	✓	5250	276	1166	210	
74	15:00:30.124	PetHome 1-25	Login	4565	✓	5250	276	1139	204	
75	15:00:29.141	PetHome 1-52	Login	5559	✓	5250	276	2137	1205	
76	15:00:29.335	PetHome 1-51	Login	5377	✓	5250	276	1272	221	
77	15:00:30.227	PetHome 1-82	Login	4494	✓	5250	276	1133	198	
78	15:00:29.937	PetHome 1-41	Login	4794	✓	5250	276	890	205	
79	15:00:30.306	PetHome 1-97	Login	4484	✓	5250	276	1172	196	
80	15:00:29.582	PetHome 1-96	Login	5209	✓	5250	276	707	188	
81	15:00:30.373	PetHome 1-75	Login	4491	✓	5250	276	1198	193	
82	15:00:29.306	PetHome 1-67	Login	5568	✓	5250	276	2140	1192	
83	15:00:29.625	PetHome 1-93	Login	5250	✓	5250	276	873	214	
84	15:00:30.329	PetHome 1-24	Login	4556	✓	5250	276	1217	215	
85	15:00:30.380	PetHome 1-6	Login	4516	✓	5250	276	1299	224	
86	15:00:30.432	PetHome 1-13	Login	4475	✓	5250	276	1756	207	
87	15:00:30.408	PetHome 1-30	Login	4545	✓	5250	276	1285	205	
88	15:00:30.535	PetHome 1-34	Login	4433	✓	5250	276	1296	192	
89	15:00:30.420	PetHome 1-49	Login	4572	✓	5250	276	1746	200	
90	15:00:29.801	PetHome 1-69	Login	5192	✓	5250	276	815	211	
91	15:00:30.040	PetHome 1-84	Login	4987	✓	5250	276	1017	209	
92	15:00:28.718	PetHome 1-35	Login	6392	✓	5250	276	726	336	
93	15:00:31.694	PetHome 1-80	Login	3537	✓	5250	276	1189	359	
94	15:00:30.045	PetHome 1-42	Login	5199	✓	5250	276	1031	214	
95	15:00:29.533	PetHome 1-4	Login	5718	✓	5250	276	2315	1210	
96	15:00:28.047	PetHome 1-77	Login	7234	✓	5250	276	4232	3325	
97	15:00:29.444	PetHome 1-29	Login	5909	✓	5250	276	3993	3191	
98	15:00:29.793	PetHome 1-33	Login	5970	✓	5250	276	771	192	
99	15:00:29.848	PetHome 1-68	Login	6191	✓	5250	276	794	198	
100	15:00:29.051	PetHome 1-9	Login	7266	✓	5250	276	1900	1189	▼
<input type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples? No of Samples 100 Latest Sample 7266 Average 4565 Deviation 709										

Figura 40. Resultados de la prueba de carga (Página de inicio de sesión)

Summary Report										
Name: Summary Report										
Comments:										
Write results to file / Read from file										
Filename										
Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>										
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received ...	Sent KB/s...	Avg. Bytes
Login	100	4565	3537	7266	709.96	0.00%	11.9/sec	60.93	3.20	5250.0
TOTAL	100	4565	3537	7266	709.96	0.00%	11.9/sec	60.93	3.20	5250.0

Figura 41. Resumen de la prueba de carga (Página de inicio de sesión)

La figura 40 y 41 muestra que las 100 peticiones aplicadas a la página de inicio de sesión de PetHome fueron procesadas sin ningún inconveniente y el margen de error es de 0%.

Como se puede observar en ninguno de los escenarios existe pérdida de peticiones y todo fue completado exitosamente, de esta manera también se cumple con el requerimiento no funcional RNF01.

6.4.1.3. Pruebas de estrés

La prueba de estrés se la realizó con el objetivo de determinar la solidez de la aplicación web con una cantidad de peticiones mayores a las esperadas. El software utilizado para realizar esta prueba fue Apache JMeter 3.2.

Prueba con 200 peticiones

Se empezó la prueba simulando la petición de 200 usuarios:

Summary Report										
Name: Summary Report										
Comments:										
Write results to file / Read from file										
Filename				Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes			Configure		
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/...	Avg. Bytes
Inicio	200	7541	4280	17488	2296.72	0.00%	10.9/sec	97.88	2.60	9169.0
TOTAL	200	7541	4280	17488	2296.72	0.00%	10.9/sec	97.88	2.60	9169.0

Figura 42. Resumen de la prueba de estrés con 200 peticiones

La figura 42 muestra que no existió error en la prueba con 200 peticiones.

Prueba con 300 peticiones

Summary Report										
Name: Summary Report										
Comments:										
Write results to file / Read from file										
Filename				Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes			Configure		
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/...	Avg. Bytes
Inicio	286	10214	5020	18008	2770.38	0.00%	15.6/sec	139.86	3.72	9169.0
TOTAL	286	10214	5020	18008	2770.38	0.00%	15.6/sec	139.86	3.72	9169.0

Figura 43. Resumen de la prueba de estrés con 300 peticiones

La figura 43 muestra que no existió error en la prueba con 300 peticiones.

Prueba con 400 peticiones

Summary Report										
Name: Summary Report										
Comments:										
Write results to file / Read from file										
Filename				Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes			Configure		
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/...	Avg. Bytes
Inicio	400	13958	5665	36783	5770.00	0.00%	10.8/sec	96.32	2.56	9168.9
TOTAL	400	13958	5665	36783	5770.00	0.00%	10.8/sec	96.32	2.56	9168.9

Figura 44. Resumen de la prueba de estrés con 400 peticiones

La figura 44 muestra que no existió error en la prueba con 400 peticiones.

Prueba con 500 peticiones

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Success

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency
46	16:06:01.470	PetHome 1-...	Inicio	6351	✖	2201	0	0
47	16:06:01.300	PetHome 1-...	Inicio	6521	✖	2201	0	0
48	16:06:01.388	PetHome 1-...	Inicio	6433	✖	2201	0	0
49	16:06:01.155	PetHome 1-...	Inicio	6735	✖	2201	0	0
50	16:06:01.145	PetHome 1-...	Inicio	6745	✖	2201	0	0
51	16:06:01.093	PetHome 1-...	Inicio	6804	✖	2201	0	0
52	16:06:01.150	PetHome 1-...	Inicio	6747	✖	2201	0	0
53	16:06:01.182	PetHome 1-...	Inicio	6715	✖	2201	0	0
54	16:06:01.136	PetHome 1-...	Inicio	6761	✖	2201	0	0
55	16:06:01.105	PetHome 1-...	Inicio	6793	✖	2201	0	0
56	16:06:01.060	PetHome 1-...	Inicio	6838	✖	2201	0	0
57	16:06:01.141	PetHome 1-...	Inicio	6757	✖	2201	0	0
58	16:06:00.908	PetHome 1-...	Inicio	7019	✔	9169	244	1875
59	16:06:00.896	PetHome 1-...	Inicio	7221	✔	9169	244	1884
60	16:06:00.787	PetHome 1-...	Inicio	7561	✔	9169	244	2257
61	16:06:00.939	PetHome 1-...	Inicio	7465	✔	9169	244	2195
62	16:06:01.329	PetHome 1-...	Inicio	7134	✔	9169	244	2446
63	16:06:00.783	PetHome 1-...	Inicio	7701	✔	9169	244	2276
64	16:06:00.763	PetHome 1-...	Inicio	7738	✔	9169	244	2697
65	16:06:01.354	PetHome 1-...	Inicio	7164	✔	9169	244	2560
66	16:06:00.758	PetHome 1-...	Inicio	7793	✔	9169	244	2719
67	16:06:01.414	PetHome 1-...	Inicio	7137	✖	2201	0	0
68	16:06:01.240	PetHome 1-...	Inicio	7313	✖	2201	0	0
69	16:06:01.416	PetHome 1-...	Inicio	7137	✖	2201	0	0
70	16:06:01.382	PetHome 1-...	Inicio	7175	✖	2201	0	0
71	16:06:01.376	PetHome 1-...	Inicio	7187	✖	2201	0	0
72	16:06:00.884	PetHome 1-...	Inicio	7724	✔	9169	244	1898
73	16:06:01.618	PetHome 1-...	Inicio	7008	✔	9169	244	2515
74	16:06:01.228	PetHome 1-...	Inicio	7423	✔	9169	244	2383
75	16:06:00.973	PetHome 1-...	Inicio	7707	✔	9169	244	1691
76	16:06:01.443	PetHome 1-...	Inicio	7246	✖	2201	0	0
77	16:06:01.441	PetHome 1-...	Inicio	7249	✖	2201	0	0
78	16:06:01.301	PetHome 1-...	Inicio	7426	✔	9169	244	2531
79	16:06:00.956	PetHome 1-...	Inicio	7838	✔	9169	244	1690
80	16:06:00.913	PetHome 1-...	Inicio	7906	✔	9169	244	1815
81	16:06:00.816	PetHome 1-...	Inicio	8029	✔	9169	244	2133
82	16:06:00.864	PetHome 1-...	Inicio	8007	✔	9169	244	2022
83	16:06:00.832	PetHome 1-...	Inicio	8061	✔	9169	244	2098
84	16:06:01.316	PetHome 1-...	Inicio	7597	✔	9169	244	3059
85	16:06:00.773	PetHome 1-...	Inicio	8176	✖	2201	0	0
86	16:06:01.052	PetHome 1-...	Inicio	7939	✔	9169	244	2538

☐ Scroll automatically? ☐ Child samples? No of Samples 500 Latest Sample 41527 Average 13372

Figura 45. Resultado de la prueba de estrés con 500 peticiones

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/...	Avg. Bytes
Inicio	500	13372	5927	41527	5197.46	12.60%	11.9/sec	96.73	2.49	8291.0
TOTAL	500	13372	5927	41527	5197.46	12.60%	11.9/sec	96.73	2.49	8291.0

Figura 46. Resumen de la prueba de estrés con 500 peticiones

Como muestra la figura 45 y 46, en la prueba realizada con 500 peticiones existen peticiones perdidas con un error del 12,60%. Esto se debe a la capacidad y recursos con que cuenta el servidor en la nube contratado que dispone de 512MB/1 CPU, 20 GB

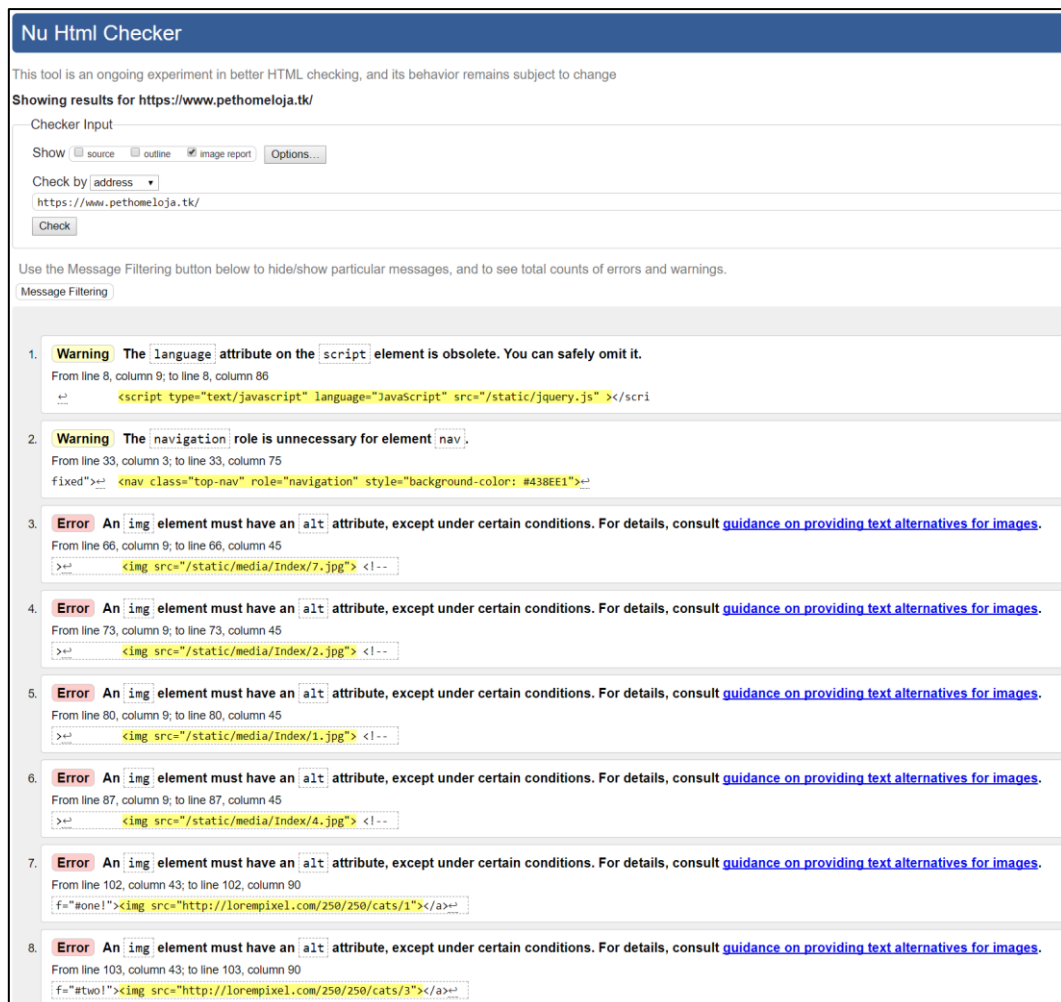
SSD disk y 1000GB de transferencia, además de la carga de imágenes y estilos que posee el sitio web produce.

6.4.1.4. Pruebas de accesibilidad

La w3c (Wide Web Consortium), permite evaluar el contenido HTML de acuerdo con estándares web. A continuación, se muestra los resultados obtenidos al aplicar la prueba UN HTML Checker.

En la primera prueba que se realizó, se puede observar que la página web contiene algunos errores en la utilización de etiquetas y elementos innecesarios.

Página de inicio de la aplicación



The screenshot displays the Nu Html Checker interface. At the top, it states: "This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change". Below this, it shows the URL being checked: "Showing results for https://www.pethomeloja.tk/". The "Checker Input" section includes a "Show" dropdown with options for "source", "outline", and "image report", and a "Check" button. The "Message Filtering" section is also visible. The main results area lists eight items:

- Warning** The `language` attribute on the `script` element is obsolete. You can safely omit it.
From line 8, column 9; to line 8, column 86
`<script type="text/javascript" language="JavaScript" src="/static/jquery.js"></scri`
- Warning** The `navigation` role is unnecessary for element `nav`.
From line 33, column 3; to line 33, column 75
`fixed"><nav class="top-nav" role="navigation" style="background-color: #438EE1">`
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 66, column 9; to line 66, column 45
`> <!--`
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 73, column 9; to line 73, column 45
`> <!--`
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 80, column 9; to line 80, column 45
`> <!--`
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 87, column 9; to line 87, column 45
`> <!--`
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 102, column 43; to line 102, column 90
`f="#one! ">`
- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 103, column 43; to line 103, column 90
`f="#two! ">`

Figura 47. Prueba de accesibilidad incorrecta de página de inicio

Como se puede observar en la Figura.47, existen diversos tipos de errores y advertencias. Estos errores fueron corregidos y subidos a GitHub en los commits 4d492e0 y aa6c25c como pueden ser chequeados en la dirección <https://github.com/nadiapc128/SistemaAdopcion/commits/prueba>.

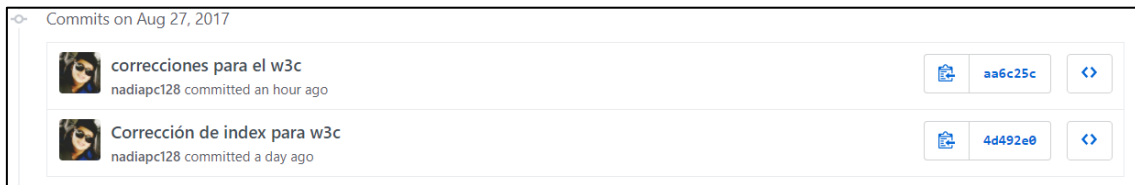


Figura 48. Commits de corrección de código de página de inicio

Una vez aplicado estos cambios, se realizó la prueba una vez más obteniendo los siguientes resultados:

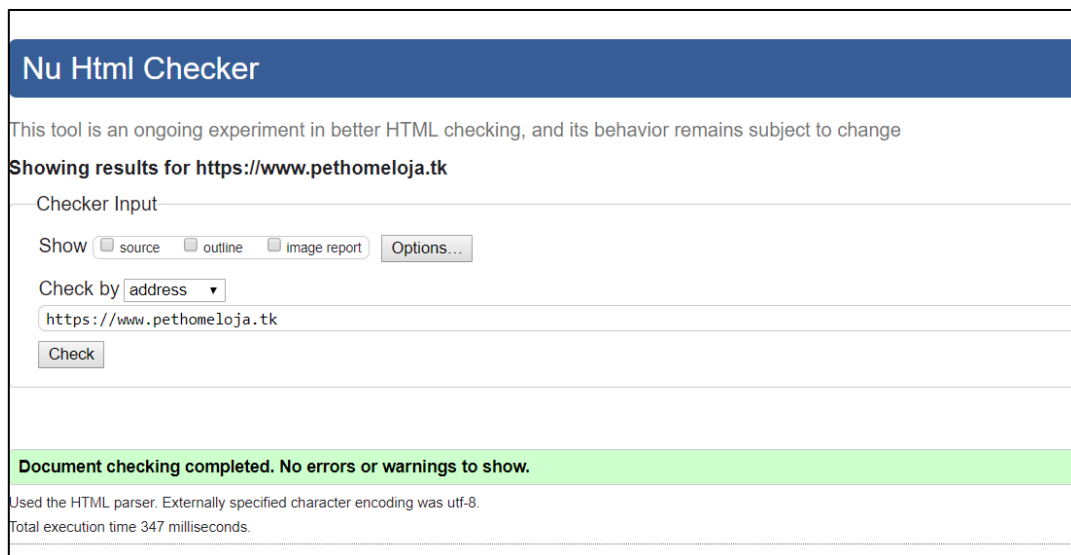


Figura 49. Prueba de accesibilidad correcta de página de inicio

Como se puede observar, el resultado no muestra ni errores ni advertencias. Por lo tanto, la aplicación web cuenta con una accesibilidad correcta con respecto a su código HTML y CSS, de este modo la prueba ha sido superada con éxito.

Página de registro de usuario

The screenshot shows the Nu Html Checker interface. At the top, it says "Nu Html Checker" and "This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change". Below this, it states "Showing results for https://www.pethomeloja.tk/usuario/registrar/". The "Checker Input" section shows the URL and a "Check" button. The "Message Filtering" section is empty. The main area displays seven errors:

- Error** Bad value `true` for attribute `required` on element `input`.
From line 79, column 13; to line 79, column 118
`<input class="form-control" id="id_username" maxlength="30" name="username" required="true" type="text" />`
- Error** Bad value `true` for attribute `required` on element `input`.
From line 91, column 13; to line 91, column 122
`<input class="form-control" id="id_first_name" maxlength="30" name="first_name" required="true" type="text" />`
- Error** Bad value `true` for attribute `required` on element `input`.
From line 95, column 13; to line 95, column 120
`<input class="form-control" id="id_last_name" maxlength="30" name="last_name" required="true" type="text" />`
- Error** Bad value `true` for attribute `required` on element `input`.
From line 102, column 14; to line 102, column 115
`<input class="form-control" id="id_email" maxlength="254" name="email" required="true" type="email" />`
- Error** The `for` attribute of the `label` element must refer to a non-hidden form control.
From line 80, column 13; to line 80, column 34
`<label for="username">Usuario`
- Error** The `for` attribute of the `label` element must refer to a non-hidden form control.
From line 92, column 13; to line 92, column 36
`<label for="first_name">Nombre`
- Error** The `for` attribute of the `label` element must refer to a non-hidden form control.
From line 96, column 13; to line 96, column 35
`<label for="last_name">Apellido`

Figura 50. Prueba de accesibilidad incorrecta de página de registro de usuario

En la Figura 50, se puede observar que la página de registro de usuario, su código HTML contiene varios errores. El código corregido se puede observar en el commit 251522e:

The screenshot shows a GitHub commit message. It includes a profile picture of a person with glasses and a yellow background. The text reads "Corrección para w3c registro de usuario" and "nadiapc128 committed 4 hours ago". To the right, there is a commit hash "251522e" and a "Compare" button with a double arrow icon.

Figura 51. Commits de corrección de código de página de registro de usuario

El resultado luego de haber corregido los errores es el siguiente:

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://www.pethomeloja.tk/usuario/regarlar/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

<https://www.pethomeloja.tk/usuario/regarlar/>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was utf-8.
Total execution time 500 milliseconds.

Figura 52. Prueba de accesibilidad correcta de página de registro de usuario

Estas pruebas fueron aplicadas a varias páginas de la aplicación “PetHome”, el resultado de este test se muestra en la siguiente tabla:

TABLA XXIV. RESULTADO DE LAS PRUEBAS DE ACCESIBILIDAD A LAS PÁGINAS DE "PETHOME"

Páginas HTML	Calificación		
	A	AA	AAA
Inicio		X	
Registro de usuario		X	
Registro de fundación		X	
Login		X	
Recuperar contraseña	X		
Mascotas en adopción		X	
Formulario de mascotas en adopción		X	
Mascotas extraviadas		X	
Formulario de mascotas extraviadas		X	

La tabla anterior muestra que la mayoría de páginas que conforman la aplicación cuentan con una calificación AA, esto significa que cumplen con las directrices de accesibilidad al contenido web dada por la W3C.

6.4.2. Pruebas de funcionalidad

Las pruebas funcionales sirven para asegurar el trabajo de los requisitos, incluyendo la navegación, entrada de datos, procesamiento y los resultados obtenidos.

Las siguientes pruebas corresponden a los casos de uso funcionales descritos anteriormente en el diagrama (Fig. 3 Diagrama de casos de uso general). El desarrollo de la misma estuvo a cargo de la autora y fueron realizadas tomando en cuenta los casos de uso (Ver anexo V. Plantilla para pruebas funcionales), ingresando valores correctos e incorrectos y así comprobar la respuesta del sistema.

6.4.2.1. Resumen de las pruebas funcionales de PetHome

Caso de uso: Gestionar usuarios (CU001)

Fecha de inicio: 12 de diciembre del 2016.

Fecha de fin: 16 de diciembre del 2016.

Para la realización de la prueba del caso de uso: Gestionar usuarios, se ingresó varios tipos de valores correctos e incorrectos, con la finalidad de comprobar la respuesta del sistema. A continuación, se presenta el resultado de la prueba:

TABLA XXV. RESULTADO DE LAS PRUEBAS FUNCIONALES DEL CU.
GESTIONAR USUARIOS

Casos Específicos	Valores Correctos			Valores Incorrectos		
	Entrada	Resultado Esperado	Evaluación de Prueba	Entrada	Resultado Esperado	Evaluación de Prueba
Crear usuario	X	Correcto	Satisface	X	Correcto	Satisface
Modificar Usuario	X	Correcto	Satisface	X	Correcto	Satisface

Caso de uso: Gestionar mascota en adopción (CU002)

Fecha de inicio: 12 de diciembre del 2016.

Fecha de fin: 16 de diciembre del 2016.

El caso de uso gestionar mascota en adopción, se ingresó varios tipos de valores correctos e incorrectos, con la finalidad de comprobar la respuesta del sistema y así realizar la prueba de funcionalidad. La tabla XXVIII representa los resultados obtenidos:

TABLA XXVI. RESULTADOS PRUEBA FUNCIONAL CU. GESTIONAR MASCOTA EN ADOPCIÓN

Casos Específicos	Valores Correctos			Valores Incorrectos		
	Entrada	Resultado Esperado	Evaluación de Prueba	Entrada	Resultado Esperado	Evaluación de Prueba
Crear mascota en adopción	X	Correcto	Satisface	X	Correcto	Satisface
Modificar mascota en adopción	X	Correcto	Satisface	X	Correcto	Satisface

Caso de uso: Realizar adopción (CU003)

Fecha de inicio: 12 de diciembre del 2016.

Fecha de fin: 16 de diciembre del 2016.

El caso de uso Solicitar adopción, se ingresó varios tipos de valores correctos e incorrectos en el formulario, así se pudo comprobar que la respuesta del sistema sea la correcta para cada caso. La tabla XXVIII representa los resultados obtenidos:

TABLA XXVII. RESULTADOS PRUEBA FUNCIONAL CU. SOLICITAR ADOPCIÓN

Casos Específicos	Valores Correctos			Valores Incorrectos		
	Entrada	Resultado Esperado	Evaluación de Prueba	Entrada	Resultado Esperado	Evaluación de Prueba
Enviar solicitud de adopción	X	Correcto	Satisface	X	Correcto	Satisface

Caso de uso: Gestionar mascota extraviada (CU004)

Fecha de inicio: 12 de diciembre del 2016.

Fecha de fin: 16 de diciembre del 2016.

El caso de uso gestionar mascota extraviada, se ingresó varios tipos de valores correctos e incorrectos, con la finalidad de comprobar la respuesta del sistema y así realizar la prueba de funcionalidad. La tabla XXX representa los resultados obtenidos:

TABLA XXVIII. RESULTADO PRUEBAS FUNCIONALES CU. GESTIONAR
MASCOTA EXTRAVIADA

Casos Específicos	Valores Correctos			Valores Incorrectos		
	Entrada	Resultado Esperado	Evaluación de Prueba	Entrada	Resultado Esperado	Evaluación de Prueba
Crear mascota extraviada	X	Correcto	Satisface	X	Correcto	Satisface
Modificar mascota extraviada	X	Correcto	Satisface	X	Correcto	Satisface

Caso de uso: Gestionar evento (CU005)

Fecha de inicio: 12 de diciembre del 2016.

Fecha de fin: 16 de diciembre del 2016.

El caso de uso gestionar evento, se ingresó varios tipos de valores correctos e incorrectos, con la finalidad de comprobar la respuesta del sistema y así realizar la prueba de funcionalidad. La tabla XXXI representa los resultados obtenidos:

TABLA XXIX. RESULTADOS PRUEBA FUNCIONAL CU. GESTIONAR EVENTO

Casos Específicos	Valores Correctos			Valores Incorrectos		
	Entrada	Resultado Esperado	Evaluación de Prueba	Entrada	Resultado Esperado	Evaluación de Prueba
Crear evento	X	Correcto	Satisface	X	Correcto	Satisface
Modificar evento	X	Correcto	Satisface	X	Correcto	Satisface
Eliminar evento	X	Correcto	Satisface	-	-	-

Caso de uso: Consulta de mascotas extraviadas (CU006)

Fecha de inicio: 12 de diciembre del 2016.

Fecha de fin: 16 de diciembre del 2016.

Esta prueba funcional se la comprobó mediante una aplicación móvil. La tabla XX

TABLA XXX. RESULTADOS PRUEBAS FUNCIONALES CU. PROMOCIONAR MASCOTA EXTRAVIADA

Casos Específicos	Valores Correctos			Valores Incorrectos		
	Entrada	Resultado Esperado	Evaluación de Prueba	Entrada	Resultado Esperado	Evaluación de Prueba
API de mascotas extraviadas	X	Correcto	Satisface	-	-	-

7. Discusión

7.1. Desarrollo de la propuesta alternativa

El presente trabajo de fin de carrera denominado “Desarrollo de una aplicación web de adopción de mascotas en la ciudad de Loja” se culminó creando la aplicación web “PetHome”, desarrollado con el framework Django, utilizando como gestor de base de datos PostgreSQL y Nginx como servidor web.

Con la aplicación de métodos, técnicas y metodología de desarrollo de software AUP (proceso unificado ágil), se pudo concluir en su totalidad el objetivo general junto a los objetivos específicos. A continuación, se evaluará los objetivos planteados en el trabajo de fin de carrera:

Objetivo específico 1

Realizar un estudio de casos de éxito que hayan permitido resolver el problema planteado

Para realizar este objetivo con éxito, se usó el método de revisión sistemática de literatura siguiendo las etapas planteadas por Kitchenham[20] y Biolchini [21]. En primer lugar, se determinó la pregunta de investigación, la misma que sirvió de pauta para obtener los estudios que fueran de utilidad. Posteriormente se seleccionó las fuentes de búsqueda y se elaboró las cadenas de búsqueda usando los operadores AND y OR. Aplicadas las cadenas de búsqueda en las diferentes bases científicas se obtuvo un total de 59 estudios, de los cuales 1 se repitió, aplicando los criterios de inclusión, exclusión y de selección se han escogido 6 artículos. De esta manera fueron que ayudan a cumplir el objetivo planteado.

Objetivo específico 2

Recolectar la documentación de requerimientos aplicando el estándar IEEE 830

Para dar cumplimiento a este objetivo, se realizó la recolección de requerimientos mediante entrevista a la fundación “Ángeles con Patas”, como se puede observar en el Anexo II, además, se ejecutó consultas a sitios web de colectivos y fundaciones para tener una visión general de la actividad que cumplen los mismos.

Se utilizó el estándar IEEE 830 para documentar la especificación de requerimiento de software (Ver anexo IV), este documento es de suma importancia para definir una visión clara de las funcionalidades que tuvo el software en base a las necesidades del cliente, así se obtuvo los requerimientos funcionales y no funcionales, que son de suma ayuda al momento de realizar la programación.

Objetivo específico 3:

Implementar un web service que permite hacer las consultas de mascotas extraviadas.

Este objetivo se lo cumplió utilizando la herramienta Django Rest Framework para construir el API Rest con la información de las mascotas extraviadas registradas. El funcionamiento del API Rest se puede comprobar en la dirección https://www.pethomeloja.tk/mascota/api_mascota/.

Objetivo específico 4:

Crea un escenario de pruebas, para verificar la funcionalidad de la aplicación

Para dar cumplimiento de este objetivo, se realizaron las pruebas de funcionalidad (Ver apartado 4.2 Pruebas de funcionalidad), probando cada caso de uso con valores correctos e incorrectos y así comprobar la respuesta del sistema. Los resultados de esta prueba permitieron corregir errores y obtener un software funcional a los requerimientos planteados desde un inicio.

También se aplicó la prueba de accesibilidad (Ver apartado 4.1.4 Pruebas de accesibilidad). La misma tuvo como finalidad valorar la estructura HTML de la aplicación web según los estándares W3C, obteniendo resultados favorables, que expresan que su accesibilidad es buena para el usuario.

7.2. Valoración técnica económica ambiental

En el desarrollo de la aplicación web de adopción de mascotas PetHome, se requirió de recurso humano para la orientación tanto técnica como para el diseño de la aplicación, además de tecnológicos de hardware y software que permitieron la culminación del trabajo de titulación.

• Recursos Humanos

El presente trabajo contó con el apoyo del director de tesis, el mismo que contribuyó con el asesoramiento y guía académico. Sus costos fueron asumidos por la Universidad Nacional de Loja.

TABLA XXXI. VALORACIÓN ECONÓMICA DE RECURSOS HUMANOS

Recursos Humanos	Cantidad	Número de horas	Valor unitario	Valor total
Desarrollador:	1	400	\$ 5	\$ 2000
• Nadia Paucar Cabrera				
• Director de tesis	1	-	.	-
Subtotal				\$ 2000

• Recursos Materiales

Los recursos materiales utilizados fueron principalmente de oficina, para el almacenamiento de la información y para la documentación final del trabajo de titulación.

TABLA XXXII. VALORACIÓN ECONÓMICA DE RECURSOS MATERIALES

Materiales	Cantidad	Valor Unitario	Valor Total
• Resma de papel A4	2	\$ 5	\$ 10
• Copias	650	\$ 0,02	\$ 13
• Caja de CD	1	\$ 14	\$ 14
• Anillado	5	\$ 2	\$ 10
• Tinta de impresión	4	\$ 10	\$ 40
• Empastado	3	\$ 12	\$ 36

• Suministros de oficina (perforadora, carpetas, lápices, grapadora, borradores)	-	\$ 15	\$ 15
Subtotal			\$ 138

- **Recursos Técnicos/Tecnológicos**

Estos recursos se los dividió en tres partes: Recursos de hardware, software y comunicación.

- **Recursos de Hardware**

En esta sección se describe los equipos utilizados en la aplicación web.

TABLA XXXIII. VALORACIÓN ECONÓMICA DE RECURSOS DE HARDWARE

Hardware	Cantidad	Valor Unitario	Valor Total
Computadora Dell Inspiron	1	\$ 800	\$ 800
Sony Xperia M4	1	\$ 300	\$ 300
Subtotal			\$ 1100

- **Recursos de Software**

A continuación, se detalla el software libre que fue utilizado en el desarrollo de PetHome, razón por la cual no hubo inversión económica.

TABLA XXXIV. VALORACIÓN ECONÓMICA DE RECURSOS DE SOFTWARE

Descripción	Cantidad	Valor Unitario	Valor Total
• Lenguaje de programación: Python	1	Gratuito	\$ 0,00
• Framework Django 1.9	1	Gratuito	\$ 0,00
• PostgreSQL	1	Gratuito	\$ 0,00
• Django Rest	1	Gratuito	\$ 0,00
• Entorno de programación PyCharm	1	Gratuito	\$ 0,00
• Ionic 2	1	Gratuito	\$ 0,00
• Servidor en la nube DigitalOcean	1	\$ 5	\$ 5
• Dominio	1	Gratuito	\$ 0,00
Subtotal			\$ 5,00

- **Recursos de Comunicaciones**

Los recursos de comunicación fueron una parte imprescindible para la realización de consultas, implementación y pruebas. En la siguiente tabla se detalla su costo.

TABLA XXXV. VALORACIÓN ECONÓMICA DE RECURSOS DE COMUNICACIONES

Descripción	Cantidad	Valor Unitario	Valor Total
• Internet	12 meses	\$ 20	\$ 20
• Celular	12 meses	\$ 10	\$ 10
Subtotal			\$ 30

- **Resumen del Presupuesto**

El resumen del presupuesto muestra la suma de los subtotales de los recursos humanos, materiales y técnico/tecnológicos detallados anteriormente.

TABLA XXXVI. RESUMEN DEL PRESUPUESTO

Subtotales	Valor Total
Recursos Humanos	\$ 2000
Recursos Materiales	\$ 138
Recursos Técnicos y Tecnológicos	\$ 1130
Subtotal	\$ 3268
Imprevistos 10%	\$ 326,8
Total	\$ 3594,8

El valor total del proyecto es de \$ 3594,8 (tres mil quinientos noventa y cuatro con ochenta centavos).

8. Conclusiones

- La recolección de requerimientos con el estándar IEEE 830 permitió obtener las necesidades del usuario como el requerimiento funcional RF14 correspondiente a la gestión de mascotas en adopción, constituyéndose en la base fundamental para el desarrollo de este proyecto, esto se logró mediante la aplicación de técnicas, métodos y una constante comunicación.
- Con la implementación del web service con Django Rest Framework, se construyó un servicio REST que puede ser consumido externamente, ya sea por una aplicación web o móvil. De esta manera se puede lograr el intercambio de información entre plataformas o tecnologías diferentes de una forma sencilla.
- En base a las pruebas de carga y estrés realizadas a la aplicación web, se puede asegurar la correcta respuesta y funcionamiento del servidor hasta con 400 peticiones al mismo tiempo, sin existir pérdidas ni retraso en la información devuelta al usuario.
- Con las pruebas de accesibilidad realizadas a la aplicación PetHome por medio de la W3C, se pudo validar la calidad del código fronted con una calificación AA, lo que asegura una experiencia del usuario óptima.

9. Recomendaciones

Al momento de realizar un proyecto siempre nos encontraremos con dificultades que nos ponen trabas o impedimentos para culminar con el desarrollo del mismo. A continuación, se detallan algunas recomendaciones que ayudarán a agilizar futuros trabajos:

- Es importante al momento de desarrollar la aplicación web tener definido su alcance, de esta forma se tiene la certeza de estar utilizando los procedimientos correctos para que posteriormente estos no resulten extensos e innecesarios.
- Previo a empezar con el proyecto de investigación como tal, es preciso hacer un preparatorio en base a como se va a estructurar el sistema que se va a realizar; es decir, tener una idea clara de qué es lo que queremos desarrollar, para ello se recomienda utilizar el estándar IEEE 830, mismo que nos da todas las pautas para realizar la recolección de requerimientos que constituye la base principal para el desarrollo de software.
- Se debe tener en cuenta que las APIs pueden ser accedidas desde cualquier lugar, por lo que se recomienda tener una conexión SSL y usar métodos de autenticación, dependiendo del grado de importancia de los datos.
- Si se va a realizar una aplicación que tenga alcance para una población mucho mayor a la de la ciudad de Loja, se recomienda que se realicen pruebas de carga y stress para cuantificar la cantidad de peticiones por segundo que puede soportar el servidor web que se haya seleccionado.
- Se recomienda que para trabajos futuros se implemente en la aplicación inteligencia artificial para la selección de mascotas en adopción en función de las necesidades y preferencias del usuario, de esta forma se podrá dar sugerencias personalizadas sobre las mascotas disponibles.

10. Bibliografía

- [1] J. Bailón Abad and F. Loján González, *ORDENANZA MUNICIPAL DE CONTROL Y PROTECCIÓN ANIMAL*. 2011.
- [2] S. E. Hill and N. C. Murphy, "Analysis of Dog Adoption Success and Failure Using Surveys With Vignettes," *J. Appl. Anim. Welf. Sci.*, vol. 8705, no. February, pp. 1–13, 2016.
- [3] J. Alcacidinho *et al.*, "Leveraging Mobile Technology to Increase the Permanent Adoption of Shelter Dogs," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '15*, 2015, pp. 463–469.
- [4] D. Garrote, J. L. Poza, J. L. Posadas, and J. E. Simó, "Mobile-Based Distributed System for Managing Abandoned or Lost Pets Daniel," *Distributed Computing and Artificial Intelligence*, vol. 217, Valencia. Spain, pp. 197–200, 2013.
- [5] S. Chutichudet, T. Kanthathasiri, I. Ritsakunchai, and D. Wongsawang, "LFD: Lost and found dog application on mobile," *Proc. 2014 3rd ICT Int. Sr. Proj. Conf. ICT-ISPC 2014*, pp. 147–150, 2014.
- [6] R. Lampe and T. H. Witte, "Speed of Dog Adoption: Impact of Online Photo Traits," *J. Appl. Anim. Welf. Sci.*, vol. 8705, no. November, pp. 1–12, 2014.
- [7] M. K. Workman and C. L. Hoffman, "An Evaluation of the Role the Internet Site Petfinder Plays in Cat Adoptions," *J. Appl. Anim. Welf. Sci.*, vol. 18, no. 4, pp. 388–397, 2015.
- [8] "Learn Python - Programming Tutorial for Beginners." [Online]. Available: <https://www.programiz.com/python-programming>. [Accessed: 01-Feb-2017].
- [9] "TurboGears Tutorial." [Online]. Available: <http://www.w3ii.com/es/turbogears/default.html>. [Accessed: 12-Aug-2017].
- [10] "TurboGears Guía rápida." [Online]. Available: http://www.w3ii.com/es/turbogears/turbogears_quick_guide.html. [Accessed: 12-Aug-2017].
- [11] "Welcome to Pyramid, a Python Web Framework." [Online]. Available: <https://trypyramid.com/>. [Accessed: 13-Aug-2017].

- [12] "The Web framework for perfectionists with deadlines | Django." [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 01-Feb-2017].
- [13] "Writing your first Django app, part 1 | Django documentation | Django." [Online]. Available: <https://docs.djangoproject.com/en/1.10/intro/tutorial01/>. [Accessed: 01-Feb-2017].
- [14] "About - Materialize." [Online]. Available: <http://materializecss.com/about.html>. [Accessed: 13-Aug-2017].
- [15] "Sobre PostgreSQL | www.postgresql.org.es." [Online]. Available: http://www.postgresql.org.es/sobre_postgresql. [Accessed: 02-Feb-2017].
- [16] "PostgreSQL | 2ndQuadrant." [Online]. Available: <https://2ndquadrant.com/es/postgresql/>. [Accessed: 02-Feb-2017].
- [17] "Django REST framework." [Online]. Available: <http://www.django-rest-framework.org/>. [Accessed: 02-Feb-2017].
- [18] "¿Cómo instalar Nginx en Ubuntu 16.04? | DigitalOcean," 2016. [Online]. Available: <https://www.digitalocean.com/community/tutorials/como-instalar-nginx-en-ubuntu-16-04-es>. [Accessed: 02-Feb-2017].
- [19] C. Acenswhitepapers, San, and Rafael, "Servidor web Nginx, una clara alternativa a Apache," *Alcobendas*, vol. 14, pp. 900–103, 2810.
- [20] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele Univ.*, vol. 33, no. TR/SE-0401, p. 28, 2004.
- [21] J. Biolchini, P. G. Mian, A. Candida, and C. Natali, "Systematic Review in Software Engineering," *Engineering*, vol. 679, no. May, pp. 1–31, 2005.
- [22] D. Calidad and G. De, "Los WEB SERVICES Y CARACTERÍSTICAS DE CALIDAD," 2005.
- [23] "U.S. Pet Ownership, Community Cat and Shelter Population Estimates: The Humane Society of the United States." 13-Jan-2016.
- [24] "Exceso de mascotas causa abandono- Noticias de Cuenca - Azuay - Ecuador - El tiempo de Cuenca." 13-Jan-2016.
- [25] J. Méndez, "Aplicación Web escalable para gestión de información geolocalizada."

- [26] M. Besteiro and M. Rodriguez, "WebServices." [Online]. Available: <http://www.ehu.eus/mrodriguez/archivos/csharp/pdf/ServiciosWeb/WebServices.pdf>. [Accessed: 11-Jan-2016].
- [27] "IBM - Arquitectura orientada a servicios (SOA) de IBM - España." [Online]. Available: <https://www-01.ibm.com/software/es/solutions/soa/>. [Accessed: 12-Jan-2016].
- [28] D. González and M.-C. Marcos, "Responsive web design: diseño multidispositivo para mejorar la experiencia de usuario," *BiD textos Univ. Bibliotecon. i Doc.*, no. 31, p. 10 p., 2013.
- [29] Ó. Beltrán, "RinconEpidemio.indd - v20n1a09," *Revisiones sistemáticas de la literatura*. 13-Jan-2016.

11. Anexos

Anexo 1: Estudios seleccionados en la revisión sistemática de literatura

#	Artículo
S01	Alcaldinho, J., Valentin, G., Tai, S., Nguyen, B., Sanders, K., Jackson, M. M., ... Starner, T. (2015). Leveraging Mobile Technology to Increase the Permanent Adoption of Shelter Dogs. In <i>Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '15</i> (pp. 463–469). http://doi.org/10.1145/2785830.2785861
S02	Garrote, D., Poza, J. L., Posadas, J. L., & Simó, J. E. (2013). Mobile-Based Distributed System for Managing Abandoned or Lost Pets Daniel. <i>Distributed Computing and Artificial Intelligence</i> , 217, 197–200. http://doi.org/10.1007/978-3-319-00551-5
S03	Garrote-Hildebrand, D., Poza, J. L., Posadas, J. L., Simó, J. E., & Garcia, J. L. (2015). Distributed System for Intelligent Management of Lost or Abandoned Pets, 15(1), 91–101.
S04	Chutichudet, S., Kanthathasiri, T., Ritsakunchai, I., & Wongsawang, D. (2014). LFD: Lost and found dog application on mobile. <i>Proceedings of the 2014 3rd ICT International Senior Project Conference, ICT-ISPC 2014</i> , 147–150. http://doi.org/10.1109/ICT-ISPC.2014.6923238
S05	Lampe, R., & Witte, T. H. (2014). Speed of Dog Adoption: Impact of Online Photo Traits. <i>Journal of Applied Animal Welfare Science: JAAWS</i> , 8705(November), 1–12. http://doi.org/10.1080/10888705.2014.982796
S06	Workman, M. K., & Hoffman, C. L. (2015). An Evaluation of the Role the Internet Site Petfinder Plays in Cat Adoptions. <i>Journal of Applied Animal Welfare Science</i> , 18(4), 388–397. http://doi.org/10.1080/10888705.2015.1043366

Anexo 2: Entrevista a la Fundación “Ángeles con Patas”

Anexo 3: Entrevista a la Fundación Conciencia Animal

Fecha: 13 de enero 2016

Entrevistador: Nadia Susana Paucar Cabrera

1. ¿A qué fundación pertenece?

Conciencia Animal

2. ¿Qué tipo de rol cumple en la fundación?

Presidente

3. ¿Qué tipo de actividades y servicios brinda la fundación?

Defensa de animales, rescate según las circunstancias, recuperación y adopción

4. ¿Con qué tipo de mascotas trabaja la fundación?

Con perros y gatos principalmente

5. ¿Posee la fundación algún tipo de sistema web para la ejecución de sus procesos de adopción?

No, no poseemos ningún sistema web ni aplicación.

6. ¿Cómo se desarrolla el proceso de adopción de una mascota?

Primero se realiza una visita a la casa que desea adoptar para conocer el lugar. Se conversa con el adoptante (mayor de edad, o si es menor con autorización) sobre cómo debe cuidarse el animalito, dentro de esto debe estar de acuerdo principalmente con:

- Las visitas periódicas.
- La esterilización en caso de no entregarse esterilizado.

Se firma una carta compromiso con copias de cédula y finalmente se entrega al animalito.

7. ¿Qué datos son necesarios proporcionar por parte de la ciudadanía para solicitar una adopción?

Ser mayor de edad, o estar autorizado por los padres, ubicación de domicilio y datos generales.

8. ¿Cómo se almacenan los datos de adopciones o información en general?

Mediante hojas de compromiso que son archivadas en carpetas.

9. ¿Existe alguna normativa o ley del país que regule los pasos para una adopción?

No.

10. ¿Cuánto tiempo tarda en concretarse una adopción?

Unos 3 días mínimo

11. ¿Qué pasos se debe realizar para notificar una mascota extraviada?

Enviar foto, datos de la mascota como se encontraba en el instante de la pérdida (corte de pelo, collar y demás), y lugar exacto de pérdida.

Anexo 4: Documento de Especificación de Requerimientos

Especificación de requisitos de software IEEE 830

Proyecto: Desarrollo de una aplicación web de adopción de mascotas para la ciudad de Loja

1. Introducción

El presente documento consta de los requerimientos completos y necesarios para el funcionamiento del proyecto “Desarrollo de una aplicación web de adopción de mascotas en la ciudad de Loja”, con el fin de desarrollar el trabajo de titulación de la carrera de Ingeniería en Sistemas.

Este documento es el resultado de aplicar técnicas como la entrevista a fundaciones y colectivos a favor de los animales, encuestas y revisión sistemática de bibliografía. Este ERS se ha realizado tomando en cuenta el estándar IEEE-830.

1.1. Propósito

El objetivo de este documento es recolectar el listado de requerimientos necesarios para la elaboración de la aplicación web de la forma más detallada posible para cada una de las funcionalidades que debe cumplir.

Está dirigido para cualquier persona que tenga como fin el cuidado de las mascotas.

1.2. Alcance

La especificación de requisitos del sistema “PetHome”, tiene como objetivo ser un medio facilitador para la adopción de mascotas abandonadas en la ciudad de Loja, para eso se plantea las siguientes funcionalidades que formarán como alcance del proyecto:

- Gestión de usuarios
- Gestión de mascotas
- Gestión de adopciones
- Servicio Web
- Gestión de eventos

1.3. Personal involucrado

TABLA XXXVII. RESPONSABLE DEL DESARROLLO DEL PROYECTO

Nombre	Nadia Susana Paucar Cabrera
Rol	Desarrollador de software
Categoría profesional	Estudiante de Ingeniería en Sistemas
Responsabilidades	Análisis, diseño y desarrollo del sistema

Información de contacto	nspaucarc@unl.edu.ec
--------------------------------	----------------------

TABLA XXXVIII – DIRECTOR DEL PROYECTO

Nombre	Alfredo Vinicio Zuñiga Tinizaray
Rol	Director de trabajo de titulación
Responsabilidades	Supervisar y asesorar en el desarrollo del proyecto
Información de contacto	alfredo.zuniga@unl.edu.ec

1.4. Definiciones, acrónimos y abreviaturas

DEFINICIONES

- **Administrador:** Usuario encargado de la administración del sitio web.
- **Fundación/colectivo:** Persona jurídica sin ánimo de lucro, que cumple la voluntad de quien la fundó.
- **Adopción:** Proceso de tomar responsabilidad de un animal que un dueño previamente ha abandonado o dejado en un refugio de animales.
- **Base de datos:** Conjunto de datos organizados para almacenamiento.
- **Web Service:** Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones

ACRÓNIMOS

- **ERS:** Especificación de requisitos de software
- **RF:** Requisito funcional

2. Descripción general

2.1. Perspectiva del producto

El presente proyecto “PetHome” tiene como finalidad la construcción de una aplicación web que facilite la difusión y agilice el proceso de adopción de una mascota. Además, contiene otras funcionalidades como reporte de mascotas extraviadas y eventos para fundaciones protectoras de animales.

La aplicación web funcionará de forma independiente sin ser parte de un sistema mayor, adaptándose a cualquier dispositivo electrónico que tenga acceso a un navegador web.

2.2. Funcionalidad del producto

La aplicación “PetHome” contará con las siguientes funcionalidades:

- Gestión de usuarios
- Gestión de mascotas
- Gestión de adopciones
- Gestión de mascotas extraviadas
- Gestión de eventos
- Servicio Web

La aplicación web permitirá interactuar entre las personas afines al cuidado y protección de las mascotas. Para esto se contará con registro, identificación y gestión de usuarios, además de los módulos para gestionar las mascotas extraviadas, adopciones y eventos, contando con un buscador que facilitará la experiencia de navegabilidad del producto.

2.3. Características de los usuarios

Tipo de usuario	Usuario administrador
Formación	Ingeniero o técnico en sistemas
Actividades	<ul style="list-style-type: none"> • Gestión de usuarios • Gestión de mascotas • Gestión de eventos

Tipo de usuario	Usuario fundación
Formación	Básica, superior o tercer nivel
Actividades	<ul style="list-style-type: none"> • Gestionar adopciones • Consulta de mascotas extraviadas • Gestionar reportes • Gestionar eventos

Tipo de usuario	Usuario general
Formación	Básica, superior o tercer nivel
Actividades	<ul style="list-style-type: none"> • Gestionar mascota • Solicitar adopción • Gestionar mascota extraviada • Consulta de mascotas en adopción

2.4. Restricciones

Se utilizará el framework Django 1.9 con el lenguaje de programación Python, la base de datos será construida en PostgreSQL y para el desarrollo del web service se utilizará Django Rest Framework. Se requerirá de internet para su funcionamiento.

2.5. Suposiciones y dependencias

Los requisitos presentados están basados desde la premisa que el usuario tendrá siempre disponible una conexión a internet de banda ancha al igual que los navegadores web actualizados.

2.6. Evolución previsible del sistema

El producto está abierto a evolución, ya que se podrá integrar nuevos módulos a la funcionalidad del sitio web según se requiera.

3. Requisitos específicos

3.1. Requisitos comunes de los interfaces

3.1.1. Interfaces de usuario

Las interfaces respetarán la resolución de cada dispositivo en el que se esté visualizando, con el fin de obtener una integración adecuada.

3.1.2. Interfaces de hardware

Monitor	El sistema deberá mostrar información sobre la relación del cliente con la empresa
Teclado	Ingresar datos en el sistema por parte de los usuarios
Mouse	Ayudará al usuario a interactuar con diferentes componentes del sistema como botones, links, menús, etc
Impresora	Fundamental para impresión de reportes

3.2. Requisitos funcionales

Número del requisito	RF001
Nombre del requisito	Autenticar usuario
Descripción del requisito	El usuario podrá iniciar sesión ingresando su identificación y contraseña
Prioridad del Requisito	Alta

Número del requisito	RF002
Nombre del requisito	Registro de usuarios
Descripción del requisito	Los usuarios deberán registrarse en el sistema para poder acceder al mismo proporcionando una cuenta de correo y contraseña o utilizando una cuenta de redes sociales como facebook, twitter y google+.
Prioridad del Requisito	Alta

Número del requisito	RF003
Nombre del requisito	Roles de usuarios
Descripción del requisito	La aplicación web debe diferenciar el tipo de usuario para la visualización de la información
Prioridad del Requisito	Alta

Número del requisito	RF004
-----------------------------	-------

Nombre del requisito	Registro de mascotas extraviadas
Descripción del requisito	Se mostrará un formulario y mapa de ubicación para registrar los datos de la mascota extraviada. Se podrá subir hasta tres imágenes por mascota.
Prioridad del Requisito	Alta

Número del requisito	RF005
Nombre del requisito	Catálogo de mascotas extraviadas
Descripción del requisito	La aplicación web debe mostrar una lista de las mascotas que se encuentran extraviadas ordenadas de manera cronológica del más reciente al más antiguo.
Prioridad del Requisito	Alta

Número del requisito	RF006
Nombre del requisito	Registro de mascotas en adopción
Descripción del requisito	Se mostrará un formulario para ingresar la información de la mascota con su respectiva foto.
Prioridad del Requisito	Alta

Número del requisito	RF007
Nombre del requisito	Galería de mascotas disponibles para adopción

Descripción del requisito	La aplicación web debe mostrar una lista de las mascotas que se encuentran disponibles para la adopción
Prioridad del Requisito	Alta

Número del requisito	RF008
Nombre del requisito	Solicitar adopción
Descripción del requisito	Una vez que el usuario haya seleccionado la mascota que desea adoptar, se mostrará un formulario para realizar la solicitud de adopción
Prioridad del Requisito	Alta

Número del requisito	RF009
Nombre del requisito	Emitir reportes de solicitudes de adopción
Descripción del requisito	Se debe emitir un reporte que indique que mascota desea ser adoptada y por quienes fueron solicitadas.
Prioridad del Requisito	Alta

Número del requisito	RF010
Nombre del requisito	Gestionar eventos

Descripción del requisito	El usuario fundación debe poder crear, modificar y eliminar eventos.
Prioridad del Requisito	Alta

Número del requisito	RF011
Nombre del requisito	Registro de mascotas encontradas
Descripción del requisito	Se emitirá reportes de casos de éxito de mascotas que han vuelto a sus respectivos dueños
Prioridad del Requisito	Alta

Número del requisito	RF012
Nombre del requisito	API para consulta de mascotas extraviadas
Descripción del requisito	Proporcionar un recurso Rest para la consulta de mascotas extraviadas
Prioridad del Requisito	Alta

3.3. Requisitos no funcionales

3.3.1. Requisitos de rendimiento

El sistema debe tener un tiempo de respuesta aceptable.

3.3.2. Seguridad

El sistema utilizará https para la navegación de los usuarios.

3.3.3. Fiabilidad

Todo tipo de operación o solicitud de información debe realizarse sin ningún tipo de conflicto y sin incidentes.

3.3.4. Disponibilidad

El sistema debe estar disponible las 24 horas del día. En caso de algún fallo, el sistema debe recuperarse en el menor tiempo posible.

3.3.5. Mantenibilidad

Una vez terminado la aplicación web, las tareas de administración y mantenimiento serán realizados por la desarrolladora. Además, el sistema podrá ampliarse con nuevas funcionalidades según como se requiera.

3.3.6. Portabilidad

El sistema será multiplataforma ya que se podrá ejecutar desde cualquier navegador web.

Anexo 5. Plantilla para pruebas funcionales

TABLA XXXIX. PLANTILLA DE PRUEBAS FUNCIONALES

Especificación de Pruebas: [Nombre de CU Específico y CU General/Código]		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
Descripción: [Breve resumen general de lo que realiza el caso de uso]		
Ejecutar: [Nombre del Caso del Uso Específico] con valores correctos		
Descripción: [Descripción detallada del caso de uso, indicando posibles riesgos, tipos de usuarios que puedan realizar la acción, etc.]		
Condiciones de Ejecución: Indicar las condiciones necesarias para que el caso de uso funcione correctamente		
Entrada: Indicar las acciones que realiza el usuario de manera ordenada para cumplir con el caso de uso. [Realizar con valores correctos]		
Resultado Esperado: [Indicar qué se espera como resultado de la ejecución del usuario]		
Evaluación de la Prueba: Indicar si la prueba fue correcta o incorrecta.		
Ejecutar: [Nombre del Caso Uso Específico] con valores incorrectos		
[Se realiza los mismos pasos anteriores, pero con valores incorrectos, para medir la cobertura del sistema.]		
Descripción: [Descripción detallada del UC Específico, indicando posibles riesgos, tipos de usuarios que puedan realizar la acción, etc.]		
Condiciones de Ejecución: Indicar las condiciones necesarias para que el caso de uso funcione correctamente.		
Entrada: [Indicar las acciones que realiza el usuario de manera ordenada para cumplir con el UC. [Realizar con valores incorrectos]		
Resultado Esperado: [Indicar qué se espera como resultado de la ejecución del usuario]		
Evaluación de la Prueba: Indicar si la prueba es aceptable o no aceptable.		

Anexo 6. Pruebas funcionales de PetHome

Caso de Uso: Gestionar usuario (Cód. CU001)

TABLA XL. PRUEBAS FUNCIONALES DE CREAR DE USUARIO

Especificación de Pruebas: Crear usuario / CU001 (Administrador)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema permite crear usuarios. De acuerdo al ingreso de los datos se valida y se almacena en la base de datos.		
Ejecutar: Crear usuarios con datos correctos		
Descripción: El administrador debe ingresar a la administración de django y seleccionar la sección de usuarios. Se mostrará los usuarios registrados con las opciones de modificar y eliminar. Además, se podrá seleccionar “Nuevo Usuario”, el cual redireccionará al formulario para el registro.		
Condiciones de Ejecución: El administrador debe estar con la sesión activa.		
Entrada: <ul style="list-style-type: none">• Iniciar sesión como usuario-administrador.• Ingresar a la administración del sistema web.• Seleccionar el campo “usuarios”.• Seleccionar “agregar usuario”.• Llenar los campos obligatorios (username, nombres, apellidos, correo electrónico, contraseña, grupo de usuario).• El sistema verifica que los campos sean correctos y que el usuario no exista anteriormente.• El sistema guarda los datos del nuevo usuario.		
Resultado Esperado: Los datos son almacenados en la base de datos.		
Evaluación de la Prueba: Prueba satisfactoria.		
Ejecutar: Crear usuarios con datos incorrectos		
Descripción:		

El administrador debe ingresar a la administración de django y seleccionar la sección de usuarios. Se mostrará los usuarios registrados con las opciones de modificar y eliminar. Además, se podrá seleccionar “Nuevo Usuario”, el cual redireccionará al formulario para el registro.
Condiciones de Ejecución: El administrador debe estar con la sesión activa.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-administrador. • Ingresar a la administración del sistema web. • Seleccionar el campo “usuarios”. • Seleccionar “agregar usuario”. • Llenar los campos obligatorios (username, nombres, apellidos, correo electrónico, contraseña, grupo de usuario). • El sistema verifica que los campos sean correctos y que el usuario no exista anteriormente. • Si los datos son incorrectos, el sistema presenta un mensaje de error.
Resultado Esperado: Los datos no son almacenados en la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.

TABLA XLI. PRUEBA FUNCIONAL DE ACTUALIZAR USUARIO

Especificación de Pruebas: Actualizar usuario / CU001 (Administrador)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema modificar usuarios. De acuerdo al ingreso de los datos se valida y se almacena en la base de datos.		
Ejecutar: Actualizar usuarios con datos correctos		
Descripción:		

El administrador debe ingresar a la administración de django y seleccionar la sección de usuarios. Seleccionar “modificar usuario” y se redireccionará a la página para editar.
Condiciones de Ejecución: El administrador debe estar con la sesión activa y haber seleccionado el usuario a editar.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-administrador. • Ingresar a la administración del sistema web. • Seleccionar el campo “usuarios”. • Seleccionar el usuario a modificar. • El administrador modificará los campos necesarios. • El sistema verifica que los campos sean correctos. • El sistema actualiza los datos del usuario.
Resultado Esperado: Los datos son actualizados y almacenados en la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.
Ejecutar: Actualizar usuarios con datos incorrectos
Descripción: El administrador debe ingresar a la administración de django y seleccionar la sección de usuarios. Seleccionar “modificar usuario” y se redireccionará a la página para editar.
Condiciones de Ejecución: El administrador debe estar con la sesión activa y haber seleccionado el usuario a editar.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-administrador. • Ingresar a la administración del sistema web. • Seleccionar el campo “usuarios”. • Seleccionar el usuario a modificar. • El administrador modificará los campos necesarios. • El sistema verifica que los campos sean correctos. • Si los datos ingresados son incorrectos, el sistema mostrará un mensaje de error.
Resultado Esperado:

Los datos no son almacenados en la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.

Caso de Uso: Gestionar mascota en adopción (Cód. CU002)

TABLA XLII. PRUEBA FUNCIONAL DE REGISTRAR MASCOTA EN ADOPCIÓN

Especificación de Pruebas: Registrar mascota en adopción / CU002 (Usuario-fundación)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema permite crear nueva mascota en adopción. De acuerdo al ingreso de los datos, se valida y se almacena en la base de datos.		
Ejecutar: Crear mascota en adopción con datos correctos		
Descripción: El usuario-fundación, una vez iniciada sesión, ingresará a mis mascotas y luego en nueva adopción. Se presentarán dos opciones de registro entre perro y gato el mismo que enviará a un formulario vacío para el nuevo registro.		
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.		
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-fundación. • Seleccionar “mis mascotas”. • Seleccionar “nueva adopción”. • Elegir el tipo de mascota a registrar (perro, gato). • Llenar el formulario de registro de mascota en adopción. • El sistema verifica que los campos sean correctos. • El sistema guarda los datos de la nueva mascota en adopción. 		
Resultado Esperado: Los datos son almacenados en la base de datos.		
Evaluación de la Prueba: Prueba satisfactoria.		
Ejecutar: Crear mascota en adopción con datos incorrectos		

Descripción: El usuario-fundación, una vez iniciada sesión, ingresará a mis mascotas y luego en nueva adopción. Se presentarán dos opciones de registro entre perro y gato el mismo que enviará a un formulario vacío para el nuevo registro.
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-fundación. • Seleccionar “mis mascotas”. • Seleccionar “nueva adopción”. • Elegir el tipo de mascota a registrar (perro, gato). • Llenar el formulario de registro de mascota en adopción. • El sistema verifica que los campos sean correctos. • Si los datos son incorrectos, el sistema muestra un mensaje de error.
Resultado Esperado: Los datos no son almacenados en la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.

TABLA XLIII. PRUEBAS FUNCIONALES DE ACTUALIZAR MASCOTA EN ADOPCIÓN

Especificación de Pruebas: Actualizar mascota en adopción / CU002 (Usuario-fundación)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema permite actualizar mascota en adopción. De acuerdo al ingreso de los datos, se valida y se actualiza en la base de datos.		
Ejecutar: Crear mascota en adopción con datos correctos		
Descripción:		

El usuario-fundación, una vez iniciada sesión, ingresará a mis mascotas y luego en el ícono de actualizar de la mascota deseada. Se presentarán un formulario con los datos de la mascota para realizar la modificación.
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-fundación. • Seleccionar “mis mascotas”. • Seleccionar ícono de actualizar de la mascota deseada. • Editar los campos deseados del formulario. • El sistema verifica que los campos sean correctos. • El sistema guarda los datos actualizados de la mascota en adopción.
Resultado Esperado: Los datos son almacenados en la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.
Ejecutar: Crear mascota en adopción con datos incorrectos
Descripción: El usuario-fundación, una vez iniciada sesión, ingresará a mis mascotas y luego en el ícono de actualizar de la mascota deseada. Se presentarán un formulario con los datos de la mascota para realizar la modificación.
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-fundación. • Seleccionar “mis mascotas”. • Seleccionar ícono de actualizar de la mascota deseada. • Editar los campos deseados del formulario. • El sistema verifica que los campos sean correctos. • Si los datos son incorrectos, el sistema muestra un mensaje de error.
Resultado Esperado: Los datos no son almacenados en la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.

Caso de Uso: Realizar adopción (Cód. CU003)

TABLA XLIV. PRUEBAS FUNCIONALES DE SOLICITAR ADOPCIÓN

Especificación de Pruebas: Solicitar mascota en adopción / CU003 (Usuario-general)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema permitirá solicitar una mascota en adopción.		
Ejecutar: Solicitar mascota en adopción con datos correctos		
Descripción: El usuario-general, una vez iniciada sesión, ingresará a mascotas en adopción y escogerá la mascota que desea adoptar. Se presentarán un formulario para realizar la solicitud. Una vez verificado que los datos sean correctos, se envían al usuario-fundación que pertenece la mascota.		
Condiciones de Ejecución: El usuario-general debe estar con la sesión activa.		
Entrada: <ul style="list-style-type: none">• Iniciar sesión como usuario-general.• Seleccionar “mascotas en adopción”.• Escoger la mascota deseada para adoptar.• Seleccionar “solicitar adopción”.• Llenar el formulario de solicitud de adopción.• El sistema verifica que los campos sean correctos.• El sistema guarda los datos de la nueva solicitud de adopción y envía la solicitud a la fundación correspondiente		
Resultado Esperado: Los datos son almacenados en la base de datos y la solicitud es enviada al usuario-fundación		
Evaluación de la Prueba: Prueba satisfactoria.		
Ejecutar: Solicitar mascota en adopción con datos correctos		
Descripción:		

El usuario-general, una vez iniciada sesión, ingresará a mascotas en adopción y escogerá la mascota que desea adoptar. Se presentarán un formulario para realizar la solicitud. Una vez verificado que los datos sean correctos, se envían al usuario-fundación que pertenece la mascota.
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-general. • Seleccionar “mascotas en adopción”. • Escoger la mascota deseada para adoptar. • Seleccionar “solicitar adopción”. • Llenar el formulario de solicitud de adopción. • El sistema verifica que los campos sean correctos. • Si los datos son incorrectos o incompletos, el sistema muestra un mensaje de error.
Resultado Esperado: Los datos no son almacenados en la base de datos ni enviados al usuario-fundación
Evaluación de la Prueba: Prueba satisfactoria.

Caso de Uso: Gestionar mascota extraviada (Cód. CU004)

TABLA XLV. PRUEBA FUNCIONAL DE CREAR MASCOTA EXTRAVIADA

Especificación de Pruebas: Crear mascota extraviada / CU004 (Usuario-general)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema permitirá crear mascota extraviada.		
Ejecutar: Crear mascota extraviada con datos correctos		
Descripción: El usuario-general, una vez iniciada sesión, podrá reportar como extraviada a su mascota registrada		
Condiciones de Ejecución:		

<p>El usuario-general debe estar con la sesión activa.</p> <p>Debe poseer una mascota registrada previamente</p>
<p>Entrada:</p> <ul style="list-style-type: none"> • Iniciar sesión como usuario-general. • Seleccionar “mascotas extraviadas”. • Seleccionar “nueva mascota extraviada”. • Seleccionar la mascota extraviada. • Llenar el formulario para reportar la mascota extraviada. • El sistema verifica que los campos sean correctos. • El sistema guarda los datos de la nueva mascota extraviada y publica en la sección de inicio de los perfiles de usuarios.
<p>Resultado Esperado:</p> <p>Los datos son almacenados en la base de datos y publicada en la sección de inicio.</p>
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>
<p>Ejecutar: Crear mascota extraviada con datos correctos</p>
<p>Descripción:</p> <p>El usuario-general, una vez iniciada sesión, podrá reportar como extraviada a su mascota registrada</p>
<p>Condiciones de Ejecución:</p> <p>El usuario-general debe estar con la sesión activa.</p> <p>Debe poseer una mascota registrada previamente</p>
<p>Entrada:</p> <ul style="list-style-type: none"> • Iniciar sesión como usuario-general. • Seleccionar “mascotas extraviadas”. • Seleccionar “nueva mascota extraviada”. • Seleccionar la mascota extraviada. • Llenar el formulario para reportar la mascota extraviada. • El sistema verifica que los campos sean correctos. • Si los datos son incorrectos o incompletos, el sistema muestra un mensaje de error.
<p>Resultado Esperado:</p> <p>Los datos no son almacenados en la base de datos ni publicados en la sección de inicio.</p>
<p>Evaluación de la Prueba:</p>

Prueba satisfactoria.

Caso de Uso: Gestionar evento (Cód. CU005)

TABLA XLVI. PRUEBA FUNCIONAL DE CREAR EVENTO

Especificación de Pruebas: Crear evento / CU005 (Usuario-fundación)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema permitirá crear eventos.		
Ejecutar: Crear eventos con datos correctos		
Descripción: El usuario-fundación, una vez iniciada sesión, seleccionará “Mis eventos” en el menú, en el mismo elegirá nuevo evento, se presentará un formulario con campos vacíos para el nuevo registro.		
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.		
Entrada: <ul style="list-style-type: none">• Iniciar sesión como usuario-fundación.• Seleccionar “mis eventos”.• Seleccionar “nueva evento”.• Llenar el formulario para evento.• El sistema verifica que los campos sean correctos.• El sistema guarda los datos del nuevo evento y publica en la sección de eventos de los perfiles de usuarios.		
Resultado Esperado: Los datos son almacenados en la base de datos y publicada en los perfiles de los usuarios.		
Evaluación de la Prueba: Prueba satisfactoria.		
Ejecutar: Crear eventos con datos incorrectos		
Descripción:		

El usuario-fundación, una vez iniciada sesión, seleccionará “Mis eventos” en el menú, en el mismo elegirá nuevo evento, se presentará un formulario con campos vacíos para el nuevo registro.
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.
Entrada: <ul style="list-style-type: none"> • Iniciar sesión como usuario-fundación. • Seleccionar “mis eventos”. • Seleccionar “nueva evento”. • Llenar el formulario para evento. • El sistema verifica que los campos sean correctos. • Si los datos son incorrectos o incompletos, el sistema muestra un mensaje de error.
Resultado Esperado: Los datos no son almacenados en la base de datos ni publicados en los perfiles de los usuarios.
Evaluación de la Prueba: Prueba satisfactoria.

TABLA XLVII. PRUEBA FUNCIONAL DE ACTUALIZAR EVENTO

Especificación de Pruebas: Actualizar evento / CU005 (Usuario-fundación)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema permitirá actualizar eventos.		
Ejecutar: Eliminar evento con datos correctos		
Descripción: El usuario-fundación, una vez iniciada sesión, seleccionará “Mis eventos” en el menú, en el mismo elegirá el evento que desee eliminar. Se presentará un mensaje de confirmación.		
Condiciones de Ejecución: El usuario-fundación debe estar con la sesión activa.		

Entrada: <ul style="list-style-type: none"> • El actor selecciona el evento a eliminar. • Seleccionar ícono de eliminar. • El sistema muestra mensaje de confirmación. • El sistema elimina el evento.
Resultado Esperado: Los datos son eliminados de la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.

Caso de Uso: Consulta de mascotas extraviadas (Cód. CU006)

TABLA XLVIII. PRUEBA FUNCIONAL DE PROMOCIONAR MASCOTA EXTRAVIADA

Especificación de Pruebas: Promocionar mascota extraviada / CU006 (WebService)		
Historial de Revisiones:		
Fecha:	Descripción:	Autor:
		Nadia Paucar Cabrera
Descripción: El sistema proporciona servicio para consumir información de mascotas extraviadas.		
Ejecutar: Promocionar mascota extraviada		
Descripción: Mediante una aplicación móvil, se consumirá un servicio web creado para la consulta de mascotas extraviadas registradas en la web.		
Condiciones de Ejecución: Debe existir mascotas extraviadas.		
Entrada: <ul style="list-style-type: none"> • Consultar servicio REST mediante la url proporcionada por el sistema 		
Resultado Esperado: Los datos de mascotas extraviadas serán presentados en la aplicación móvil.		
Evaluación de la Prueba: Prueba satisfactoria.		

Anexo 7: Pruebas de accesibilidad

Página de registro de fundación

La prueba de accesibilidad realizada a la página de registro de fundación muestra algunos errores en la estructura HTML de la misma como se puede observar a continuación:

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change.

Showing results for https://www.pethomeloja.tk/usuario/registrar_fundacion/

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

- Error** Bad value 'true' for attribute 'required' on element 'input'.
From line 80, column 23; to line 80, column 128
`<input class="form-control" id="id_username" maxlength="30" name="username" required="true" type="text" />+</code>`
- Error** Bad value 'true' for attribute 'required' on element 'input'.
From line 89, column 25; to line 89, column 126
`<input class="form-control" id="id_email" maxlength="254" name="email" required="true" type="email" />+</code>`
- Error** Bad value 'true' for attribute 'required' on element 'input'.
From line 117, column 23; to line 117, column 132
`<input class="form-control" id="id_first_name" maxlength="30" name="first_name" required="true" type="text" />+</code>`
- Error** Bad value 'true' for attribute 'required' on element 'input'.
From line 121, column 23; to line 121, column 130
`<input class="form-control" id="id_last_name" maxlength="30" name="last_name" required="true" type="text" />+</code>`
- Error** The 'for' attribute of the 'label' element must refer to a non-hidden form control.
From line 81, column 23; to line 81, column 44
`<label for="username">Usuari</code>`
- Error** The 'for' attribute of the 'label' element must refer to a non-hidden form control.
From line 90, column 25; to line 90, column 43
`<label for="email">Email</code>`
- Error** The 'for' attribute of the 'label' element must refer to a non-hidden form control.
From line 101, column 25; to line 101, column 46
`<label for="password">Contra</code>`
- Error** The 'for' attribute of the 'label' element must refer to a non-hidden form control.
From line 108, column 25; to line 108, column 46
`<label for="password">Repeti</code>`
- Error** The 'for' attribute of the 'label' element must refer to a non-hidden form control.
From line 118, column 23; to line 118, column 46
`<label for="first_name">Nombre</code>`
- Error** The 'for' attribute of the 'label' element must refer to a non-hidden form control.
From line 122, column 23; to line 122, column 45
`<label for="last_name">Apellido</code>`

Figura 53. Prueba de accesibilidad incorrecta página de registro de fundación

Se realizó la corrección del código y se aplicó nuevamente la prueba, dando como resultado lo siguiente:

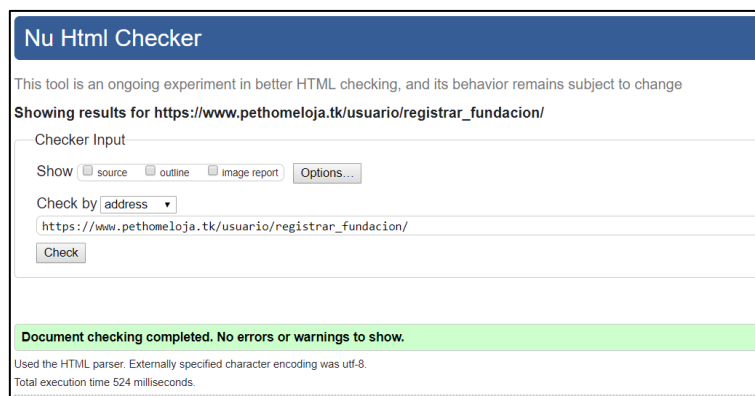


Figura 54. Prueba de accesibilidad correcta de registro de fundación

La figura 54 muestra que errores ni advertencias, por lo que la prueba fue superada satisfactoriamente.

Página de inicio de sesión



Figura 55. Prueba de accesibilidad incorrecta página de inicio de sesión

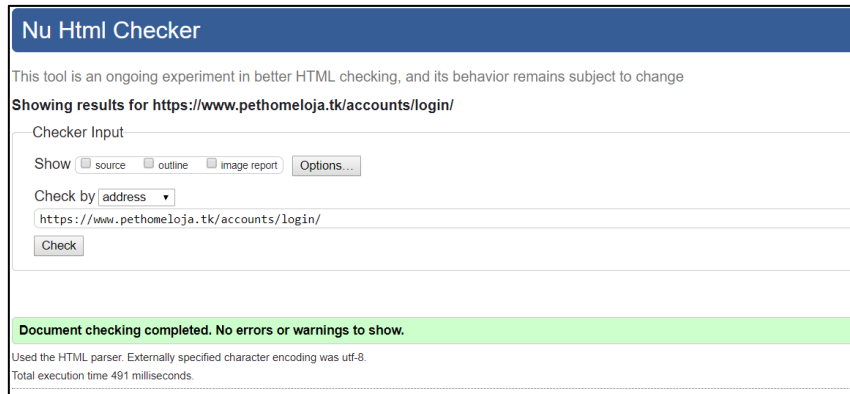


Figura 56. Prueba de accesibilidad correcta de página de inicio de sesión

La figura 56 muestra que errores ni advertencias, por lo que la prueba fue superada satisfactoriamente.

Página de recuperar contraseña

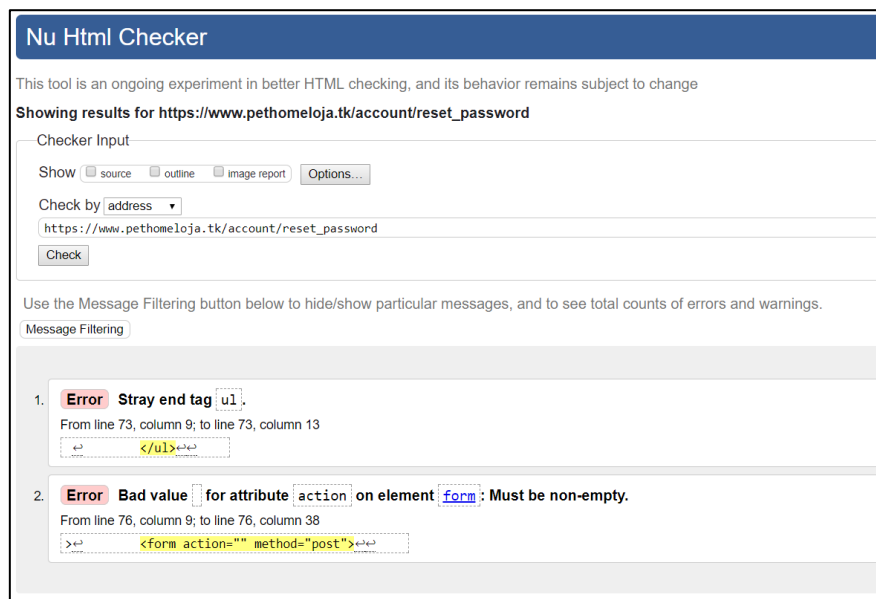


Figura 57. Prueba de accesibilidad incorrecta página de recuperar contraseña

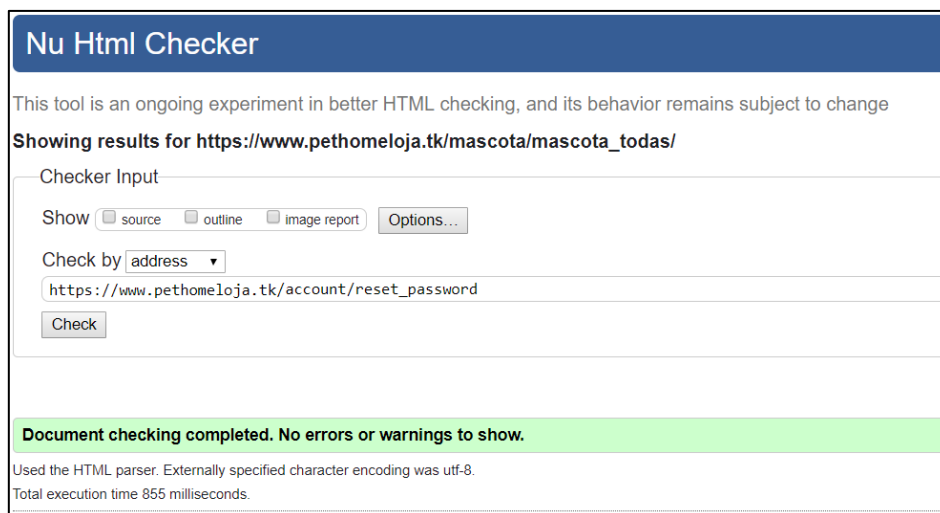


Figura 58. Prueba de accesibilidad correcta de página de recuperar contraseña

La figura 58 muestra que errores ni advertencias, por lo que la prueba fue superada satisfactoriamente.

Página de mascotas en adopción

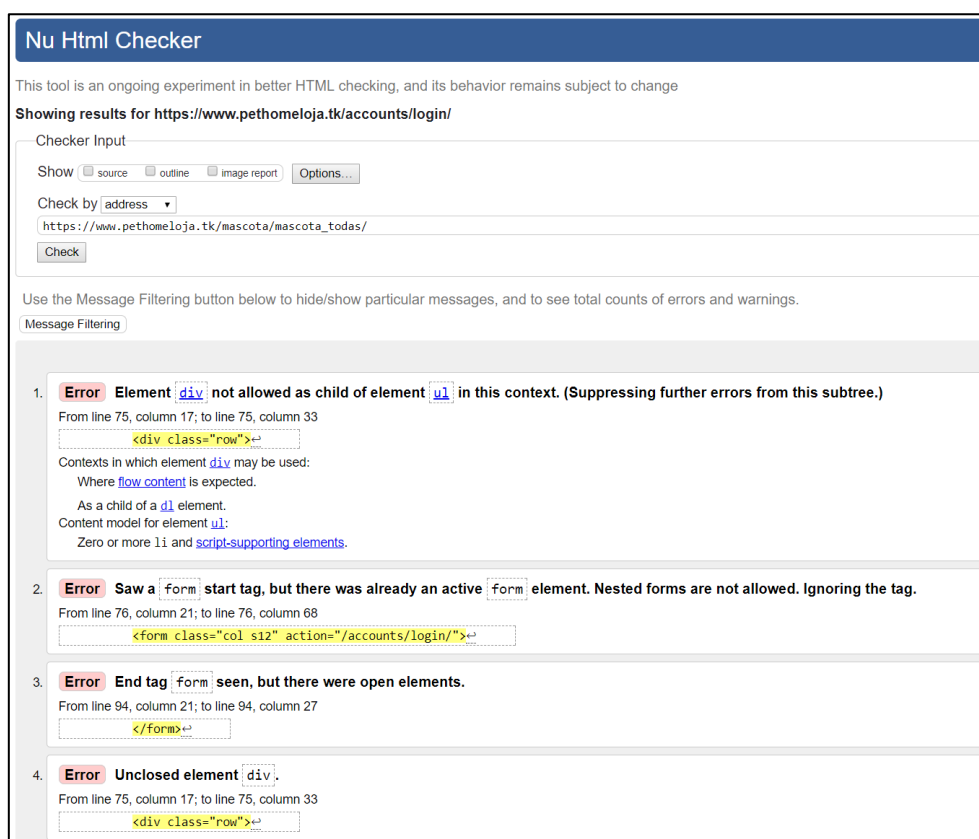


Figura 59. Prueba de accesibilidad incorrecta página de mascotas en adopción

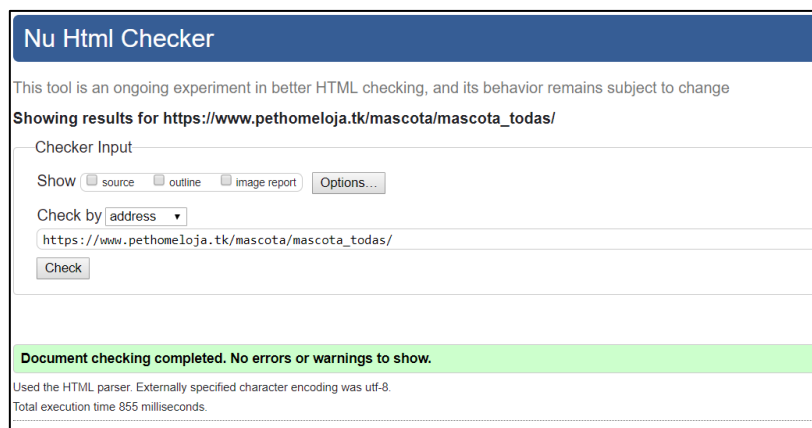


Figura 60. Prueba de accesibilidad correcta de página de mascotas en adopción

La figura 60 muestra que errores ni advertencias, por lo que la prueba fue superada satisfactoriamente.

Los commits de corrección del código para las pruebas de accesibilidad pueden ser observadas en el repositorio GitHub <https://github.com/nadiapc128/SistemaAdopcion>.