REVISTA
ODIGOS

Período octubre 2020 - enero 2021
Vol. 1, Núm. 3
rodigos@uisrael.edu.ec
https://doi.org/10.35290/ro.v1n3.2020.372 • e-ISSN: 2697-3405

Prototipo móvil para la geolocalización de mascotas callejeras

Fecha de recepción: 2020-08-27 • Fecha de aceptación: 2020-09-24 • Fecha de publicación: 2020-10-10

Esteban Alejandro Burbano Ulloa [1]
Sherwin Williams, Ecuador
teban_burb@hotmail.com
https://orcid.org/0000-0001-5180-7486

Mayra Alexandra Constante Molina [2]
Centro Educativo "Marqués de la Fayette", Ecuador
alexa_constant@hotmail.com
https://orcid.org/0000-0003-2661-058X

Lesly Maribel Hidalgo Guamán [3]
Fundación Emprender el Futuro, Ecuador
lesly.hidalgo@hotmail.com
https://orcid.org/0000-0002-4466-8411

Fernando Alexander Moya Chiluiza [4]
Edwin Vargas Consultores, Ecuador
fernandoalex_92@hotmail.com
https://orcid.org/0000-0001-6188-4928

RESUMEN

En Ecuador existen fundaciones que se encargan de velar por los derechos de los animales, además de la promoción sobre la tenencia responsable de mascotas; sin embargo, esto no es suficiente, ya que estas organizaciones se auto sustentan y muchas veces no dan abasto con los animales que tienen a cargo. Con la creación de la ordenanza 048 se establecieron penalizaciones para controlar el problema de fauna urbana, tomando en consideración aquellos actos que ponen en riesgo la vida de los animales, y que a su vez afectan la sociedad. Con el uso de las tecnologías esta problemática se ha hecho mucho más evidente en las redes sociales, por lo que el presente trabajo propone la creación de una aplicación móvil que permita en tiempo real identificar y ubicar perros callejeros

en diferentes sectores de la ciudad de Quito, para que los organismos encargados puedan tener un registro actualizado. Esta app funcionaria mediante la georreferenciación por medio del uso de un sensor de geolocalización, obteniendo latitud y longitud que permita alimentar la base de datos.

PALABRAS CLAVE: tenencia responsable, desarrollo de aplicaciones, geolocalización, fauna urbana, mascotas callejeras.

ABSTRACT

In Ecuador there are foundations that are in charge of looking after the rights of animals, as well as promoting responsible pet ownership. However, this is not enough, as these organizations are self-sustaining and many times cannot cope with the animals they have in their care. With the creation of ordinance 048, penalties were established to control the problem of urban fauna, taking into consideration those acts that put the lives of animals at risk, and that in turn affect society. With the use of technology this problem has become much more evident in social networks, so this work proposes the creation of a mobile application that allows in real time to identify and locate stray dogs in different sectors of the city of Quito, so that the agencies in charge can have an updated record. This app will work by means of georeferencing through the use of a geolocation sensor, obtaining latitude and longitude that will allow feeding the database.

KEYWORDS: responsible tenure, applications development, geolocation, urban fauna, stray dogs

Introducción

En los últimos años el cuidado hacia los animales ha sido más evidente, pudiendo ser el uso de redes sociales lo que ha ayudado a difundir información para concientizar y conocer más detalles sobre la responsabilidad que existe cuando se decide incluir a una mascota en el hogar (Calderón Pazmiño, 2016)from USD 2.27 millions to USD 12,68 millions. (Banco Central del Ecuador, 2014; sin embargo, a pesar de las campañas por parte de organizaciones locales, extranjeras o información que circula en Internet, la presencia de animales callejeros aún es un problema social que existe en la ciudad de Quito.

En el 2011 se creó la ordenanza 048 (Municipio de Quito) para tratar de mitigar esta problemática, por lo que establece que “mantener un número mayor de animales de compañía al que le permita cumplir satisfactoriamente con las normas de bienestar animal” se considera una infracción grave; así como “no mantener animales de compañía dentro de su domicilio con las debidas seguridades, o dejarlos transitar por espacios públicos o comunitarios, sin la compañía de una persona responsable del animal, a fin de evitar situaciones de peligro tanto para las personas como para el animal”. Estas faltas generan una multa que va desde el 45% al 90% de una remuneración básica unificada.

Con el trabajo conjunto de organizaciones, fundaciones y el Gobierno se trata que el problema de animales callejeros disminuya, pero las cifras siguen en aumento, según las proyecciones poblacionales del Instituto Nacional de Estadística y Censos (INEC) para Quito y las estimaciones de los censos, en el 2013 había 41.676 perros callejeros y para el 2018 la población sería de alrededor de 122.280 (Camacho Pardo & Albornoz Naranjo, 2018). Lo que se convierte en una problemática que afecta tanto la vida de los animales, como la salud de las personas, ya que los animales que viven en la calle se ven obligados a buscar comida en la basura, lo que es perjudicial para todos, debido a la transmisión de bacterias y virus. Además, esto desencadena que la población de animales en la calle aumente, ya que estos no pueden ser fácilmente sometidos a procesos de esterilización.

En Ecuador ya se han realizado iniciativas para controlar este mal que afecta a toda la sociedad, según Del Castillo Garzón & Ordoñez Charpentier (2020) en el presente año realizan un estudio que evidencia los motivos del abandono y sobrepoblación indiscriminada de animales domésticos en las calles de la ciudad de Quito, y mencionan ciertas propuestas y soluciones para poder afrontar esto. Por otro lado, Basantes Serrano & Noboa Jaramillo (2016) realizaron un análisis legal referente al maltrato y abandono de animales domésticos, donde se determina que, en el Distrito Metropolitano de Quito no existe tutela judicial efectiva para los animales de compañía en caso de maltrato o muerte, también se ha podido identificar la poca o nula importancia que dan las entidades gubernamentales competentes.

Por su parte, Falconí et al., (2013) iniciaron la campaña “Anímate no tengo raza, acógeme en tu casa”, que se promovió como una campaña de responsabilidad social que busca incentivar la adopción canina, resaltando que este procedimiento es un aporte para disminuir la problemática de animales callejeros.

De acuerdo a lo mencionado, el presente trabajo propone el desarrollo de una aplicación móvil para dispositivos Android que permita obtener la ubicación de mascotas callejeras en tiempo real y notificar a alguna institución para que se haga cargo, esto contribuirá a mitigar la situación actual de animales en la calle, ya que se tendrá un dato real para que los organismos de control puedan gestionar sus planes sobre datos verídicos.

Metodología

En esta sección se detalla el proceso de construcción de la aplicación móvil, por lo que el primer paso es definir como se realiza el proceso actual y el que será propuesto en este trabajo.

La *Figura 1* muestra el proceso que actualmente se realiza para evidenciar la situación de calle de los perros. Esta se realiza mediante la toma de una fotografía que posteriormente será subida en las redes sociales, esto no permite un registro apropiado de todos los animales que se encuentran en esta situación, lo que dificulta tomar acción sobre esta problemática.

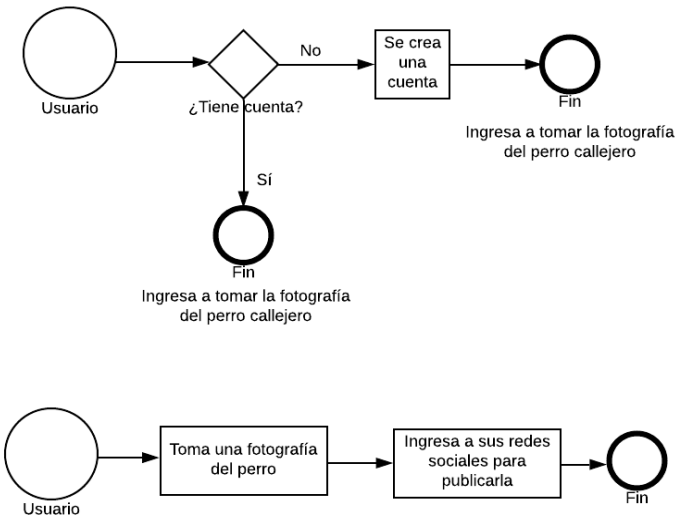

Figura 1. *Proceso actual - no automatizado*

Fuente: elaboración propia

Mientras que en la *Figura 2* se describe cómo sería el proceso automatizado con la aplicación propuesta. En esta el usuario debe ingresar a la aplicación, tomar una fotografía que será almacenada junto a una breve descripción, **más su ubicación (latitud, longitud) quedando** así un registro grabado en la base de datos en tiempo real Firebase (Moroney, 2017).

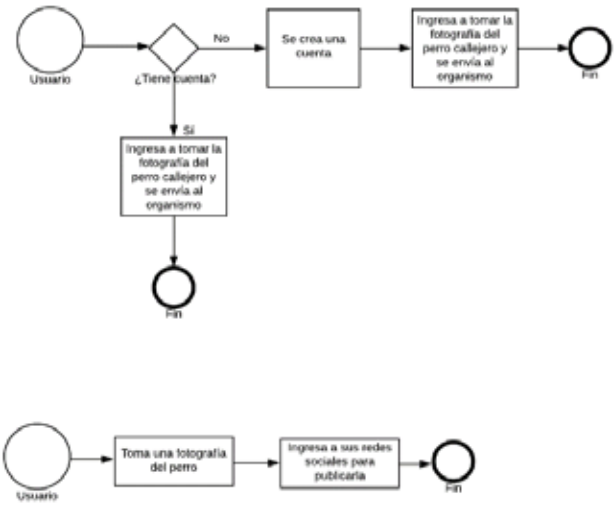

Figura 2. *Proceso actual automatizado*
Fuente: elaboración propia

Para el desarrollo de la aplicación móvil se planteó lo siguiente:

- Simplicidad

La funcionalidad de la aplicación debe ser sencilla para el fácil manejo del usuario, implica también que el diseño de las interfaces sea amigable y comprensible. La aplicación debe ser de rápida respuesta para que los usuarios puedan utilizarla en cualquier momento y circunstancia (Medina et al., 2016).

- Programación

El uso de Android Studio se eligió por su capacidad de ejecución en varios dispositivos del mercado, editor inteligente de código, emulador de funciones, integración de marcos de desarrollo y por los bajos costos que implica un desarrollo para Android. Se facilita la ejecución de pruebas y es un IDE amigable para la solución de errores (Desarrolladores de Android, 2020).

- Diseño Adaptable

Al ser una aplicación para una población extensa, es menos probable que se puedan controlar los dispositivos en los que se va a utilizar la aplicación, por lo que es importante que se adapte a cualquier tamaño de pantalla, cualquier marca de dispositivo y que no tenga errores de compatibilidad al momento de ejecución, por lo cual se desarrollara con la API 16 de Android la cual es el sistema operativo Jelly Bean 4.1 o superior para que funcione en la mayor cantidad de dispositivos (Medina et al., 2016).

En cuanto a los requisitos, se plantea una lista de requerimientos funcionales y no funcionales mostrados a continuación:

Funcionales

RF01: Con respecto al inicio de sesión, la aplicación permitirá autenticar al usuario mediante el envío de las credenciales de este al servidor, el cual se encargará en último momento de la autenticación y responderá a la aplicación afirmativamente si las credenciales son correctas o negativamente, dependiendo el caso. La aplicación ha de ser capaz de diferenciar entre los dos casos.

RF02: permitirá tomar y guardar las imágenes con la cámara del dispositivo

RF03: obtendrá las coordenadas del lugar donde el dispositivo se encuentra, existiendo consentimiento por parte del usuario. Además, se mostrará la localización obtenida al usuario mediante un mapa en tiempo real.

No funcionales

RNF04: la aplicación se deberá recuperar de los errores siempre y evitará que se cierre inesperadamente siempre que sea posible, ofreciendo al usuario mensajes de error e información de lo que ha ocurrido.

RNF05: tendrá que ser capaz de mantener la información introducida por el usuario y poder recuperarse de posibles cambios sin perder esta. Estos posibles cambios incluyen, entre otros, el cambio de orientación del dispositivo o llamadas entrantes durante el uso de la aplicación.

RNF06: deberá funcionar correctamente en diferentes dispositivos Android, abarcando el máximo número de versiones del sistema operativo. Las interfaces además han de poder adaptarse a diferentes tamaños de pantalla

Con este antecedente, para el proyecto se utilizó la metodología Scrum (Kuz et al., 2018), para lo que se utilizó la herramienta Trello (Kaur, 2018) que es una aplicación que permite gestionar las actividades que se realizan durante el proyecto mediante tarjetas, lo que facilita la comunicación y el avance progresivo del proyecto (ver *Figura 3* y *Figura 4*).

Al pasar el tiempo, la industria del desarrollo se ha dado cuenta que esta metodología funciona muy bien para la entrega de proyectos que se están realizando porque se puede observar claramente cuáles son los puntos más débiles y en los que se tarda más tiempo al momento de desarrollar un programa, esto facilita que se pueda resolver lo antes posible dichos contratiempos, para así poder entregar en la fecha establecida lo que se está realizando.

Para el control de código se utilizó Git, con el cliente GitHub (Blischak et al., 2016), esta es una

plataforma de desarrollo colaborativa que permite alojar proyectos utilizando un sistema de control de versiones Git (ver *Figura 5*). En esta plataforma un grupo de desarrolladores pueden aportar de manera ordenada en el proyecto sin dañar las versiones funcionales que se van generando durante el desarrollo.

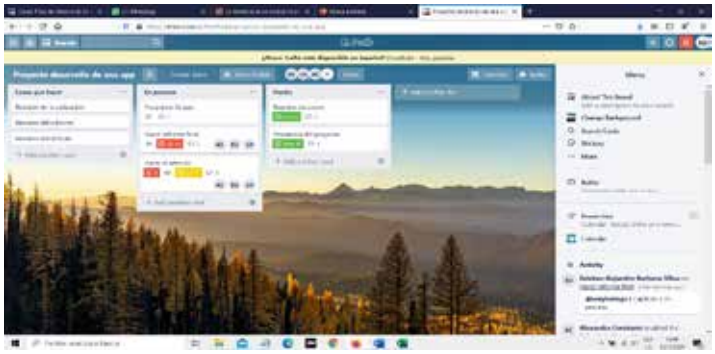

Figura 3. *Plataforma de Trello*
Fuente: elaboración propia

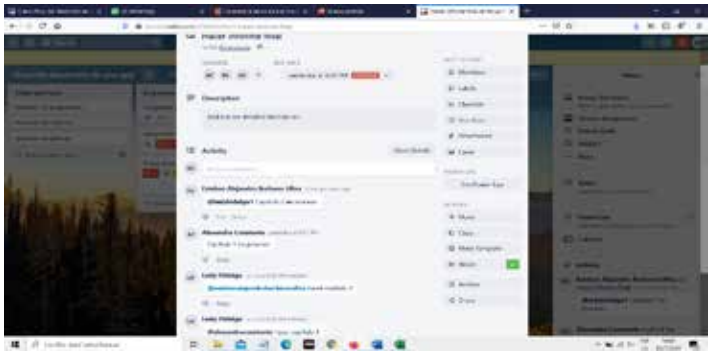

Figura 4. *Lista de actividades de Trello*
Fuente: elaboración propia

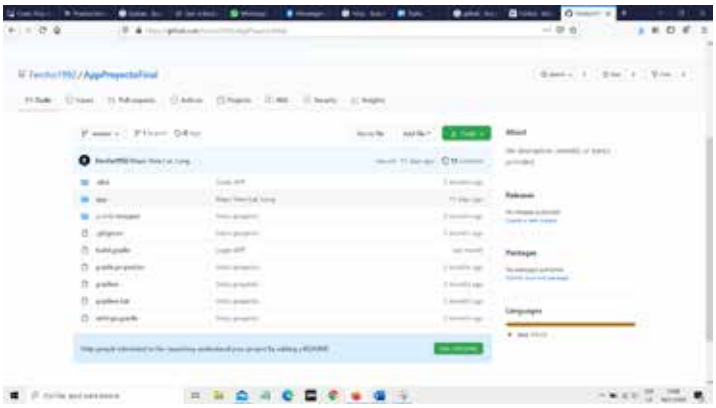

Figura 5. *Plataforma de GitHub*
Fuente: elaboración propia

1. Base de datos

La base de datos que se utiliza para realizar el aplicativo es FireBase (ver *Figura 6*). En esta los objetos se almacenan como objetos JSON, se actualiza como un árbol JSON que se encuentra alojado en la nube. Cuando se agregan datos al árbol se convierten en un nodo dentro del estructura Jason. Esta base de datos funciona en tiempo real y aloja los datos en la nube permitiendo acceder a ellos en cualquier momento y desde cualquier lugar (Moroney, 2017).

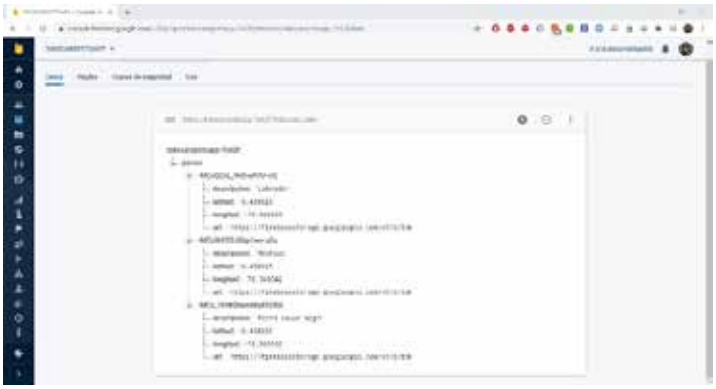

Figura 6. *Plataforma de FireBase*
Fuente: elaboración propia

2. Diseño de interfaces

En el diseño de las interfaces se realizaron en primera instancia Mockup para tener una idea de cómo se quiere que se vea el diseño de la aplicación.

La *Figura 7* muestra un Mockup donde se realizó el *login* en el cual se requiere que exista dos campos en los cuales se puede ingresar el nombre de usuario y la contraseña.

Ya en la *Figura 8* se ve el segundo Mockup que es el menú de opciones, en el cual se va a poder elegir si se desea subir una nueva fotografía o buscar algún perro dentro de la base de datos.

Figura 7. *Login con Mockup*
Fuente: elaboración propia

Figura 8. *Menú de opciones*
Fuente: elaboración propia

El tercer Mockup se describe en la *Figura 9*, aquí se realiza del ingreso de un nuevo perro, en el cual tiene que haber un lugar donde se pueda ingresar la fotografía y la descripción para subirlo a la base de datos.

El cuarto Mockup se muestra en la *Figura 10*, con la lista de avistamientos, es decir, dónde se va a poder ver los diferentes perros que se han subido en la base de datos.

Figura 9. *Nuevo ingreso*
Fuente: elaboración propia

Figura 10. *Lista de perros callejeros*
Fuente: elaboración propia

Y en la *Figura 11* se evidencia la información del desarrollo de la aplicación.

Figura 11. *Información de la aplicación*
Fuente: elaboración propia

3. Programación de la aplicación

En cuanto a los estándares de programación que se utilizaron en el desarrollo son los siguientes:

- Estilo de escritura CamelCase
- Lenguaje de programación Java
- Ide de desarrollo Android Studio 3.6.2
- Base de datos en tiempo real: FireBase
- Metodología Kanban

Para poder implementar el proyecto tendremos que subir la aplicación a la tienda virtual de Android, es decir al Play Store, y podrá funcionar en teléfonos con sistema operativo Jelly Bean 4.1 en adelante.

Los requerimientos de Hardware y Software que se requieren para la implementación se detallan en la *Tabla 1*:

Tabla 1.

Requerimientos Hardware y Software

Requerimiento	Detalle
Software para servidor	FireBase PostMan Xampp Google Maps
Software para usuario	Api 16: Android Jelly Bean 4.1 Google Maps
Hardware para servidor	RAM de 8Gb Procesador Core I7 Tarjeta de Red 1Gbps
Hardware para usuario	RAM al menos de 1Gb Cámara Sensor de Geolocalización Memoria disponible de 10m

Fuente: elaboración propia

La aplicación se desarrolló en base al modelo MVC. En la *Figura 12* se puede ver el Modelo (Base de datos Firebase).

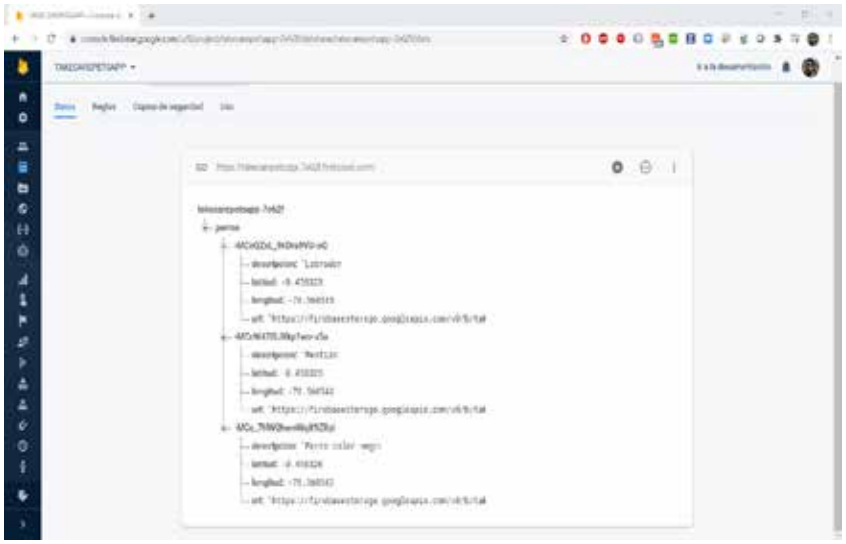

Figura 12. *Firebase*
Fuente: elaboración propia

La vista son archivos xml generados en Android y se muestran en la *Figura 13* y *Figura 14*.

Figura 13. *Vista del login*
Fuente: elaboración propia

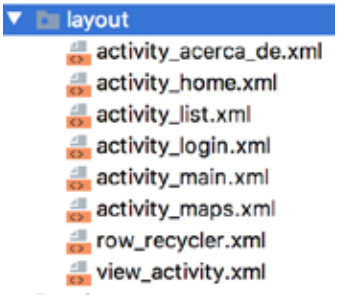

Figura 14. *Archivos xml*
Fuente: elaboración propia

Mientras que para el controlador son las Clases en java (ver *Figura 15* y *Figura 16*).

Figura 15. *Clase del login*
Fuente: elaboración propia

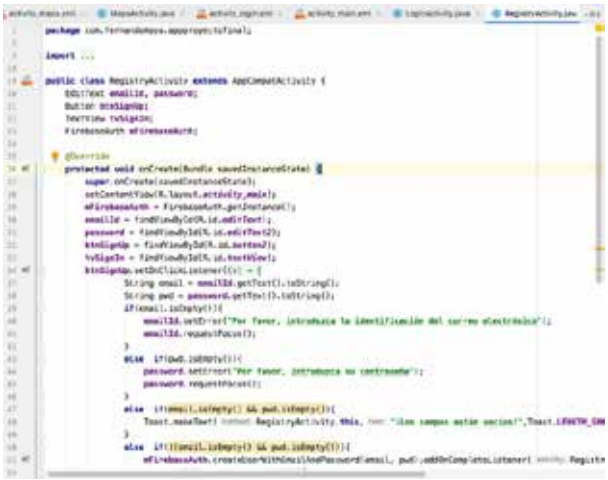

Figura 16. *Clase del registro*
Fuente: elaboración propia

Resultados

Se realizaron pruebas funcionales dado que se basa en los requisitos del sistema, con esto se pueden validar y verificar si el programa funciona correctamente y de manera óptima para tener una idea de la calidad que se aplicó al momento de desarrollar el programa.

En las siguientes *Figuras 17, 18, 19 y 20* se puede observar en funcionamiento el código cuando el usuario decide guardar un nuevo registro, además de observar lo que sucede en la base de datos Firebase.

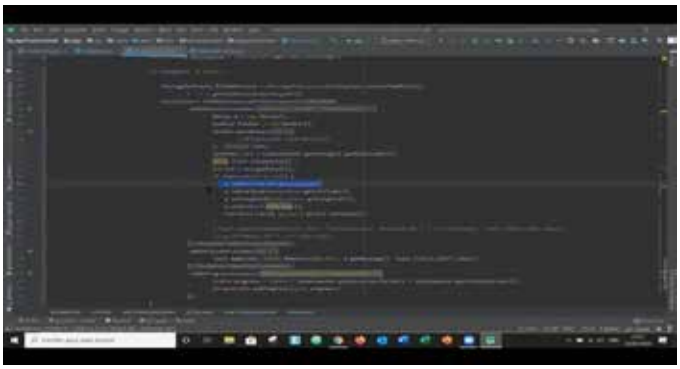

Figura 17. *Código aplicación*
Fuente: elaboración propia

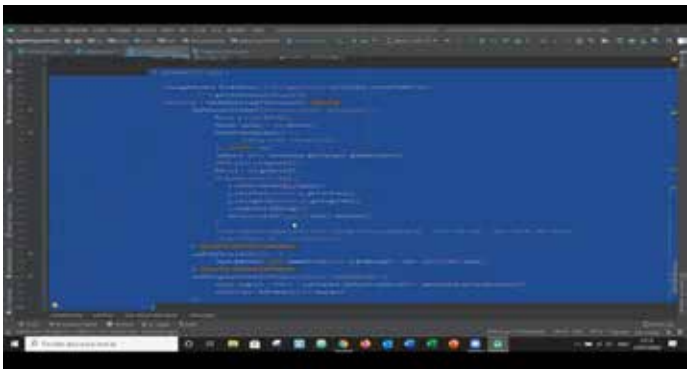

Figura 18. *Código aplicación*
Fuente: elaboración propia

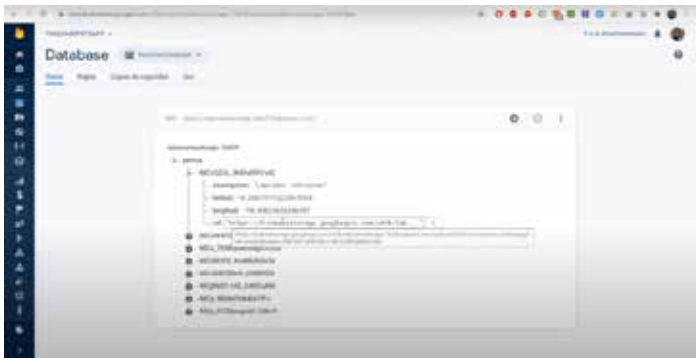

Figura 19. *Base de datos Firebase*
Fuente: elaboración propia

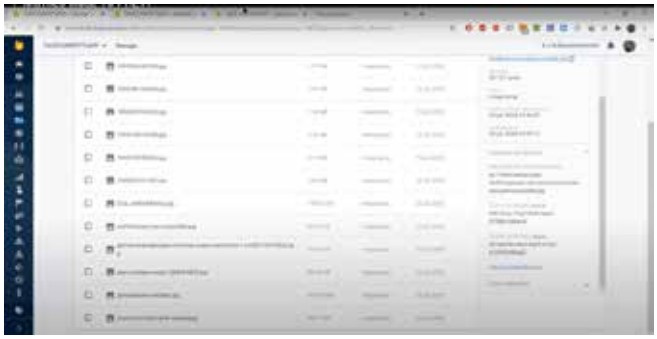

Figura 20. *Base de datos Firebase*
Fuente: elaboración propia

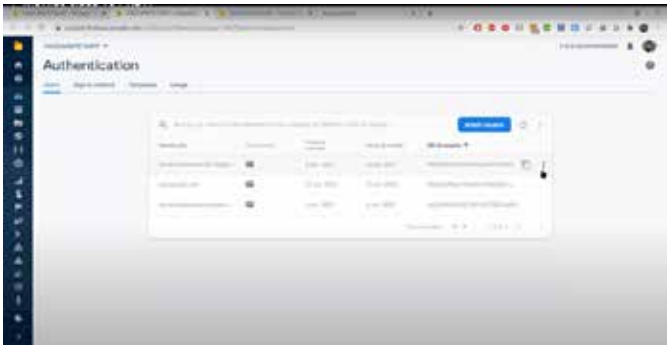

Figura 21. *Base de datos Firebase*
Fuente: elaboración propia

A su vez, se realizaron pruebas de mono para de esta manera poder observar si es que el programa tenía algún inconveniente con la nevera navegación en diferentes direcciones dentro del programa para observar si se encuentran fallos o inconsistencias al momento en que el usuario se equivoque al ingresar en una ventana o dirección diferente.

Los resultados que se obtuvieron en esta prueba fueron alentadores, dado que no se presentó ningún inconveniente durante la navegación en diferentes direcciones, lo cual nos permite asumir que el programa es adecuado para los usuarios.

En las siguientes *Figuras 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 y 32* se puede observar como un usuario interactúa en la aplicación.

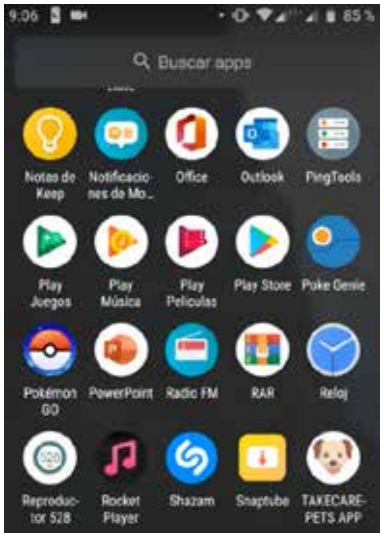

Figura 22. *Aplicación ambiente celular*
Fuente: elaboración propia

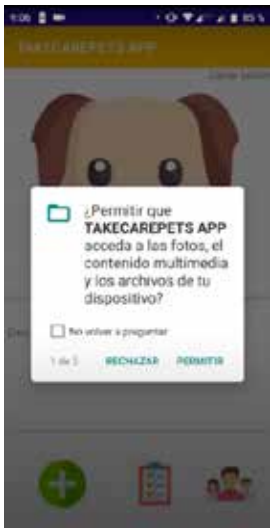

Figura 23. *Aplicación solicitando acceso al dispositivo*
Fuente: elaboración propia

Figura 24. *Primera imagen de ingreso*
Fuente: elaboración propia

Figura 25. *Toma de fotografía desde la app*
Fuente: elaboración propia

Figura 26. *Adjunto de la fotografía tomada*
Fuente: elaboración propia

Figura 27. *Escritura de la descripción*
Fuente: elaboración propia

Figura 28. *Ingreso de lo descrito*
Fuente: elaboración propia

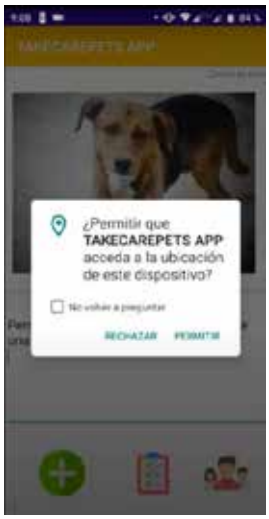

Figura 29. *Permitir que la aplicación acceda a la ubicación*
Fuente: elaboración propia

Figura 30. *Agregado con éxito*
Fuente: elaboración propia

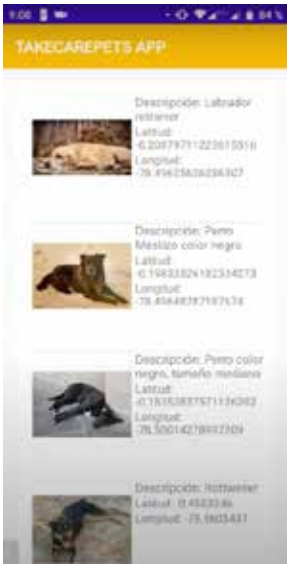

Figura 31. *Listado de animales en situación de calle*
Fuente: elaboración propia

Figura 32. *Vista de un animal callejero desde el mapa*
Fuente: elaboración propia

Conclusiones

Tras la investigación realizada se puede apreciar que en la ciudad de Quito el tema de los perros callejeros es una problemática que afecta a los ciudadanos, debido a que las herramientas que utilizan para mitigar esta situación no son viables. En su mayoría los usuarios reportan animales en situación de calle desde las redes sociales.

La creación de esta aplicación permite la localización exacta mediante la georreferenciación,

obteniendo latitud y longitud del lugar donde se encuentre el animal y con la toma de una fotografía se definirán las coordenadas y datos específicos para su búsqueda.

El establecimiento de los requerimientos funcionales y no funcionales fue importante ya que se tuvo una idea general del funcionamiento que tiene la aplicación. El uso de Android Studio para la construcción de la aplicación facilitó su desarrollo, ya que es un entorno que ofrece variedad de opciones que son aplicables para cualquier proyecto y debido a la compatibilidad puede ser ejecutado en varios dispositivos diferentes.

En referencia a la implementación de esta aplicación, esta hará que los usuarios obtengan la ubicación en tiempo real de perros callejeros facilitando su rápida búsqueda y evitando información inexacta.

Durante todo el desarrollo se realizaron pruebas del funcionamiento de la aplicación que permitieron hacerle mejoras para obtener un óptimo resultado final, ya que los usuarios podrán descargarse y hacer uso correcto de ella aprovechando las funcionalidades que tiene como herramienta de ayuda.

Referencias

Basantes Serrano, G. T., & Noboa Jaramillo, I. V. (2016). *Tutela judicial a los animales de compañía en caso de maltrato y muerte en el Distrito Metropolitano de Quito*. Quito: Universidad de las Américas, 2016. http://dspace.udla.edu.ec/handle/33000/6347

Blischak, J. D., Davenport, E. R., & Wilson, G. (2016). A Quick Introduction to Version Control with Git and GitHub. *PLOS Computational Biology*, *12*(1), e1004668. https://doi.org/10.1371/journal.pcbi.1004668

Calderón Pazmiño, M. A. (2016). *Plan de negocios para la creación de un centro de cuidado integral para mascotas en la ciudad de Quito*. Quito: Universidad de las Américas, 2016. http://dspace.udla.edu.ec/handle/33000/5521

Camacho Pardo, M. A., & Albornoz Naranjo, O. P. (2018). *Caracterización demográfica de la población canina callejera en 6 mercados del sur de Quito*. Quito: Universidad de las Américas, 2018. http://dspace.udla.edu.ec/handle/33000/8864

Del Castillo Garzón, C. R., & Ordoñez Charpentier, R. E. (2020). *Animales de la calle en Quito: propuestas y soluciones*. Quito: Universidad de las Américas, 2020. http://dspace.udla.edu.ec/handle/33000/882

Desarrolladores de Android. (2020). *Developer*.android https://developer.android.com/?hl=es

Falconí, G., Guerrero Freire, A. J., & Tigreros Quintero, P. A. (2013). *Camapaña de mercadeo social anímate no tengo raza, acógeme en tu casa*. Quito, 2013. http://repositorio.usfq.edu.ec/handle/23000/2142

Kaur, A. (2018). App Review: Trello. *Journal of Hospital Librarianship*, *18*(1), 95–101. https://doi.org/10.1080/15323269.2018.1400840

Kuz, A., Falco, M., & Giandini, R. S. (2018). Comprendiendo la Aplicabilidad de Scrum en el Aula: Herramientas y Ejemplos. *Revista Iberoamericana de Tecnología En Educación y Educación En Tecnología*, *21*(21), e07. https://doi.org/10.24215/18509959.21.e07

Medina, O. C., Marciszack, M. M., & Groppo, M. A. (2016). Trazabilidad y validación de requerimientos funcionales de sistemas informáticos mediante la transformación de modelos conceptuales - Traceability and validation for functional requirements of information systems using conceptual model transformation. *Revista Electrónica de Computación, Informática y Electrónica RECIBE* , *5*(1). http://recibe.cucei.udg.mx/ojs/index.php/ReCIBE/article/view/53

Moroney, L. (2017). Firebase Cloud Messaging. In *The Definitive Guide to Firebase* (pp. 163–188). Apress. https://doi.org/10.1007/978-1-4842-2943-9_9

Municipio de Quito. (2011). *Ordenanza Municipal 048*. http://www7.quito.gob.ec/mdmq_ordenanzas/Ordenanzas/ORDENANZAS MUNICIPALES 2011/ORDM-0048 TENENCIA, PROTECCIÓN Y CONTROL DE FAUNA URBANA.pdf