



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

**DESARROLLO DE UNA APLICACIÓN WEB PROGRESIVA PARA
LA IDENTIFICACIÓN Y BÚSQUEDA DE MASCOTAS
EXTRAVIADAS EN LA CIUDAD DE RIOBAMBA**

Trabajo de Integración Curricular

TIPO: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO DE SOFTWARE

AUTOR:

MATHEW PATRICIO AVILÉS CHÁVEZ

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

DESARROLLO DE UNA APLICACIÓN WEB PROGRESIVA PARA LA IDENTIFICACIÓN Y BÚSQUEDA DE MASCOTAS EXTRAVIADAS EN LA CIUDAD DE RIOBAMBA

Trabajo de Integración Curricular

TIPO: Proyecto Técnico

Presentado para optar el grado académico de:

INGENIERO EN SOFTWARE

AUTOR: MATHEW PATRICIO AVILÉS CHÁVEZ

DIRECTOR: MSC. JULIO ROBERTO SANTILLÁN CASTILLO

Riobamba – Ecuador

2023

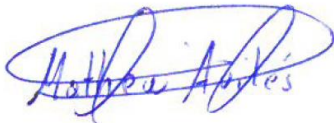
© 2023, Mathew Patricio Avilés Chávez

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

Yo, Mathew Patricio Avilés Chávez, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 11/12/2023



Mathew Patricio Avilés Chávez

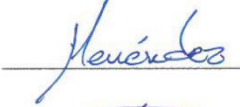

0604834549

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; tipo Proyecto Técnico , **DESARROLLO DE UNA APLICACIÓN WEB PROGRESIVA PARA LA IDENTIFICACIÓN Y BÚSQUEDA DE MASCOTAS EXTRAVIADAS EN LA CIUDAD DE RIOBAMBA**, realizado por el señor: **Mathew Patricio Avilés Chávez**, ha sido minuciosamente revisado por los Miembros del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Jorge Ariel Menéndez Verdecia PRESIDENTE DEL TRIBUNAL		2023-12-11
Msc. Julio Roberto Santillán Castillo DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-12-11
Ing. Jaime David Camacho Castillo ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-12-11

DEDICATORIA

A Dios por la vida, a mi familia porque me ayudaron a ser una persona buena y me inculcaron el estudio.

Mathew Avilés.

AGRADECIMIENTO

El más sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo, por darme la oportunidad de obtener una profesión y ser una ayuda a la sociedad.

Mathew Avilés.

ÍNDICE DE CONTENIDO

INDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xii
INDICE DE GRÁFICOS.....	xiii
INDICE DE ANEXOS	xiv
RESUMEN.....	xv
SUMMARY	xvi
INTRODUCCIÓN	1
CAPÍTULO I	
1 DIAGNÓSTICO DEL PROBLEMA.....	2
1.1 Planteamiento del problema.....	2
1.1.1 Antecedentes	2
1.1.2 Formulación del problema	3
1.1.3 Sistematización del problema.....	3
1.2 Justificación.....	4
1.2.1 Justificación teórica	4
1.2.2 Justificación aplicativa.....	4
1.3 Objetivos.....	5
1.3.1 Objetivo general	5
1.3.2 Objetivos específicos	6
CAPÍTULO II	
2 MARCO TEÓRICO	7
2.1 Aplicaciones web progresivas	7

2.2	Comparación de imágenes	7
2.3	Aprendizaje automático	7
2.3.1	Teachable Machine	8
2.4	Metodología de desarrollo software	9
2.4.1	Scrumban	9
2.5	Lenguajes de programación	12
2.5.1	Python	12
2.5.2	Python para comparar imágenes	13
2.5.3	Tensorflow	13
2.5.4	Keras	14
2.6	Base de datos	15
2.6.1	SQLite	15
2.7	Lenguajes de estilo	16
2.7.1	CSS	16
2.8	Variable de estudio	16
2.8.1	Usabilidad	16
2.8.2	Escala de Likert.....	19
2.9	Evaluación del producto software.....	20
2.9.1	ISO/IEC 25010.....	20
2.10	Trabajos relacionados	20
CAPÍTULO III		
3	MARCO METODOLÓGICO.....	23
3.1	Introducción	23
3.2	Tipo de estudio	23

3.2.1	Métodos y técnicas.....	23
3.3	Metodología para la determinación de la usabilidad	24
3.3.1	Operacionalización de variables	26
3.4	Población y muestra	27
3.4.1	Población y muestra.....	27
3.4.2	Análisis económico	27
3.4.3	Fuente de financiamiento	28
3.4.4	Recursos Hardware.....	28
3.4.5	Recursos Software.....	29
3.4.6	Riesgos.....	30
3.5	Desarrollo del sistema web utilizando la metodología de desarrollo ágil SCRUMBAN.....	32
3.5.1	Meta.....	32
3.5.2	Cola de historias	32
3.5.3	Análisis	32
3.5.4	Desarrollo.....	34
3.5.5	Pruebas.....	45
3.5.6	Implementación.....	45
3.5.7	Terminado.....	45
 CAPÍTULO IV		
4	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	46
4.1	Introducción.....	46
4.2	Resultados de usabilidad.....	46
4.2.1	Operabilidad.....	46

4.2.2	Reconocibilidad de la adecuación	48
4.2.3	Aprendizabilidad.....	49
4.2.4	Estética de la interfaz de usuario	51
4.3	Análisis de resultados de la variable usabilidad	52
4.4	Niveles de evaluación de la usabilidad	54
CAPÍTULO V		
5	CONCLUSIONES Y RECOMENDACIONES	55
BIBLIOGRAFÍA		
ANEXOS		

INDICE DE TABLAS

Tabla 1-2 Escala de Likert	19
Tabla 2-3 Métodos y técnicas.....	23
Tabla 3-3 Métricas de evaluación de la usabilidad	25
Tabla 4-3 Operacionalización conceptual de variables	26
Tabla 5-3 Operacionalización metodológica de variables.....	26
Tabla 6-3 Presupuesto estimado.....	28
Tabla 7-3 Recursos Hardware	28
Tabla 8-3 Recursos Software	29
Tabla 9-3 Riesgos.....	30
Tabla 10-3 Tareas por hacer.....	32
Tabla 11-3 Historias de usuario	34
Tabla 12-3 Diccionario de datos de la tabla registro_u	41
Tabla 13-4 Resultados del cuestionario de la subcaracterística Operabilidad.	46
Tabla 14-4 Resultados del cuestionario de la subcaracterística Reconocibilidad de la adecuación.	48
Tabla 15-4 Resultados del cuestionario de la subcaracterística aprendizabilidad.....	49
Tabla 16-4 Resultados del cuestionario de la subcaracterística Estética de la interfaz de usuario.	51
Tabla 17-4 Resultados usabilidad	52

ÍNDICE DE FIGURAS

Figura 1-2 Fases de Scrumban	10
Figura 2-3 Miembros y equipos de desarrollo	33
Figura 3-3 Tablero De tareas	34
Figura 4-3 Conceptualización del sistema	36
Figura 5-3 Diagrama físico de la base de datos	37
Figura 6-3 Diagrama de casos de uso	38
Figura 7-3 Diagrama de componentes	38
Figura 8-3 Diagrama de actividades para el registro de usuario	39
Figura 9-3 Diagrama de actividades para el registro de la mascota	40
Figura 10-3 Prototipo de la página de inicio	43
Figura 11-3 Lista de tareas completadas	45
Figura 12-4 Calificaciones de adjetivos, puntajes de aceptabilidad y escalas de calificaciones en relación con los porcentajes promedios del cuestionario SUS	54

INDICE DE GRÁFICOS

Gráfico 1-4 Operabilidad de la aplicación sobre 5 puntos	47
Gráfico 2-4 Promedio de Reconocibilidad de la adecuación sobre 5 puntos	48
Gráfico 3-4 Promedio de Aprendizabilidad sobre 5 puntos	50
Gráfico 4-4 Promedio de la Estética de la interfaz de usuario sobre 5 puntos	51
Gráfico 5-4 Nivel de usabilidad.	53
Gráfico 6-4 Nivel de usabilidad de la aplicación	53

INDICE DE ANEXOS

Anexo A: Requisitos funcionales

Anexo B: Requisitos no funcionales

Anexo C: Diagramas de procesos

Anexo D: Diccionario de datos

Anexo E: Pruebas

Anexo F: Cuestionario Sumi Usabilidad

Anexo G: Respuestas de los usuarios al cuestionario

Anexo H: Implementación mediante un contenedor en Docker

RESUMEN

En el marco de este trabajo de integración curricular, se afrontó el reto de crear una aplicación web progresiva diseñada para localizar y encontrar mascotas perdidas en Riobamba. Siguiendo un enfoque aplicativo, se detallaron de manera exhaustiva tanto los requerimientos funcionales como los no funcionales de la aplicación. Los módulos esenciales se concibieron e implementaron con éxito, abarcando desde el sistema de ingreso hasta la sección dedicada a las mascotas, focalizada en su identificación y búsqueda. La evaluación de la usabilidad, llevada a cabo conforme a los lineamientos de la norma ISO 25010, evidenció que la aplicación alcanza un nivel de usabilidad del 85.66%, un valor considerado satisfactorio en este contexto. Las conclusiones de este estudio resaltan la precisa identificación de requerimientos a través del análisis de historias de usuarios, la ejecución exitosa de los módulos clave, incorporando todas las funciones requeridas, y la adhesión a los estándares de usabilidad. Asimismo, se plantean recomendaciones con miras a mejoras futuras, como la potencial implementación de un sistema de rastreo basado en microchips para las mascotas, la adaptación de versiones móviles para plataformas Android, iOS y la exploración de posibles colaboraciones con instituciones gubernamentales o fundaciones de protección animal para ampliar el impacto y la utilidad del sistema. En esencia, esta tesis enriquece el campo al introducir una solución pragmática y técnicamente sólida para un problema social, validando la eficacia de una aplicación web progresiva que facilita la ubicación y rescate de mascotas perdidas, al tiempo que proporciona perspectivas prometedoras para futuros desarrollos y colaboraciones que expandan su alcance y utilidad.

Palabras clave: <DESARROLLO WEB>, <APLICACIÓN WEB>, <INTELIGENCIA ARTIFICIAL>, <REDES NEURONALES CONVOLUCIONALES>, <PYTHON (SOFTWARE)>, <BÚSQUEDA>, <MASCOTAS>, <RIOBAMBA (CANTÓN)>.

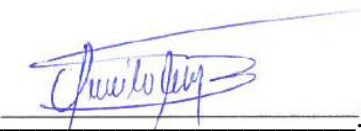


0002-DBRA-UPT-2024

SUMMARY

Within the framework of this curricular integration work, the challenge of creating a progressive web application designed to locate and find lost pets in Riobamba was faced. Following an applicative approach, both the functional and non-functional requirements of the application were exhaustively detailed. The essential modules were conceived and successfully implemented, ranging from the entry system to the section dedicated to pets, focused on their identification and search. The usability evaluation, carried out in accordance with the guidelines of the ISO 25010 standard, showed that the application reaches a usability level of 85.66%, a value considered satisfactory in this context. The conclusions of this study highlight the precise identification of requirements through the analysis of user stories, the successful execution of key modules, incorporating all required functions, and adherence to usability standards. Likewise, recommendations are made for future improvements, such as the potential implementation of a tracking system based on microchips for pets, the adaptation of mobile versions for Android and iOS platforms, and the exploration of possible collaborations with government institutions or animal protection foundations to expand the impact and utility of the system. In essence, this degree work enriches the field by introducing a pragmatic and technically sound solution to a social problem, validating the effectiveness of a progressive web application that facilitates the location and rescue of lost pets, while providing promising prospects for future developments and collaborations. that expand its reach and usefulness.

Keywords: <WEB DEVELOPMENT>, <WEB APPLICATION>, <ARTIFICIAL INTELLIGENCE>, <CONVOLUTIONAL NEURAL NETWORKS>, <PYTHON (SOFTWARE)>, <SEARCH>, <PETS>, <RIOBAMBA (CANTON)>.



Lic. Nelly Padilla P. Mgs.

0603818717

DOCENTE FIE

INTRODUCCIÓN

En el contexto de la ingeniería en software, donde la tecnología evoluciona constantemente y la necesidad de soluciones innovadoras crece, se abre paso a la búsqueda de aplicaciones prácticas que aborden problemáticas sociales y potencien la calidad de vida. En esta perspectiva, el presente estudio se sumerge en el desarrollo de una aplicación web progresiva diseñada para identificar y localizar mascotas extraviadas en la ciudad de Riobamba. El cruce entre la creciente utilización del software, la atención puesta en la usabilidad y la cuestión de las mascotas desaparecidas converge en un proyecto que aspira no solo a aplicar conocimientos en ingeniería, sino también a generar un impacto positivo en la comunidad.

La concepción de esta aplicación web progresiva surge en respuesta a la imperiosa necesidad de facilitar la ubicación y el rescate de mascotas perdidas, un desafío que afecta tanto a los dueños de las mascotas como a los propios animales. Con el objetivo general de desarrollar una herramienta que cumpla con métricas de usabilidad según la norma ISO 25010, se trazan objetivos específicos que abarcan desde la definición pormenorizada de requisitos funcionales y no funcionales, hasta el diseño e implementación de módulos esenciales y la evaluación de la usabilidad del sistema.

Este enfoque se cimienta en la aplicación práctica de los saberes adquiridos durante la formación universitaria, para abordar problemas reales y pertinentes. Además, introduce un elemento innovador al incorporar redes neuronales convolucionales para el reconocimiento de mascotas, con la intención de potenciar la eficiencia en la identificación.

CAPÍTULO I

1 DIAGNÓSTICO DEL PROBLEMA

1.1 Planteamiento del problema

1.1.1 *Antecedentes*

En la actualidad las principales plataformas y herramientas que usan las personas para la búsqueda de animales extraviados son el uso de redes sociales, foros de internet y la creación de carteles con información de la mascota. Sin embargo, estos métodos presentan algunas limitaciones, como la falta de unificación y persistencia en la información y la poca efectividad en la localización de la mascota (Fancypets 2022).

La fundación de rescate animal FRAE (2022) menciona los problemas culturales importantes como la falta de uso de collar con placa, la costumbre de dejar deambular solos a los animales y una de la más importantes, por su repercusión, la falta de esterilizaciones termina aumentando la población de animales en situación de calle.

Llanos y Avelino (2019) propusieron la implementación de una aplicación web y móvil para ubicar y rescatar animales domésticos extraviados o abandonados para la fundación de rescate animal Ecuador, la cual utiliza como principal medio de geolocalización mapas con las coordenadas en las que fueron vistos o encontrados las mascotas.

Orozco Villacres (2022) propone la implementación de collares con códigos QR para la identificación de las mascotas, los mismos que al momento de ser escaneados lanzan una alerta mediante notificación al celular y correo electrónico de la ubicación de la mascota mediante el uso de Google Maps.

Para mejorar la búsqueda y la identificación de mascotas perdidas, se propone el uso de redes neuronales convolucionales las cuales son un tipo de algoritmo de aprendizaje automático que se utiliza en la identificación y clasificación de imágenes. Son inspiradas en la forma en que el cerebro humano procesa la información visual.

Estas redes reciben una imagen como entrada y la descomponen en pequeñas partes llamadas "filtros". Cada filtro procesa una sección de la imagen y la convierte en una característica o patrón que la red puede utilizar para clasificar la imagen en una categoría específica (por ejemplo, identificar si una imagen corresponde a un perro o un gato).

A medida que la imagen se procesa a través de la red, los filtros se combinan en capas para detectar patrones cada vez más complejos. Finalmente, la red produce una salida que indica la categoría a la que la imagen pertenece.

Este enfoque permite una mayor precisión y eficacia en la búsqueda, ya que se utiliza la inteligencia artificial para la identificación de patrones y características de cada mascota (Liu et al. 2017).

En este sentido, el presente trabajo de titulación busca desarrollar una aplicación web progresiva que contenga toda la información de la mascota extraviada y de su dueño, así como también un algoritmo de comparación de imágenes mediante redes neuronales convolucionales. Para ello, se trabajará en la creación de un data set de datos de animales perdidos registrados mediante la aplicación, lo que permitirá mejorar la efectividad de la búsqueda e identificación de las mascotas perdidas.

Para la implementación del sistema se ha propuesto a la clínica veterinaria Vita Farm la cual es un centro especializado en servicios médicos para las mascotas como: vacunas, laboratorio, ecografía, cirugía y consulta externa, además de proporcionar alimento especializado para cada mascota y accesorios para las mismas como collares, juguetes, etc.

1.1.2 *Formulación del problema*

¿Cómo desarrollar una aplicación web para la identificación y búsqueda de mascotas extraviadas, la misma que deberá tener un adecuado nivel de usabilidad?

1.1.3 *Sistematización del problema*

¿Qué pasos se debe seguir para determinar los requisitos funcionales y no funcionales para el desarrollo de la aplicación web de identificación y búsqueda de mascotas?

¿Cuáles son los módulos que se desarrollaran en el sistema web?

¿Cuál es el nivel de operabilidad de la aplicación web según las métricas de usabilidad de la ISO 25010?

1.2 Justificación

1.2.1 *Justificación teórica*

La tecnología de reconocimiento de imágenes ha experimentado un gran avance en los últimos años, permitiendo la creación de herramientas poderosas para la identificación y búsqueda de objetos en imágenes y videos. Una aplicación de gran importancia en esta área es la identificación de mascotas extraviadas, donde la rapidez y precisión son fundamentales para lograr su ubicación y devolución a sus dueños.

Este trabajo de integración curricular tiene como objetivo el desarrollo de una aplicación web progresiva que permita la identificación y búsqueda de mascotas extraviadas en la ciudad de Riobamba, mediante el uso de técnicas de reconocimiento de imágenes. Esta aplicación se basará en plataformas de aprendizaje profundo, como Teachable Machine, TensorFlow y Keras, y se implementará en el lenguaje de programación Python.

La aplicación permitirá a los usuarios subir una imagen de la mascota extraviada y, mediante técnicas de procesamiento de imágenes y redes neuronales, identificar si la mascota se encuentra en alguna base de datos de mascotas perdidas y encontradas. Además, la aplicación también permitirá a los usuarios reportar el extravío de sus mascotas, lo que contribuirá a aumentar la cantidad de datos disponibles para el entrenamiento del modelo de reconocimiento de imágenes.

Este trabajo de titulación se va a realizar con el propósito de ayudar a las personas a que sus mascotas que por alguna razón desaparecieron de su hogar puedan regresar al mismo, para esto se va a analizar las causas que lo provocan. Además del gran inconveniente que es para la sociedad que existan más perros callejeros.

1.2.2 *Justificación aplicativa*

Para mejorar la identificación y búsqueda de mascotas y agilizar los procesos de identificación de semejanzas se propone crear una herramienta para comparar características similares para localizar a la mascota rápidamente. Para el desarrollo de la aplicación web se utilizará como base el lenguaje Python con un gestor de base de datos SQLite, esta aplicación debe ser muy intuitiva y fácil de usar, con un gestor de archivos para poder subir imágenes de la mascota extraviado y de las mascotas que se encuentren sin su respectivo dueño o las quieran dar en adopción.

La importancia de este trabajo radica en la necesidad de contar con herramientas efectivas para la identificación y búsqueda de mascotas extraviadas, lo que puede significar la diferencia entre su

rescate o su desaparición definitiva. La aplicación web progresiva desarrollada en este trabajo permitirá una rápida identificación y búsqueda de mascotas extraviadas, lo que contribuirá a la disminución de la tasa de mascotas perdidas y a su pronta recuperación. Para ello contaremos con los siguientes módulos:

- **Login:** la aplicación contará con un módulo de registro mediante el ingreso de credenciales del usuario.
- **Mascotas:** este módulo contará con un menú de opciones, tales como:
 - **Ver mascotas perdidas:** aquí se mostrarán todas las mascotas que han sido registradas en el sistema como perdidas.
 - **Ver mascotas encontradas:** aquí se mostrarán todas las mascotas que los usuarios las hayan encontrado han sido registradas al sistema.
 - **Ver mascotas en adopción:** aquí se mostrarán las macotas que están en estado de adopción.
 - **Registrar una mascota:** en este módulo vamos a poder registrar una mascota ya sea como perdida, encontrada o en adopción.
 - **Buscar mascota:** este módulo permite mediante el uso de la cámara poder identificar la mascota que ha sido registrada como encontrada o perdida.
- **Nosotros:** este módulo brinda información de la clínica veterinaria Vita farm, su historia, a que se dedica, los servicios que frece, su ubicación y demás datos de la institución.

El trabajo de integración curricular incluye la línea y programa de investigación de la ESPOCH en el eje de las TICs, en la línea de investigación de tecnologías de la información y comunicación con el programa de ingeniería de software. Mientras que en la línea de investigación de la Escuela de ingeniería en Software en el área de Ciencias con el programa de investigación de Ingeniería de Software y con el ámbito de estudio análisis y diseño de software. Este trabajo cumple con el objetivo 5 del Plan Nacional de desarrollo el cual nos ayuda a impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria.

1.3 Objetivos

1.3.1 *Objetivo general*

Desarrollar una aplicación web para la identificación y búsqueda de mascotas extraviadas de acuerdo con las métricas de usabilidad de la ISO 25010

1.3.2 *Objetivos específicos*

- Determinar los requisitos funcionales y no funcionales para el desarrollo de la aplicación web.
- Desarrollar los módulos de login, mascotas, nosotros planteados para el aplicativo web.
- Evaluar la usabilidad de la aplicación web para la identificación y búsqueda de mascotas extraviadas con la norma ISO 25010

CAPÍTULO II

2 MARCO TEÓRICO

En este capítulo se hace una descripción general de los conceptos que se van a manejar dentro del trabajo de integración curricular como las herramientas a utilizar, lenguaje de programación, metodología, bases de datos, frameworks y las métricas para medir la operabilidad de la aplicación según la norma ISO/IEC 25010.

2.1 Aplicaciones web progresivas

Rodríguez et al (2019) mencionan que una Aplicación Web Progresiva (PWA, por sus siglas en inglés: Progressive Web App) es una forma de aplicación web que combina características de aplicaciones móviles y sitios web tradicionales. Las PWAs son accesibles a través de navegadores y ofrecen una experiencia progresiva, adaptándose a diferentes dispositivos y navegadores.

2.2 Comparación de imágenes

La comparación de imágenes según De Giusti y Tarrío (2000) mencionan que se refiere al proceso de analizar y comparar dos o más imágenes para determinar si son similares o no. Esto puede ser útil en una variedad de aplicaciones, como la seguridad, la inteligencia artificial, la robótica y la visión por computadora. Hay varios métodos para comparar imágenes, como la comparación de características, la comparación de patrones y la comparación de contenido. La comparación de características implica analizar y comparar características específicas de las imágenes, como las formas, los colores y las texturas. La comparación de patrones implica buscar patrones similares en las imágenes, mientras que la comparación de contenido implica analizar el contenido de las imágenes.

Cada método tiene sus propias ventajas y desventajas y es mejor para ciertos casos de uso. Por ejemplo, la comparación de características es adecuada para imágenes con pocos cambios sutiles, mientras que la comparación de contenido es mejor para imágenes con muchos cambios sutiles como es nuestro caso para comparar las diferentes mascotas (De Giusti y Tarrío 2000).

2.3 Aprendizaje automático

Smarandache (2022) menciona que el aprendizaje automático es una rama de la inteligencia artificial que se enfoca en el diseño y desarrollo de algoritmos y modelos que permiten a las computadoras

aprender a partir de datos, sin la necesidad de ser programadas explícitamente para cada tarea específica. En otras palabras, el aprendizaje automático permite a las máquinas aprender por sí mismas a través de la experiencia, de manera similar a cómo los humanos aprendemos a partir de la experiencia.

Existen diferentes tipos de aprendizaje automático, incluyendo el aprendizaje supervisado, no supervisado y por refuerzo. En el aprendizaje supervisado, se entrena al modelo con ejemplos etiquetados, es decir, datos que tienen una respuesta conocida. En el aprendizaje no supervisado, el modelo debe encontrar patrones y relaciones en los datos sin una respuesta conocida previamente. En el aprendizaje por refuerzo, el modelo aprende a través de la retroalimentación proporcionada por un entorno en el que está interactuando QuinnRadich(2021).

2.3.1 Teachable Machine

Es una herramienta de aprendizaje automático en línea desarrollada por Google que permite a los usuarios crear modelos de aprendizaje personalizados utilizando una interfaz gráfica fácil de usar. La herramienta utiliza un enfoque de aprendizaje automático de transferencia. El aprendizaje automático de transferencia es un enfoque de aprendizaje automático en el que se aprovechan los conocimientos adquiridos por un modelo previamente entrenado en una tarea, para resolver un problema diferente pero relacionado según la página oficial de Teachable machine (<https://teachablemachine.withgoogle.com> 2022).

En lugar de entrenar un modelo desde cero para cada tarea específica, el aprendizaje automático de transferencia utiliza los conocimientos previamente adquiridos por el modelo en una tarea anterior, para mejorar su rendimiento en una tarea diferente pero relacionada Carney et al (2020).

Por ejemplo, un modelo entrenado para reconocer imágenes de gatos podría utilizarse como base para entrenar un nuevo modelo para reconocer imágenes de perros. En lugar de comenzar el entrenamiento desde cero, el nuevo modelo podría aprovechar los conocimientos adquiridos por el modelo anterior sobre la detección de características importantes de las imágenes, como las formas y los contornos, para acelerar el proceso de entrenamiento y mejorar el rendimiento general. El aprendizaje automático de transferencia puede ser particularmente útil en casos donde no se cuenta con grandes cantidades de datos de entrenamiento, o cuando el costo de entrenar un modelo desde cero es prohibitivo. También puede ser útil en casos donde se necesita adaptar un modelo previamente entrenado para una tarea específica, como la detección de objetos en imágenes médicas (Garcia et al. 2022).

La herramienta es compatible con imágenes, sonidos y datos de sensores, y permite a los usuarios etiquetar cada muestra para entrenar el modelo. Una vez que se ha entrenado el modelo, se puede utilizar para una variedad de aplicaciones, como la clasificación de imágenes o sonidos en tiempo real. Teachable Machine se ha utilizado en una variedad de aplicaciones, desde la creación de juegos interactivos hasta la identificación de objetos y el reconocimiento de voz. La herramienta también se ha utilizado en la educación, donde los maestros pueden utilizarla para enseñar a los estudiantes sobre el aprendizaje automático y la inteligencia artificial (Bajaña Moran, y Paladines Gallardo 2021).

La herramienta está disponible de forma gratuita en línea y es compatible con una variedad de plataformas, incluyendo computadoras de escritorio, tabletas y teléfonos inteligentes. Además, Google ofrece una serie de tutoriales y guías para ayudar a los usuarios a comenzar a utilizar la herramienta.

2.4 Metodología de desarrollo software

Una metodología de desarrollo de software es un enfoque sistemático y estructurado que se utiliza para planificar, diseñar, implementar, probar y mantener el software.

La metodología de desarrollo de software proporciona un marco de trabajo que guía a los equipos de desarrollo de software en la manera en que trabajan juntos para construir y entregar software. La metodología describe los procesos, las prácticas, las herramientas y los roles que se utilizan en el desarrollo de software (Maida 2015).

2.4.1 *Scrumban*

La metodología Scrumban es una combinación de las metodologías Scrum y Kanban. Scrum es un marco ágil para el desarrollo de productos y Kanban es un enfoque para la gestión de proyectos. Juntas, estas metodologías se utilizan para mejorar la eficiencia y la eficacia en el desarrollo de productos (Laoyan 2022).

Scrumban tiene las siguientes fases:

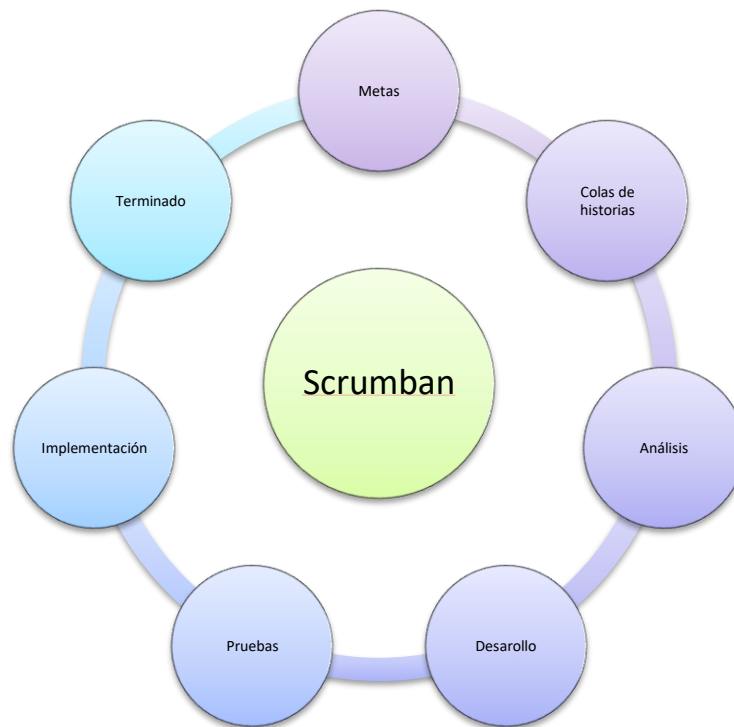


Figura 1-2 Fases de Scrumban

Fuente: (Team 2015)

Según Team Yodiz (2015), las fases de SCRUMBAN son las siguientes:

- **Metas.** En esta etapa, el equipo define ampliamente sus metas. Una meta puede ser un objetivo general que el equipo espera lograr realizando múltiples tareas más pequeñas.
- **Cola de historias.** En esta etapa, las metas se descomponen en múltiples historias. Se crea una larga lista de historias en esta etapa.
- **Análisis.** Aquí es donde Scrumban se diferencia de otros enfoques. Las historias creadas en la etapa anterior ahora se analizan y se seleccionan algunas para continuar trabajando en ellas.
- **Desarrollo.** Se comienza a trabajar en las historias seleccionadas.
- **Pruebas.** Una vez que se ha trabajado en las historias, los resultados son probados por los equipos de control de calidad (QA).
- **Implementación.** Luego se ponen en práctica los resultados.
- **Terminado.** Todas las historias completadas se marcan como terminadas.

La metodología Scrumban ayuda a mejorar la planificación y la gestión del trabajo, permitiendo a los equipos ser más ágiles y adaptarse mejor a los cambios en el proyecto Ibarra Guzmán et al (2014).

Scrum es una metodología ágil para la gestión de proyectos que se enfoca en la entrega incremental y continua de valor al cliente. Kanban es otra metodología ágil que se enfoca en la optimización del flujo de trabajo y la mejora continua. Scrumban es una combinación de ambas metodologías, utiliza los principios de Scrum para planificar y priorizar el trabajo, y los principios de Kanban para optimizar el flujo de trabajo. Es adecuado para equipos que necesitan un enfoque ágil para la planificación y priorización de trabajo, pero también quieren optimizar el flujo de trabajo para mejorar la eficiencia y la entrega continua (Azabache y Caruajulca 2019).

Algunos de los conceptos más importantes de Scrumban son:

2.4.1.1 *Iteraciones (Sprints)*

El objetivo principal de estas divisiones es el tener contacto y revisión continua con el cliente. De esta manera cualquier tipo de problema se puede detectar y tratar con rapidez, y cualquier requisito puede ser fácilmente refinado con tal de adaptarse a las necesidades, quizás cambiantes, del cliente. Esta filosofía de revisiones semanales es ideal para el desarrollo de proyectos de investigación e innovación, ya que las tecnologías evolucionan muy rápidamente y en cualquier momento las prioridades del proyecto pueden cambiar, y ésta facilita este tipo de cambio y adaptación (Azabache Medina y Caruajulca Sánchez 2019).

2.4.1.2 *Tablero de tareas (Task Board)*

Es un tablero que permite ver el estado de cada tarea en el momento. Al estar todo en un mismo tablero permite la visualización de este completamente con tan solo una ojeada. Permite ver qué tareas se están llevando a cabo en el momento, quién las está desarrollando, las tareas pendientes y las tareas ya completadas. El tablero está formado por 4 columnas: Backlog: Tareas listas para ser realizadas. Doing: Tareas que cada miembro está llevando a cabo. Review: Tareas completadas pendientes de ser revisadas, Done: Tareas completadas y revisadas (Rodríguez Franco 2014).

2.4.1.3 *Historias de usuario*

Los requisitos funcionales del proyecto se han definido usando las historias de usuario. Es decir, cada uno de estos elementos define una funcionalidad que ha de tener el sistema. Cada historia de usuario consiste en un par de frases descriptivas sobre la funcionalidad desde el punto de vista del usuario. El título de esta ha de ser corto y conciso (cuánto más conciso mejor). La historia de

usuario también incluye los criterios de aceptación expresados desde el punto de vista del usuario. En toda historia de usuario ha de quedar claro el motivo y la funcionalidad esperada. Las historias de usuario pueden ser más complejas o menos, dependiendo de la granularidad con que se quiera tratar (Rodríguez Franco 2014).

Se ha seleccionado la metodología Scrumban ya que permite planificar y priorizar el trabajo, pero al mismo tiempo permite optimizar el flujo de trabajo y mejorar la eficiencia.

2.5 Lenguajes de programación

2.5.1 Python

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos (González 2016).

Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados). La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo, los lenguajes interpretados son más flexibles y más portables. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta, generando archivos .pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones. La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo (González Duque 2011).

El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS.) por lo que si no utilizamos librerías específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios. La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos en nuestro programa. La ejecución del programa

consiste en una serie de interacciones entre los objetos. Python también permite la programación imperativa, programación funcional y programación orientada a aspectos (Hurtado 2014).

2.5.2 Python para comparar imágenes

Python es un lenguaje de programación muy potente y versátil que se utiliza en una variedad de aplicaciones de procesamiento de imágenes, incluyendo la comparación de imágenes. Algunas de las bibliotecas más populares en Python para comparar imágenes incluyen:

- **OpenCV:** es una biblioteca de visión por computadora de código abierto que proporciona una gran cantidad de funciones para el procesamiento de imágenes y vídeo, incluyendo funciones para comparar imágenes.
- **scikit-image:** es una biblioteca de procesamiento de imágenes en Python que proporciona una amplia gama de funciones para el análisis de imágenes, incluyendo la comparación de imágenes.
- **ImageHash:** es una biblioteca que proporciona una forma fácil de calcular un "hash" o resumen numérico de una imagen, que luego se puede comparar con otras imágenes para determinar si son similares.
- **PIL o Pillow:** Es una biblioteca de procesamiento de imágenes que proporciona una amplia gama de funciones para el procesamiento y comparación de imágenes.
- **Keras** es una biblioteca de redes neuronales de código abierto escrita en Python que se ejecuta sobre TensorFlow. Fue desarrollada con el objetivo de facilitar la creación de redes neuronales para aplicaciones de aprendizaje automático y deep learning. Keras proporciona una API de alto nivel y modular para la construcción de redes neuronales. Permite construir modelos de aprendizaje profundo de forma rápida y sencilla, sin tener que preocuparse demasiado por los detalles de implementación.

Por supuesto, hay muchas otras opciones disponibles, pero estas son algunas de las más populares y ampliamente utilizadas. Es importante evaluar las diferentes opciones y seleccionar la que mejor se adapte a sus necesidades específicas.

2.5.3 Tensorflow

TensorFlow es una biblioteca de código abierto de aprendizaje automático desarrollada por Google Brain Team. Fue lanzada en 2015 y se ha convertido en una de las herramientas más populares y ampliamente utilizadas para la creación y entrenamiento de modelos de aprendizaje automático y de inteligencia artificial (Torres 2020).

TensorFlow permite a los desarrolladores crear modelos de aprendizaje profundo y entrenarlos para que puedan realizar tareas específicas, como clasificación de imágenes, reconocimiento de voz, detección de objetos, traducción automática y muchas otras aplicaciones. La biblioteca funciona mediante la creación de gráficos computacionales que representan la estructura de un modelo de aprendizaje automático, y luego utiliza la optimización de gradiente descendente para ajustar los parámetros del modelo durante el entrenamiento (Ortiz y Jiménez 2018).

La principal característica de TensorFlow es su capacidad para manejar grandes conjuntos de datos y modelos complejos. TensorFlow es compatible con la ejecución en CPU y GPU, lo que permite una mayor escalabilidad y velocidad de procesamiento. También es compatible con la ejecución en múltiples dispositivos, lo que permite que los modelos se ejecuten en múltiples servidores o dispositivos móviles.

Además, TensorFlow es compatible con una amplia gama de lenguajes de programación, incluyendo Python, C++, Java y otros, lo que permite a los desarrolladores integrar la biblioteca en sus aplicaciones existentes (Torres 2020).

2.5.4 Keras

Keras es una biblioteca de aprendizaje profundo de código abierto escrita en Python que funciona como una interfaz de alto nivel para otros frameworks de aprendizaje automático, como TensorFlow, Theano y Microsoft Cognitive Toolkit (CNTK). Fue desarrollado para simplificar el proceso de creación y entrenamiento de redes neuronales profundas (Palacios De La Cruz y Pinargote Tomalá 2021).

Las principales ventajas de Keras son:

- **Fácil de usar:** Keras es una herramienta fácil de usar que permite a los desarrolladores crear modelos de aprendizaje profundo con una sintaxis simple y fácil de entender. Esto reduce el tiempo y la complejidad del desarrollo de modelos.
- **Modularidad:** Keras está diseñado de forma modular, lo que permite a los desarrolladores crear modelos complejos mediante la combinación de capas individuales. También permite a los desarrolladores agregar fácilmente nuevas capas personalizadas.
- **Flexibilidad:** Keras es compatible con múltiples backends de aprendizaje profundo, incluidos TensorFlow, Theano y CNTK, lo que permite a los desarrolladores elegir el backend que mejor se adapte a sus necesidades.

- Velocidad: Keras está optimizado para aprovechar la capacidad de procesamiento de las unidades de procesamiento gráfico (GPU), lo que acelera el tiempo de entrenamiento de los modelos.

Las principales desventajas de Keras son:

- Falta de personalización: aunque Keras es una herramienta fácil de usar, su simplicidad puede limitar la personalización de los modelos.
- Falta de documentación: aunque Keras es una herramienta popular, su documentación es limitada en comparación con otros frameworks.
- Menor control: debido a que Keras es una biblioteca de alto nivel, puede proporcionar a los desarrolladores menos control sobre los detalles técnicos del modelo que otras bibliotecas de aprendizaje profundo.

2.6 Base de datos

Una base de datos es un conjunto de información organizada de manera que pueda ser utilizada eficientemente. Un directorio telefónico, un diccionario, un calendario o un libro de recetas son ejemplos de bases de datos. La información en una base está organizada en forma de registros. Cada registro contiene toda la información sobre una persona o un elemento de la base (Barcenes Llanos y Cedeño Avelino 2019).

2.6.1 *SQLite*

SQLite es un sistema de gestión de bases de datos relacional que te permite almacenar y recuperar información de manera eficiente. A diferencia de otros sistemas de gestión de bases de datos, SQLite es ligero y no requiere una configuración complicada. Además, no funciona como un servidor independiente, sino que se ejecuta directamente dentro de la aplicación que lo utiliza SQLite(2023).

SQLite utiliza un lenguaje llamado SQL (Structured Query Language), que es un estándar utilizado en muchas bases de datos. SQL te permite crear, modificar y consultar la base de datos de forma sencilla y legible. Con SQL, puedes crear tablas para almacenar tus datos y ejecutar consultas para recuperar información específica de la base de datos (Owens 2006).

Una de las ventajas de SQLite es su amplia compatibilidad. Puedes utilizar SQLite en una variedad de entornos y lenguajes de programación, incluyendo Python, Java, C#, C++, entre otros. Esto significa que puedes integrar SQLite en tus aplicaciones sin problemas y aprovechar sus

características sin tener que aprender un nuevo sistema de gestión de bases de datos en cada lenguaje (Kreibich 2010).

Además, SQLite es confiable y rápido en operaciones de lectura y escritura. Aunque está diseñado principalmente para aplicaciones de un solo usuario o pequeñas aplicaciones en red, también puede manejar cargas de trabajo más pesadas con eficacia. Si bien no es ideal para aplicaciones de alta concurrencia con múltiples usuarios simultáneos, para proyectos más pequeños y personales, SQLite es una excelente opción (Owens 2006).

2.7 Lenguajes de estilo

2.7.1 CSS

Se trata de un lenguaje de estilo para la web que permite definir el aspecto de las páginas. Este lenguaje se utiliza para establecer la presentación de un archivo escrito en HTML. CSS también se emplea para controlar el formato y diseño de los documentos web. Por ejemplo, es posible definir el tamaño, color y grosor de línea de un elemento, así como el espacio entre líneas o párrafos. Sin embargo, es importante tener en cuenta el siguiente punto: 'Al igual que HTML, CSS no es en realidad un lenguaje de programación. Se trata de un lenguaje de hojas de estilo que posibilita la aplicación selectiva de estilos a elementos en documentos HTML'.

A pesar de no tener la naturaleza de un lenguaje de programación, la utilización de CSS en el diseño del sistema resulta sumamente crucial, ya que brinda una amplia diversidad en cuanto a personalización. Se podría decir que contribuye a conferir un aspecto más llamativo al sistema. Fundamentalmente, CSS fue desarrollado con el propósito de proveer un método eficiente para presentar y estilizar el contenido HTML, logrando así que las páginas web cuenten con una interfaz placentera para el usuario (Durango 2015).

2.8 Variable de estudio

2.8.1 Usabilidad

Según Hassan Montero (2002), la usabilidad (dentro del campo del desarrollo web) es la disciplina que estudia la forma de diseñar sitios web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible.

La mejor forma de crear un sitio web usable es realizando un diseño centrado en el usuario, diseñando para y por el usuario, en contraposición a lo que podría ser un diseño centrado en la tecnología o uno centrado en la creatividad u originalidad.

Alrededor del concepto de usabilidad existe una gran cantidad de conceptos relacionados semánticamente, que a veces son confundidos entre sí. La Usabilidad de un sitio web está estrechamente relacionada con la "findability", ya que cuanto más sencillo resulte para el usuario navegar por un sitio web, mayores posibilidades tendrá de encontrar lo que busca.

La usabilidad refleja la calidad de los sistemas interactivos de información, evaluando el equilibrio entre productividad, navegabilidad, accesibilidad y optimización en función del logro de objetivos comerciales. Este enfoque no solo persigue equilibrar desarrollos tecnológicos óptimos con requisitos del cliente, el mercado, y metas empresariales, sino que también fomenta una interacción continua y actualizada entre la herramienta web, sus administradores y los usuarios. La usabilidad no es una ciencia precisa, sino un arte preciso, ya que un ligero cambio en el texto de un enlace, el contraste con el fondo o el espacio circundante de un elemento, pueden marcar la diferencia entre el éxito y la mediocridad. Todo se fundamenta en la ergonomía, la facilidad de aprendizaje, uso y adaptación a las necesidades del usuario, así como la experiencia placentera del sistema durante su uso. El usuario no debe enfocarse en la interfaz, sino en su labor. El diseño debe garantizar una experiencia óptima y una interacción positiva (Hassan Montero 2002). Las subcaracterísticas que vamos a utilizar son:

2.8.1.1 *Operabilidad*

Según ISO/IEC(2023), es una subcaracterística de la usabilidad que se enfoca en la facilidad con la que los usuarios pueden operar el software y realizar tareas. En otras palabras, la operabilidad se centra en la interacción entre el usuario y la aplicación, y en la eficacia y eficiencia con la que se puede operar el software.

La operabilidad es esencial para el éxito de cualquier aplicación, ya que una mala operabilidad puede provocar una experiencia de usuario negativa y disminuir la utilidad del software. Por lo tanto, es fundamental evaluar la operabilidad de una aplicación para asegurarse de que cumpla con los estándares de calidad y que los usuarios puedan operarla de manera fácil y efectiva (Mora 2017).

La operabilidad se relaciona con la facilidad de uso y aprendizaje de la aplicación. La facilidad de uso se refiere a la capacidad del software para permitir que los usuarios realicen tareas de manera eficiente y efectiva, mientras que la facilidad de aprendizaje se refiere a la capacidad del software para permitir que los usuarios aprendan a usarlo de manera rápida y efectiva. Existen varias técnicas y herramientas para evaluar la operabilidad, como las pruebas de usuario, la evaluación heurística y el análisis de la interfaz de usuario (Mora 2017).

2.8.1.2 *Reconocibilidad de la adecuación*

Según ISO/IEC(2023), esta característica se refiere a la capacidad del software para ser comprendido, aprendido y utilizado de manera efectiva por los usuarios cuando se realiza una tarea específica. La reconocibilidad de la adecuación implica que el software debe presentar elementos de interfaz de usuario y señales claras que permitan a los usuarios identificar y comprender cómo utilizarlo de manera apropiada en diferentes situaciones. Esto incluye el uso de convenciones reconocibles, terminología familiar, estructuras consistentes y elementos de interfaz intuitivos.

2.8.1.3 *Aprendizabilidad*

Según ISO/IEC(2023), la aprendizabilidad implica que el software debe ser fácil de aprender, permitiendo a los usuarios adquirir rápidamente el conocimiento y las habilidades necesarias para utilizarlo. Además, el software debe ser comprensible, de modo que los usuarios puedan entender fácilmente su funcionamiento y las acciones disponibles.

2.8.1.4 *Estética de la interfaz de usuario*

Según ISO/IEC(2023), la apariencia estética de la interfaz de usuario se refiere al aspecto visual y al diseño general de la interfaz de usuario de un software, aplicación o sitio web. Su enfoque principal es crear una interfaz atractiva y agradable a la vista para los usuarios. La apariencia estética engloba diversos elementos del diseño visual de la interfaz, como la selección de colores, la tipografía, el diseño de iconos y gráficos, la disposición de elementos y la coherencia visual en todo el software. Estos elementos se combinan para formar una interfaz visualmente agradable que captura la atención del usuario de manera positiva.

2.8.1.5 *Método de evaluación*

- **Pruebas de usuario:** Esta técnica consiste en pedir a un grupo de usuarios que realicen tareas específicas en la aplicación, mientras se observa su interacción. Esta técnica permite identificar problemas de operabilidad, como tareas complejas o confusiones en la interfaz de usuario (Perurena Cancio y Moráguez Bergues 2013).
- **Análisis de la interfaz de usuario:** Esta técnica se enfoca en analizar la interfaz de usuario de la aplicación, y evalúa cómo los usuarios interactúan con ella. Esto permite identificar

problemas de operabilidad relacionados con la interfaz, como la falta de consistencia o la complejidad de la navegación (Catalán Vega 2006).

- Cuestionarios SUMI es un cuestionario utilizado para la evaluación de la calidad de un conjunto software desde el punto de vista del usuario final. Este cuestionario puede ser utilizado para evaluar nuevos productos, efectuar comparaciones con versiones previas y establecer objetivos para desarrollos futuros (Laterio Martinez 2016).

2.8.2 Escala de Likert

Según Muguira (2016) la escala de Likert es un método ampliamente utilizado para medir actitudes, opiniones o percepciones de las personas con respecto a una serie de afirmaciones o preguntas. Consiste en una serie de afirmaciones o enunciados a los que los encuestados deben responder indicando su grado de acuerdo o desacuerdo. Por lo general, la escala de Likert consta de 5 puntos, donde cada punto representa un nivel de acuerdo o desacuerdo específico. Los puntos pueden estar etiquetados, por ejemplo, desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo" como se ve en la siguiente tabla **1-2**.

Tabla 1-2 Escala de Likert

Característica	Puntuación
Totalmente de acuerdo	5
De acuerdo	4
Indeciso	3
Desacuerdo	2
Totalmente en desacuerdo	1

Realizado por: Mathew Avilés, 2023

2.9 Evaluación del producto software

2.9.1 ISO/IEC 25010

Según ISO/IEC (2023) la usabilidad es la capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, también La usabilidad es la cualidad de una página web o programa informático de ser de fácil acceso, legible, descarga rápida de información, contar con funciones y menús entendibles, por los usuarios pueden realizar tareas específicas fáciles y en menos tiempo. Esta característica se subdivide a su vez en las siguientes sub características:

- Operabilidad. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Reconocibilidad de la adecuación. Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Aprendizabilidad. Capacidad del producto que permite al usuario aprender su aplicación.
- Estética de la interfaz de usuario. Capacidad de la interfaz de usuario para agrandar y satisfacer la interacción con el usuario.

2.10 Trabajos relacionados

En los trabajos previos internacionales tenemos el de Peña Anticona (2020) con el tema Sistema web basado en la gestión de mascotas y su geolocalización en caso de extravío en la Municipalidad Distrital de Puente Piedra tuvo como objetivo general determinar la influencia de un sistema web basado en la gestión de mascotas y su geolocalización en caso de extravío en la municipalidad distrital de Puente Piedra. Se usó PHP y Laravel como lenguaje de programación y MySQL Workbench como gestor de base de datos, la metodología de investigación utilizada fue experimental, mientras que la metodología de desarrollo de software fue Scrum, como resultado se obtuvo que el sistema web mejoró la gestión de mascotas en la municipalidad distrital de Puente Piedra, pues posibilitó el incremento en el nivel de captación de clientes, lo que permitió el buen funcionamiento sobre el registro completo de cada vecino captado de la localidad en la plataforma online. Así mismo, se logró un incremento en el índice de efectividad del servicio, pues posibilitó poder encontrar rápida y eficazmente las mascotas de la localidad buscadas por parte de sus mascotas. Se logró una mejoría a nivel municipal ya que se tuvo una notable y

significativa mejoría con respecto a los procedimientos de geolocalización en el momento de llevar a cabo las tareas designadas por el administrador del área.

En los trabajos previos nacionales realizados está el de Barcenos Llanos y Cedeño Avelino (2019) en su proyecto de titulación con el tema Desarrollo de un servicio web y aplicación móvil en Android que permita ubicar y rescatar animales domésticos extraviados o abandonados para la fundación rescate animal ecuador. Tuvo como objetivo Desarrollar un Servicio Web y aplicación móvil en Android que les permita ubicar y rescatar animales domésticos extraviados o abandonados para la Fundación Rescate Animal Ecuador vía online a través de sus teléfonos inteligentes en el momento que los usuarios requieran. Se usó (Java , Android Studio, Laravel) como lenguaje de programación y (MySQL) como gestor de base de datos, la metodología de investigación utilizada fue la observación participativa, Investigación descriptiva , mientras que la metodología de desarrollo de software fue (Scrum), como resultado se obtuvo que la Fundación cuenta ahora, y está a disposición de la ciudadanía, con la Aplicación Móvil Cola que facilitará al dueño de mascota el poder reportar y ubicar en tiempo real mediante un mapa Geolocalizado de las mascotas que se encuentren abandonadas o extraviadas en la ciudad de Guayaquil. El desarrollo del Servicio Web que se realizó en el presente proyecto se ajusta a las necesidades de la Fundación de manera que agiliza los procesos que se mantenían de forma manual con el fin de automatizarlos, sean más eficiente y se puedan obtener los resultados esperados. A través del nuevo Sistema Web, puede contar y generar informes actuales o históricos en formato PDF sobre las mascotas que son reportadas por medio de la aplicación las mismas que son obtenidas mediante filtros como: especie, raza, sexo, estado y fecha.

Orozco Villacrés (2022) en su trabajo de titulación Implementación de aplicación web para el registro y control de perros extraviados utilizando código QR en el centro veterinario Animedica tuvo como objetivo Implementar la aplicación web para el registro de perros extraviados mediante la utilización de tecnología basada en códigos (QR), de tal manera que contribuyan al desarrollo de la veterinaria Animedica. Se usó PHP y Laravel como lenguaje de programación y (MySQL) como gestor de base de datos, la metodología de investigación utilizada fue de tipo descriptiva con un diseño de investigación no experimental, mientras que la metodología de desarrollo de software fue Cascada, como resultado se obtuvo que se llegó a la conclusión de que el sistema se ajusta al entorno del centro veterinario y cumple con cada una de las necesidades identificadas en un principio. Se establecieron de manera ideal los requerimientos del software, esto se logró a través de la utilización de técnicas de recolección de datos, como son la entrevista personalizada y encuestas, lo que permitió seguir un modelo estructurado al momento del diseño y construcción del software. En lo que respecta al desarrollo, las herramientas utilizadas permitieron la

construcción de un sistema eficiente, accesible a los usuarios y con altas prestaciones; la utilización de la interfaz de línea de comandos Artisan que integra Laravel permitió crear recursos y administrar el sistema de manera rápida y eficiente.

CAPÍTULO III:

3 MARCO METODOLÓGICO

3.1 Introducción

Este capítulo contiene el tipo de estudio, métodos y técnicas, población y muestra, hipótesis y desarrollo de la aplicación web utilizando la metodología ágil Scrumban.

3.2 Tipo de estudio

Para este trabajo se empleó el enfoque de estudio aplicativo, ya que se utilizaron todos los conocimientos aprendidos durante la universidad en resolver problemas, específicamente en el desarrollo de una aplicación web progresiva para la identificación y búsqueda de mascotas.

3.2.1 Métodos y técnicas

De acuerdo con los objetivos planteados en este trabajo de integración curricular en la tabla 2-3 se detallan los métodos y las técnicas que permitan cumplir con los objetivos planteados

Tabla 2-3 Métodos y técnicas

Objetivos	Métodos	Descripción	Técnicas	Fuentes
Determinar los requisitos funcionales y no funcionales para el desarrollo de la aplicación web.	Analítico.	Mediante el análisis podemos determinar cuáles son los requisitos funcionales y no funcionales de la aplicación a desarrollar.	Revisión de documentación. Observación. Fichas de requerimientos.	Usuario. Tesis. Papers.
Desarrollar los módulos de login, mascotas, nosotros planteados para el aplicativo web.	SCRUMBAN.	Es una metodología ágil que nos permite integrar lo mejor de Scrum y Kanban para desarrollar la aplicación definiendo lo esprints y las tareas a realizar en cada uno mediante el uso	Lluvia de ideas. Tableros Kanban.	Papers. Tesis.

		de los tableros Kanban.		
Evaluar la usabilidad de la aplicación web para la identificación y búsqueda de mascotas extraviadas con la norma ISO 25010.	Descriptivo	Permite la recolección de datos mediante el uso de cuestionarios los mismos que permiten evaluar el nivel de usabilidad de la aplicación web.	Cuestionario SUMI adaptado. Observación.	Documentación ISO/IEC 25010. Papers. Cuestionario SUMI.

Realizado por: Mathew Avilés, 2023

La metodología SCRUMBAN se escogió por la flexibilidad que tiene para dividir el trabajo en entregables pequeños y concretos, aparte fracciona el tiempo y optimiza el plan de entregas y esto combinado con los tableros Kanban los cuales nos permiten visualizar el flujo de trabajo de una manera muy simple y ver el progreso de cada uno de las tareas y su estado, por todos estos aportes es que he decidido implementar esta metodología de desarrollo de software.

Para determinar los requisitos funcionales y no funcionales de la aplicación se ha decidido realizar un análisis tanto de documentación, como de requerimientos para lograr cumplir este objetivo y una vez analizados diseñar tantos requisitos funcionales como no funcionales.

Para la evaluación de la usabilidad utilizaremos el cuestionario SUMI seleccionando las preguntas más adecuadas, adaptándolas a nuestro idioma y a cada subcaracterística a evaluar, los mismos que nos ayudaron a determinar el nivel de usabilidad del aplicativo web.

3.3 Metodología para la determinación de la usabilidad

En esta sección se establece los indicadores para mediar la usabilidad de la aplicación web mediante la aplicación de cuestionario SUMI adaptado para evaluar la usabilidad de la aplicación. Se han seleccionado cuatro indicadores de usabilidad y se dejó fuera la protección contra errores de usuarios y la accesibilidad, el primer indicador se lo quitó porque es un sistema el cual no tiene muchos pasos para realizar una acción y por consiguiente es difícil que el usuario se equivoque y el segundo indicador no se lo tomó en cuenta por que el sistema no está especializado para usuarios con determinadas características especiales ni discapacidades.

Tabla 2-3 Indicadores para medir la usabilidad

Variable	Indicador	Descripción
Usabilidad	Operabilidad.	Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad ISO/IEC(2023).
	Reconocibilidad de la adecuación.	Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades ISO/IEC(2023).
	Aprendizabilidad	Capacidad del producto que permite al usuario aprender su aplicación ISO/IEC(2023).
	Estética de le interfaz de usuario.	Capacidad de la interfaz de usuario para agradar y satisfacer la interacción con el usuario ISO/IEC(2023).

Realizado por: Mathew Avilés, 2023

Tabla representativa de las métricas de evaluación de la usabilidad

Tabla 3-3 Métricas de evaluación de la usabilidad

Subcaracterísticas	Porcentaje de valoración
Operabilidad	36%
Reconocibilidad de la adecuación	25%
Aprendizabilidad	19%
Estética de la interfaz de usuario	20%
Total	100%

Realizado por: Mathew Avilés, 2023

3.3.1 Operacionalización de variables

3.3.1.1 Operacionalización conceptual de variables

Tabla 4-3: Operacionalización conceptual de variables

Formulación del problema	Variable	Tipo	Concepto
¿Cómo desarrollar una aplicación web para la identificación y búsqueda de mascotas extraviadas, la misma que deberá tener un adecuado nivel de usabilidad?	Usabilidad	Cualitativa	Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones (ISO 2023).
		Compleja	
		Dependiente	

Realizado por: Mathew Avilés, 2023

3.3.1.2 Operacionalización metodológica de variables

Tabla 5-3: Operacionalización metodológica de variables

Formulación del problema	Variable	Categoría	Indicador	Técnica	Fuente
¿Cómo desarrollar una aplicación web para la identificación y búsqueda de mascotas extraviadas, la misma que deberá tener	Usabilidad	Operabilidad	Simplicidad en la navegación	Encuesta	Usuarios finales
			Intuitividad de la interfaz.		Cuestionario SUMI
		Reconocibilidad de la adecuación		Encuestas	Usuarios finales

un adecuado nivel de usabilidad?			Satisfacer las necesidades del usuario.		Cuestionario SUMI
		Aprendizabilidad	Tiempo para aprender a utilizar el sistema.	Encuestas	Usuarios finales
			Realizar tareas básicas después de un breve período de aprendizaje		Cuestionario SUMI
		Estética de la interfaz de usuario	Colores atractivos.	Encuesta	Aplicación web
					Usuarios finales
			Diseño agradable	Encuesta	Cuestionario SUMI

Realizado por: Mathew Avilés, 2023

3.4 Población y muestra

3.4.1 Población y muestra

Para determinar la muestra vamos a utilizar una técnica llamada muestreo aleatorio simple. Esta técnica garantiza que todos los individuos que componen la población tienen la misma oportunidad de ser incluidos en la muestra. Esto significa que la probabilidad de selección de un sujeto a estudio "x" es independiente de la probabilidad que tienen el resto de los sujetos que integran forman parte de la población blanco (Otzen y Manterola 2017).

Como criterio de estratificación se ha tomado en cuenta la edad de los usuarios definiendo rangos de edades de entre 20 y 30, 31 y 40, 41 y 50, 51 y en adelante y en cada rango seleccionando aleatoriamente a 5 personas y en total se han seleccionado a 20 personas como una muestra significativa.

Análisis previo al desarrollo del proyecto

3.4.2 Análisis económico

En esta tabla se muestra el presupuesto estimado necesario para la realización del trabajo de titulación.

Tabla 6-3 Presupuesto estimado

Descripción	Valor Unitario	Valor Total (6 meses)
Computadora Asus Vivobook X570UD	\$1000	\$ 1000
Alojamiento en la nube	\$5	\$30
Servicios básicos	\$30	\$ 180
Servicio de internet	\$30	\$ 180
Materiales de oficina	\$ 3	\$ 18
Horas de ingeniería (otros, horas de trabajo)	\$ 90	\$ 540
Transporte	\$ 24	\$ 144
Total		\$ 2092

Realizado por: Mathew Avilés, 2023

3.4.3 Fuente de financiamiento

El 100% del financiamiento para el trabajo de integración curricular será cubierto por el proponente

3.4.4 Recursos Hardware

Tabla 7-3 Recursos Hardware

Hardware	Características	Utilidad
Equipo de computo	<ul style="list-style-type: none">• Asus Vivobook 15• Intel Core i7-8550U CPU• 1.8GHz• 8 GB RAM• 512 GB disco duro	<ul style="list-style-type: none">• Realizar la aplicación web• Realizar el trabajo de integración curricular• Revisión de literatura

Realizado por: Mathew Avilés, 2023

3.4.5 Recursos Software

Tabla 8-3 Recursos Software

Software	Características	Utilidad
Sistema operativo	Windows 11 Home	Gestionar todos los recursos del computador
Ide de desarrollo	Visual Studio Code	Editor de código en varios lenguajes, necesario para la programación en Python
Gestor de base de datos	SQLite	Crear, Gestionar y almacenar datos a través de varios motores de almacenamiento
Mockups	Balsamiq	Realizar mockups de los módulos a realizar
Tecnología de programación	HTML5	Sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, juegos, entre otros
Lenguaje de programación para páginas web	Python	Utilizado por empresas de todo el mundo para construir aplicaciones web, analizar datos, automatizar operaciones y crear aplicaciones empresariales fiables y escalables

Realizado por: Mathew Avilés, 2023

3.4.6 Riesgos

Tabla 9-3 Riesgos

ID Riesgo	Descripción	Porcentaje de probabilidad	Porcentaje de impacto	Técnicas de mitigación	Plan de contingencia	Consecuencias
R1	Requisitos incompletos o ambiguos.	50%	85%	1.El estudiante debe tener en cuenta todo lo que va a realizar en el trabajo de titulación.	Determinar qué es lo que falta o falla para integrarlo como un nuevo requisito y cumplir con lo que se desea	Se entrega un proyecto que no cumple con los objetivos requisitos planteados en el documento
				2. Agregar el/los nuevos requisitos necesarios para complementar correctamente el trabajo de titulación.		
R2	Falta de información al momento de investigar	40%	50%	1.Realizar una investigación preliminar del tema.	Hacer una búsqueda diversificada en varias fuentes como: libros, revistas científicas, tesis, papers, etc. Y en diferentes idiomas para obtener mucha mas información y apoyarse en la experiencia del tutor y asesor a cargo	El trabajo es muy pobre en información y a veces la información es errónea, lo que genera pérdida de tiempo y mucho más esfuerzo
				2.Hacer revisión de literatura.		
				3. Consultar al director y asesor del trabajo de titulación.		
R3	Diseño deficiente de la estructura de la base de datos.	40%	90%	1. Realizar un análisis exhaustivo de los requisitos del sistema.	Realizar modificaciones iterativas para resolver los problemas, realizar copias de seguridad de las bases de datos al momento de realizar los cambios,	Un rendimiento ineficiente, problemas de integridad de datos, dificultad para realizar cambios o actualizaciones
				2. Utilizar técnicas de modelado de datos.		

				3, Aplicar reglas de normalización.		
R4	Averías en el equipo de cómputo.	30%	100%	1.Realizar un mantenimiento del equipo.	Llevar inmediatamente el equipo de cómputo a realizar un mantenimiento u arreglo del problema	Retraso en la realización del trabajo de titulación y entrega de avances de este.
				2.Realizar copias de seguridad periódicamente del avance en la nube.		
R5	Cambio de tecnologías de desarrollo.	30%	80%	1. Realizar un análisis de cual lenguaje, IDE, framework utilizar.	Revisar la documentación, examinar cursos o tutoriales relacionados con las tecnologías de desarrollo	Retraso en la realización del trabajo de titulación y entrega de avances de este.
				2. Tener conocimientos previos en las tecnologías implementadas.		
				3. No entregar en el tiempo especificado.		
R6	Falta de tiempo en el desarrollo del trabajo de integración curricular	70%	100%	1.Realizar un cronograma de las actividades y fecha de realización de estas.	Priorizar las tareas más críticas y priorizarlas en función de su importancia, reducir los requisitos y quedarse solo con los realmente importantes y necesarios	No entrega de los avances del trabajo de titulación y finalmente no concluir a la final y perder la materia.
				2. Organizar el tiempo del desarrollador al momento de realizar el trabajo de integración curricular.		
				3. Enfocarse y tener motivación en la actividad a realizar.		

Realizado por: Mathew Avilés, 2023

3.5 Desarrollo del sistema web utilizando la metodología de desarrollo ágil SCRUMBAN

3.5.1 Meta

Desarrollar una aplicación web para la identificación y búsqueda de mascotas extraviadas de acuerdo con las métricas de usabilidad de la ISO 25010.

3.5.2 Cola de historias

Tabla 10-3 Tareas por hacer

Nro.	Tareas por hacer
1	Elicitación de requerimientos funcionales y no funcionales.
2	Definir arquitectura del sistema
3	Definir el lenguaje de programación
4	Definir el entorno de desarrollo
5	Definir el sistema de gestión de base de datos
6	Instalar todas las librerías necesarias en el IDE
7	Diseñar el esquema de la base de datos
8	Establecer la conexión con la base de datos
9	Desarrollar mockups
10	Diseñar las interfaces de la aplicación
11	Desarrollar los módulos (login, mascotas, nosotros)
12	Realizar las pruebas de la aplicación
13	Evaluar el nivel de usabilidad de la aplicación

Realizado por: Mathew Avilés, 2023

3.5.3 Análisis

Se empleó la plataforma en línea llamada 'Meister Task' porque es una herramienta web fácil de utilizar para gestionar tareas y proyectos, se utiliza para organizar y administrar tareas tanto individuales como proyectos en equipos, ofrece la función de creación de tableros Kanban lo que nos ayuda especialmente en el proyecto ya que estamos utilizando la metodología Scrumban. Brindando la posibilidad de organizar, clasificar y gestionar tareas según su estado. Dichos estados se dividen en Backlog (tareas pendientes por realizar), Doing (tareas en proceso), Review (tareas finalizadas pero que requieren una revisión final) y Done (tareas completamente finalizadas).

3.5.3.1 Equipo de desarrollo

Definimos cual es el equipo de desarrollo, como es un trabajo de integración curricular donde solo hay un estudiante, este deberá realizar todas las tareas y desempeñar todos los roles.

Miembros y Grupos

Invitar Importar Descargar CSV

Buscar

Nombre ^ Correo electrónico

Todos los usuarios (1)

Mathew. A. :3 (Administración)juanaviles12345@gmail.com

Grupos

Trabajo de integrac... (1)

+ Añadir Grupo

Figura 2-3 Miembros y equipos de desarrollo

Realizado por: Mathew Avilés, 2023

3.5.3.2 Flujo de trabajo

El flujo de trabajo se organiza utilizando un tablero Kanban, que es una herramienta visual que muestra el progreso de las tareas a medida que avanzan a través de diferentes etapas. Las etapas típicas en el flujo de trabajo de Scrum incluyen:

1. **Backlog:** Esta es la etapa inicial donde se encuentran las tareas pendientes de realizar. Aquí se registran todas las tareas que deben completarse, y se priorizan y estiman según su importancia y esfuerzo requerido.
2. **Doing (En proceso):** Cuando un miembro del equipo comienza a trabajar en una tarea, se mueve a esta etapa. Aquí se realiza el trabajo real, y el equipo se enfoca en completar la tarea de acuerdo con las estimaciones y objetivos establecidos.
3. **Review (Revisión):** Después de que una tarea se completa, se mueve a esta etapa para su revisión. Aquí se realiza una evaluación y una revisión final para asegurarse de que la tarea cumpla con los requisitos y está lista para su entrega o próxima etapa.
4. **Done (Completado):** Una vez que una tarea ha pasado la revisión final y se considera finalizada, se mueve a esta etapa. Aquí se registra oficialmente como completada y se archiva.

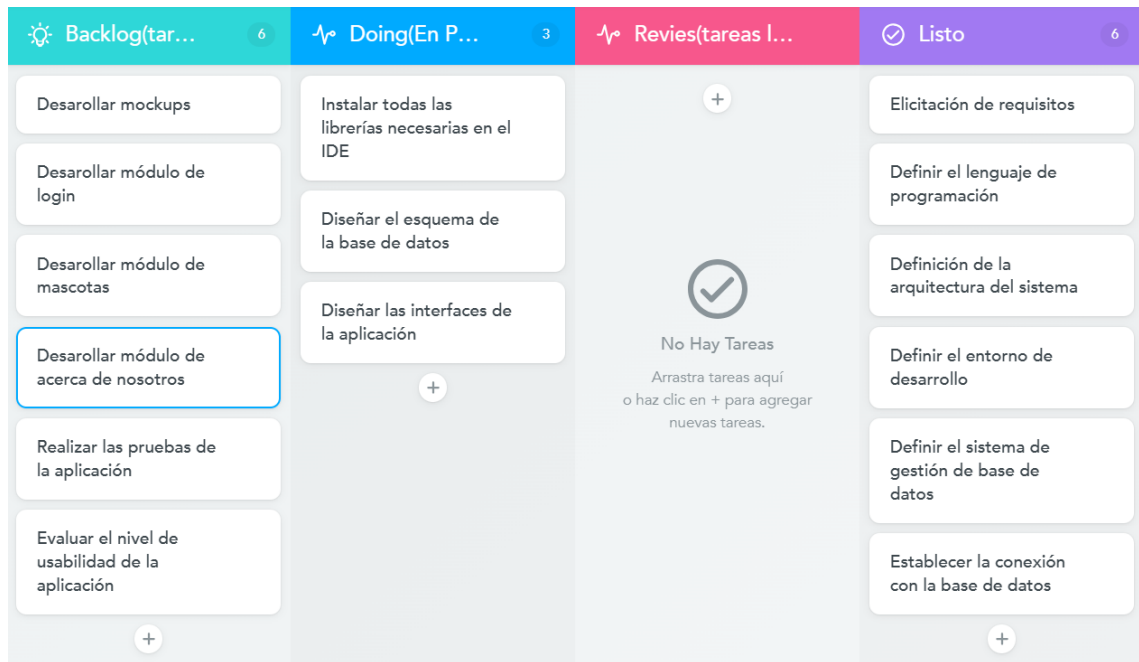


Figura 3-3 Tablero De tareas

Realizado por: Mathew Avilés, 2023

3.5.4 Desarrollo

3.5.4.1 Elicitación de requerimientos

En la metodología de SCRUMBAN, la elicitación de requisitos se hace mediante historias de usuario, las mismas que son una breve descripción de la funcionalidad desde el punto de vista del usuario, el título debe ser corto y conciso, las historias de usuario pueden ser más complejas o menos, dependiendo de la granularidad con que se quiera tratar (Azabache Medina y Caruajulca Sánchez 2019).

Tabla 11-3 Historias de usuario

Historias de Usuario				
Nro. de historia	Rol	Característica/ Funcionalidad	Razón/ Resultado	Criterio de aceptación
H01	Como usuario	Quiero poder registrarme e iniciar sesión.	Para ingresar al sistema.	Registro en el sistema.
H02	Como usuario	Quiero poder registrar a mi mascota según su estado (Perdido, en adopción).	Para poder visualizarlo en la aplicación y que los demás usuarios la vean.	Página de mascotas/ registro mascota.
H03	Como usuario	Quiero poder editar los datos de la mascota.	Para poder actualizar los datos de la mascota.	Página de mascotas/mi mascota/editar.

H04	Como usuario	Quiero poder eliminar una mascota.	Para poder quitar a la mascota de la lista de mascotas que se encuentran registradas en la aplicación.	Página de mascotas/mi mascota/eliminar.
H05	Como usuario	Quiero poder visualizar las mascotas según indicadores como (sexo, ubicación, estado).	Para visualizar mas rápidamente en base a las preferencias del usuario.	Página de mascotas.
H06	Como usuario	Quiero poder subir imágenes de mi mascota extraviada.	Para determinar si la mascota está desaparecida y le pertenece a alguien o solo es un animal en estado de calle.	Página de mascotas/ registrar mascota.
H07	Como usuario	Quiero poder buscar una mascota mediante el ingreso de una imagen.	Para determinar si la mascota se encuentra en la base de datos de la aplicación y determinar su estado (perdida, en adopción)	Página de mascotas/ reconocimiento imágenes.
H08		Quiero poder buscar una mascota mediante el reconocimiento por cámara.	Para determinar si la mascota se encuentra en la base de datos de la aplicación y determinar su estado (perdida, en adopción)	Página de mascotas/ reconocimiento con cámara.
H09	Como usuario	Quiero poder cerrar sesión después de utilizar el sistema.	Para volver a ingresar cuando sea necesario.	Botón de salir.
H10	Como administrador	Quiero poder agregar datos de mascotas como (sexo, ubicación, tipo).	Para ingresar al sistema.	Registro en el sistema.
H11	Como administrador	Quiero poder eliminar mascotas.	Para que en los resultados de los informes se utilicen datos recientes.	Eliminar un usuario.
H12	Como administrador	Quiero poder visualizar las imágenes que los usuarios suben para revisión y posterior implementación en el modelo de reconocimiento.	Para analizar las imágenes y poderlas incluir en el modelo para nutrir el sistema de reconocimiento de mascotas.	Agregar mascotas al modelo.
H13	Como administrador	Quiero poder cerrar sesión.	Para volver a ingresar en caso de que sea necesario.	Botón de cerrar sesión.

Realizado por: Mathew Avilés, 2023

Los requisitos funcionales y no funcionales se encuentran descritos en: **Anexo A** y **Anexo B**.

3.5.4.2 Conceptualización del sistema

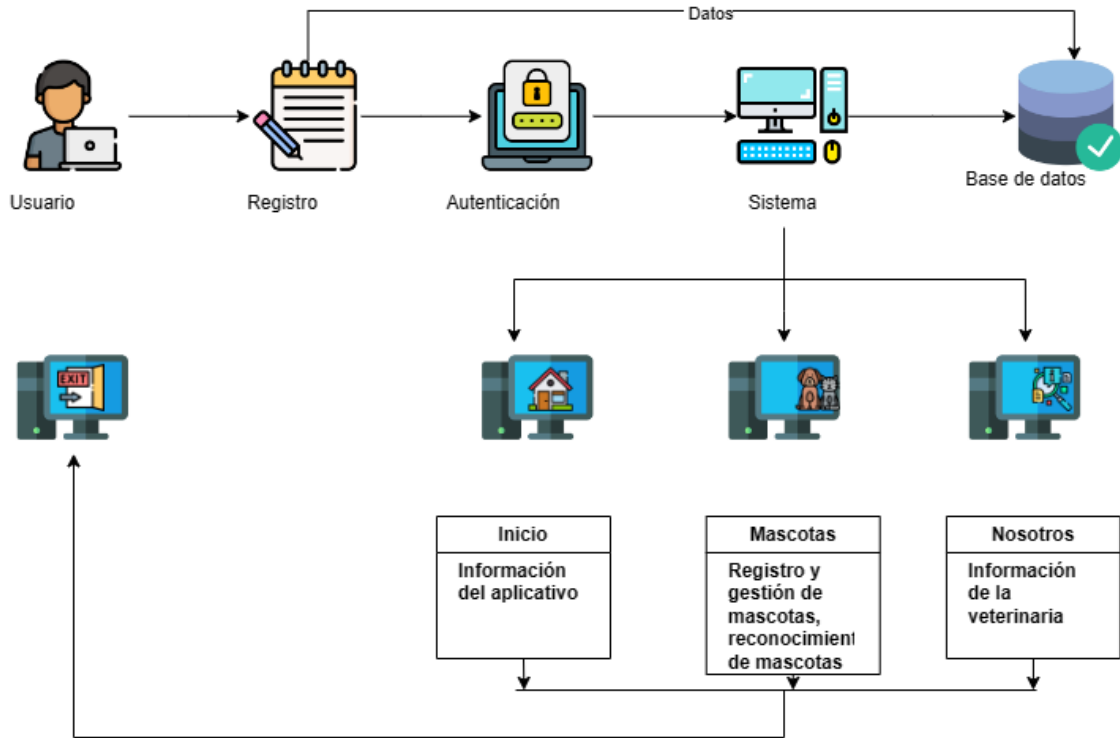


Figura 4-3 Conceptualización del sistema

Realizado por: Mathew Avilés, 2023

3.5.4.3 Arquitectura de la aplicación

MVC (Model View Controller), que es un patrón arquitectónico para implementar interfaces de usuario. Divide una aplicación en tres capas interconectadas que permiten separar el cómo el usuario ve los datos a cómo estos se gestionan e interpretan, y cómo se almacenan. Este patrón incrementa la posibilidad de reutilización del proyecto y la flexibilidad del mismo (Rodríguez Franco 2014).

Según Azabache Medina Y Caruajulca Sánchez (2019) el patrón MVC separa la lógica (y acceso a datos) de una aplicación de su presentación, usando 3 componentes:

1. Modelo: Representa las reglas de negocio de la aplicación (y el acceso a datos subyacente).
2. Vistas: Representan la presentación de la aplicación.

3. Controlador: Actúan de intermediario entre el usuario y el Modelo y las Vistas.

3.5.4.4 Diagrama físico de la base de datos

Aquí se muestra el diagrama físico de la base de datos de la aplicación huellitas perdidas

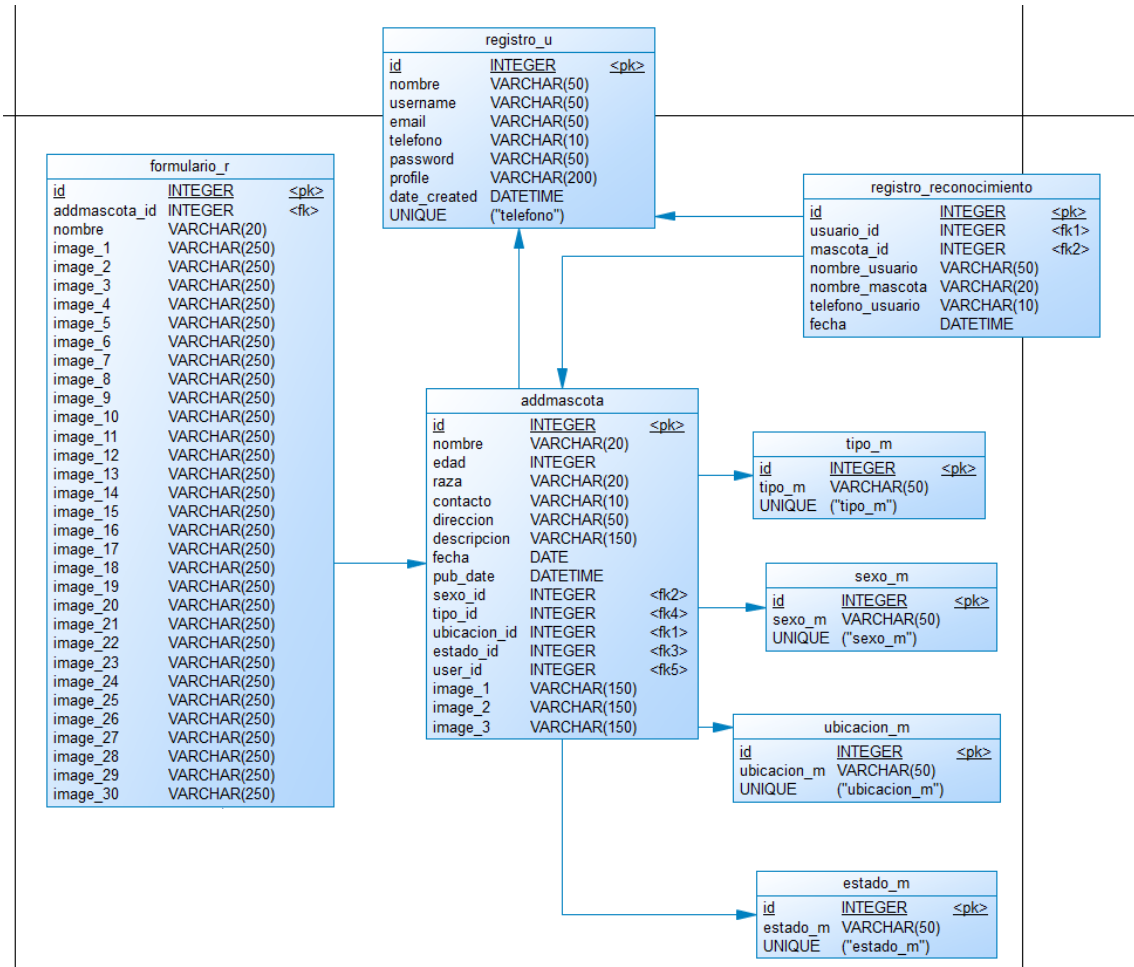


Figura 5-3 Diagrama físico de la base de datos

Realizado por: Mathew Avilés, 2023

3.5.4.5 Escenario

Para describir el escenario se ha utilizado un diagrama de casos de uso en el mismo que se puede ver el funcionamiento del sistema y la interacción con los módulos.

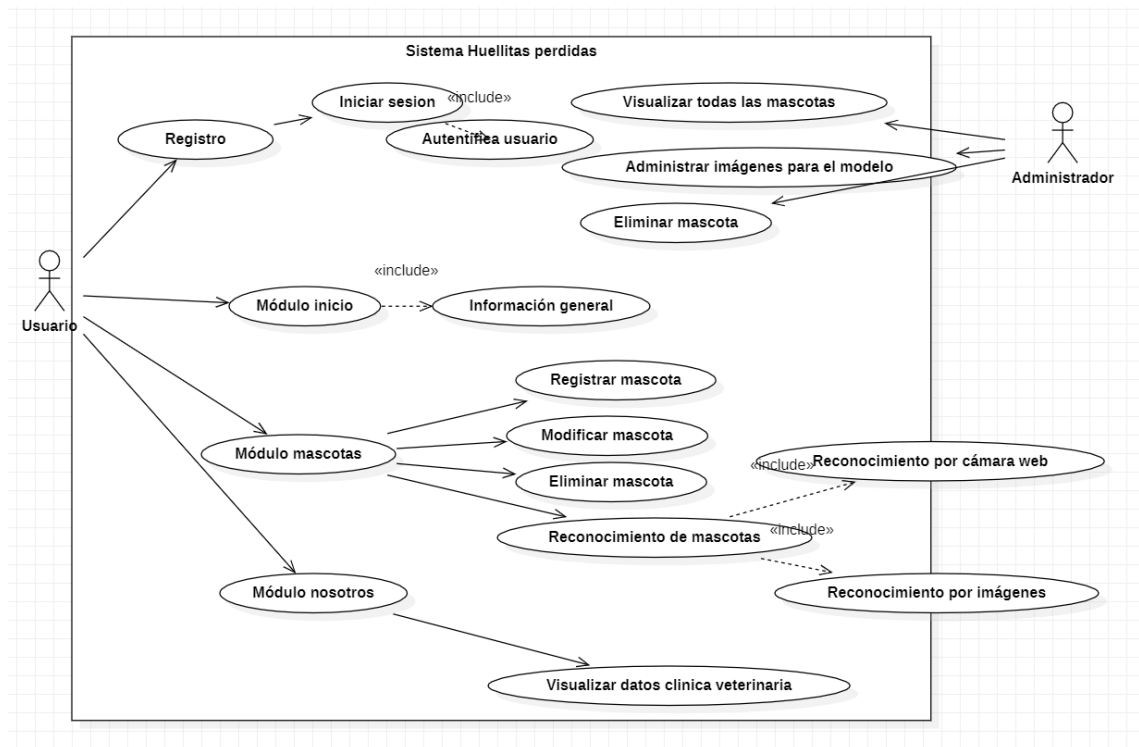


Figura 6-3 Diagrama de casos de uso

Realizado por: Mathew Avilés, 2023

3.5.4.6 Vista de despliegue

Para mostrar la vista de despliegue de la aplicación se usó el diagrama de componentes que se ve en la ilustración

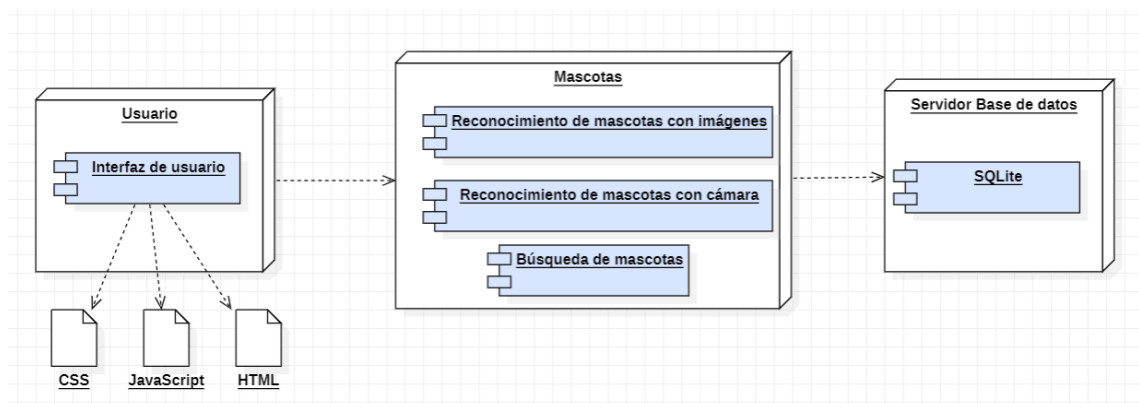


Figura 7-3 Diagrama de componentes

Realizado por: Mathew Avilés, 2023

3.5.4.7 Vista de procesos

Mediante los diagramas de actividades se muestra los procesos que cumple el usuario o el sistema para realizar una actividad específica, a continuación, se muestra el diagrama de actividades para el registro del usuario como se ve en la ilustración 8-3.

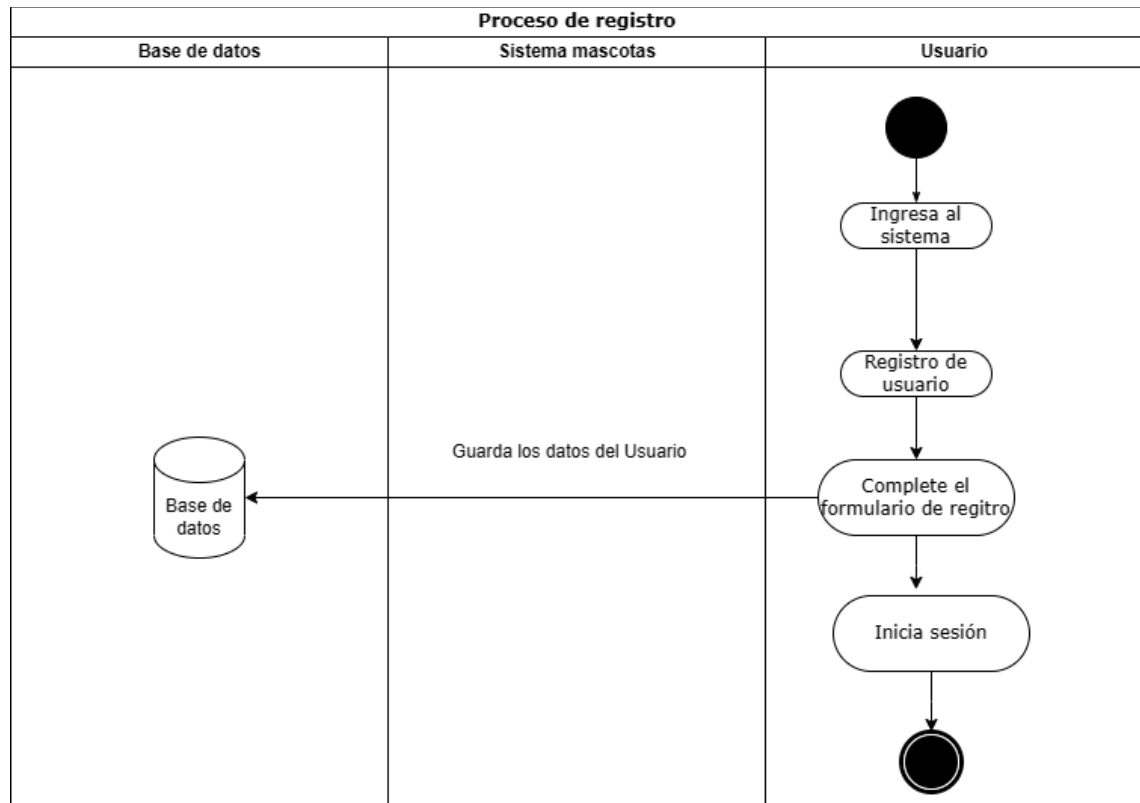


Figura 8-3 Diagrama de actividades para el registro de usuario

Realizado por: Mathew Avilés, 2023

- **Diagrama de procesos para registrar una mascota**

En este diagrama se muestra las actividades para el registro del usuario como se ve en la ilustración 9-3. Los demás diagramas de procesos van a estar en el **Anexo C**.

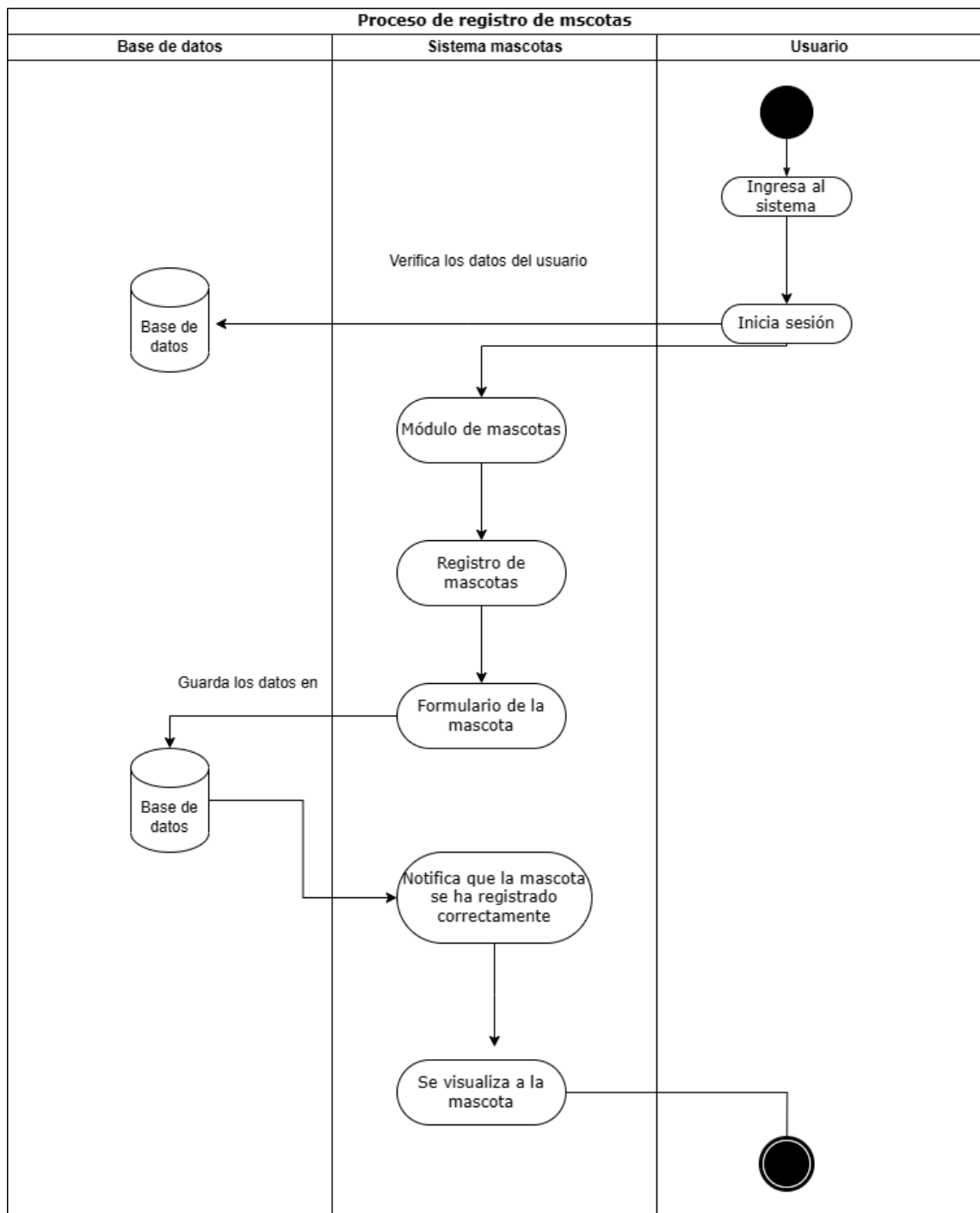


Figura 9-3 Diagrama de actividades para el registro de la mascota

Realizado por: Mathew Avilés, 2023

3.5.4.8 Estándar y convención de la codificación

Con el fin de desarrollar la aplicación para buscar e identificar mascotas perdidas, es necesario establecer un conjunto de reglas de programación para garantizar que el equipo pueda trabajar con un código organizado y fácil de leer. En este caso, se ha decidido adoptar PEP8, que son recomendaciones específicas para el lenguaje de programación Python. Estas recomendaciones abarcan diferentes aspectos, como la forma de nombrar variables, funciones y clases, la forma de indentación del código, el uso de espacios en blanco, la longitud de las líneas y otros detalles de estilo.

Para los objetos de la base de datos, se ha establecido la convención SnakeCase, que es una forma de nombrar variables, funciones y otros elementos en programación. Esta convención utiliza una combinación de palabras en minúsculas separadas por guiones bajos “_”.

3.5.4.9 Diccionario de datos

En esta tabla se encuentra el diccionario de la tabla usuario registro

Tabla 12-3 Diccionario de datos de la tabla registro_u

Nombre del archivo: registro_u				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0 *Es auto incremental 1,1
nombre	Nombre y apellido del usuario.	Varchar(100)	No	primer nombre + (primer apellido)) = { [A-Z a-z] }

username	Nombre de usuario o username.	Varchar(100)	No	Nombre de usuario = { [AZ a-z] } + [0 a 9]
correo	Correo del usuario.	Varchar(100)	No	correo@example.com
password	Contraseña del usuario.	Varchar(100)	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,}
fecha_creacion	Fecha en que se registra el usuario.	Datetime	No	* formato: aaaa-mm-dd *
is_admin (FK)	Rol de usuario.	Integer	No	[1 2] * significado: 1: Admin 2: Usuario *

Realizado por: Mathew Avilés, 2023

El diccionario completo se lo puede observar específicamente en el **Anexo D**

3.5.4.10 Prototipad de la página principal

En la **Figura 10-3** Se puede observar el prototipo planteado para la página de inicio de la aplicación.

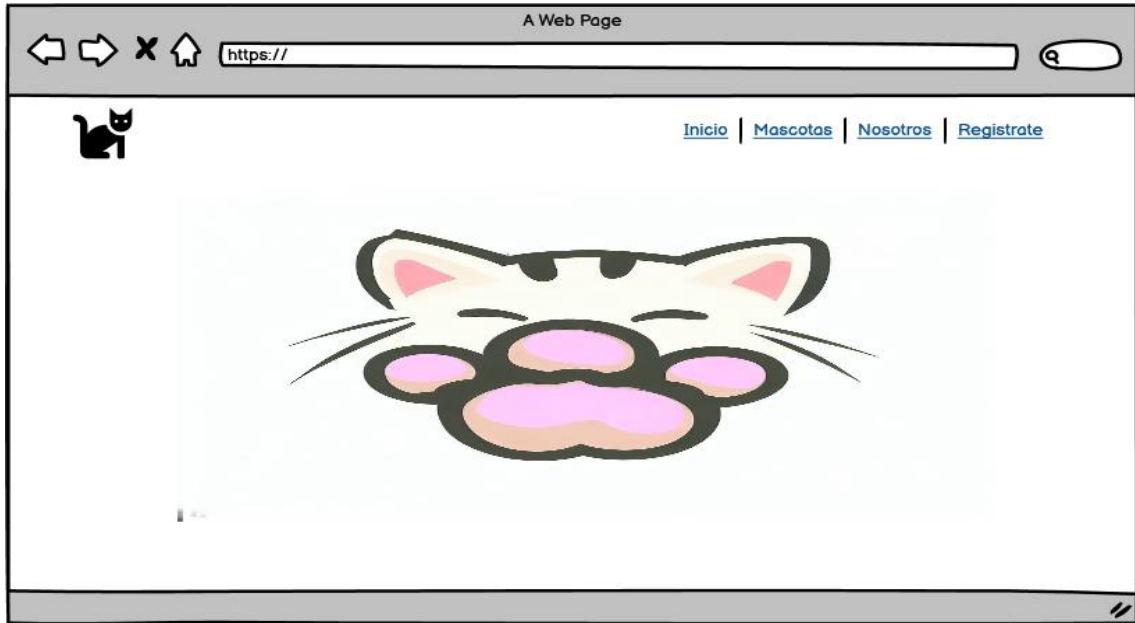


Figura 10-3 Prototipo de la página de inicio

Realizado por: Mathew Avilés, 2023

En la **Figura 11-3** Se puede observar el prototipo planteado para la página de mascotas

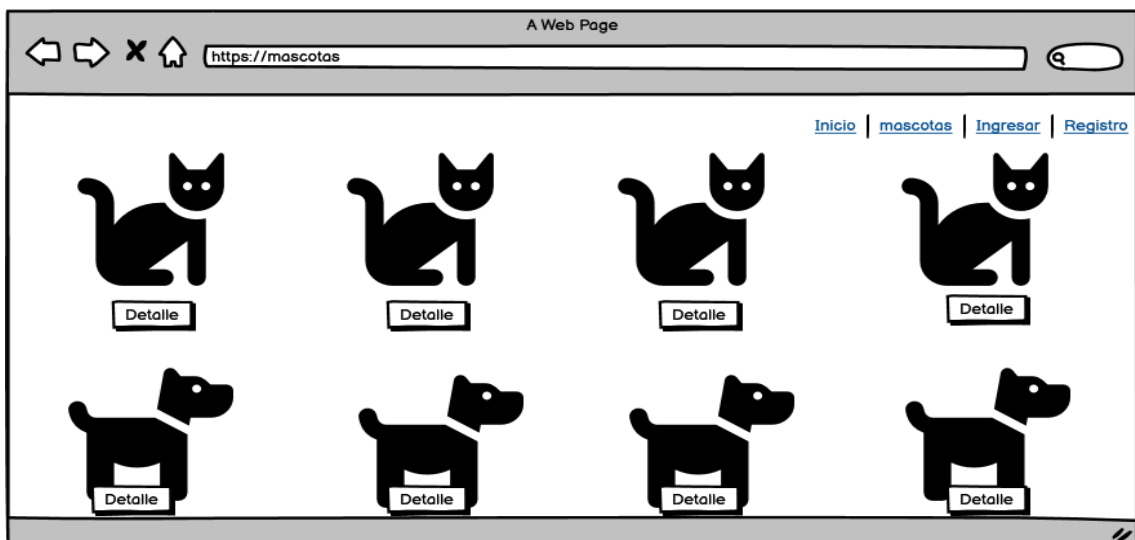


Figura 11 Prototipo de la página de mascotas

Realizado por: Mathew Avilés, 2023

En la **Figura 12-3** Se puede observar el prototipo planteado para la página de mascotas

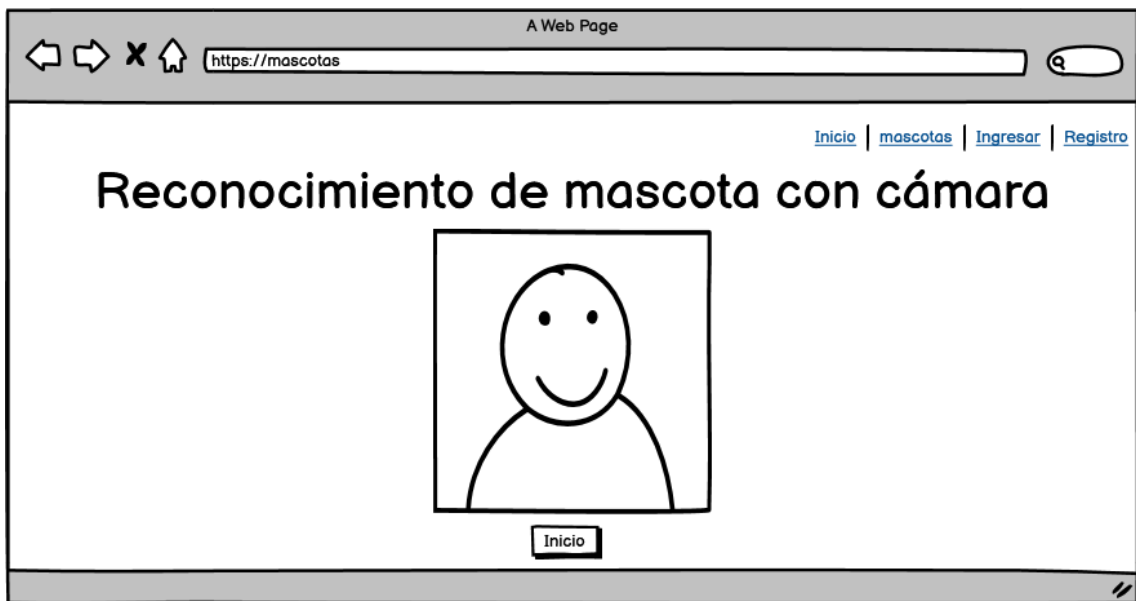


Figura 12 Prototipo de la página de reconocimiento de mascotas mediante cámara

Realizado por: Mathew Avilés, 2023

En la **Figura 13-3** Se puede observar el prototipo planteado para la página de mascotas

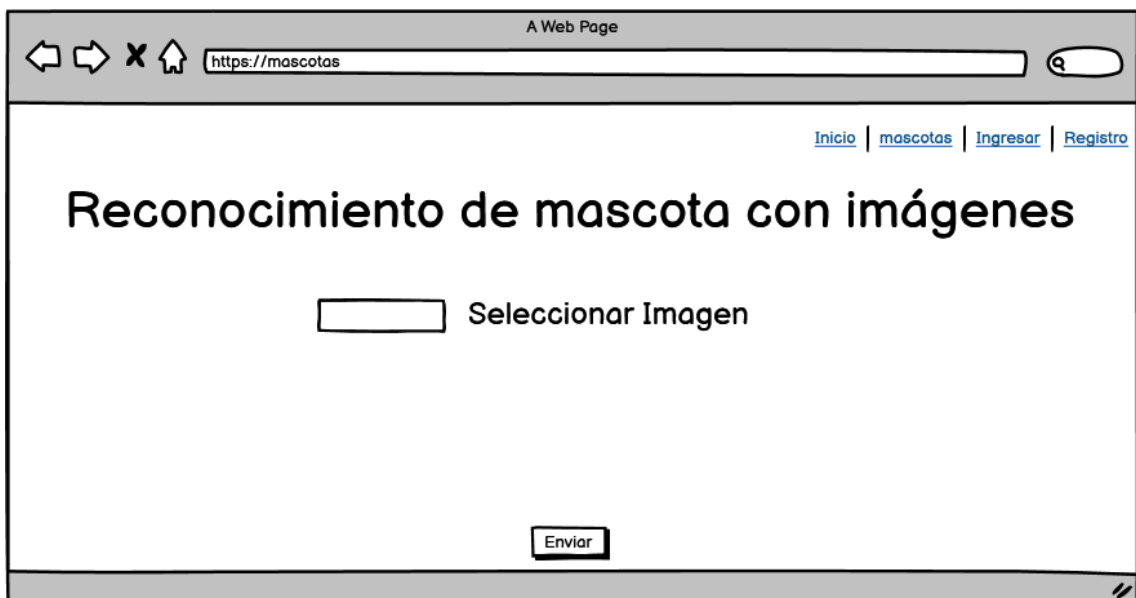


Figura 13 Prototipo de la página de reconocimiento de mascotas mediante imágenes

Realizado por: Mathew Avilés, 2023

3.5.5 Pruebas

En esta etapa, se realizan las pruebas correspondientes al sistema basadas en las historias de usuario. El objetivo es verificar el cumplimiento de todos los requisitos y asegurarse de que la aplicación esté libre de errores antes de su implementación. En caso de que se detecte algún error, se garantiza que no afecte la usabilidad de la aplicación, gracias a la implementación de un sólido manejo de errores en los distintos módulos. Para observar las pruebas completas observar el **Anexo E**.

3.5.6 Implementación

Una vez finalizado con todos los módulos planteados se procede con el despliegue de la aplicación, el cual fue realizado mediante una imagen en un contenedor de Docker como se puede ver en el **Anexo H**.

3.5.7 Terminado

Una vez finalizadas todas las tareas planteadas mediante la elicitación de requisitos y las historias de usuarios, las ubicamos en la fase de listo en el tablero Kanban como se ve en la siguiente ilustración.

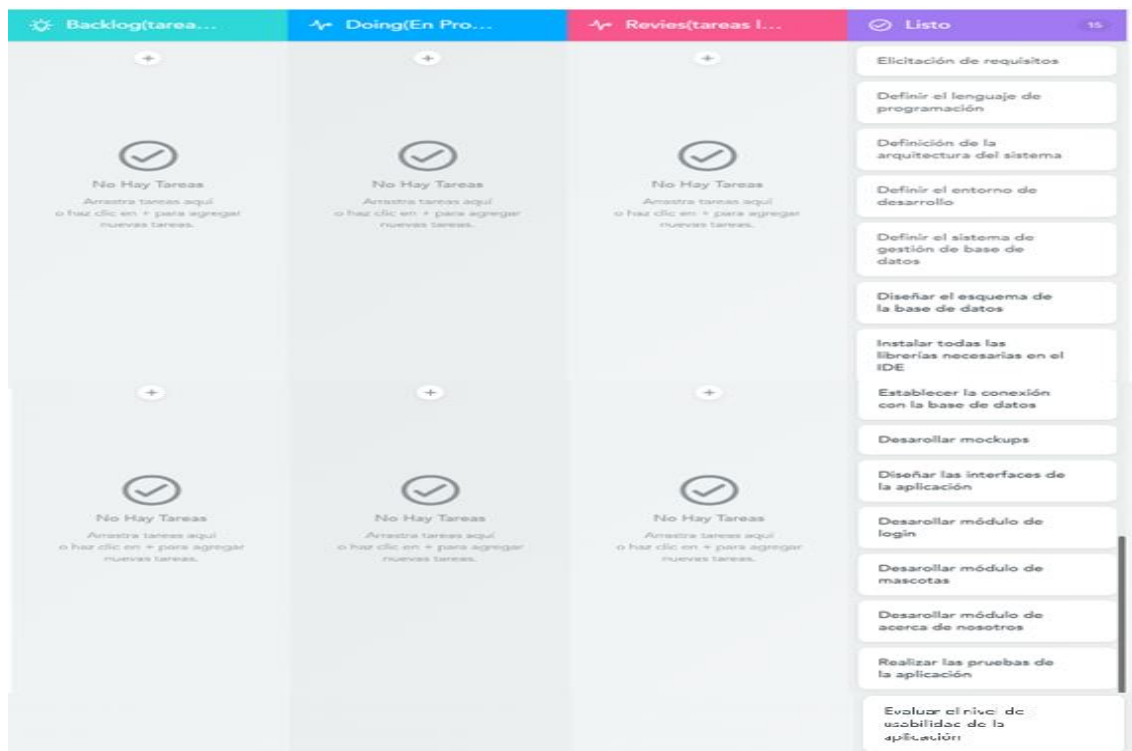


Figura 14-3 Lista de tareas completadas

Realizado por: Mathew Avilés, 2023

CAPÍTULO IV:

4 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Introducción

En este capítulo se realizan las respectivas pruebas de usabilidad del aplicativo web y se exponen los resultados obtenidos de la aplicación.

4.2 Resultados de usabilidad

Para medir la usabilidad de la aplicación, se realizaron modificaciones y adaptaciones en las preguntas del cuestionario SUMI, ver **Anexo F** y se lo aplicó a una muestra de 20 personas como se indica en el capítulo anterior. El resultado de este cuestionario aplicado a los clientes de la clínica veterinaria se lo puede ver en el **Anexo G**.

4.2.1 Operabilidad

Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad (ISO 2023).

En la **tabla 12-3** se puede observar en la primera columna los números de las preguntas, en la segunda columna se observan las preguntas relacionadas con la subcaracterística de operabilidad, en la tercera columna se observa el valor en promedio de las 20 respuestas por partes de los usuarios sobre un valor máximo de 5 puntos, en la cuarta columna se observa el valor en promedio sobre un máximo de 30 puntos.

Tabla 13-4 Resultados del cuestionario de la subcaracterística Operabilidad.

Número	Preguntas	Promedio (_/5)	Resultado (_/30)
Operabilidad			
1	El uso de este software para el reconocimiento de mascotas resultó satisfactorio.	4,60	27,60
2	Este software se ha detenido en algún momento inesperadamente de tal manera que afecte su experiencia como usuario.	4,05	24,30
3	Hay demasiados pasos necesarios para realizar un proceso como registrar una mascota.	4,25	25,50
4	Este software responde demasiado lento al momento de realizar una acción.	4,20	25,20
5	Este software ocasionalmente se comporta de una manera que no se puede entender lo que se debe hacer.	4,15	24,90

6	La velocidad de este software es lo suficientemente rápida.	4,60	27,60
Promedio		4,31	25,85

Realizado por: Mathew Avilés, 2023

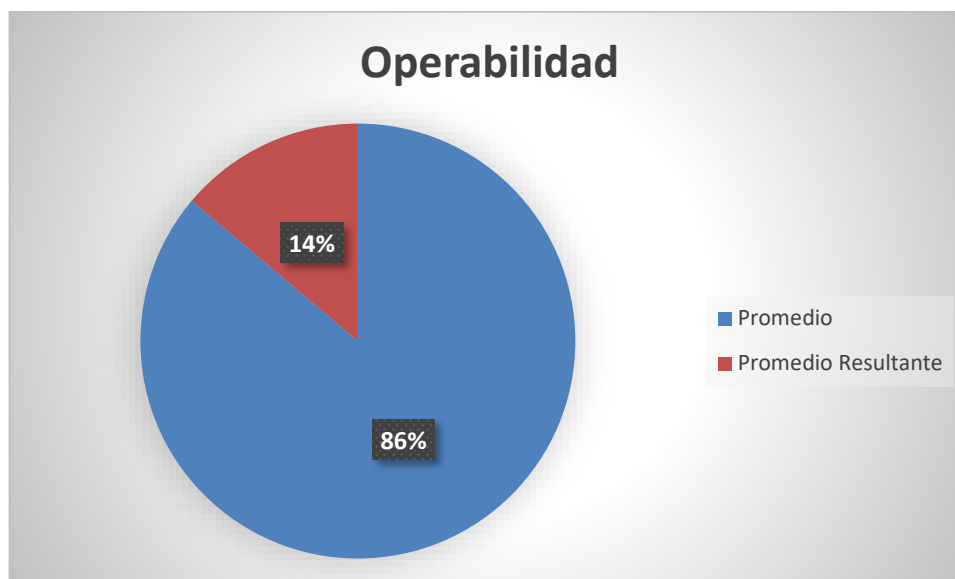


Gráfico 1-4 Operabilidad de la aplicación sobre 5 puntos

Realizado por: Mathew Avilés, 2023

En la **tabla 12-4**, se puede observar que la calificación más baja corresponde a la pregunta 2, con una puntuación de 4,05 de 5 puntos. Al analizar las razones detrás de esta calificación, se identifica que, al momento de iniciar el sistema de reconocimiento de mascotas, existe una demora de aproximadamente 2 segundos para obtener los resultados debido a la carga del modelo desde la web mediante el sistema de Teachable Machine para el reconocimiento de imágenes. Por otro lado, las preguntas 1 y 6 obtienen las mejores puntuaciones debido a que el control del sistema es muy sencillo y cumple todas las funcionalidades para las que fue diseñado. Además, el sistema responde rápidamente a las acciones realizadas por los usuarios como registrarse, iniciar sesión, registrar una mascota.

En la **ilustración 10-4**, se observa que el promedio de operabilidad obtenido mediante las encuestas es del 86%, pero existe un 14% de margen de mejora. Para lograr una calificación más alta, se sugiere ajustar algunas características, como mejorar la velocidad del sistema al momento de realizar la búsqueda mediante imágenes o con cámara, también se podría mejorar la organización del contenido y agregar nuevas características como por ejemplo la funcionalidad de compartir en redes sociales los anuncios de mascotas perdidas.

4.2.2 Reconocibilidad de la adecuación

Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades ISO (2023).

Tabla 14-4 Resultados del cuestionario de la subcaracterística Reconocibilidad de la adecuación.

Número	Preguntas	Promedio (_/5)	Resultado (/_30)
Reconocibilidad de la adecuación			
7	El software no siempre ha hecho lo que esperaba o no ha cumplido tus expectativas como usuario.	3,80	22,80
8	Puede entender y actuar sobre la información proporcionada por este software.	4,75	28,50
9	Las tareas se pueden realizar de manera sencilla utilizando este software.	4,75	28,50
10	Cree que este software es inconsistente, es decir no realiza correctamente los procesos para lo que fue diseñado.	4,20	25,20
Promedio		4,38	26,25

Realizado por: Mathew Avilés, 2023

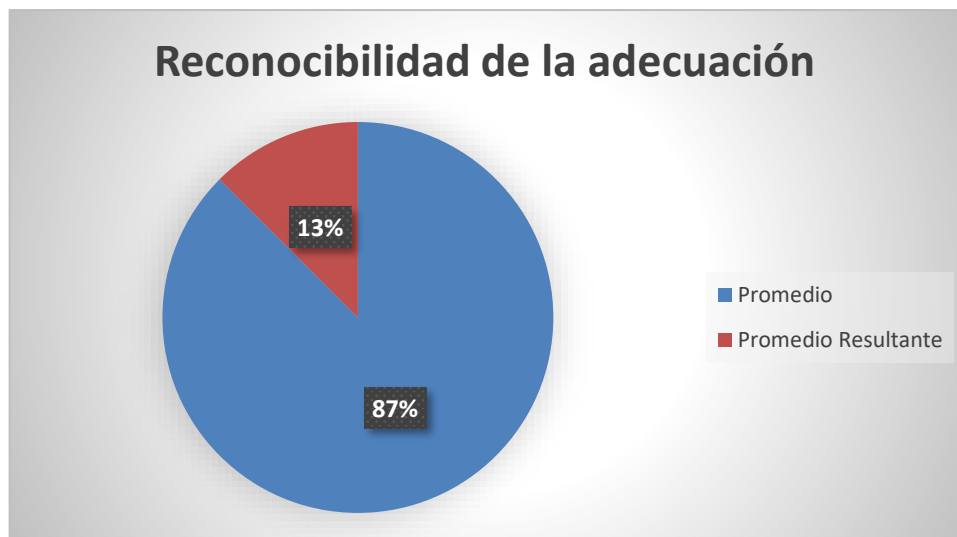


Gráfico 2-4 Promedio de Reconocibilidad de la adecuación sobre 5 puntos

Realizado por: Mathew Avilés, 2023

En la **tabla 13-4**, se observa que la calificación más baja corresponde a la pregunta 7, con un resultado de 3,80 sobre 5. Al analizar los datos de las encuestas, podemos percatarnos de que los usuarios al utilizar la aplicación consideran que su funcionamiento es correcto. Sin embargo, algunos mencionaron que, inicialmente, tenían la expectativa de que la aplicación operara mediante un sistema de rastreo GPS con el uso de chips para las mascotas. Esta era su primera impresión de la aplicación antes de la capacitación previa a el uso del aplicativo. No obstante, al utilizarla, comprendieron que, en realidad, el sistema se basa en el reconocimiento mediante imágenes y no en un sistema de rastreo GPS.

En la **ilustración 11-4**, se muestra que el promedio de reconocibilidad de la adecuación es del 87% sobre 5 puntos. Aunque existe un 13% que puede ser mejorado para obtener una calificación más alta en esta subcaracterística. Una opción para lograrlo sería implementar más funcionalidades en la aplicación como lo que proponen los usuarios al realizar un rastreo mediante GPS y la utilización de chips, esto es un proyecto mucho ambicioso y completo, pero logrará aumentar la funcionalidad y versatilidad del aplicativo, pero requerirá de la inmersión de muchas más personas que estén especializadas y conozcan del tema de chips para mascotas y sistemas de rastreo GPS.

4.2.3 *Aprendizabilidad*

Capacidad del producto que permite al usuario aprender su aplicación ISO (2023).

Tabla 15-4 Resultados del cuestionario de la subcaracterística aprendizabilidad.

Número	Preguntas	Promedio (_/5)	Resultado (/30)
Aprendizabilidad			
11	Se necesita demasiado tiempo para aprender las funciones del software.	3,95	23,7
12	Es fácil olvidar cómo hacer las cosas con este software.	3,95	23,7
13	Los mensajes de error no son adecuados.	3,65	21,9
Promedio		3,85	23,10

Realizado por: Mathew Avilés, 2023

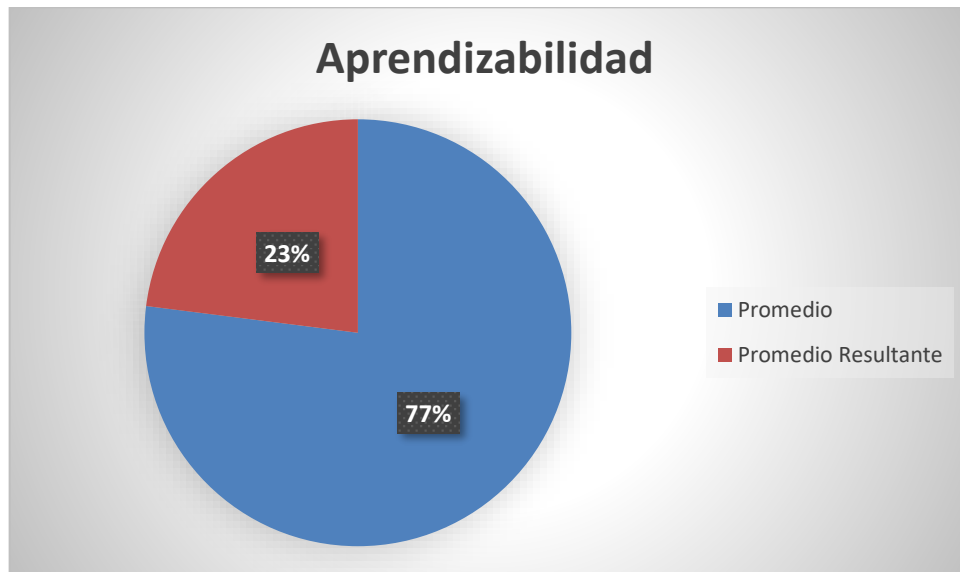


Gráfico 3-4 Promedio de Aprendizabilidad sobre 5 puntos

Realizado por: Mathew Avilés, 2023

En la **tabla 14-4** se observa que el promedio más bajo corresponde a la pregunta 13, que se refiere a que los mensajes de error no son los adecuados. Tras analizar la pregunta y las respuestas proporcionadas por los usuarios, se concluye que esto se debe a que al momento de realizar la socialización del funcionamiento de la aplicación a los usuarios mostramos los principales mensajes de error que mostraba la aplicación para mitigar los errores por parte del usuario y al momento de que los usuarios usaron la aplicación, estos no presentaron dichos errores, por qué ya sabían cómo funcionaba y no cometieron los mismos errores que se presentaron al momento de la socialización como ingresar correos que no tienen la estructura de un correo o nombres de usuarios que ya existan en la aplicación o que el número de teléfono estuviese incorrecto, pero la calificación de 3,65 se podría deber más a que una minoría de mensajes de error no estaban en español, por la utilización del framework Flask donde si no ponemos un mensaje personalizado por cada control de error, el mensaje lo muestra en inglés.

Por otro lado, en la **ilustración 12-4** se muestra que el promedio obtenido en la subcaracterística de Reconocibilidad de la adecuación sobre 5 puntos es del 77%.

Esto indica que existe margen de mejora en aspectos como el correcto manejo de mensajes de errores, esto se lo logró personalizando los mensajes de error y poniéndolos en español para que todos los usuarios puedan entenderlos y comprenderlos.

4.2.4 Estética de la interfaz de usuario

Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario ISO (2023).

Tabla 16-4 Resultados del cuestionario de la subcaracterística Estética de la interfaz de usuario.

Número	Preguntas	Promedio (_/5)	Resultado (/_30)
Estética de la interfaz de usuario			
14	La forma en que se presenta la información del sistema es clara y comprensible.	4,70	28,20
15	El software se presenta de una manera muy atractiva visualmente.	4,65	27,90
16	Nunca hay suficiente información en la pantalla cuando se necesita.	4,25	25,50
Promedio		4,53	27,20

Realizado por: Mathew Avilés, 2023

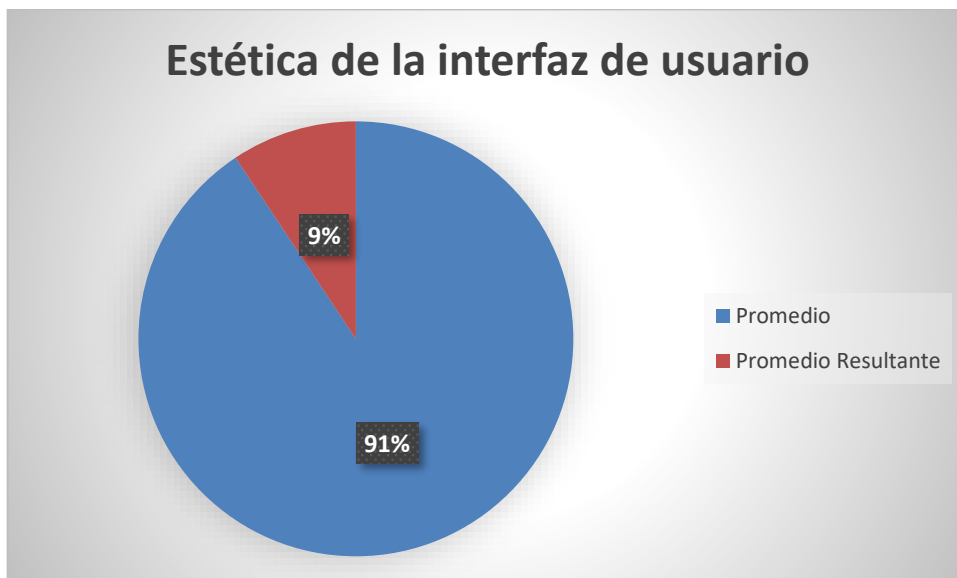


Gráfico 4-4 Promedio de la Estética de la interfaz de usuario sobre 5 puntos

Realizado por: Mathew Avilés, 2023

En la **tabla 15-4**, se observa que la pregunta 16 obtiene el promedio más bajo que es de 4,25 sobre 5 puntos. Al analizar dicha pregunta y las respuestas por parte de los usuarios, se puede determinar que el sistema no proporciona suficiente información cuando es necesaria. Esto podría deberse a un diseño no complicado, donde la pantalla contiene solo información estrictamente relevante, evitando así una saturación con contenido que podrían resultar poco útiles o interferir con la verdadera funcionalidad del aplicativo. Además, es posible que las personas estén influenciadas

por su familiaridad con modelos de redes sociales los mismos que presentan una gran cantidad de opciones y contenido en pantalla para mantener al usuario enganchado en dicha red social.

Por otro lado, en la **ilustración 13-4** se muestra que la subcaracterística de estética obtiene un promedio del 91% sobre 5, lo que indica que a los usuarios les agrada el diseño de la aplicación. Sin embargo, se identifica la necesidad de mejorar ciertos aspectos, como aumentar la cantidad de información en las interfaces para lograr una mejor puntuación en esta área.

4.3 Análisis de resultados de la variable usabilidad

En la siguiente tabla, se muestra el promedio de cada subcaracterística, que refleja el nivel de usabilidad de la aplicación en forma de porcentaje.

Tabla 17-4 Resultados usabilidad

Subcaracterística	Promedio (_/5)	Resultado en porcentaje	Porcentaje de valoración
Operabilidad	4,31	31,02%	36%
Reconocibilidad de la adecuación	4,38	21,88%	25%
Aprendizabilidad	3,85	14,63%	19%
Estética de la interfaz de usuario	4,53	18,13%	20%
Total		85,66%	100%

Realizado por: Mathew Avilés, 2023

En la **ilustración 17-4** se realiza una comparación entre el grado de usabilidad de cada subcaracterística y su ponderación total. Es notable que la subcaracterística de estética de la interfaz de usuario es la que más se acerca al cumplimiento del porcentaje total indicado, mientras que la subcaracterística de aprendizaje es la que se aleja más del cumplimiento del porcentaje total requerido.

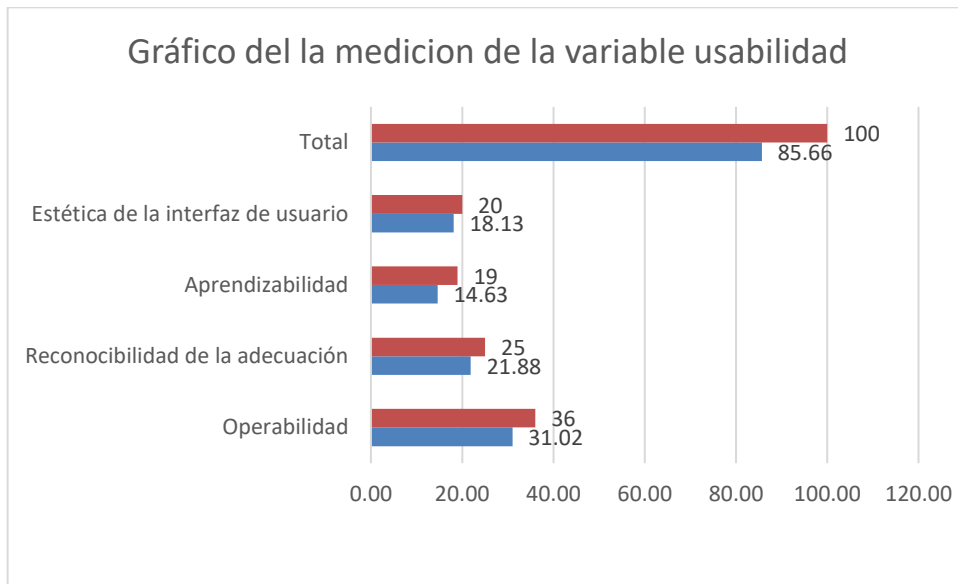


Gráfico 5-4 Nivel de usabilidad.

Realizado por: Mathew Avilés, 2023

En la **ilustración 11-4** se muestra el nivel de usabilidad del aplicativo web diseñado para la búsqueda e identificación de mascotas, el cual alcanza un 85,66%. Este resultado evidencia la capacidad del sistema para ser utilizado de manera efectiva por los usuarios.



Gráfico 6-4 Nivel de usabilidad de la aplicación

Realizado por: Mathew Avilés, 2023

4.4 Niveles de evaluación de la usabilidad

En la **ilustración 18-4** se detallan los niveles de puntuación porcentual para la determinación del grado de usabilidad del aplicativo de búsqueda e identificación de mascotas.

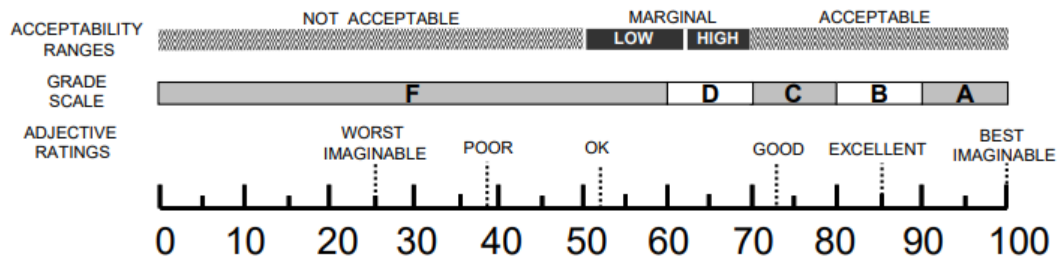


Figura 15-4 Calificaciones de adjetivos, puntajes de aceptabilidad y escalas de calificaciones en relación con los porcentajes promedios del cuestionario SUS.

Fuente: (Bangor 2009)

Los resultados obtenidos se van a contrastar con la **tabla 18-4** que indican la escala para medir el nivel de usabilidad de software (SUS), basándonos en esta escala podemos observar que el promedio obtenido es del 85,56% cayendo en un rango de aceptabilidad aceptable y un grado de B en la escala logrando concluir que el nivel de usabilidad de la aplicación está por encima del rango de aceptabilidad en cuanto a la usabilidad.

CAPÍTULO V:

5 CONCLUSIONES Y RECOMENDACIONES

Concluido el trabajo de titulación, se llegó a las siguientes conclusiones:

- La adopción de la metodología SCRUMBAN en conjunto con el enfoque en historias de usuario ha demostrado ser altamente beneficiosa en la redacción de las funcionalidades primordiales de la aplicación. Mediante la implementación de historias de usuario, se logró una identificación concisa de las características esenciales, estableciendo así una conexión más estrecha con las necesidades de los usuarios finales. Este enfoque permitió la determinación tanto de los requisitos funcionales como de los no funcionales de la aplicación de manera efectiva.
- Los módulos de login, mascotas y sobre nosotros se desarrollaron de manera satisfactoria y completa al aplicar las mejores prácticas de usabilidad como simplicidad, claridad, interfaces agradables, etc. El módulo de inicio de sesión garantiza un acceso seguro y personalizado, mientras que el módulo de mascotas ofrece un registro eficiente que incluye opciones de identificación basada en imágenes. Por su parte, el apartado 'Sobre nosotros' promueve los servicios esenciales de la clínica veterinaria y proporciona información relevante para nuestros usuarios.
- La evaluación de la usabilidad de la aplicación web, en línea con la norma ISO 25010, se llevó a cabo a través del análisis y la interpretación de los datos obtenidos mediante la aplicación del cuestionario de usabilidad. Este proceso, respaldado por los criterios de la norma, arrojó resultados concluyentes que indican que la aplicación diseñada para la identificación y búsqueda de mascotas extraviadas logra un nivel de usabilidad del 85.66%. Esta calificación se encuentra en un rango aceptable según la escala de medición de software (SUS), lo que refuerza la efectividad de la aplicación en términos de facilidad de uso, comprensión intuitiva y satisfacción del usuario. La adopción de la norma ISO 25010 ha proporcionado un marco sólido para esta evaluación, permitiendo una medición objetiva de la usabilidad y respaldando la conclusión de que la aplicación no solo cumple con sus funcionalidades, sino que también ofrece una experiencia de usuario satisfactoria y eficiente.

Las recomendaciones que se determinó son las siguientes:

- Para fortalecer y ampliar las funcionalidades de la aplicación, se sugiere incorporar un sistema de rastreo mediante la implementación de chips a las mascotas los mismo que puedan emitir frecuencias las cuales nos permitan saber la ubicación de la mascota mediante el uso de la tecnología GPS, lo que brindaría beneficios significativos al incorporar dicha tecnología para la búsqueda de mascotas extraviadas.
- Para avanzar en el uso de redes neuronales convolucionales, es recomendable investigar cómo usar lo que ya se ha aprendido de otros modelos similares para mejorar nuevos modelos. Esto se llama "transferencia de aprendizaje". Al aplicar esto, se lograría hacer que el entrenamiento de nuevos modelos sea más rápido y preciso, incluso si no se tiene muchos datos disponibles. También sería útil explorar nuevas formas de diseñar las redes neuronales. Por ejemplo, se podría usar arquitecturas más avanzadas, como redes en cascada o modelos de atención, para resolver problemas específicos en la visión por computadora. Estas exploraciones podrían traer nuevas perspectivas y mejoras en el campo de las redes neuronales convolucionales.
- Se recomienda que el sistema en un futuro se pueda implementar en fundaciones de animales o en algún municipio de la ciudad para poder llegar a más personas y lograr mejores resultados en la búsqueda y que más mascotas se encuentren identificadas en un sistema que ayude a una centralización y crecimiento de datos de mascotas perdidas, encontradas y en adopción.

BIBLIOGRAFÍA

- AZABACHE MEDINA, B.J. y CARUAJULCA SÁNCHEZ, J.L.D.**, 2019. Sistema web móvil de gestión de accesos del personal de la empresa TAL S.A. utilizando ICONIX SCRUMBAN y tecnología RFID. ,
- BAJAÑA MORAN, L.A. y PALADINES GALLARDO, J.A.**, 2021. Diseño e implementación de un prototipo clasificador de granos de cacao usando Visión Artificial y Machine Learning. ,
- BANGOR, A.**, 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. , vol. 4, no. 3,
- BARCENES LLANOS, C.J. y CEDEÑO AVELINO, M.F.**, 2019. Desarrollo de un servicio web y aplicación móvil en android que permita ubicar y rescatar animales domésticos extraviados o abandonados para la fundación rescate animal ecuador. ,
- CARNEY, M., WEBSTER, B., ALVARADO, I., PHILLIPS, K., HOWELL, N., GRIFFITH, J., JONGEJAN, J., PITARU, A. y CHEN, A.**, 2020. Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* [en línea]. New York, NY, USA: Association for Computing Machinery, pp. 1-8. [consulta: 24 abril 2023]. CHI EA '20, ISBN 978-1-4503-6819-3. DOI 10.1145/3334480.3382839. Disponible en: <https://doi.org/10.1145/3334480.3382839>.
- CATALÁN VEGA, M.A.**, 2006. *Metodologías de evaluación de interfaces gráficas de usuario*. 2006. S.l.: s.n.
- DE GIUSTI, L. y TARRÍO, D.**, 2000. Algoritmo de Análisis de Similitud de Imágenes. ,
- DURANGO, A.**, 2015. *Diseño Web con CSS: 2ª Edición*. S.l.: IT Campus Academy. ISBN 978-1-5150-5278-4.
- FANCYPETS**, 2022. Cómo encontrar a un perro perdido: consejos para que vuelva pronto a casa - Fancypets. [en línea]. [consulta: 24 abril 2023]. Disponible en: <https://fancypets.pe/como-encontrar-a-un-perro-perdido/>.
- FRAE**, 2022. Estadísticas de animales extraviados GYE, de enero a junio de 2022. *FRAE* [en línea]. [consulta: 13 enero 2023]. Disponible en: <https://rescateanimal.org.ec/estadisticaampsej2022/>.
- GARCIA, M., ALEJANDRO, P., COROBARRUTIA, M. y RICARDO, A.**, 2022. Aplicativo Móvil Híbrido Apoyado en Reconocimiento de Imágenes para el Proceso de Adopción y Búsqueda de Mascotas. ,
- GONZÁLEZ DUQUE, R.**, 2011. *Python para todos*. S.l.: s.n.

- GONZÁLEZ, G.M.**, 2016. *Aprende a Desarrollar con Spring Framework: 2ª Edición*. S.l.: IT Campus Academy. ISBN 978-1-5411-5737-8.
- HASSAN MONTERO, Y.**, 2002. Introducción a la Usabilidad. *No Solo Usabilidad* [en línea], no. 1, [consulta: 10 enero 2023]. Disponible en: https://www.nosolousabilidad.com/articulos/introduccion_usabilidad.htm.
- TEACHABLEMACHINE.WITHGOOGLE.COM**, 2022. Teachable Machine. [en línea]. [consulta: 18 abril 2023]. Disponible en: <https://teachablemachine.withgoogle.com/>.
- HURTADO, M.**, 2014. Python. [en línea]. [consulta: 23 mayo 2023]. Disponible en: <https://sites.google.com/site/xtmichaelhurtado/proyecto-10>.
- IBARRA GUZMÁN, D., CASTAÑEDA ISLAS, U., PÉREZ CORONA, C. y PEDROZA MÉNDEZ, B.E.**, 2014. Metodología ágil Scrumban en el proceso de desarrollo y mantenimiento de software de la norma MoProSoft. *Research in Computing Science*, vol. 79, no. 1, ISSN 1870-4069. DOI 10.13053/rcs-79-1-8.
- ISO**, 2023. ISO 25010. [en línea]. [consulta: 12 enero 2023]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&start=2>.
- JHON EDWARD, P.A.**, 2020. Sistema web basado en la gestión de mascotas y su geolocalización en caso de extravío en la Municipalidad Distrital de Puente Piedra. ,
- KREIBICH, J.**, 2010. *Using SQLite*. S.l.: O'Reilly Media, Inc. ISBN 978-0-596-52118-9.
- LAOYAN**, 2022. Scrumban: lo mejor de dos metodologías ágiles • Asana. *Asana* [en línea]. [consulta: 13 enero 2023]. Disponible en: <https://asana.com/es/resources/scrumban>.
- LATERIO MARTINEZ, J.M.**, 2016. Usabilidad en un sistema de gestión del conocimiento: Caso consultoría KM Solucion. ,
- LIU, W., WANG, Z., LIU, X., ZENG, N., LIU, Y. y ALSAADI, F.E.**, 2017. A survey of deep neural network architectures and their applications. *Neurocomputing*, vol. 234, ISSN 0925-2312. DOI 10.1016/j.neucom.2016.12.038.
- MAIDA, E.G.**, 2015. Metodologías de desarrollo de software. ,
- MORA, N.L.**, 2017. Métricas de calidad para el desarrollo de aplicaciones web. , vol. 11,
- MUGUIRA, A.**, 2016. Escala de Likert: Qué es y cómo utilizarla en tus encuestas. *QuestionPro* [en línea]. [consulta: 18 julio 2023]. Disponible en: <https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/>.
- OROZCO VILLACRÉS, A.R.**, 2022. Implementación de aplicación web para el registro y control de perros extraviados utilizando código QR en el centro veterinario animedic propuesta tecnológica. ,
- ORTIZ, D.C. y JIMÉNEZ, F.J.F.**, 2018. Inteligencia Artificial con TensorFlow para predicción de comportamientos. ,
- OTZEN, T. y MANTEROLA, C.**, 2017. Técnicas de Muestreo sobre una Población a Estudio. *International Journal of Morphology*, vol. 35, no. 1, ISSN 0717-9502. DOI 10.4067/S0717-95022017000100037.

- OWENS, M.**, 2006. Introducing SQLite. En: , *The Definitive Guide to SQLite* [en línea]. Berkeley, CA: Apress, pp. 1-16. [consulta: 23 mayo 2023]. ISBN 978-1-4302-0172-4. Disponible en: https://doi.org/10.1007/978-1-4302-0172-4_1.
- PALACIOS DE LA CRUZ, M.B. y PINARGOTE TOMALÁ, A.D.**, 2021. Implementación de una aplicación web para el reconocimiento de patrones de diagnóstico del COVID 19 en rayos x mediante una red neuronal convolucional para la universidad técnica de Cotopaxi extensión la maná. ,
- PERURENA CANCIO, L. y MORÁGUEZ BERGUES, M.**, 2013. *Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación*. 2013. S.l.: s.n.
- QUINN RADICH**, 2021. What is a machine learning model? [en línea]. [consulta: 18 abril 2023]. Disponible en: <https://learn.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model>.
- RODRÍGUEZ FRANCO, M.J.**, 2014. Desarrollo de componentes de software con tecnologías emergentes. ,
- RODRÍGUEZ, R.A., VERA, P.M., MARTÍNEZ, M.R., BELTRÁN, F.A.P., TRIGUEROS, A. y DOGLIOTTI, M.G.**, 2019. Aplicaciones Web Progresivas Impulsadas por el Avance de los Estándares Web. ,
- SMARANDACHE, F.**, 2022. *Collected Papers. Volume X: On Neutrosophics, Plithogenics, Hypersoft Set, Hypergraphs, and other topics*. S.l.: Infinite Study.
- SQLITE**, 2023. About SQLite. [en línea]. [consulta: 23 mayo 2023]. Disponible en: <https://www.sqlite.org/about.html>.
- TEAM, Y.**, 2015. Scrumban – An amalgamation of Scrum and Kanban. *Yodiz Project Management Blog* [en línea]. [consulta: 6 junio 2023]. Disponible en: <http://www.yodiz.com/blog/scrumban-an-amalgamation-of-scrum-and-kanban/>.
- TORRES, J.**, 2020. *Python Deep Learning: Introducción práctica con Keras y TensorFlow 2*. S.l.: Alpha Editorial. ISBN 978-958-778-641-5.

ANEXOS

Anexo A: Requisitos funcionales

Requisitos Funcionales			
Nro. de requisito	Nombre	Característica	Descripción
RF1	Registro e inicio de sesión	El sistema debe permitir a los usuarios registrarse e iniciar sesión.	Los usuarios deben poder registrarse proporcionando nombre, correo electrónico y contraseña, y luego iniciar sesión utilizando sus credenciales.
RF2	Registro de mascota	El sistema debe proporcionar una página de registro de mascota.	Los usuarios deben poder registrar los datos de su mascota, incluyendo estado, y visualizarlo en la aplicación para que otros usuarios lo vean.
RF3	Edición de datos de mascota	El sistema debe permitir a los usuarios editar los datos de la mascota registrada.	Los usuarios deben poder acceder a la página de mascota y realizar cambios en los datos de su mascota.
RF4	Eliminación de mascota	El sistema debe permitir a los usuarios eliminar una mascota registrada.	Los usuarios deben tener la opción de eliminar una mascota
RF5	Visualización de mascotas según indicadores	El sistema debe permitir a los usuarios visualizar las mascotas según indicadores como sexo, ubicación y estado.	Los usuarios deben tener acceso a una página de mascotas donde puedan filtrar y visualizar las mascotas basadas en indicadores específicos.
RF6	Subida de imágenes de mascotas extraviadas	El sistema debe permitir a los usuarios subir imágenes de sus mascotas extraviadas.	Los usuarios deben poder adjuntar imágenes de sus mascotas extraviadas en la página de registro de datos de mascotas
RF7	Búsqueda de mascota mediante ingreso de imagen	El sistema debe permitir a los usuarios buscar una mascota mediante el ingreso de una imagen.	Los usuarios deben tener la opción de subir una imagen y utilizarla para buscar si la mascota se encuentra en la base de datos de la aplicación y determinar su estado y datos generales de la mascota.

Anexo B: Requisitos no funcionales

Requisitos no Funcionales			
ID Requisito	Nombre del requisito	Característica	Descripción
RNF1	Usabilidad	Sistema agradable para el usuario	El sistema deberá ser sencillo, intuitivo, de fácil aprendizaje. Y cumplir con los requerimientos.
RNF2	Seguridad	Sistema que protege la información	El sistema contará con un control de acceso seguro de la información y datos personales, recuperación de contraseñas

Anexo C: Diagramas de procesos

Diagrama de procesos para identificación de mascotas mediante cámara

En este diagrama de muestra las actividades para identificar a la mascota mediante el uso de la cámara.

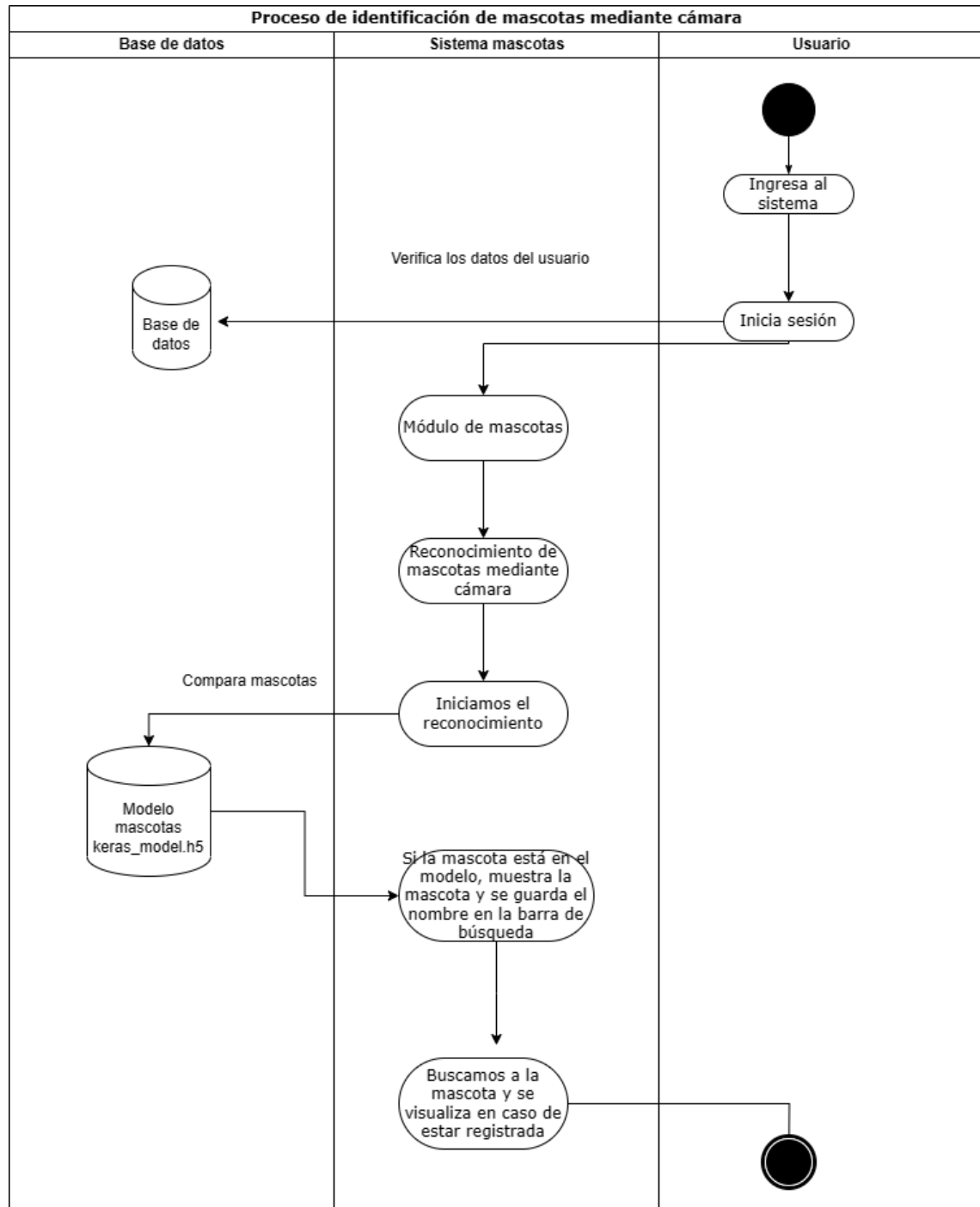


Diagrama de procesos para identificación de mascotas mediante imágenes

En este diagrama se muestra las actividades para identificar a la mascota mediante el ingreso de una imagen de una mascota.

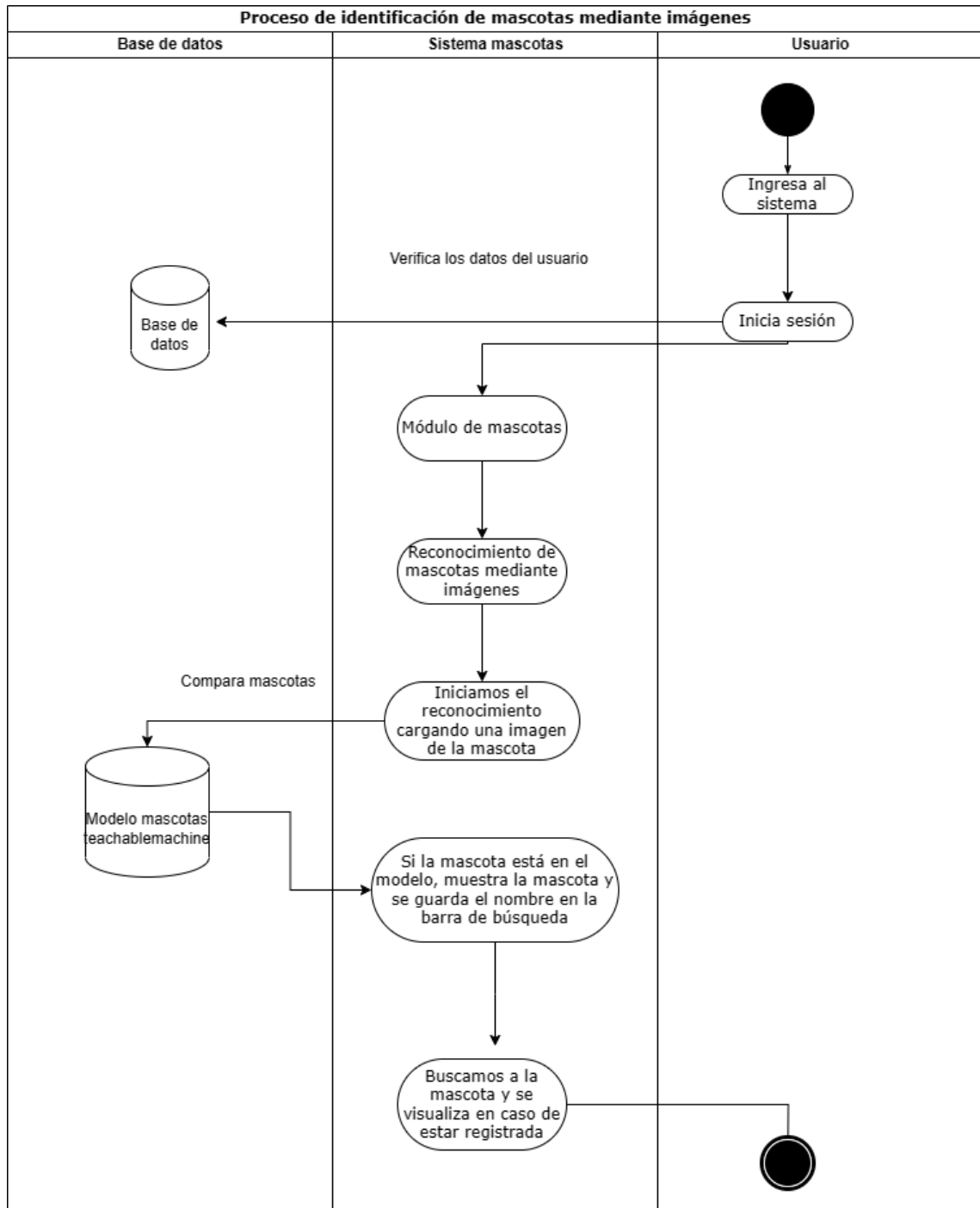
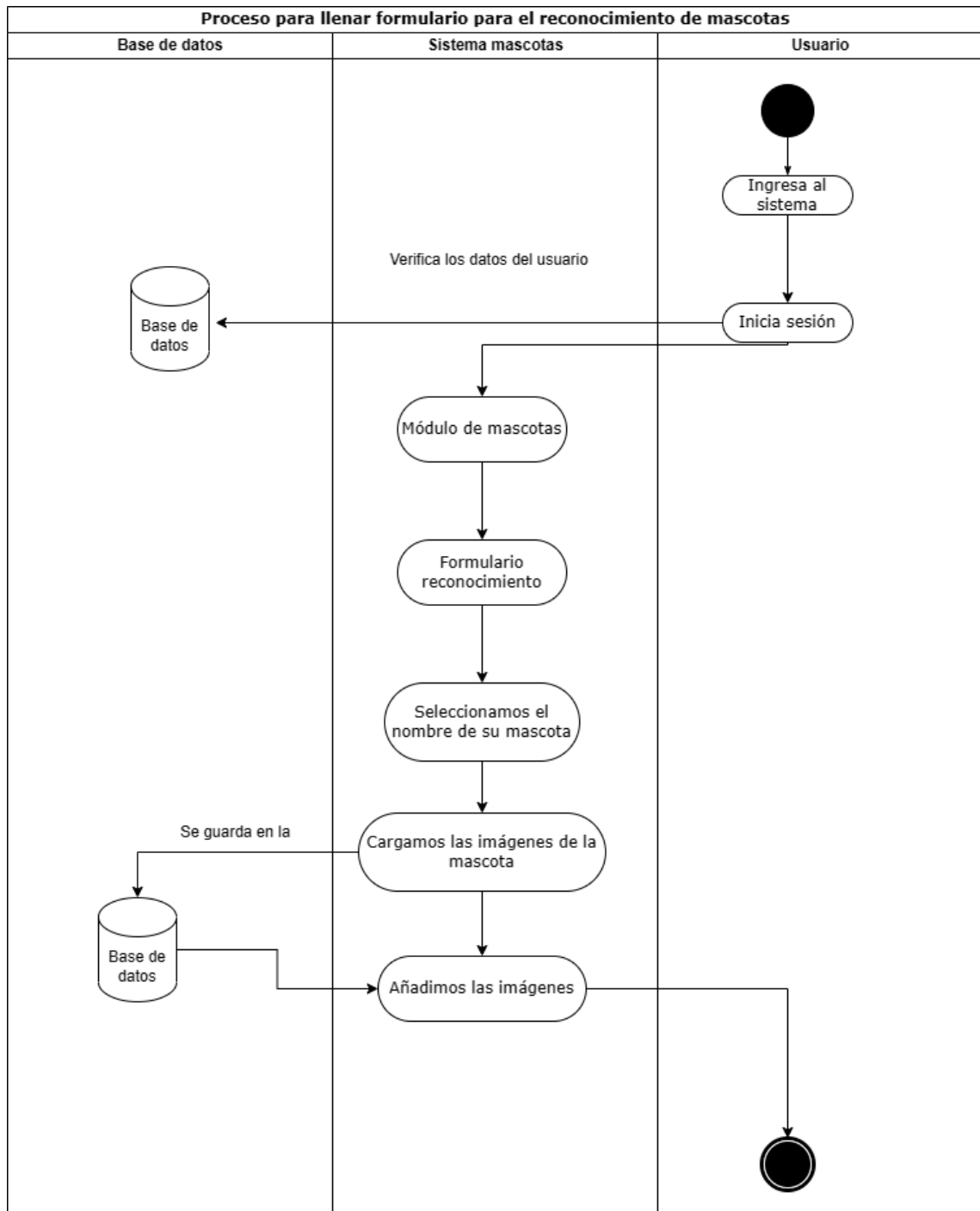


Diagrama de procesos para llenar formulario de identificación de mascotas

En este diagrama de muestra las actividades para llenar el formulario para ingresar las imágenes al modelo de reconocimiento de mascotas.



Anexo D: Diccionario de datos

Diccionario de datos registro_reconocimiento

Nombre del archivo: registro_reconocimiento				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,1
usuario_id	Identificador del usuario	Integer	No	Solo números
				[0,1,2,.....]
mascota_id	Identificador de la mascota	Integer	No	Solo números [0,1,2,.....]
nombre_usuario	Nombre del usuario	Varchar(50)	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,}
nombre_mascota	Nombre de la mascota	Varchar(20)	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,}
telefono_usuario	Teléfono del usuario	Varchar(10)	No	solo números [0,1,2,.....]
fecha	Fecha en que se registra el usuario.	Datetime	No	* formato: aaaa-mm-dd *

Diccionario de datos tipo_m

Nombre del archivo: tipo_m				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,2...
tipo_m	Identificador del usuario	Varchar	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,}

Diccionario de datos sexo_m

Nombre del archivo: sexo_m				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,2...
sexo_m	Identificador del usuario	Varchar	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,}

Diccionario de datos ubicacion_m

Nombre del archivo: ubicacion_m				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,2...
ubicacion_m	Identificador del usuario	Varchar	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,}

Diccionario de datos estado_m

Nombre del archivo: estado_m				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,2...
estado_m	Identificador del usuario	Varchar	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,}

Diccionario de datos addmascota

Nombre del archivo: addmascota				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,1
nombre	Nombre de la mascota	Varchar(20)	No	=.*[a-z])(?=. *[A-Z])(?=. *d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,}
edad	Edad de la mascota	Integer	No	[0,1,2,.....]
raza	Raza de la mascota	Varchar(20)	No	=.*[a-z])(?=. *[A-Z])(?=. *d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,}
contacto	Teléfono del usuario	Varchar(10)	No	=.*[a-z])(?=. *[A-Z])(?=. *d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,}
direccion	Dirección donde se perdió la mascota	Varchar(50)	No	=.*[a-z])(?=. *[A-Z])(?=. *d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,}

descripcion	Descripción de la mascota	Varchar(150)	No	=.*[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,}
fecha	Fecha en que se perdió la mascota	Date	No	* formato: aaaa-mm-dd *
pub_date	Fecha en que se registra el usuario.	Datetime	No	* formato: aaaa-mm-dd *
sexo_id	Identificador del sexo	Integer	No	Valores numéricos [0,1,2,3,.....]
tipo_id	Identificador del tipo	Integer	No	Valores numéricos [0,1,2,3,.....]
ubicación_id	Identificador de la ubicación	Integer	No	Valores numéricos [0,1,2,3,.....]
estado_id	Identificador del estado	Integer	No	Valores numéricos [0,1,2,3,.....]
user_id	Identificador del usuario	Integer	No	Valores numéricos [0,1,2,3,.....]
image_1	Imagen 1 de la mascota	Varchar(150)	No	[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg,jpeg

image_2	Imagen de la mascota	Varchar(150)	No	[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_3	Imagen de la mascota	Varchar(150)	No	[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,} mas imágenes en formato jpg, jpeg

Diccionario de datos usuario

Nombre del archivo: usuario				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permite NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,2...
nombre	Nombre del usuario administrador	Varchar	No	[a-z])(?=. *[A-Z])(?=. *\d)(?=. *[@!%*?&])[A-Za-z\d@!%*?&]{8,}

correo	correo del usuario administrador	Varchar	No	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@!%*?&])[A-Za-z\d@!%*?&]{8,}
password	contraseña del usuario administrador	Varchar	No	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@!%*?&])[A-Za-z\d@!%*?&]{8,}

Diccionario de datos formulario_r

Nombre del archivo: formulario_r				
Descripción del archivo: Persona natural que utiliza el sistema.				
Nombre del campo	Descripción	Tipo de dato y tamaño	Permit e NULL	Valor permitido del dato
id (PK)	Identificador.	Integer	No	Mayores a 0
				*Es auto incremental 1,2...
nombre	Nombre del usuario administrador	Varchar(20)	No	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@!%*?&])[A-Za-z\d@!%*?&]{8,}
addmascota_id	Identificador de la mascota	Integer	No	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@!%*?&])[A-Za-z\d@!%*?&]{8,}

image_1	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_2	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_3	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_4	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg

image_5	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_6	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_7	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_8	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg

image_9	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_10	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_11	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_12	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg

image_13	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_14	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_15	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_16	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg

image_17	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_18	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_19	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_20	Imagen de la mascota	Varchar(250))	Si	[a-z](?=. *[A-Z])(?=. *\d)(?=. *[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg

image_21	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_22	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_23	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_24	Imagen de la mascota	Varchar(250)	Si	[a-z](?=[A-Z])(?=[\d])(?=[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg

image_25	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_26	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_27	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg
image_28	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg, jpeg

image_29	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg,jpeg
image_30	Imagen de la mascota	Varchar(250))	Si	[a-z](?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,} mas imágenes en formato jpg,jpeg

Anexo E: Pruebas

1. El primer control de validación se presenta al momento de registrar un usuario, como se observa en la ilustración a continuación, que el correo ingresado debe ser válido.



The image shows a user registration form titled "Registro de usuario". The form contains several input fields and a button. The fields are labeled "Nombre:", "Usuario", "Correo:", "Teléfono:", "Contraseña:", and "Repita la contraseña:". The "Nombre:" field contains "Juan". The "Usuario" field contains "juanmartinez". The "Correo:" field contains "qqquuul123", and below it, a red error message reads "• Invalid email address.". The "Teléfono:" field contains "0980237364". The "Contraseña:" and "Repita la contraseña:" fields are empty. At the bottom, there is a blue button labeled "Iniciar Sesión".

Registro de usuario

Nombre:

Usuario

Correo:

- Invalid email address.

Teléfono:

Contraseña:

Repita la contraseña:

Iniciar Sesión

2. Otro control de validación se presenta al momento de ingresar la contraseña , como se observa en la ilustración a continuación, la contraseña debe contener 8 caracteres como mínimo, entre mayúsculas, minúsculas, y números.

The image shows a login form with the following fields and values:

- Nombre:** Juan
- Usuario:** juanmartinez
- Correo:** juanaviles12345@gmail.com
 - Invalid email address.
- Teléfono:** 0995027724
- Contraseña:** (5 characters visible)

A validation error message is displayed below the password field:

! Prolonga este texto a 8 caracteres o más (actualmente tiene 5 caracteres).

.....

Iniciar Sesión

3. Otro control de validación se presenta al momento de ingresar la contraseña, como se observa en la ilustración a continuación, la contraseña debe coincidir al momento de repetirla, si no es el caso, me muestra un mensaje de que las contraseñas no coinciden.

Registro de usuario

Nombre:

Usuario

Correo:

Teléfono:

Contraseña:

- Las contraseñas no coinciden

Repita la contraseña:

[Iniciar Sesión](#)

4. Otro control de validación se presenta al momento de ingresar el nombre de usuario y el correo electrónico, como se observa en la ilustración a continuación, el nombre de usuario y el correo ya están registrados en la base de datos, por lo que procede a mostrar un mensaje de error que indica que el usuario y el correo deben ser diferentes a los antes ingresados.

Registro de usuario

Nombre:

Usuario

- Este nombre de usuario ya está registrado, ingrese otro

Correo:

- Este correo ya está registrado, ingrese otro

Teléfono:

Contraseña:

Repita la contraseña:

[Iniciar Sesión](#)

5. Cuando ingresemos correctamente las credenciales del usuario que se va a registrar al sistema como se muestra a continuación y dar clic en el botón de registrarme, el sistema automáticamente nos redije al formulario para iniciar sesión y nos muestra un mensaje de que el usuario Daysi Chávez se ha registrado y le da las gracias.

Registro de usuario

Nombre:

Daysi Chavez

Usuario

daysi

Correo:

daysi.chavez@epoch.edu.ec

Teléfono:

0979986286

Contraseña:

.....

Repita la contraseña:

.....

Registrarme

Inicio de Sesión



Correo:

Contraseña:

Iniciar Sesión

Regístrate

[Olvidé mi contraseña](#)

6. Otro control de validación es al momento de iniciar sesión se presenta cuando ingresamos un email invalido como se muestra en la ilustración

Inicio de Sesión



Correo:

- Invalid email address.

Contraseña:

Iniciar Sesión

Regístrate

Olvidé mi contraseña

7. Otro control de validación es al momento de iniciar sesión, se presenta cuando ingresamos una contraseña o correo que no son los correctos, mostrándonos el mensaje de correo o contraseña incorrecta.

Correo o contraseña incorrectos

Inicio de Sesión



Correo:


Contraseña:

[Iniciar Sesión](#)


[Regístrate](#)

8. Cuando ingresamos las credenciales correctamente al formulario para iniciar sesión y damos clic en iniciar sesión, inmediatamente accederemos al sistema de Huellitas.

Inicio de Sesión



Correo:

Contraseña:
 

[Iniciar Sesión](#)


[Regístrate](#)



9. Otro control de validación es al momento de cambiar la contraseña, se presenta cuando ingresamos un correo que no es válido, mostrándonos un mensaje de error indicando este error.

Correo electrónico no válido

Cambio de contraseña



Correo Electrónico

Nueva Contraseña


Confirmar Contraseña

[Restablecer Contraseña](#)

10. Otro control de validación es al momento de cambiar la contraseña, se presenta cuando las contraseñas que ingresamos no coinciden y al momento de ocurrir esto se presenta el siguiente

Las contraseñas no coinciden

Cambio de contraseña



Correo Electrónico

Nueva Contraseña

Confirmar Contraseña

[Restablecer Contraseña](#)

Anexo F: Cuestionario Sumi Usabilidad

Operabilidad

1. El uso de este software para el reconocimiento de mascotas resultó satisfactorio.
2. Este software se ha detenido en algún momento inesperadamente de tal manera que afecte su experiencia como usuario.
3. Hay demasiados pasos necesarios para realizar un proceso como registrar una mascota.
4. Este software responde demasiado lento al momento de realizar una acción.
5. Este software ocasionalmente se comporta de una manera que no se puede entender lo que se debe hacer.
6. La velocidad de este software es lo suficientemente rápida.

Reconocibilidad de la adecuación

7. El software no siempre ha hecho lo que esperaba o no ha cumplido tus expectativas como usuario.
8. Puede entender y actuar sobre la información proporcionada por este software.
9. Las tareas se pueden realizar de manera sencilla utilizando este software.
10. Cree que este software es inconsistente, es decir no realiza correctamente los procesos para lo que fue diseñado.

Aprendizabilidad

11. Se necesita demasiado tiempo para aprender las funciones del software.
12. Es fácil olvidar cómo hacer las cosas con este software.
13. Los mensajes de error no son adecuados.

Estética de la interfaz de usuario

14. La forma en que se presenta la información del sistema es clara y comprensible.
15. El software se presenta de una manera muy atractiva visualmente.
16. Nunca hay suficiente información en la pantalla cuando se necesita.

Anexo G: Respuestas de los usuarios al cuestionario

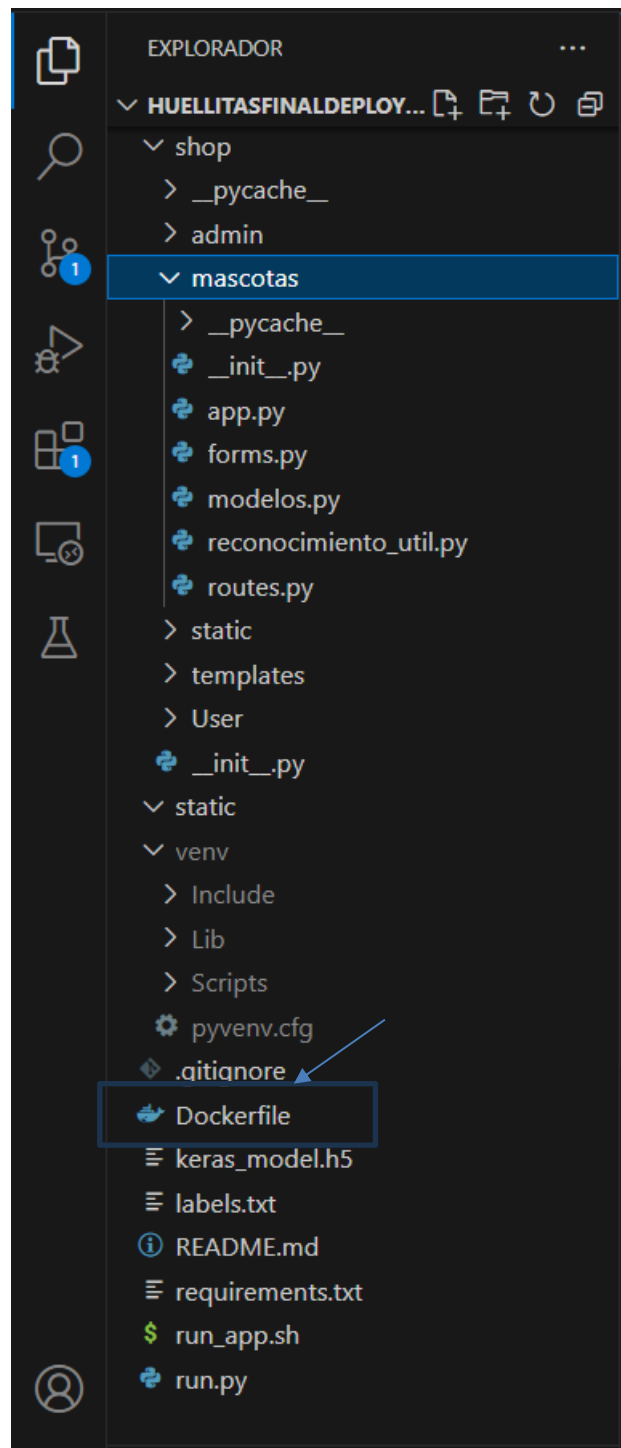
Encuestado	El uso de este software para el reconocimiento de mascotas resultó satisfactorio.	Este software se ha detenido en algún momento inesperadamente de tal manera que afecte su experiencia como usuario.	Hay demasiados pasos necesarios para realizar un proceso como registrar una mascota.	Este software responde demasiado lento al momento de realizar una acción.	Este software ocasionalmente se comporta de una manera que no se puede entender que se debe hacer.	La velocidad de este software es lo suficientemente rápida.	El software no siempre ha hecho lo que esperaba o no ha cumplido tus expectativas como usuario.	Puede entender y actuar sobre la información proporcionada por este software.	Las tareas se pueden realizar de manera sencilla utilizando este software.	Cree que este software es inconsistente, es decir no realiza correctamente los procesos para lo que fué diseñado.	Se necesita demasiado tiempo para aprender las funciones del software.	Es fácil olvidar cómo hacer las cosas con este software.	Los mensajes de error no son adecuados.	La forma en que se presenta la información del sistema es clara y comprensible.	El software se presenta de una manera atractiva visualmente.	Nunca hay suficiente información en la pantalla cuando se necesita.
1	De acuerdo	De acuerdo	Desacuerdo	Totalmente de acuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	Desacuerdo	Desacuerdo	Indeciso	Totalmente en desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo
2	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
3	Totalmente de acuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Indeciso	Desacuerdo	De acuerdo	De acuerdo	Desacuerdo	Desacuerdo	Totalmente en desacuerdo	Indeciso	De acuerdo	De acuerdo	Desacuerdo

11	Totalmente de acuerdo	Totalmente en desacuerdo	Desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	Desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo
12	De acuerdo	Totalmente en desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	Desacuerdo	Totalmente de acuerdo	De acuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Desacuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	De acuerdo	Totalmente en desacuerdo
13	Totalmente de acuerdo	Totalmente en desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Totalmente de acuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	De acuerdo	Desacuerdo
14	De acuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	Desacuerdo	De acuerdo	De acuerdo	Desacuerdo	Indeciso	Desacuerdo	Desacuerdo	De acuerdo	De acuerdo	Desacuerdo
15	Totalmente de acuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Indeciso	Totalmente de acuerdo	Desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo
16	De acuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	Desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo
17	De acuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Indeciso	Totalmente de acuerdo	Desacuerdo	De acuerdo	De acuerdo	Desacuerdo	Desacuerdo	Totalmente en desacuerdo	Indeciso	Totalmente de acuerdo	De acuerdo	Totalmente en desacuerdo

18	De acuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	Desacuerdo	De acuerdo	De acuerdo	Totalmente en desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	De acuerdo	Desacuerdo
19	Totalmente de acuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente en desacuerdo
20	Totalmente de acuerdo	Totalmente en desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Totalmente de acuerdo	Desacuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Desacuerdo	Desacuerdo	Desacuerdo	Desacuerdo	De acuerdo	Totalmente de acuerdo	Desacuerdo

Anexo H: Implementación mediante un contenedor en Docker

Primero se debe crear un archivo sin extensión en la raíz donde se encuentra la aplicación



En dicho archivo Docker vamos a configurarlo mediante el siguiente código

```
Dockerfile
1 # Usa una imagen de Python como base
2 FROM python:3.9
3
4 # Establece el directorio de trabajo en /run
5 WORKDIR /run
6
7 # Copia el archivo de requisitos al contenedor
8 COPY requirements.txt .
9
10 # Instala las dependencias necesarias
11 RUN pip install --no-cache-dir -r requirements.txt
12
13 # Copia el contenido actual del directorio al contenedor en /run
14 COPY . .
15
16 # Reemplaza la línea de código en flask_uploads.py
17 RUN sed -i 's/from werkzeug import secure_filename, FileStorage/from werkzeug.utils import secure_filename\nfrom werkzeug.datastructures'
18
19 # Expón el puerto 6000
20 EXPOSE 5000
21
22 # Comando para ejecutar tu aplicación cuando el contenedor se inicia
23 CMD ["python", "run.py"]
24
```

Se configura en el archivo app.py para que inicie en el puerto 5000

```
run.py
1 from shop import app
2
3 if __name__ == "__main__":
4     app.run(host="0.0.0.0", port=5000, debug=True)
```

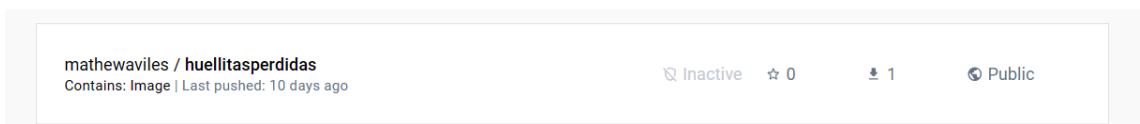
Después se ingresa el siguiente comando para crear la imagen Docker

```
PS C:\Users\user11\Documents\tesis\HuellitasFinalDeployment>
PS C:\Users\user11\Documents\tesis\HuellitasFinalDeployment> docker build -t huellitasperdidas .
>> docker run -p 5000:5000 huellitasperdidas
```

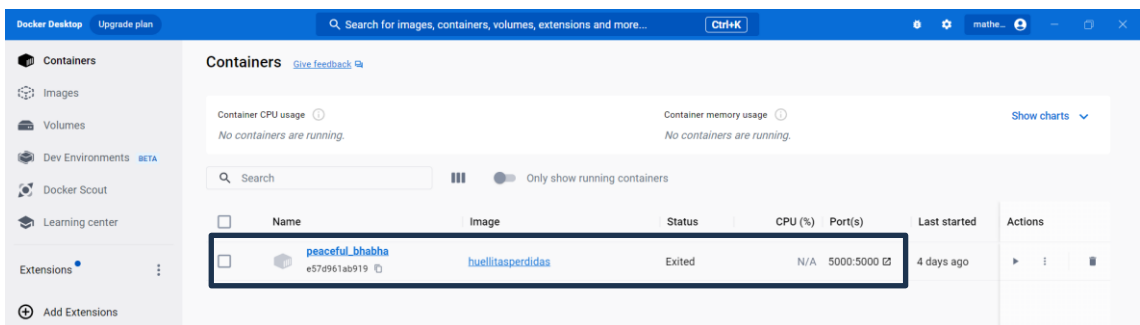
Por último, se realiza un push para subir la imagen a Dockerhub

```
PS C:\Users\user11\Documents\tesis\HuellitasFinalDeployment>
PS C:\Users\user11\Documents\tesis\HuellitasFinalDeployment> docker push mathewaviles/huellitasperdidas:huellitasperdidas
```

Entramos a Dockerhub para ver que nuestra imagen se encuentra subida



En la aplicación de Imagen Docker de Windows se puede observar que se ha creado la imagen





ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO



DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE

UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 08/01/2023

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: Mathew Patricio Avilés Chávez
INFORMACIÓN INSTITUCIONAL
Facultad: Informática y Electrónica
Carrera: Software
Título a optar: Ingeniero en Software
f. Analista de Biblioteca responsable: Ing. Fernanda Arévalo M.

