

“HiDog” Aplicación Android para la gestión de tus mascotas

Sergi Comeche Nolla

Máster universitario de Desarrollo de aplicaciones para dispositivos móviles

Eduard Martín Lineros

7/06/2017

© Sergi Comeche Nolla

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	"HiDog" Aplicación Android para la gestión de tus mascotas
Nombre del autor:	Sergi Comeche Nolla
Nombre del consultor:	Eduard Martín Lineros
Fecha de entrega (mm/aaaa):	06/2017
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Esta aplicación pretende cubrir las necesidades de un colectivo concreto de usuarios, así como hacer su día a día más sencillo y divertido.</p> <p>Con esta aplicación, se podrá interactuar con otros usuarios mediante un mapa sujeto a eventos del servidor, donde el usuario indicará en qué zona sacará su perro a pasear, y cuánto tiempo estará. De la misma forma, podrá consultar ubicaciones de otros perros cercanos para coincidir con ellos.</p> <p>Además, se dispondrá de un creador y gestor de perfiles, para añadir toda la información que se desee sobre tus mascotas.</p> <p>Y se podrá añadir "mascotas amigas" en la agenda para ver sus perfiles y consultar su último paseo.</p> <p>Por último, se implementarán notificaciones para avisar de la actividad de los "perros amigos".</p> <p>Como funcionalidad extra, se pueden añadir "lugares preferidos", con valoraciones de otros usuarios y su ubicación para dar a conocer nuevos sitios donde salir a pasear con tu mascota.</p>	
Abstract (in English, 250 words or less):	
<p>This application is intended to cover the needs of a specific group of users, as well as making their daily life easier and funnier.</p>	

With this application, you can interact with other users through a map subject to events of the server, where the user will indicate in which area will take his dog for a walk, and how long they will be. Likewise, you can check locations of other nearby dogs to match them.

In addition, you will have a creator and profile manager, to add all the information you want on your pets.

And you can add "pet friends" in the calendar to see their profiles and consult their last ride.

Finally, notifications will be implemented to warn of the activity of "friendly dogs".

As extra functionality, you can add "favourite places", with ratings of other users and their location to give you new places to go for a walk with your pet.

Palabras clave (entre 4 y 8):

Mascota, perro amigo, agenda, paseo, perfil.

Índice

Contenido

1. INTRODUCCIÓN	1
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	1
1.1.1 <i>Necesidades a cubrir</i>	1
1.1.2 <i>Por qué es un tema relevante?</i>	1
1.1.3 <i>Cómo se resuelve el problema de momento?</i>	2
1.1.4 <i>Qué resultado se quiere obtener?</i>	2
1.2 OBJETIVOS DEL TRABAJO	2
1.2.1 <i>Objetivos del proyecto</i>	2
1.2.2 <i>Objetivos personales</i>	3
1.3 ENFOQUE Y MÉTODO SEGUIDO	3
1.4 PLANIFICACIÓN DEL TRABAJO	3
1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS	6
1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA	7
2. DISEÑO	8
2.1 DISEÑO CENTRADO EN EL USUARIO (DCU)	8
2.1.1 <i>Usuarios y contexto de uso</i>	8
2.1.2 <i>Diseño conceptual</i>	8
2.1.3 <i>Prototipado</i>	10
2.1.4 <i>Evaluación</i>	16
2.2 DISEÑO TÉCNICO	17
2.2.1 <i>Definición de los casos de uso</i>	17
2.2.1.1 Diagrama UML de actores y flujo	17
2.2.1.2 Listado de los casos de uso	17
2.2.2 <i>Diseño de la arquitectura de la aplicación</i>	20
2.2.2.1 Diagrama UML correspondiente al diseño de la base de datos	20
2.2.2.2 Diagrama UML correspondiente al diseño de las entidades y clases	21
2.2.2.3 Diagrama explicativo de la arquitectura del sistema	22
3. IMPLEMENTACIÓN	23
2.1 TECNOLOGÍAS	23
2.2 HERRAMIENTAS	23
2.3 DETALLES DE LA IMPLEMENTACIÓN	23
1.2.1 <i>Estructura</i>	23
1.2.1.1 Estructura de clases	26
1.2.1.2 Estructura de Firebase	28
1.2.1.3 Estructura de interfaces	30
1.2.2 <i>Implementación de las funcionalidades</i>	30
1.2.2.1 Funcionalidad My Pets	31
1.2.2.2 Funcionalidad My Friends	33
1.2.2.3 Funcionalidad Walking Time	34
1.2.2.4 Otras funciones	36
4. PRUEBAS	38
4.1 PRUEBAS CON TERMINALES VIRTUALES	38
4.2 PRUEBAS CON TERMINALES REALES	39
5. CONCLUSIONES	42
6. BIBLIOGRAFÍA	44

Lista de figuras

FIGURA 1.1 DESCRIPCIÓN ETAPAS Y TAREAS DEL PROYECTO	4
FIGURA 1.2 DIAGRAMA DE GANTT DEL PROYECTO	5
FIGURA 1.3 LEYENDA DEL DIAGRAMA DE GANTT	5
FIGURA 1.4 DIAGRAMA DE GANTT COMPLETO	5
FIGURA 2.1 PANTALLA PROTOTIPO: "LOGIN"	10
FIGURA 2.2 PANTALLA PROTOTIPO: "MENÚ PRINCIPAL"	11
FIGURA 2.3 PANTALLA PROTOTIPO: "MI PERFIL"	12
FIGURA 2.4 PANTALLA PROTOTIPO: "VER MAPA"	13
FIGURA 2.5 PANTALLA PROTOTIPO: "VER MAPA LONG CLICK"	13
FIGURA 2.6 PANTALLA PROTOTIPO: "MIS AMIGOS"	14
FIGURA 2.7 PANTALLA PROTOTIPO: "AJUSTES"	15
FIGURA 2.8 PANTALLA PROTOTIPO: "MENÚ DESPLEGABLE"	16
FIGURA 2.9 PANTALLA PROTOTIPO: "NOTIFICACIONES"	16
FIGURA 2.10 DIAGRAMA UML ACTORES Y FLUJO	17
FIGURA 2.11 DIAGRAMA UML BASES DE DATOS	20
FIGURA 2.12 DIAGRAMA UML ENTIDADES Y CLASES.....	21
FIGURA 2.13 DIAGRAMA MVC ARQUITECTURA DEL SISTEMA	22

1. Introducción

1.1 Contexto y justificación del Trabajo

1.1.1 Necesidades a cubrir

Los dueños de mascotas, en especial de perros, sacan a pasear diariamente sus animales, y se encuentran a menudo con falta de tiempo para encontrar otros perros para que éstos puedan jugar. Esto provoca un desorden en la actividad física del animal, y también en su “vida social”, lo que se traduce en muchos casos en destrozos en el hogar debido a su gran energía no consumida, o peleas con otros perros a causa de su poca interacción.

Con esta app, podrás consultar el mapa donde otros usuarios habrán señalado su ubicación, y sabrás donde llevar a tu perro para que juegue y cuánto tiempo estarán “sus amigos”.

También es necesario llevar un control de sus vacunas y su peso, o simplemente tener un perfil con toda la información que se crea conveniente sobre la mascota, para poder consultarla en el momento de llevarlo al veterinario, o en compañía de amigos para ver algunas fotos suyas, por ejemplo. Todo esto será posible con el gestor de perfiles que incorporará la aplicación.

Gracias a estos perfiles, también se dispondrá de una agenda donde se almacenarán los “perros amigos” que el usuario quiera añadir, y donde se podrán consultar sus datos públicos y sus imágenes. Con esta funcionalidad se agruparán los amigos de nuestros animales y nos llegarán notificaciones en cuánto lleven a cabo alguna actividad de interés (como salir a pasear).

Además, será posible ahorrar búsquedas por Internet y dudas sobre qué sitios son ideales para llevar a tus mascotas, o donde frecuentan otras personas para sacar a pasear a sus perros, si se lleva a cabo la funcionalidad extra de “lugares preferidos”, donde se podrá ver ubicación y valoraciones de otros usuarios de distintos lugares.

1.1.2 Por qué es un tema relevante?

- Me parece un tema de interés actual, puesto que mucha gente adopta una mascota y no dispone del tiempo ni las herramientas para gastar la enorme energía que tienen y conseguir que se lo pasen bien y nos permitan a nosotros lidiar con nuestras actividades y tareas diarias.
- Saber dónde encontrarás otros perros antes de salir de casa sería útil para dirigirse directamente allí y aprovechar el tiempo justo que estarás a la vez que tu mascota juega y se lo pasa bien.
- Tener toda la información en tu móvil organizada y agrupada, en lugar de buscar los papeles de las vacunas, acordarte de algunos datos o buscar fotografías por la galería de fotos, es una herramienta práctica que facilitaría muchas situaciones.

1.1.3 Cómo se resuelve el problema de momento?

Si observas a la mayoría de dueños que sacan a pasear a sus perros, verás que suelen tener el tiempo justo para que haga sus necesidades y poder volver a casa. Esto es perjudicial también a nivel social del animal, ya que se volverá más reacio a interactuar con otros animales con el paso del tiempo.

Para la gestión de datos de las mascotas, simplemente se usa la memoria o tener que consultar los papeles físicamente, con el tiempo que conlleva el hecho de buscarlos.

Por último, para consultar amigos o lugares frecuentados, hay que buscar fotografías en el teléfono móvil, o por Internet, hecho que no es tan práctico si se desea hacer múltiples veces.

Se han encontrado diversas aplicaciones con funciones relacionadas con perros, pero ninguna que reúna todo lo que se propone en esta app.

“Mascotas Bienvenidas” es una app que ofrece lugares donde los perros son bien recibidos.

“Rate my pets” es una especie de red social de mascotas, para interactuar con los dueños y chatear.

“Pet master pro” te permite gestionar toda la información de tu mascota, pero no es gratuita.

Ninguna de ellas ofrece las dos grandes opciones que se implementan en nuestra app. Por un lado, la gestión de los perfiles de todas nuestras mascotas, y por otra la localización para salir a pasear con perros “amigos” y saber en qué momento han salido, cuánto rato estarán, y poder añadir a tus “amigos” en una especie de agenda.

1.1.4 Qué resultado se quiere obtener?

Con esta aplicación se pretende ahorrar tiempo para la gestión de datos del animal, aprovechar el poco tiempo que se aplica en el paseo de éstos y agrupar información sobre lugares y amigos que permita una consulta más práctica en menos tiempo.

1.2 Objetivos del Trabajo

1.2.1 Objetivos del proyecto

- Desarrollar un plan de trabajo y cumplir sus plazos.
- Implementar una aplicación funcional para Android.
- Cubrir las necesidades del colectivo relacionado con las mascotas.
- Generar un vídeo de muestra para explicar el funcionamiento de la aplicación.

1.2.2 Objetivos personales

- Gestionar eventos de red con Firebase.
- Crear y gestionar una Base de Datos local.
- Desarrollar de principio a fin una aplicación para Android.
- Aprender a realizar pruebas una vez esté acabada la aplicación.

1.3 Enfoque y método seguido

Este proyecto trata de desarrollar un producto nuevo, en un sector donde ya hay aplicaciones similares, pero se considera que no agrupan todas las funciones que se pretende implementar, y que no son tan prácticas ni atractivas a nivel de usuario, por lo que las mejoras aplicadas pueden atraer a más usuarios.

La estrategia utilizada será la metodología ágil.

El desarrollo ágil se basa en los principios del manifiesto ágil y sus valores éticos, que tratan de dar más valor a algunos conceptos, pero sin dejar de lado los demás. Son los siguientes:

- 1) Dar más valor a los individuos y a sus interacciones que a los procesos y herramientas.
- 2) Dar más valor al software que funciona que a la documentación exhaustiva.
- 3) Dar más valor a la colaboración con el cliente que a la negociación contractual.
- 4) Dar más valor a la respuesta al cambio que al seguimiento de un plan.

En esta aplicación es de vital importancia adaptar las necesidades que vayan surgiendo por parte de los usuarios, para añadir funciones prácticas que les solucionen problemas cotidianos del día a día, por lo que los requisitos y soluciones van evolucionando con el tiempo según las necesidades encontradas.

Además, siguiendo el desarrollo ágil, se prioriza la entrega final sin errores en el producto, en lugar de terminar cada iteración rápido y justificar después su lanzamiento al mercado.

El desarrollo Waterfall no es una opción puesto que los requisitos y necesidades no están absolutamente fijados, y pueden cambiar durante el proceso.

Por último, el desarrollo rápido no se contempla ya que los plazos se han planificado para seguir un proceso completo sin excesiva prisa en la entrega, y teniendo en cuenta posible variaciones en algunas funcionalidades.

1.4 Planificación del Trabajo

En la figura 1.1 se observan las 4 etapas en las que se divide el proyecto, cada una con sus tareas correspondientes.

En total se han cuantificado 358 horas de trabajo, repartidas proporcionalmente entre sus etapas como se ve en la figura 1.1.

Teniendo en cuenta las horas que paso en el trabajo actual, tiempo de ocio y otras actividades, se han asignado 2 horas al día los días laborables y 6 horas los días festivos, marcados en la figura 1.2 de color gris.

Así, se han asignado las horas a cada tarea, y por consecuente a cada etapa, y el resultado es el diagrama de Gantt de la figura 1.2, con su leyenda correspondiente en la figura 1.3.

Por último, en la figura 1.4 podemos observar el diagrama de Gantt completo.

Etapas del proyecto	Horas	Estado	
Plan de Trabajo	40	Terminado	▼
Definir objetivos	12	Terminado	
Contextualizar el trabajo	2	Terminado	
Elegir estrategias a seguir	4	Terminado	
Planificación del trabajo	22	Terminado	
Diseño	66	Terminado	▼
Investigar usuarios potenciales, definir perfiles y analizar necesidades	16	Terminado	
Elaborar escenarios de uso	6	Terminado	
Definir flujos de interacción del sistema	16	Terminado	
Diseñar y construir un prototipo de la app	28	Terminado	
Implementación	186	En proceso	▼
Construir menú principal	10	En proceso	
Crear una interfaz básica de las principales funcionalidades	12	En proceso	
Implementar código de las principales funcionalidades	60		
Crear BD local para la función de "agenda de amigos"	14		
Crear y gestionar eventos de servidor para la función "salir a pasear"	34		
Añadir posibles funcionalidades extra	22		
Decorar y mejorar interfaz de usuario	16		
Realizar pruebas	18		
Entrega Final	66		▼
Depurar código final	22		
Repasar y actualizar la memoria	26		
Preparar vídeo de presentación del trabajo	14		
Realizar entrega final	4		

Figura 1.1

Figura 1.2

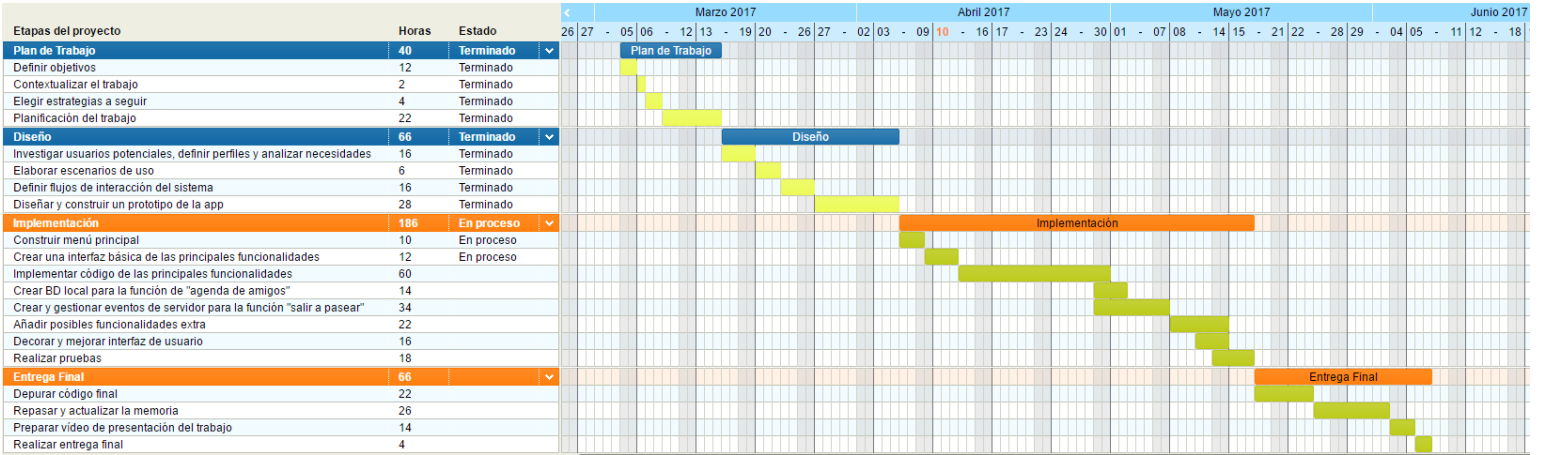
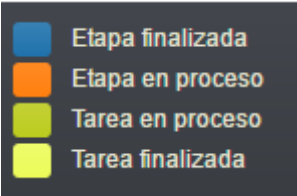


Figura 1.3

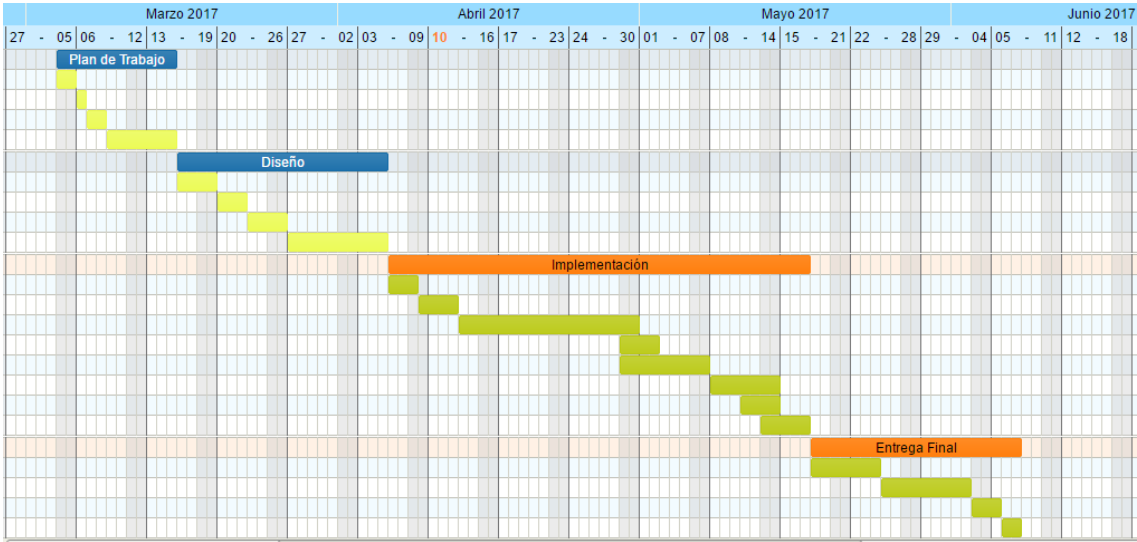


Figura 1.4

1.5 Breve sumario de productos obtenidos

El producto final obtenido en este proyecto será una aplicación Android, donde sus principales funciones serán:

La aplicación te permitirá registrarte y tener un perfil de usuario único, para gestionar todos los datos de tus mascotas.

Podrás añadir tantos perfiles como desees, y anotar toda la información que consideres relevante de tus perros.

Podrás acceder al mapa para ver a los perros que han salido a pasear y han indicado donde estarán paseando.

Dispondrás de un evento para marcar tu posición en el mapa, en el lugar donde vayas a salir con tu perro, para que los demás lo puedan ver.

Tendrás la opción de ampliar la información de los perros que aparecen en el mapa, así como añadirlos como “amigos”.

Recibirás notificaciones cuando un perro “amigo” salga a pasear, si así lo desees.

Tendrás una “agenda” con tus perros “amigos” para ver cuando han salido a pasear, y podrás acceder a su perfil si lo desees.

Dispondrás de un menú “ajustes” para establecer las preferencias.

1.6 Breve descripción de los otros capítulos de la memoria

En el capítulo 2 aplicaremos el Diseño Centrado en el Usuario en el análisis, diseño, desarrollo y evaluación de un sistema móvil.

Además, desarrollaremos diagramas UML para ofrecer una visión global del proyecto, tanto a nivel técnico, como de arquitectura del sistema.

2. Diseño

2.1 Diseño Centrado en el Usuario (DCU)

2.1.1 Usuarios y contexto de uso

Hoy en día la mayoría de gente que cuida perros en casa, tiene poco tiempo para sacarlos a pasear, y debe hacerlo rápido hasta que éste haga sus necesidades y poder volver, no les da tiempo a encontrar a otros dueños con sus perros para jugar y que estos disfruten, corran, se cansen e interactúen entre ellos.

Esto provoca en muchos casos que luego rompan cosas en casa cuando están solos, en otros casos que se muestren reticentes y agresivos con otros perros al no estar acostumbrados a acercarse a ellos, por no decir que influye negativamente en la felicidad del animal.

Por otra parte, también es muy incómodo buscar la cartilla de las vacunas cada vez que es necesaria, encontrar los papeles del chip, fotos de la mascota, o simplemente acordarse de todos los perros con los que juega. Todo eso sería mucho más sencillo con esta aplicación, ya que gestiona los perfiles con toda la información de los animales, y además dispone de una agenda con todos sus amigos y sus perfiles.

Gracias a esta agenda, se almacenarán los “perros amigos” que el usuario quiera añadir, y se podrán consultar sus datos públicos y sus imágenes. Con esta funcionalidad se agruparán los amigos de nuestros animales y nos llegarán notificaciones en cuánto lleven a cabo alguna actividad de interés (como salir a pasear).

Además, será posible ahorrar búsquedas por Internet y dudas sobre qué sitios son ideales para llevar a tus mascotas, o donde frecuentan otras personas para sacar a pasear a sus perros, si se lleva a cabo la funcionalidad extra de “lugares preferidos”, donde se podrá ver ubicación y valoraciones de otros usuarios de distintos lugares.

2.1.2 Diseño conceptual

Se han elaborado los escenarios de uso a partir de la información recopilada en la primera fase.

Nombre: Victor

Lugar de residencia: Barcelona

Lugar de trabajo: Barcelona

Edad: 27

Sexo: Hombre

S.O. del móvil: Android

Ocupación: Graduado en Arquitectura, estudiante de máster en Arquitectura Sostenible y Eficiencia Energética y trabajador a jornada completa como diseñador gráfico 3D en una consultoría.

Nivel de relación esperado con la app: Alto, puesto que saca a pasear a su perro cada día y ocasionalmente lo lleva al veterinario.

Resumen persona: Victor se pasa cada día 9 horas en el trabajo, es apasionado del arte y el poco tiempo del que dispone lo invierte en sus aficiones, tales como jugar a videojuegos, tomar algo con sus amigos o estudiar para el máster. Los fines de semana aprovecha el tiempo para estar con su pareja y, cuando puede, sacar al perro a algún lugar espacioso para que corra y juegue. Entre semana tiene poco tiempo y sería ideal disponer de HiDog para consultar donde hay otros perros en la calle y aprovechar ese tiempo para que juegue. Los fines de semana puede buscar por su ciudad las zonas con más afluencia de perros y pasar una tarde muy amena.

Escenario: Es un lunes a las 21:00h, y antes de sentarse a la mesa para cenar Victor dispone de 20 minutos para salir a pasear a su perro. Enciende HiDog, consulta donde están los amigos de su perro, si han salido, y donde hay perros paseando cerca suyo. Coloca su ubicación y sale a la calle para aprovechar el tiempo al máximo. Allí coincide con dos vecinos, uno ya era conocido, pero no tenía ni idea de la existencia de la app, le explica cómo funciona, le parece interesante y empieza a usarla. Al otro no lo había visto nunca, pero los perros se llevan bien y como son usuarios de HiDog se agregan y quedan registrados para futuros paseos. Cuando Victor llega a casa le pica la curiosidad y decide cotillear el perfil de Poncho, el perro que acaba de conocer.

Nombre: Ana

Lugar de residencia: Barcelona

Lugar de trabajo: Barcelona

Edad: 23

Sexo: Mujer

S.O. del móvil: Android

Ocupación: Graduada en Ingeniería informática, estudiante de máster de desarrollo de aplicaciones para dispositivos móviles y trabajadora a jornada completa como programadora junior en una consultoría.

Nivel de relación esperado con la app: Frecuente, tiene que sacar a su perro 3 veces al día, va justa de tiempo siempre y casi nunca deja a su perro suelto para que juegue, además es un desastre con la organización de los documentos y le iría muy bien gestionar todos los datos de su mascota en una app en su Smartphone.

Resumen persona: Ana trabaja en horario partido, y aparte estudia cuando puede para las asignaturas de la UOC. Cuando tiene tiempo se toma una cerveza con sus amigas o mira la televisión, siempre está cansada y saca a su perro lo justo y necesario. Últimamente ha tenido que ir al veterinario a menudo, y nunca se acuerda de llevar consigo la cartilla de las vacunas ni el resto de datos de su perro. Para esta situación le vendría genial consultarlo todo en su Smartphone. Y respecto a los paseos, sería un

lujo poder coincidir con otros perros sin preocuparse de quedar con nadie y hacerlo con un clic en el móvil.

Escenario: Es miércoles al mediodía, Ana vuelve del trabajo para comer en casa y sacar al perro a pasear, dispone solamente de 1 hora para hacerlo todo. Accede a HiDog mientras se prepara la comida, y descubre un parque muy cerca de donde vive con 3 personas en el mapa, termina rápido y baja corriendo. Una vez abajo puede dejar al perro suelto para que corra y juegue con los demás. Uno de ellos no se lleva bien con el suyo, pero los otros dos los va a agregar a HiDog. Su perro Lucas llega exhausto a casa y se tumba en su cama, ya no se levantará hasta la noche. Al llegar al trabajo les enseña a sus compañeros la aplicación y los perfiles de sus dos nuevos “amigos”, y por supuesto todas las fotos de “postureo” que ha añadido en el perfil de Lucas, para presumir un poco.

2.1.3 Prototipado

Tomando los flujos de interacción definidos en la segunda fase, se realizará un prototipo horizontal de la aplicación.

Un prototipo es una representación de la aplicación que permite mostrar decisiones de diseño y que éstas sean evaluadas antes de desarrollar el producto final.

Enlace al prototipo publicado en Internet y diseñado con “JustInMind”:

<https://www.justinmind.com/usernote/tests/23062062/23062370/23062372/index.html>



Figura 2.1

Pantalla de inicio. El usuario introduce sus credenciales o se registra. Aparece el icono de la aplicación. También se puede “logear” con las redes sociales. Con cualquiera de los 3 botones se accede al menú principal de la app.



Figura 2.2

Menú principal. Primera pantalla que se verá siempre después de “logearse” en la aplicación. Se observan todas las opciones que hay: Ver y editar el perfil de tus mascotas, visualizar el mapa con los perros “amigos” y poder colocar tu ubicación, Ver la agenda donde están los perros “amigos” y cambiar los ajustes de la app. Las 4 son “clickables” y te llevan a su pantalla particular.



Figura 2.3

Ver mi perfil. Pantalla para editar y visualizar el perfil de cada mascota. Se puede editar, borrar y añadir perfiles con los botones de la “action bar”. Además, se puede cambiar de perfil deslizando el dedo de un lado a otro de la pantalla. En el caso de ver el perfil de otro perro, sólo se verán los datos públicos. Al añadir o borrar un perfil se observa la transición hacia otro perfil, al editar un perfil entramos en el menú de edición.



Figura 2.4

Ver mapa. Aquí se observa el mapa con Google Maps con los perros que ya se han ubicado a través del servidor propio de la aplicación para dar a conocer su ubicación. En la parte inferior puedes seleccionar con qué mascota saldrás a pasear y cuantos minutos estarás, guardando los datos con el botón de “save”. Haciendo un “long click” en el mapa añadiremos un icono de un perro con los datos guardados anteriormente.



Figura 2.5

Si clicas a otro perro puedes ver el tiempo que lleva y el tiempo que tiene pensado estar, su foto, y también añadirlo a amigos. Aquí aparecerán todos los usuarios que den a conocer su ubicación, sean amigos o no. Al clicar encima de la foto de un perro accederemos a su perfil.



Figura 2.6

Mis amigos. En esta pantalla puedes visualizar los amigos de tus mascotas. En el buscador puedes buscar por nombre para llegar antes. En la parte superior e inferior de su foto podrás ver su perfil y eliminarlo de amigos, respectivamente. También aparecerá cuando fue la última vez que bajó a pasear, y se ordenarán por orden de tiempo. Podemos clicar en el “Edit Text” para el buscador, y también borrar y acceder al perfil de uno de los “amigos”.



Figura 2.7

Ajustes. Aquí podrás activar o desactivar notificaciones, compartir cuales son los amigos de tus mascotas en las redes sociales, y también borrar la cuenta. Podemos clicar simbólicamente el “switch” de notificaciones, y también borrar la cuenta para acceder otra vez al menú de “login”.



Figura 2.8

Menú desplegable. Desde cualquier pantalla podrás activarlo para desplazarte a la opción que desees.



Figura 2.9

Notificaciones. Aparecerán notificaciones de tus perros amigos cuando éstos salgan a pasear y coloquen su ubicación, para avisarte por si es de tu interés. Si las clicas te desplazará a la pantalla de Amigos y verás más detalles. Están programadas para aparecer en todas las pantallas a los 10 segundos.

2.1.4 Evaluación

El objetivo de este apartado es planificar la evaluación del prototipo. El proceso de DCU es un proceso iterativo y, por tanto, hay que ir evaluando los diseños y corrigiendo los errores de manera iterativa.

El prototipo será analizado y modificado conforme surjan mejoras y cambios en las necesidades de la aplicación. Este primer prototipo es una idea básica para que el usuario se haga una idea del funcionamiento de la aplicación, y de las funcionalidades que se van a implementar.

En próximas iteraciones se actualizará para acercarse más al diseño final y tener un prototipo sólido en el que basarse en el momento de la implementación.

2.2 Diseño Técnico

2.2.1 Definición de los casos de uso

El objetivo de este apartado, una vez acabada la fase de DCU, es el de definir formalmente los casos de uso. Estos casos de uso servirán para establecer las funcionalidades de la aplicación. Dichas funcionalidades serán derivadas de los escenarios de uso y el prototipo definidos durante el DCU. La definición de casos de uso incluye:

2.2.1.1 Diagrama UML de actores y flujo

Servirá para identificar de forma gráfica los actores que intervienen y el flujo de acción.

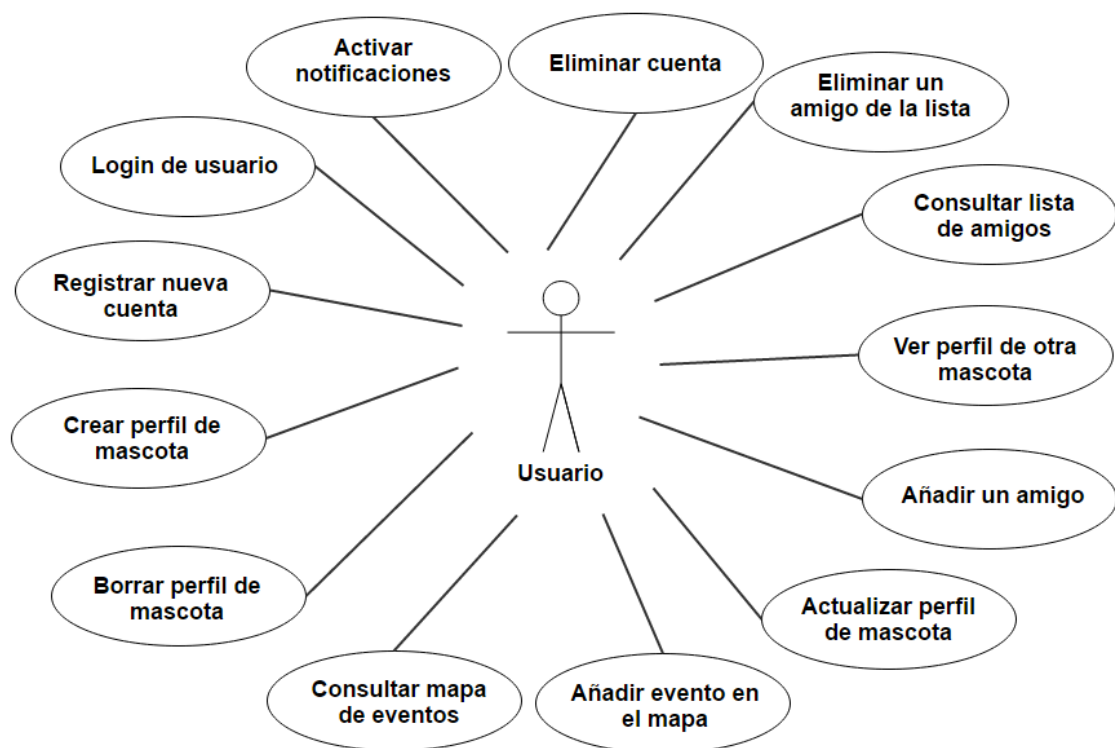


Figura 2.10

2.2.1.2 Listado de los casos de uso

Se especifican los actores, precondiciones, flujo y postcondiciones.

Nombre	Registrar nueva cuenta
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla de acceso.
Trigger	Cuando el usuario decida.
Flujo	1) El usuario selecciona “nuevo registro” en la pantalla de acceso 2) El usuario rellena el formulario 3) El usuario finaliza su registro y ya puede “loguearse”
Postcondiciones	El usuario ya dispone de una cuenta registrada.

Nombre	“Login” de usuario
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla de acceso.
Trigger	Cuando el usuario decida.
Flujo	1) El usuario rellena los dos campos de “login”, o el usuario se conecta con los datos de Gmail o Facebook 2) El usuario selecciona “submit” y ya está “logueado”
Postcondiciones	El usuario accede al menú principal

Nombre	Crear perfil de mascota
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Mis perfiles.
Trigger	Cuando el usuario seleccione el botón Añadir perfil.
Flujo	1) El usuario añade la información que crea pertinente 2) El usuario envía sus cambios
Postcondiciones	Un nuevo perfil de mascota ha sido creado.

Nombre	Borrar perfil de mascota
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Mis perfiles.
Trigger	Cuando el usuario seleccione el botón Borrar perfil.
Flujo	1) El usuario acepta la alerta de borrado
Postcondiciones	El perfil de mascota ha sido eliminado

Nombre	Consultar mapa de eventos
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Ver Mapa.
Trigger	Cuando se cargue la actividad del mapa con sus eventos.
Flujo	1) El usuario consulta los perros que aparecen en el mapa
Postcondiciones	El usuario puede ver el tiempo o añadir a los perros que aparecen.

Nombre	Añadir evento en el mapa
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Ver Mapa.
Trigger	Cuando el usuario añada un marcador en el mapa con un “long tap”
Flujo	1) El usuario elige el perro con el que saldrá a pasear 2) El usuario elige el tiempo que estará paseando 3) El usuario guarda esos datos 4) El usuario marca un “long click” en el mapa
Postcondiciones	Un nuevo marcador con la ubicación y tiempo de paseo del usuario se ha añadido en el mapa.

Nombre	Actualizar perfil de mascota
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Mis perfiles.
Trigger	Cuando el usuario seleccione el botón Editar perfil.
Flujo	1) El usuario edita la información que crea pertinente 2) El usuario envía sus cambios
Postcondiciones	El perfil de mascota se ha actualizado

Nombre	Añadir a un amigo
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Ver Mapa.
Trigger	Cuando el usuario seleccione el botón Añadir amigo
Flujo	1) El usuario clic en un perro del mapa 2) El usuario clic en Añadir amigo (icono estrella)
Postcondiciones	Un nuevo amigo ha sido añadido a la agenda.

Nombre	Ver perfil de otra mascota
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Mis amigos.
Trigger	Cuando el usuario selecciona el botón Ver perfil (icono huella)
Flujo	1) El usuario selecciona el icono de la huella de perro 2) El usuario puede ver los datos públicos de la mascota
Postcondiciones	Los datos de la mascota son visibles para el usuario

Nombre	Consultar lista de amigos
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Mis amigos.
Trigger	Cuando la actividad de Mis amigos esté cargada
Flujo	1) Los datos de Mis amigos se cargan del servidor 2) El usuario puede ver la imagen y nombre de sus amigos 3) El usuario puede usar el buscador para encontrar a un amigo
Postcondiciones	El usuario ve la imagen, último paseo y nombre de sus amigos

Nombre	Eliminar un amigo de la lista
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Mis amigos.
Trigger	Cuando el usuario selecciona el botón Borrar amigo (icono papelera)
Flujo	1) El usuario selecciona el icono de la papelera 2) El usuario elimina ese amigo de su lista
Postcondiciones	El amigo del usuario es eliminado de la lista de amigos.

Nombre	Eliminar cuenta
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Ajustes.
Trigger	Cuando el usuario quiera eliminar la cuenta
Flujo	1) El usuario selecciona Borrar mi cuenta 2) El usuario acepta la alerta de borrado
Postcondiciones	La cuenta del usuario es eliminada y se vuelve a la pantalla de acceso

Nombre	Activar notificaciones
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla Ajustes.
Trigger	Cuando el usuario active las notificaciones
Flujo	1) El usuario activa la palanca de notificaciones
Postcondiciones	Las notificaciones son permitidas y el usuario recibirá eventos de sus amigos.

2.2.2 Diseño de la arquitectura de la aplicación

El objetivo de este apartado es el de definir la arquitectura del sistema, identificando las entidades que se representarán en la base de datos, las clases y objetos que se utilizarán para gestionar los diferentes procesos y la estructura de la API que servirá para realizar las peticiones enviados al servidor desde un cliente.

2.2.2.1 Diagrama UML correspondiente al diseño de la base de datos

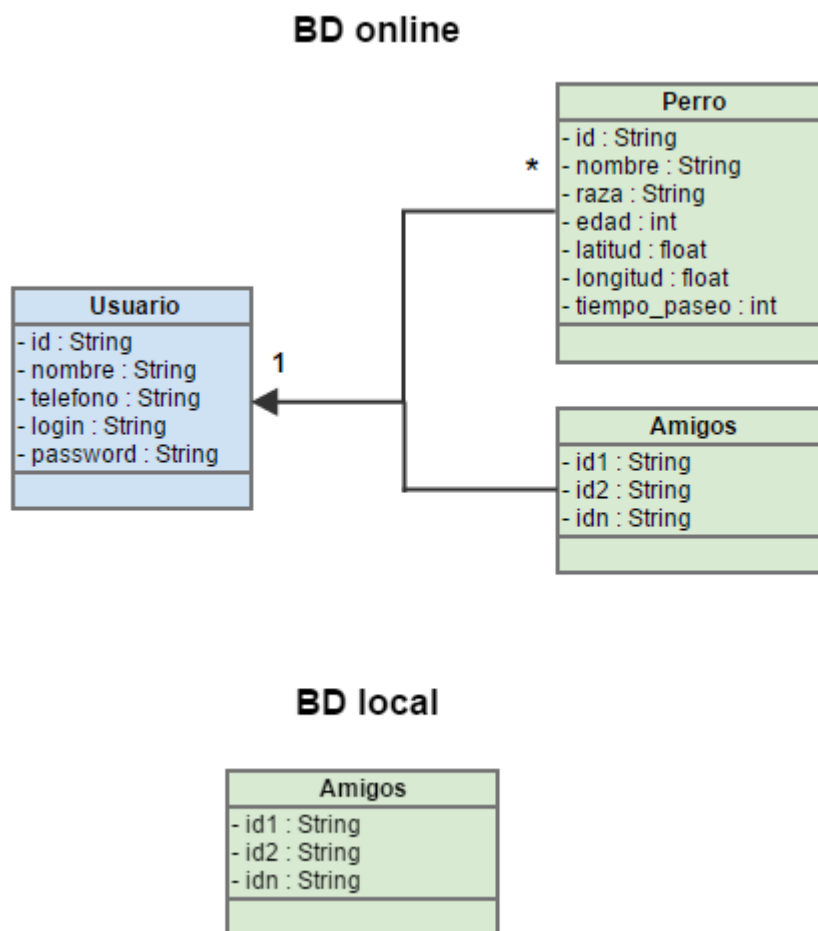


Figura 2.11

2.2.2.2 Diagrama UML correspondiente al diseño de las entidades y clases

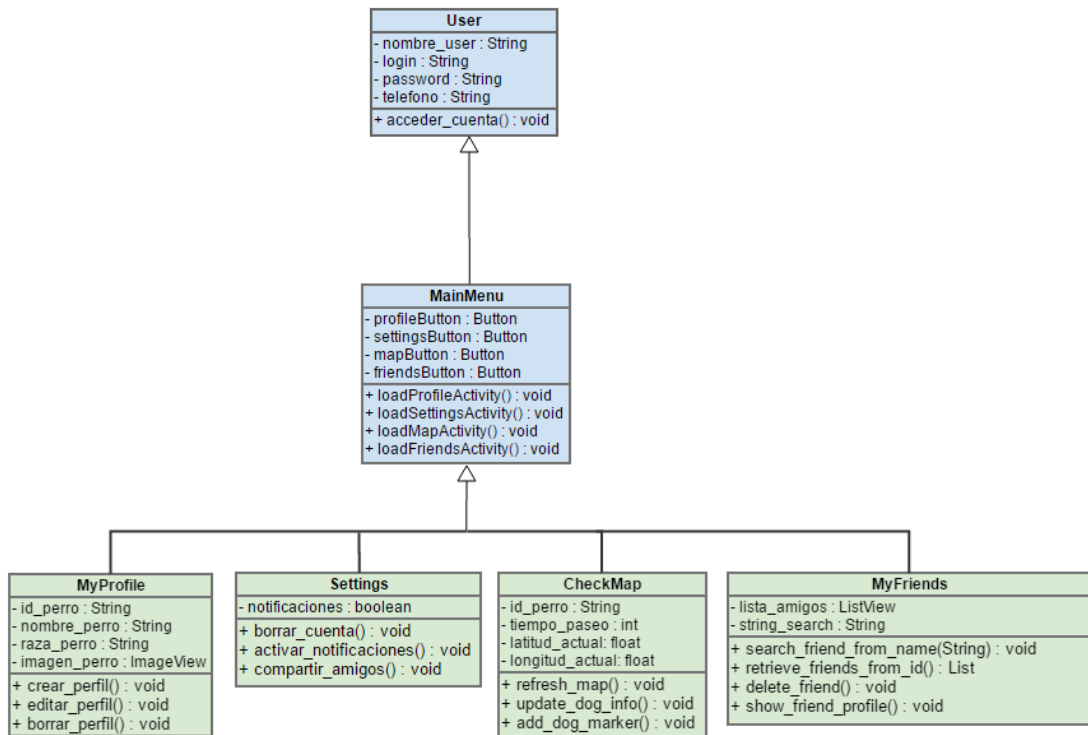


Figura 2.12

2.2.2.3 Diagrama explicativo de la arquitectura del sistema

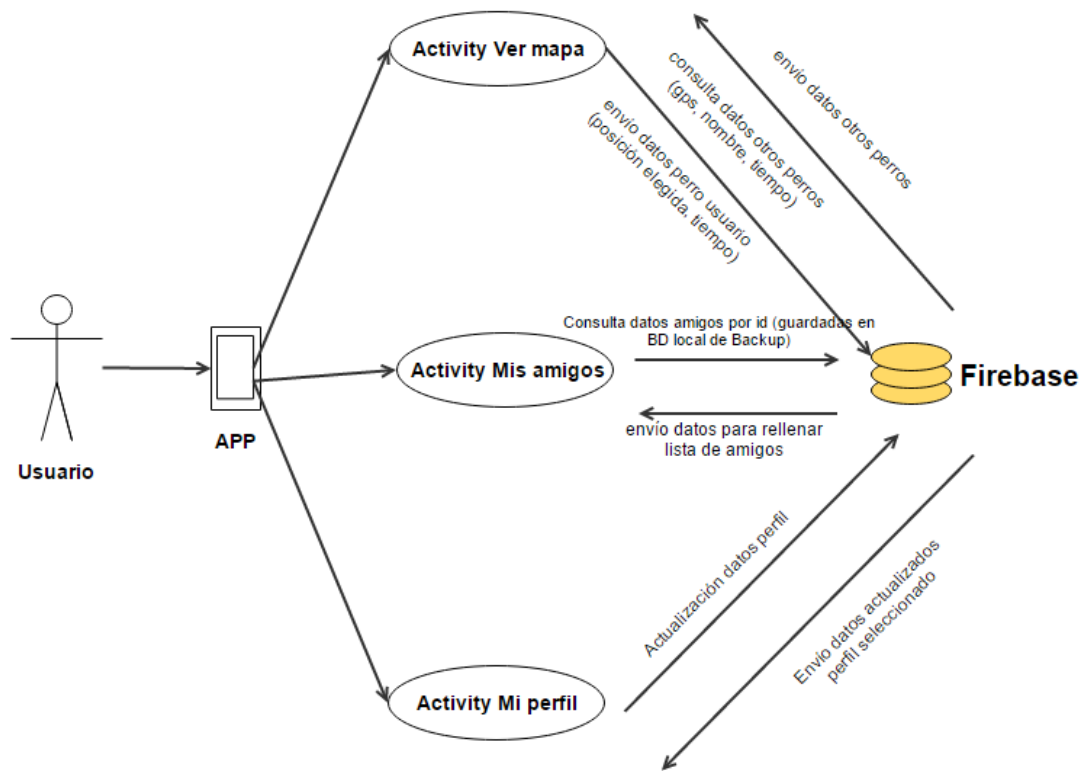


Figura 2.13

3. Implementación

2.1 Tecnologías

Para el desarrollo de la aplicación se ha utilizado un ordenador personal. A continuación, se detallan las principales características técnicas:

PC sobremesa, procesador Intel Core i7-6700 a 3.40GHz con 16 GB de memoria RAM y Windows 10 Pro de sistema operativo.

Para la depuración y realización de pruebas se ha utilizado un dispositivo móvil:

One Plus 3: procesador 2.15GHz Qualcomm Snapdragon 820 (cuatro núcleos) con 6GB de memoria RAM, pantalla de 5.5'' y Android 7.1.1 Nougat.

Android de la empresa Google como sistema operativo bajo el cual la aplicación se ejecuta. En la actualidad, junto con iOS, Android es el sistema operativo móvil más utilizado y con un mayor número de aplicaciones disponibles y desarrolladores, lo que lo convierte en un sistema operativo con un amplio número de usuarios.

Firebase es una plataforma fundada en 2011 y comprada posteriormente por Google en 2014, que permite crear y desarrollar aplicaciones para dispositivos móviles de alta calidad y de forma muy rápida. Su objetivo es ayudar a facilitar la gestión de la app para que los desarrolladores no tengan que preocuparse de desarrollar y mantener un servidor con su backend, además de ofrecer herramientas para la interconexión entre la app y el backend. En este proyecto, su uso ha sido esencial para cumplir con los objetivos y aplicar de forma cómoda la conectividad entre el usuario y el servidor.

2.2 Herramientas

En cuanto a las herramientas usadas para el desarrollo de la aplicación son las siguientes: Para el desarrollo del código fuente de la aplicación se ha empleado Android Studio como entorno de desarrollo integrado. Además, se ha usado el SDK de Android, el cual funciona con Android Studio a la perfección, gracias a la capacidad del entorno para permitir el uso de plugins.

2.3 Detalles de la implementación

1.2.1 Estructura

Con el fin de permitir el análisis del proyecto, su continuación o mejora, por parte de personas ajenas al desarrollo inicial se explicará la estructura que tiene, para así poder facilitar su entendimiento.

En primer lugar, se explicará cómo estructura Android los recursos necesarios para crear la interfaz de la aplicación. Para ello, al crear un proyecto se crea una carpeta

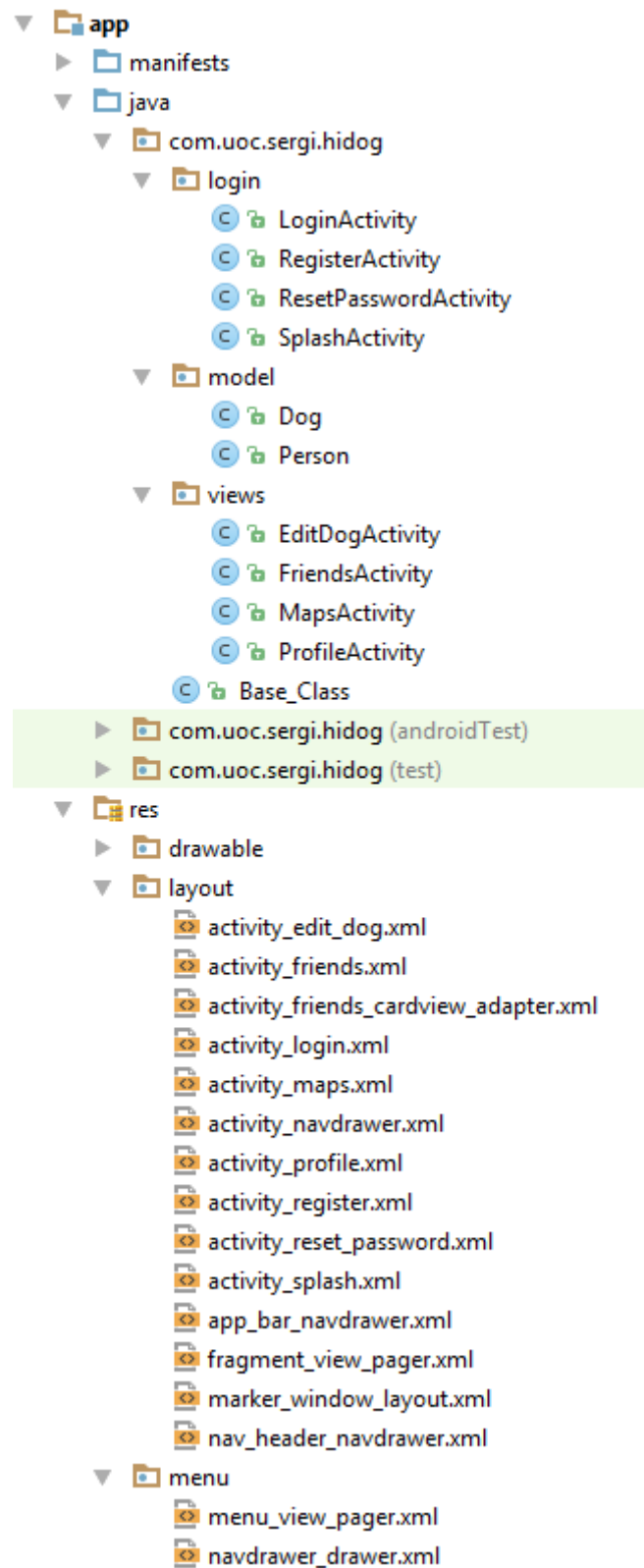
llamada res que contiene varias carpetas dentro de ella. Las tres carpetas principales dentro de res son drawable, layout y values.

Drawable contiene las imágenes utilizadas por la aplicación, así como ficheros XML utilizados para crear el estilo de los botones y de las pestañas y poder mostrar una imagen cuando están pulsados y otra cuando no. Así pues, esta carpeta contiene las imágenes del proyecto y los ficheros XML utilizados para crear los estilos de las pestañas del apartado de consulta de datos.

Por su parte la carpeta layout contiene las vistas encargadas de crear la interfaz, es decir, los ficheros XML a partir de los cuales se crea la interfaz.

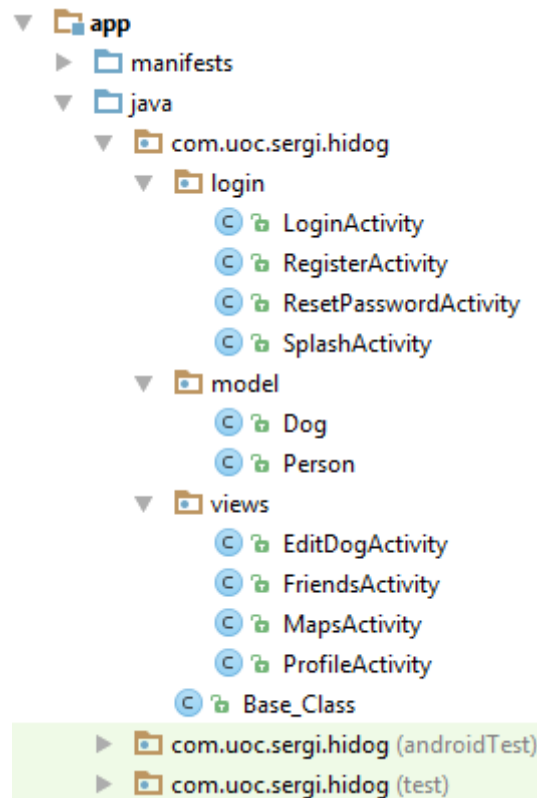
La última carpeta importante dentro de res es la carpeta values, utilizada para almacenar valores de la forma clave valor. Se utiliza principalmente para almacenar cadenas de caracteres y accederlas posteriormente haciendo uso del id. Es útil a la hora de realizar una aplicación puesto que permite modificar una cadena repetida varias veces a lo largo de la aplicación con solo cambiarla en un fichero.

En cuanto al código que implementa las funcionalidades puede encontrarse en la carpeta java. Aquí hay que hacer una distinción entre archivos que actúan como controladores y archivos que actúan como objetos. Los primeros forman parte del patrón modelo-vista-controlador y se utilizan para comunicar las interacciones del usuario con el dispositivo con los cambios que se requieren hacer en el modelo. Los segundos se utilizan para crear los objetos utilizados en la funcionalidad.



1.2.1.1 Estructura de clases

Para HiDog hemos implementado la siguiente estructura de clases:



Están divididas en 3 carpetas principales, más una clase base de la que heredarán todas las demás. Se ha intentado elegir siempre nombres explicativos, y estructurar el proyecto y el código de forma que lo pueda entender cualquiera. A continuación, se define cada una de las clases:

Paquete login:

- LoginActivity.java: Implementa la pantalla de login de la app. Está configurada como actividad principal, y siempre se ejecutará primero. Si el usuario ya está acreditado entrará a las funciones principales. Desde esta actividad el usuario podrá acreditarse para acceder a la app, cambiar la contraseña y acceder a la pantalla de registro de usuario. Además, también se accederá a esta actividad si el usuario elimina su cuenta.
- RegisterActivity: Implementa la pantalla de registro de un usuario. El usuario accederá a esta actividad cuando elija registrarse desde la actividad de Login. Una vez se añadan los datos y se acepten, el usuario entrará como registrado.
- ResetPasswordActivity: Implementa la interfaz de reenvío de contraseña. El usuario podrá usar esta función cuando se le olvide o quiera cambiar la contraseña, y así recibir una nueva en su correo.

- **SplashActivity:** Implementa la pantalla de splash de la app. En ella se cargarán datos relevantes de Firebase mientras el usuario ve durante pocos segundos el nombre y logo de la app.

Paquete model:

- **Dog.java:** Define la clase Dog. Aquí se declaran los atributos y métodos referentes a las mascotas que añadirán los usuarios a la base de datos. Se usará para gestionar toda la información relevante de los animales.
- **Person.java:** Define la clase Person. Aquí se declaran los atributos y métodos referentes a los usuarios, así como la subclase Friends, que guarda la información de los amigos del usuario. Se usará para gestionar toda la información importante del usuario.

Paquete views:

- **EditDogActivity.java:** Implementa la pantalla de edición y adición de los perros. Será posible editar la información de un perro existente y guardarla, así como añadir nuevos perros.
- **FriendsActivity.java:** Implementa la pantalla de agenda de amigos. En ella serán visibles los amigos que han sido agregados, así como sus datos más relevantes. Si se desea, es posible eliminar a un amigo de la agenda o ir a visitar su perfil.
- **MapsActivity.java:** Implementa la pantalla de salir a pasear con el perro. Aquí será posible elegir un perro y un tiempo de paseo y guardarlo, y posteriormente añadir la posición en el mapa con un long-click, para que el resto de usuarios puedan ver dónde encontrar a otros perros. Para escanear la zona circundante en el mapa se dispone de un botón con el dibujo de un perro. Al presionarlo, aparecerán los perros que hayan marcado su posición dentro del radio del usuario, y aún no haya terminado su tiempo de paseo. Desde aquí se podrá añadir un amigo al clicar la ventana de marcador de un perro. Por último, existen botones de zoom y un botón para centrar el mapa en la posición actual del usuario.
- **ProfileActivity.java:** Implementa la pantalla de visualización de perfiles. Está diseñada con un ViewPager, para deslizar de izquierda a derecha y así observar los perfiles más cómodamente. En esta actividad será posible añadir un nuevo perro o editar uno existente, así como eliminar uno ya existente. Además, si se accede desde el perfil de un amigo, también se podrá llamar al número de teléfono de su dueño.

Clase Base:

- **Base_Class.java:** Implementa la mayoría de funciones y variables globales para que el resto de clases hereden de ella. De esta forma se consigue un encapsulamiento de gran parte del código en una sola función, y permite leer y entender el resto de clases mucho más rápido. Principalmente consta de funciones para consultar y escribir en Firebase.

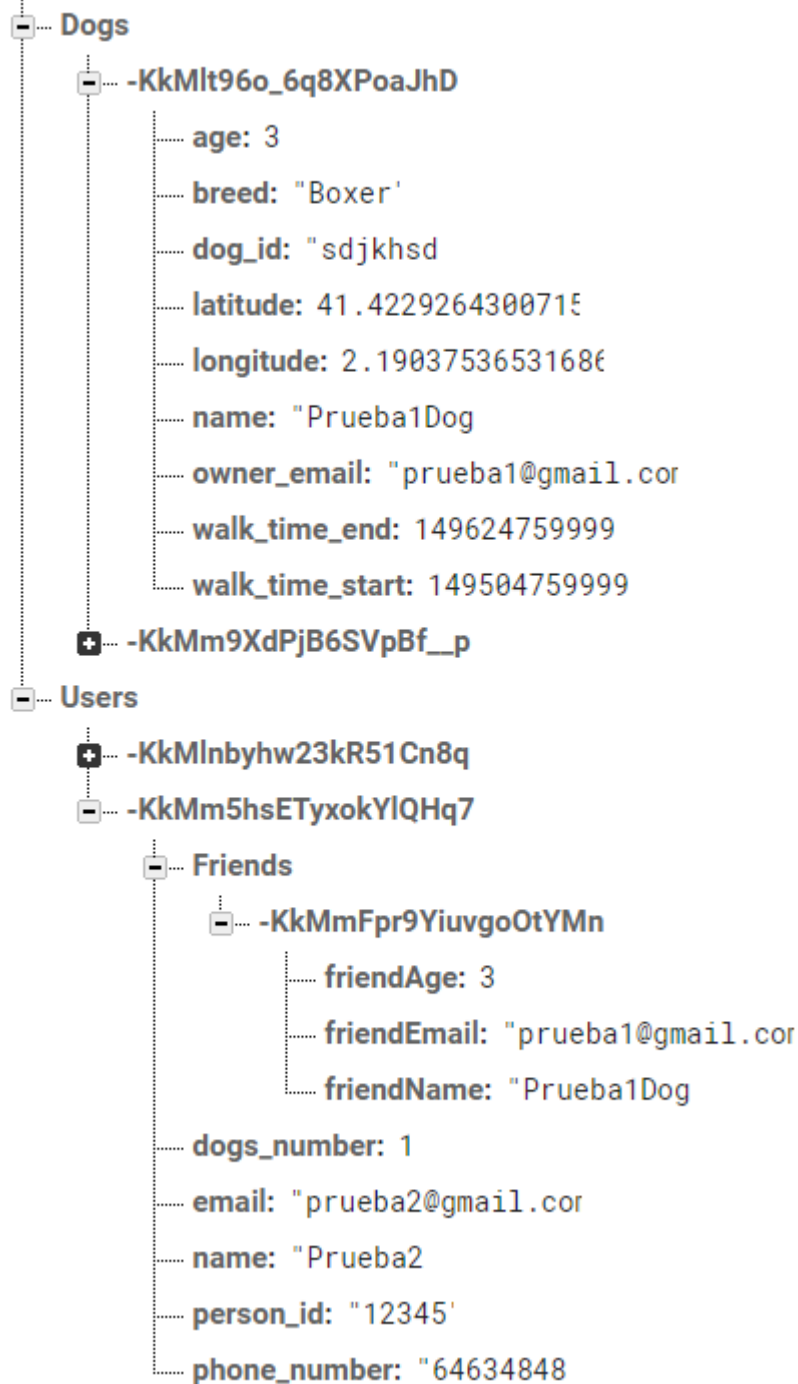
1.2.1.2 Estructura de Firebase

Firebase es una herramienta indispensable para este proyecto. Con él se ha estructurado los datos con claves únicas, separando usuarios y perros para poder leer transversalmente. De esta forma, un perro está asignado a un usuario por el email, que es un dato único, y además cada elemento tiene una clave única asignada por Firebase automáticamente.

Los amigos van atados a cada usuario, por lo tanto, han sido añadidos como un campo más en los datos del usuario.

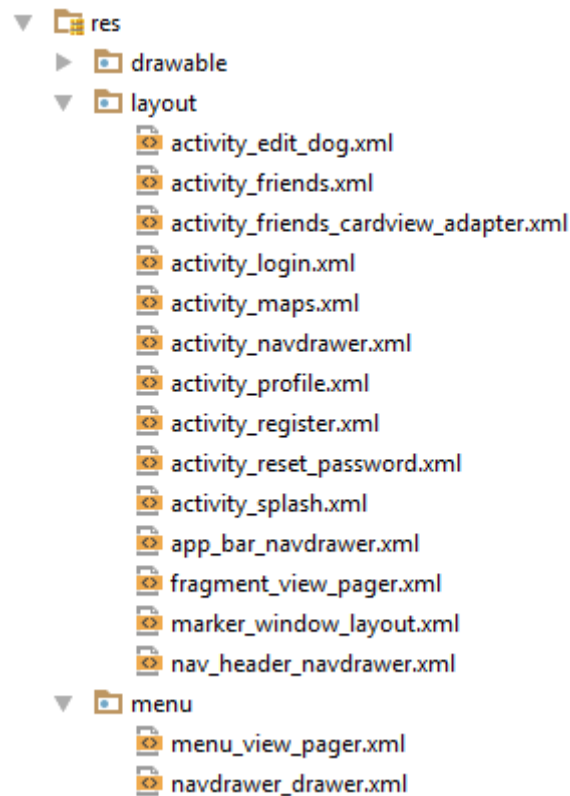
Este es un ejemplo de la estructura que sigue HiDog en Firebase:

hidog-bd30f



1.2.1.3 Estructura de interfaces

Los layout correspondientes a las pantallas se han creado con nombres parecidos al de la actividad donde están llamados. En la siguiente imagen se pueden observar todos los usados así como los menús que se han usado en la Navigation Drawer y en la Toolbar.



1.2.2 Implementación de las funcionalidades

En HiDog se dispone de 3 funcionalidades principales:

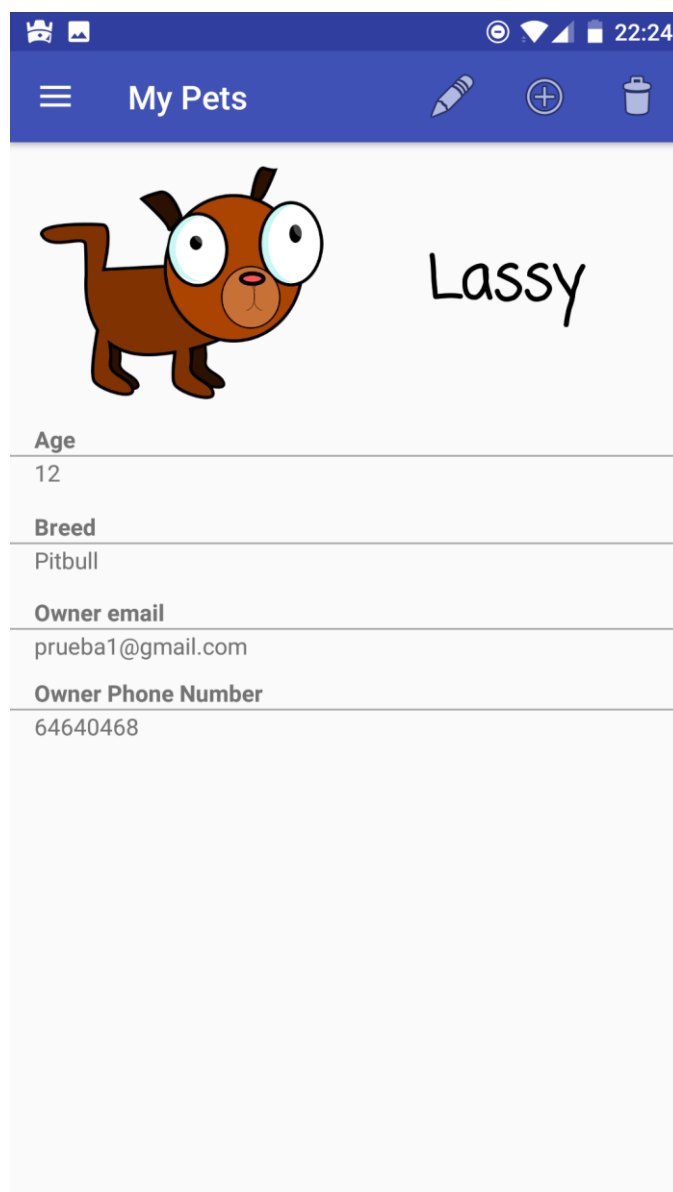
- My Pets
- My Friends
- Walking time

Además, también se han añadido elementos visuales como una barra de navegación y acciones de cuenta de usuario tales como logout y delete account.

En este apartado se explicará sobre cómo están implementadas cada una de ellas, y se ampliará información para entender qué nos permite hacer cada actividad.

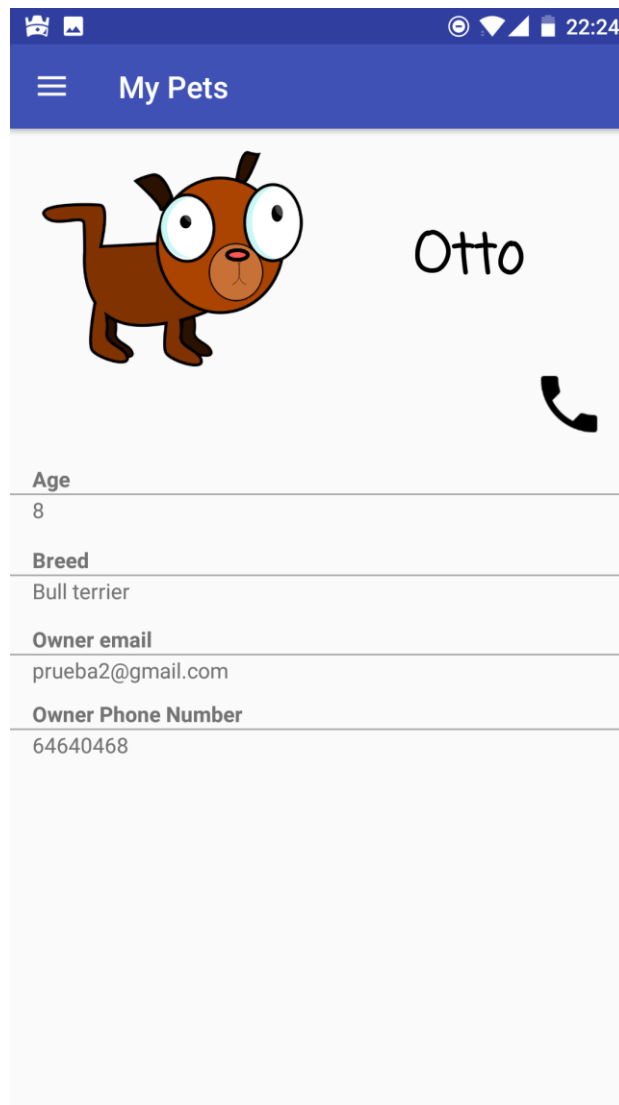
1.2.2.1 Funcionalidad My Pets

Esta actividad será usada para gestionar la información de nuestras mascotas. Esta pantalla será la primera que se verá al iniciar sesión en HiDog, y aparecerá vacía si no tenemos ningún perro registrado.



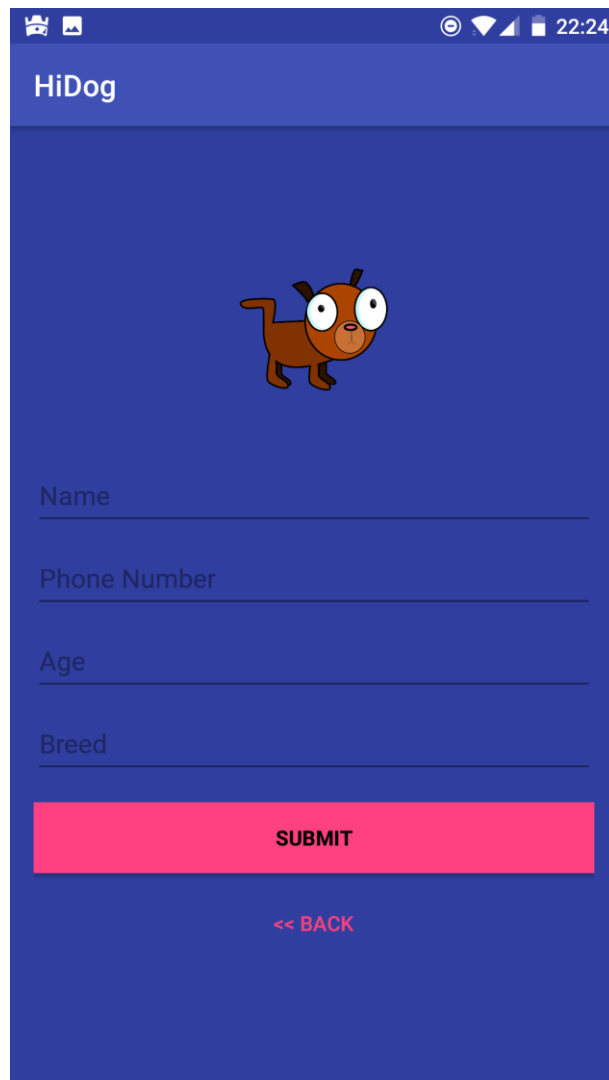
Aquí se podrá añadir, editar o eliminar perfiles de mascotas con los botones de la barra superior. Para moverse por los distintos perfiles, el usuario deberá deslizar de un lado a otro con el dedo, ya que está montado con un ViewPager.

Esta actividad tiene también un modo de visualización de amigos, cuando el usuario accede para ver un perfil de un amigo desde la actividad My Friends.




En este caso aparecerá el icono de llamada y desaparecerán los botones de gestión de perfiles, ya que en este caso el perro no es del usuario.

En la siguiente imagen se puede ver el menú de añadir/editar perros, la única variación es que el nombre es único en cada usuario, por lo tanto, no se puede editar.



HiDog



Name

Phone Number

Age

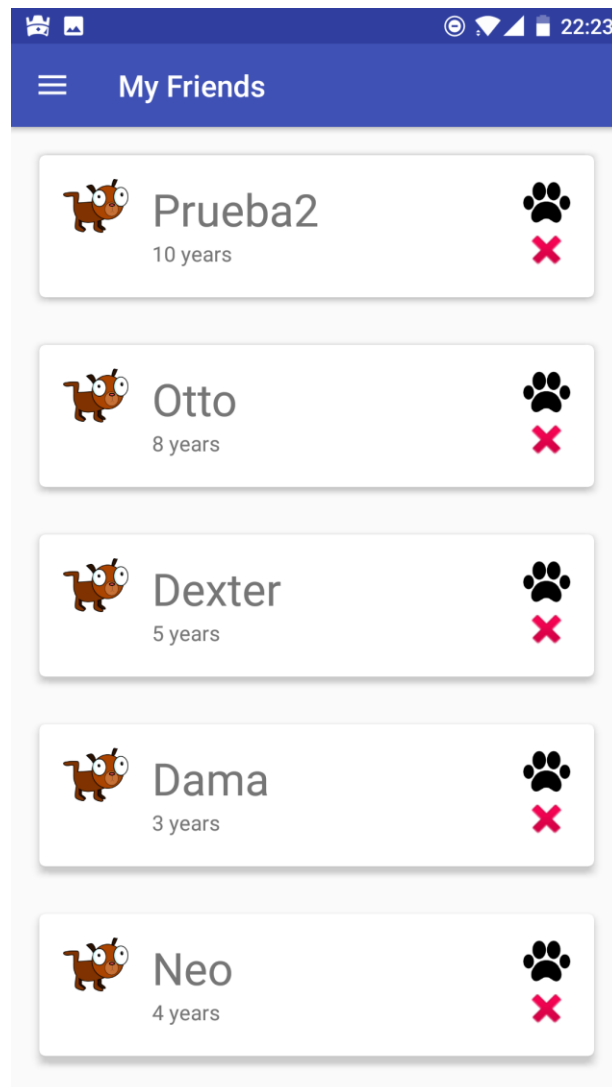
Breed

SUBMIT

<< BACK

1.2.2.2 Funcionalidad My Friends

Esta actividad será usada para gestionar la agenda de amigos de nuestros animales. Aquí visualizaremos en forma de lista todos los amigos agregados, con algunos de sus datos relevantes como su edad y su nombre (en la próxima actualización se añadirán las imágenes).



Como se observa en la imagen, existen dos botones para acceder al perfil del perro y para eliminar al perro de la lista de amigos. Como se explicará en la siguiente funcionalidad, un amigo se agrega desde el mapa, una vez se observa que saldrá a pasear.

1.2.2.3 Funcionalidad Walking Time

Esta actividad es la más compleja del proyecto y a la vez la más interesante. En esta pantalla aparecerá un menú para seleccionar ciertos datos para pasear al perro en la parte inferior, y un `FrameLayout` que ocupa el resto de la pantalla con la API de Google Maps.

El objetivo es poder colocar la ubicación propia a la hora de sacar a pasear a los animales, y a su vez ser capaz de ubicar a otras personas que han salido a pasear y han dado a conocer su ubicación también. De este modo, cuando se escanea el mapa, se

obtendrá los perros que han salido a pasear por nuestra zona, y se podrá saber el tiempo que estará esa persona en la calle.



Para poder poner la ubicación, el usuario debe elegir el tiempo que estará paseando, y el perro con el que estará, para después marcar con un long-click la zona donde saldrá a pasear. De esta forma la información se almacena en Firebase, y otras personas lo pueden consultar.

En el momento de clicar el botón de la parte superior izquierda, se obtienen todos los perros que están paseando en la zona del usuario. Si se clicca el marcador se ampliará la información de la mascota y al clicar en la ventana, se agregará como amigo. En la imagen de a continuación se observa este último paso.



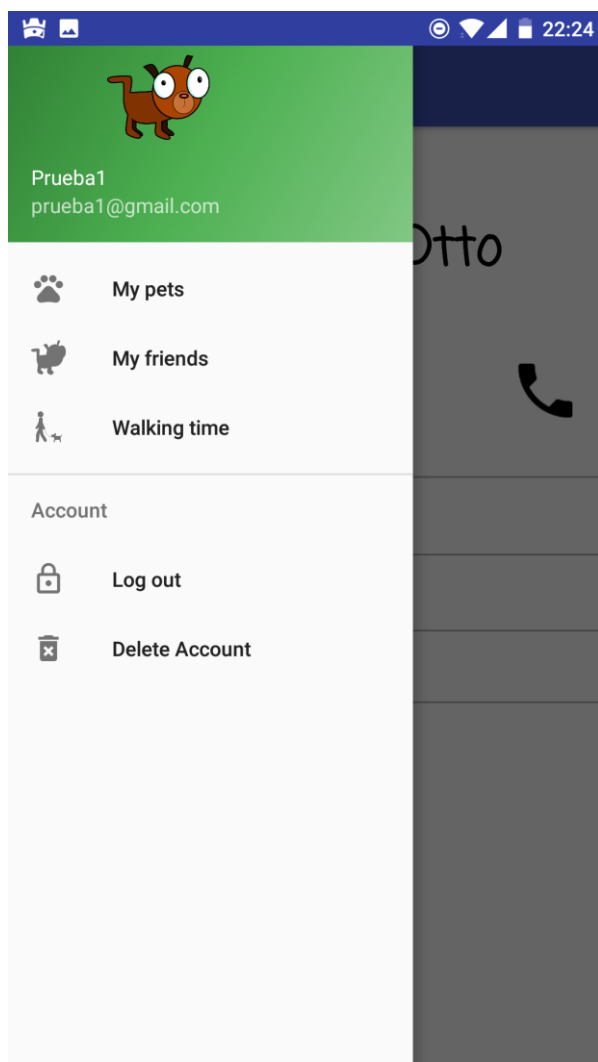
Además, cabe destacar que el escaneo también filtra por tiempo de paseo. Si un usuario ha seleccionado 20 minutos de paseo, cuando se escaneen las posiciones se comprobará que no hayan pasado ya esos 20 minutos. Si es así, no se mostrará su ubicación. De esta forma se asegura el usuario que los perros que ve en el mapa están en su zona y además todavía están en la calle.

1.2.2.4 Otras funciones

Por último, se ha creado un NavigationDrawer con el que nos moveremos de una actividad a otra, ya que estará presente en todas ellas. En él aparecen los datos del usuario actual.

En esta barra lateral se puede observar las otras dos funcionalidades de usuario disponibles: logout y delete user.

Como su nombre indica, serán útiles para desconectarse de la sesión de usuario y para eliminar la cuenta definitivamente.



4. Pruebas

4.1 Pruebas con terminales virtuales

Para el desarrollo de la aplicación se ha utilizado un ordenador personal. A continuación, se detallan las



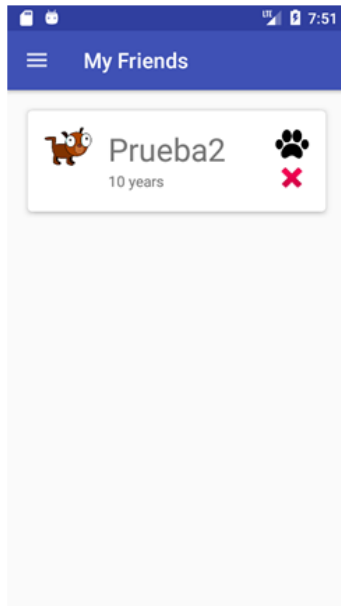
FWVGA API 23



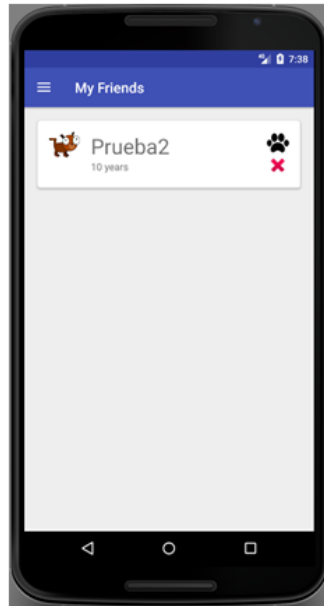
Nexus 6 API 22



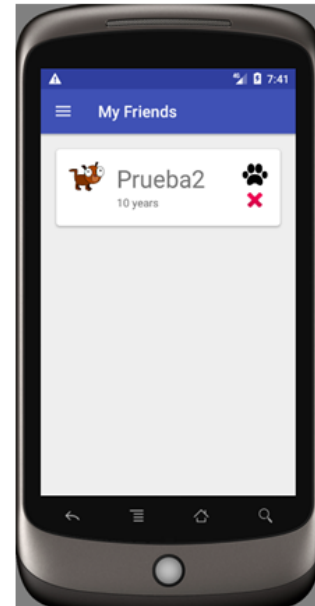
Nexus One API 22



FWVGA API 23



Nexus 6 API 22



Nexus One API 22

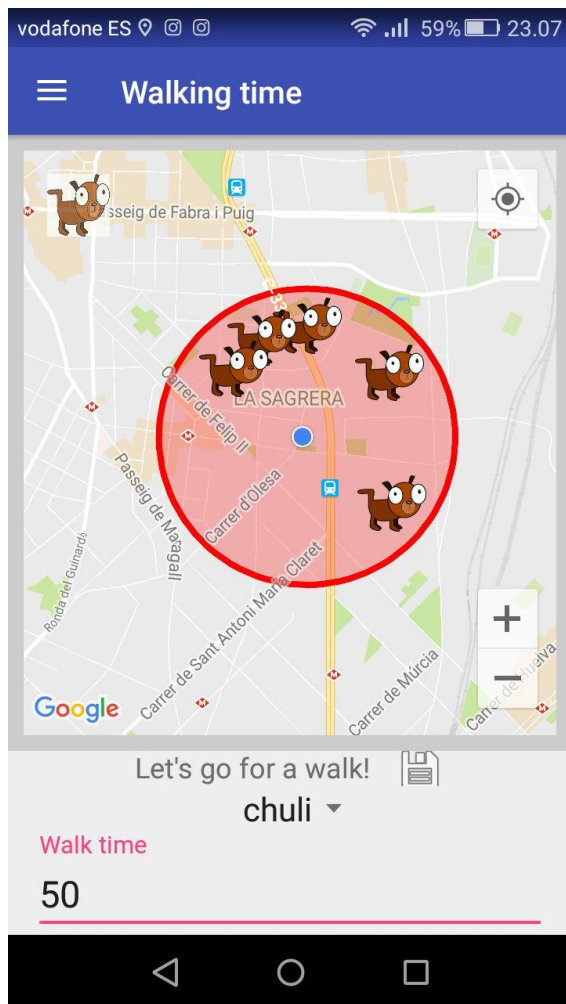
4.2 Pruebas con terminales reales

Para las pruebas con terminales reales se han usado dos dispositivos distintos. En ambos se han probado todas las funciones de la app e incluso moviéndose por el mapa por distintos lugares captando las posiciones de perros ubicados previamente.

En todos los casos las pruebas han sido satisfactorias.



One Plus 3 – Android 7.1.1 Nougat



Huawei P8 Lite – Android 5.1

5. Conclusiones

En este apartado del proyecto se expondrán las conclusiones y retos que ha supuesto la realización de este proyecto por parte del alumno, así como una serie de funcionalidades que han ido surgiendo durante la realización del mismo, pero que por falta de tiempo o por no realizar una replanificación del mismo, se ha decidido finalmente no incluir.

Este apartado se compone de las siguientes secciones:

- Evaluación de objetivos y conclusión
- Tiempo dedicado
- Mejoras y ampliaciones
- Conocimientos adquiridos

A falta de las mejoras que vendrán en la entrega final, se puede decir que el proyecto cumple en gran medida con los objetivos planteados en un inicio.

Las 3 funcionalidades principales deseadas han sido implementadas, quizás con menos detalle del que se esperaba en un inicio, pero totalmente operativas y funcionales.

Personalmente tengo que decir que me he centrado al 100% en la funcionalidad y en su buen funcionamiento, dejando un poco más al margen el diseño y una memoria que verá su versión mejorada en la entrega final.

Actualmente estoy trabajando y realizando otras actividades en mi tiempo libre, por lo que el tiempo que se le ha podido dedicar ha sido reducido, y entre dejar una app lista y funcional con todo lo que quería hacer, o ir trabajando la memoria poco a poco y con más detalle he elegido siempre la primera, por lo que la entrega final mejorará considerablemente el trabajo actual.

En cuanto a mejoras futuras hay muchas pensadas, solo nombrando las principales que están pensadas para corto plazo:

- Elección por parte del usuario del rango de búsqueda de perros
- Añadir imágenes para usuario y perros, y gestionarlas con Firebase para que sean descargables para todos los usuarios
- Mejorar el diseño con imágenes más nítidas, y los layouts más bien ajustados
- Añadir un buscador en la actividad de My Friends para encontrar rápidamente el deseado
- Implementar la funcionalidad extra para puntuación y lugares favoritos.

Por último, lo más importante en este proyecto han sido los conocimientos adquiridos, sin ninguna duda. He ampliado y aprendido mucho en programación Android y en concreto sobre las siguientes herramientas:

- Firebase y su comunicación con Android
- Google Maps API y todas sus características
- ViewPager y todo el manejo de su adaptador

- RecyclerView junto con CardView, tanto en implementación como en diseño
- NavigationDrawer
- Acciones de usuario con Firebase (login, logout, etc)
- Tareas asíncronas para no entorpecer la interacción del usuario

6. Bibliografía

[1]Developer Android <https://developer.android.com>

[2]Stackoverflow <https://stackoverflow.com/>