

# **PROYECTO FINAL DE INGENIERÍA**

## **APLICACIÓN MOBILE PARA BÚSQUEDA DE PASEADORES DE MASCOTAS BAJO DEMANDA**

**Balmaceda, Luciano – LU 1011644**

Ingeniería Informática

**Piiego, Martín Eduardo – LU 1010350**

Ingeniería Informática

Tutor:

**Faguaga, Alejandro, UADE**

**Junio 06, 2017**



**UNIVERSIDAD ARGENTINA DE LA EMPRESA  
FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS**

## **Resumen**

Las personas hoy en día viven conectadas con el mundo a través de internet. El hecho de que los smartphones se volvieran tan avanzados y accesibles hace que sean utilizados a diario para informarnos y comunicarnos entre nosotros. Nuestro proyecto busca conectar paseadores de mascotas con los dueños de las mismas a través de una aplicación mobile, con el objeto de simplificar la programación de paseos y facilitar su seguimiento en tiempo real.

## **Abstract**

People today live connected to the world through the internet. The fact that smartphones became so advanced and affordable makes them used daily to inform and communicate with each other. Our project seeks to connect pet walkers with their owners through a mobile application, in order to simplify the scheduling of rides and facilitate their monitoring in real time.

## Tabla de Contenidos

Objetivo.....	5
Alcance.....	5
Antecedentes.....	5
Aportes.....	6
Descripción.....	6
Solución Propuesta .....	6
Análisis .....	6
Requisitos para ser Paseador .....	10
Arquitectura.....	11
Aplicación Back-End y Base De Datos .....	12
Servicio de Notificación .....	12
Servicio de Localización .....	12
Estructura de Datos .....	13
Vista de Despliegue.....	13
Aplicación Back-end.....	14
Especificación de la API.....	16
Aplicación Mobile .....	34
Diagrama de Flujo de UX.....	36
Variante “Paseadores” .....	36
Variante “Dueños” .....	37
Lista de requerimientos encontrados.....	38
Identificación de los casos de uso.....	39
Trazabilidad entre requerimientos y casos de uso .....	40
Diagrama de casos de uso.....	41
Listado de casos de uso detallados.....	42
Diagrama de clases .....	80
Variante “Paseadores” .....	80
Variante “Dueños” .....	81
Conclusiones .....	82
Bibliografía .....	83

## Objetivo

El objetivo principal del PFI es la construcción de un sistema que permita buscar y coordinar paseos de mascotas.

Como objetivos específicos del trabajo, se mencionan:

- Realizar aplicación mobile que permita la búsqueda, reserva y seguimiento de un paseador de mascotas.
- Realizar aplicación back-end que permita coordinar los paseos en los clientes mobile.
- Mejorar el servicio existente de paseadores de mascotas en la ciudad.

## Alcance

La aplicación mobile será realizada únicamente para la plataforma Android. Contará con el apoyo de una plataforma back-end en la cual se almacenarán todos los datos utilizados y se coordinarán los eventos asociados al paseo.

Al finalizar el PFI se espera tener los siguientes módulos funcionando:

- Módulo Cliente desarrollado sobre una plataforma mobile y utilizado directamente por el usuario para solicitar, reservar y gestionar un paseo.
- Módulo Back-end que brinde soporte a la aplicación mobile.

## Antecedentes

Encontramos servicios similares aplicados a otros rubros, como lo es Uber y su servicio de transporte privado. Este producto permite dada una ubicación, encontrar el vehículo más cercano e informarle que un usuario requiere el servicio de transporte privado. Una vez que el conductor acepta el viaje y se dirige a recoger al usuario, ambas partes quedan en contacto permitiendo consultar en tiempo real el recorrido del vehículo y comunicarse para facilitar el encuentro. La tarifa del viaje va a depender del tiempo que duró y los kilómetros recorridos, similar a la tarifa de un Taxi porteño. El usuario luego puede consultar su historial de viajes y evaluar el servicio provisto por el conductor, para que otros usuarios puedan decidir con confianza si aceptan al conductor antes de comenzar el viaje.

También encontramos servicios simples de seguimiento de la ubicación de personas, y foros sociales donde los usuarios se comunican para solicitar recomendaciones de paseadores de mascotas en su zona; pero no encontramos un servicio que una ambos como el propuesto por este trabajo.

## Aportes

Mejorar la calidad de vida de las personas al simplificar la búsqueda y el contacto de paseadores de mascotas cercanos, al proveer de una interfaz simple en un entorno actual y conocido como lo es el smartphone. A su vez, aumentar su productividad al eliminar de la ecuación el tiempo que de otra forma estarían dedicando a pasear a sus mascotas.

## Descripción

Las personas que habitualmente contratan a un tercero para pasear a sus mascotas no saben con certeza si sus animales fueron tratados adecuadamente, o si fueron abandonados en una esquina por varios minutos mientras el paseador se ausenta para descansar o realizar otras tareas. Tampoco tienen la certeza de que las mascotas hicieron sus necesidades durante el paseo, lo cual en caso negativo implicaría tener que esperar a que las hagan antes de poder devolverlos por varias horas a un ambiente cerrado como puede ser un departamento.

El Gobierno de la Ciudad de Buenos Aires provee de un “Registro de Paseadores de perros” en el que deben inscribirse aquellos que lleven más de 3 animales, pero no proporciona capacitación ni verificación alguna de las aptitudes del paseador.

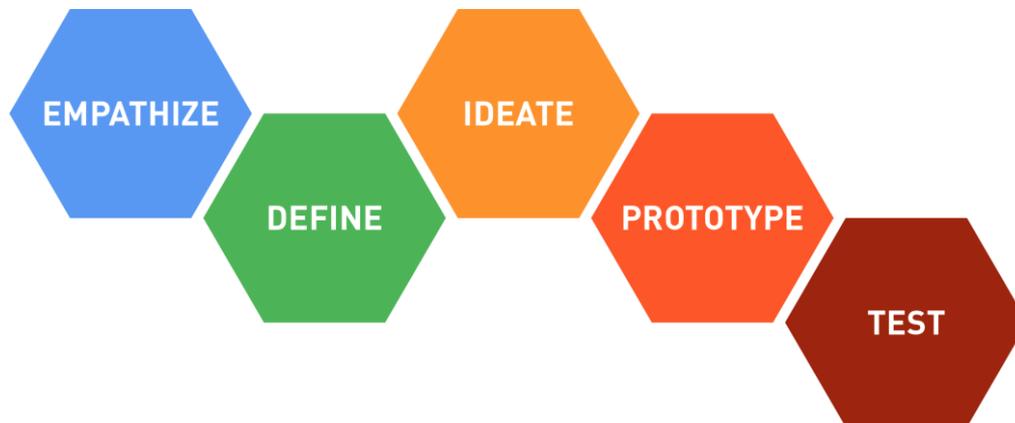
## Solución Propuesta

Ofrecer un servicio que ayude a los dueños de mascotas a encontrar personas dispuestas a pasear a las mismas a cambio de una tarifa. Los paseadores deben estar registrados en nuestra aplicación, ser capacitados y verificados de alguna forma para brindar confianza al dueño que prefiere no dejar el animal en manos de un desconocido. Desde el momento que el paseo es iniciado y hasta el mismo finaliza, el dueño puede seguir en tiempo real la ubicación del paseador, y por ende la de la mascota, para determinar tiempos de arribo y caminos tomados. El paseador por su parte puede informarle eventos que considere importantes como por ejemplo que el animal realice sus necesidades fisiológicas o se pelee con otro animal. Una vez finalizado el paseo el dueño puede evaluar la calidad del servicio brindado para que otros usuarios consideren elegir al mismo paseador, y el historial de paseos realizados se actualiza para mostrar a modo de reporte el recorrido realizado y la duración.

## Análisis

Para analizar la necesidad y determinar estos requisitos utilizamos Design Thinking. La técnica propone un conjunto de fases iterativas que buscan entender al usuario y sus

necesidades con el fin de identificar soluciones. Nos basamos en el modelo de la universidad de Stanford ya que es el más extendido hoy en día. Las fases son 5:



1. **Empatizar:** Desde un rol primero de observador y sin realizar preguntas, hay que meterse en la piel del usuario para lograr empatía y entender lo que le sucede a diario desde su perspectiva. Obtenido el contexto se puede tomar un rol de entrevistador y preguntar los detalles.
2. **Definir:** Habiendo estudiado los datos recolectados se identifica el problema para el cual se va a buscar una solución. Una forma simple de realizarlo es definir en una oración las necesidades, el problema y por qué pensamos que se necesita.
3. **Idear:** La creatividad entra en juego y se proponen soluciones innovadoras a los problemas encontrados. Aquellas ideas que seleccionadas como candidatas deben acompañarse de un plan de negocio que permita evaluar la implementación.
4. **Prototipar:** Tomando las ideas elegidas, se realizan prototipos muy simples utilizando poco tiempo y presupuesto, pero que permitan demostrar con un alto nivel qué tan alcanzables son.
5. **Probar:** Se hace entrega del prototipo al usuario para que nos diga si resuelve su problema

Las fases son iterativas porque cada una de ellas puede ser utilizada para retroalimentar una fase anterior, pero no necesariamente son lineales. De los problemas encontrados en los prototipos pueden surgir nuevas ideas, y de las pruebas realizadas por los usuarios se puede aprender que el problema era otro. Aplicamos la técnica a nuestro caso y obtuvimos lo siguiente:

1. Nos dirigimos a la Ciudad Autónoma de Buenos Aires para observar a los paseadores de mascotas en su rutina callejera. Los eventos interesantes del paseo resultaron ser el retiro y entrega de la mascota, ya que ahí pudimos observar la interacción del paseador

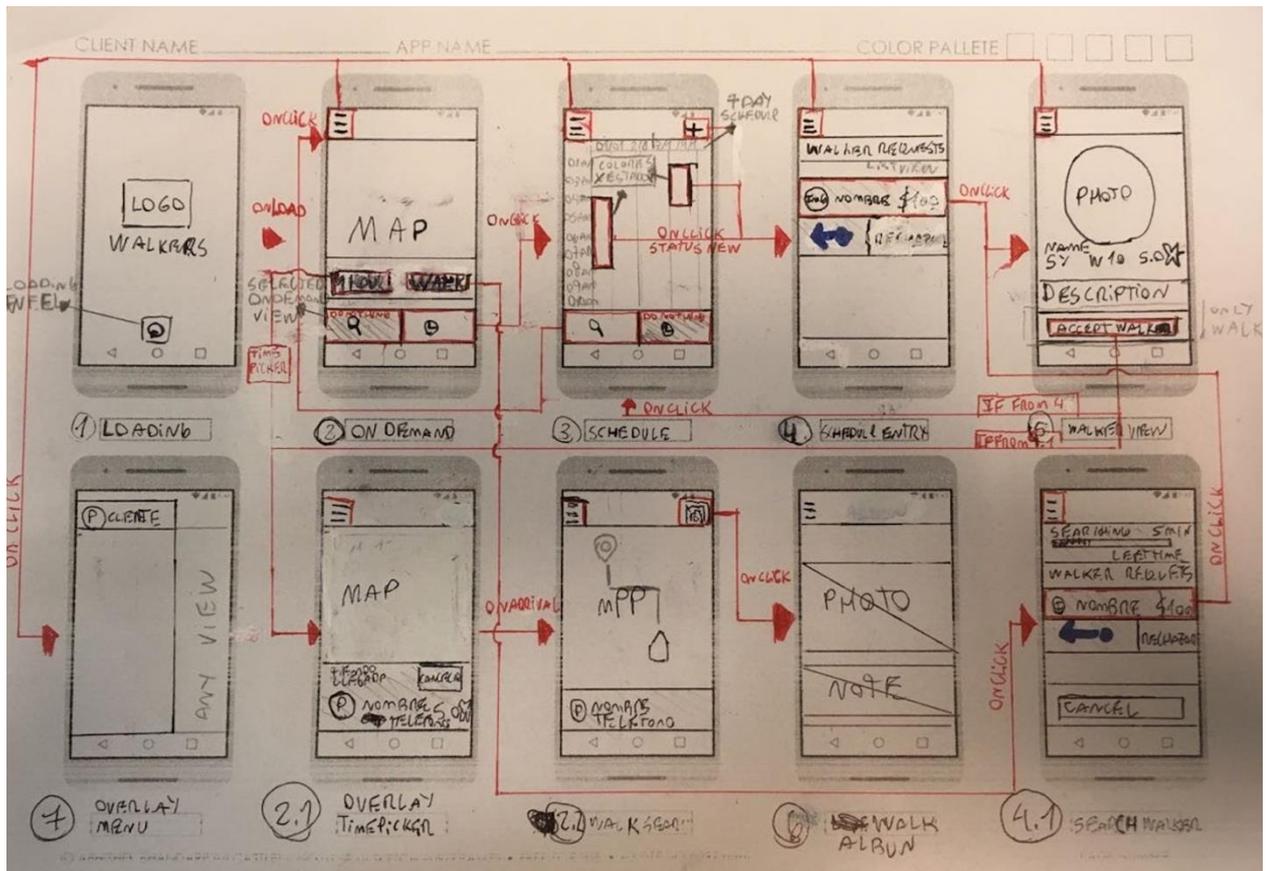
con los clientes. Luego de unas semanas de seguimiento participamos de reuniones sociales en donde algunas personas comenzaron a hablar acerca del servicio de paseadores de mascotas que se brindaba en la ciudad. Nosotros desde el rol de observadores, escuchamos los beneficios y las falencias del sistema y tomamos nota.

2. Con los datos recolectados enumeramos un conjunto de hechos:
  - La mayoría de los paseadores elige pasear una cantidad de perros significativa para incrementar las ganancias.
  - Los paseadores encontrados en la calle estaban inscriptos en el Registro de Paseadores de Mascotas de la ciudad de Buenos Aires.
  - Los paseadores al momento de devolver la mascota son consultados si la misma realizó sus necesidades durante el viaje, y en reiteradas ocasiones ellos no lo recordaban.
  - Los dueños de las mascotas prefieren que sus perros sean paseados en pequeños grupos para que las mismas puedan circular libremente y no de manera hacinada.
  - Los dueños de las mascotas en algunas ocasiones dudan de que realmente se las esté paseando o que no se les brinde el trato adecuado.
  - Algunas personas cercanas a los dueños se ofrecían como voluntarios para pasear a sus mascotas a cambio del mismo o incluso menos dinero del que los paseadores les cobran. Los dueños se mostraban seguros ya que confiaban más en ellos que en los paseadores.

Con todo lo anterior, definimos que el problema de los dueños es conseguir una persona confiable que pueda pasear a su mascota, y que la necesidad no es que aquella persona sea cercana al círculo de amigos del dueño o el animal, sino que cualquiera que pueda ser capaz de brindar detalles del paseo mientras es llevado a cabo va a calmar sus dudas.

3. Realizamos una tormenta de ideas para proponer opciones. Los resultados fueron los siguientes:
  - Una aplicación web que permita realizar el seguimiento del paseo de una mascota en tiempo real gracias a un collar especial portado por el animal.
  - Una red social donde los paseadores de mascotas puedan ofrecer sus servicios y ser contactados por usuarios que luego de evaluar sus calificaciones y comentarios pasados, deciden pasear a sus mascotas con ellos.

- Una aplicación para smartphones que permita la búsqueda de paseadores y el seguimiento de paseos en tiempo real.
4. De las ideas anteriores descartamos la del collar porque requiere desarrollar un dispositivo adicional sólo para realizar el seguimiento de la posición física del perro o paseador. Considerando que hoy en día todas las personas tienen un smartphone, la posición se puede obtener mediante el dispositivo del paseador instalando una simple aplicación. Pensamos entonces que las dos ideas restantes se pueden unir en una misma y realizamos el prototipo. Las primeras pantallas obtenidas fueron las siguientes:



5. Realizamos una implementación rápida y al mostrar la misma a los usuarios, los resultados fueron muy positivos. Estaban contentos de la transparencia que se agregaba al servicio de paseo de mascotas, y mostraron interés en utilizar la aplicación de ser publicada ya que resolvía sus necesidades.



### Requisitos para ser Paseador

El Gobierno de la Ciudad de Buenos Aires pone a disposición las normas que cualquier persona que pasee perros debe cumplir.

- Los animales domésticos deben ser conducidos mediante el empleo de correa y collar.
- Deben recogerse las deyecciones de los animales con una escobilla y una bolsita.
- No atar los animales a árboles, monumentos, semáforos, postes o cualquier tipo de mobiliario urbano.
- Los perros solamente podrán permanecer en el espacio público sin el empleo de rienda o collar cuando se encuentren en los lugares exclusivamente autorizados para el uso de animales domésticos (por ejemplo, caniles).
- Si se pasea a más de tres perros en forma simultánea (aun cuando los animales sean propios), es necesario inscribirse en el Registro de paseadores de perros.
- Los paseadores inscriptos en el Registro no pueden llevar más de 8 perros a la vez.

A su vez, aquellos que deseen obtener el carnet del “Registro de paseadores de perros” de la Ciudad de Buenos Aires, deben cumplir con las siguientes condiciones:

- Ser mayor de 18 años, acreditado con la presentación de documento de identidad y fotocopia.
- Tener domicilio legal en la Ciudad de Buenos Aires, acreditado con certificado expedido por la Policía Federal Argentina.

- Entregar dos fotos color 4 x 4.
- Firmar la declaración jurada.
- Pagar el arancel correspondiente a la suscripción (\$300) o la renovación (\$100).
- Dirigirse de manera personal a la sede del Registro de paseadores de perros, Cerrito 268, piso 11°, de 11:00 a 17:00 hs para realizar el trámite.

Tomando como base los requisitos impuestos por el Gobierno de la Ciudad, definimos los propios:

- El paseador debe ser mayor de 18 años.
- El número máximo de mascotas que puede pasear al mismo tiempo es de 3.
- El paseador no necesita estar en el “Registro de paseadores de perros” de la Ciudad de Buenos Aires, pero de hacerlo y tener la habilitación al día, el límite de mascotas que puede pasear al mismo tiempo incrementa de 3 a 8.
- El paseador debe cumplir con una pequeña capacitación/evaluación para demostrar su aptitud con el servicio.
- La identidad del paseador será verificada al momento del registro, confirmando que es una persona real con un domicilio conocido.

El servicio propuesto en este trabajo no abarca la etapa de verificación. Sin embargo, se busca incluirla en su versión final de alguna de las siguientes maneras:

- Utilizando el servicio de un tercero como Blockscore o NetVerify.
- De forma manual y personal, solicitando el documento de identidad y recibo de un servicio a su nombre, en donde también figure el domicilio particular declarado.

Para el propósito de la demostración del funcionamiento del servicio, el usuario de prueba “paseador” tendrá el estado “verificado” configurado previamente desde la base de datos.

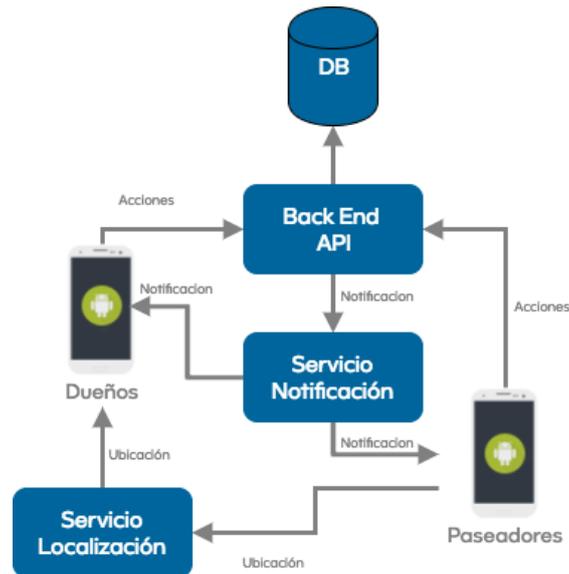
## Arquitectura

Basado en los requerimientos funcionales y no funcionales se identificó una arquitectura que podrá satisfacer a una gran cantidad de Usuarios compuesta por:

- Aplicación mobile.
- Aplicación Back-End.

- Servicio de Localización.
- Servicio de Notificaciones.
- DB o Base de Datos.

El sistema deberá proveer a los aplicativos del Dueño y Paseador por igual de herramientas administrativas que faciliten al usuario obtener paseos y hacer un seguimiento del mismo.



### Aplicación Back-End y Base De Datos

Ambos componentes facilitarán a las aplicaciones móviles de herramientas administrativas propias del modelo de negocio desarrollado. Entre sus responsabilidades se encuentran:

- Registro de datos del usuario (Email, Imagen, Nombre, Token de Notificación)
- Creación, inicio, finalización y control de paseos.

### Servicio de Notificación

Con el objetivo de ofrecer una comunicación fluida entre los paseadores y sus clientes sin importar que estén utilizando o no la aplicación, los dispositivos móviles centralizarán el manejo de estados en segundo plano.

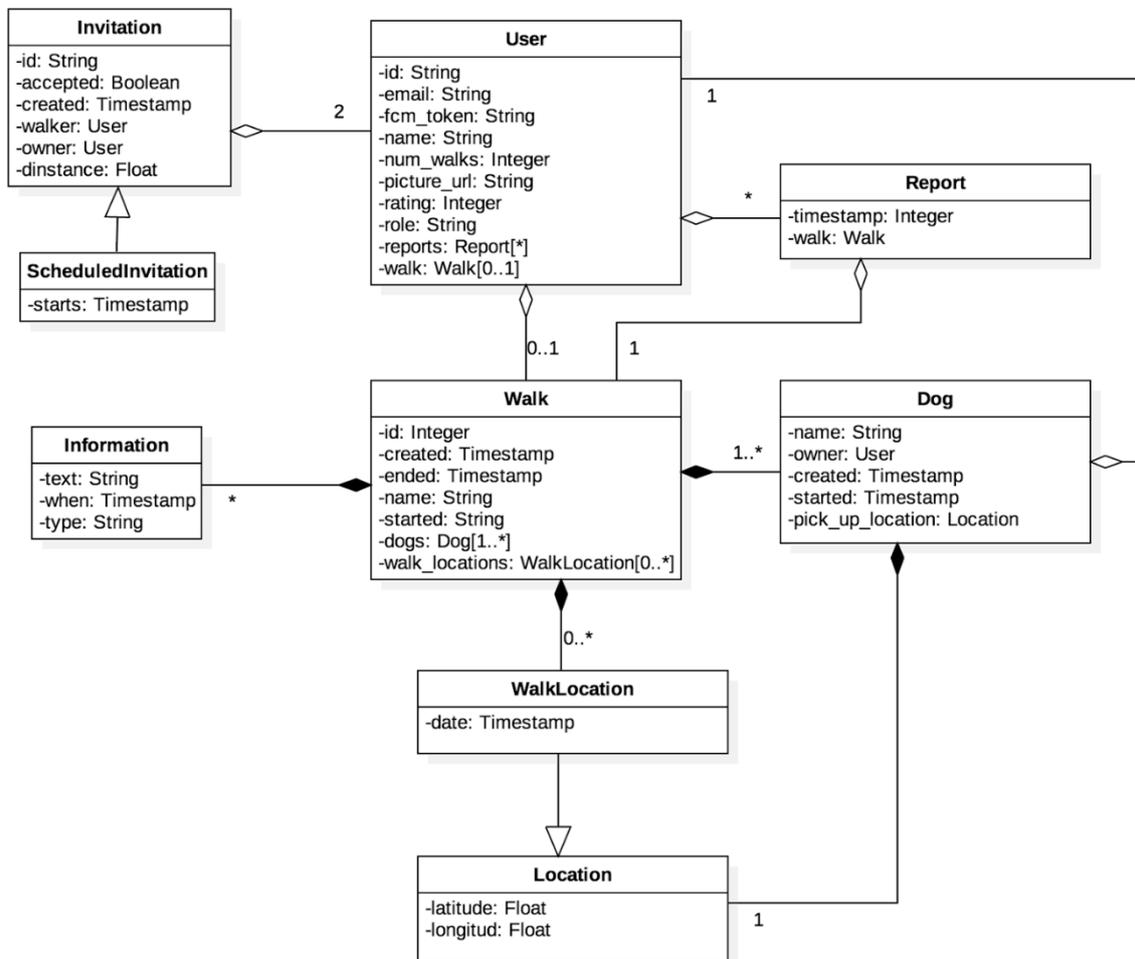
Google, para poder proveer a los desarrolladores de dichas herramientas, construyó Firebase Cloud Messaging, una multi-plataforma Firebase que permite enviar mensajes ilimitados sin costo alguno.

### Servicio de Localización

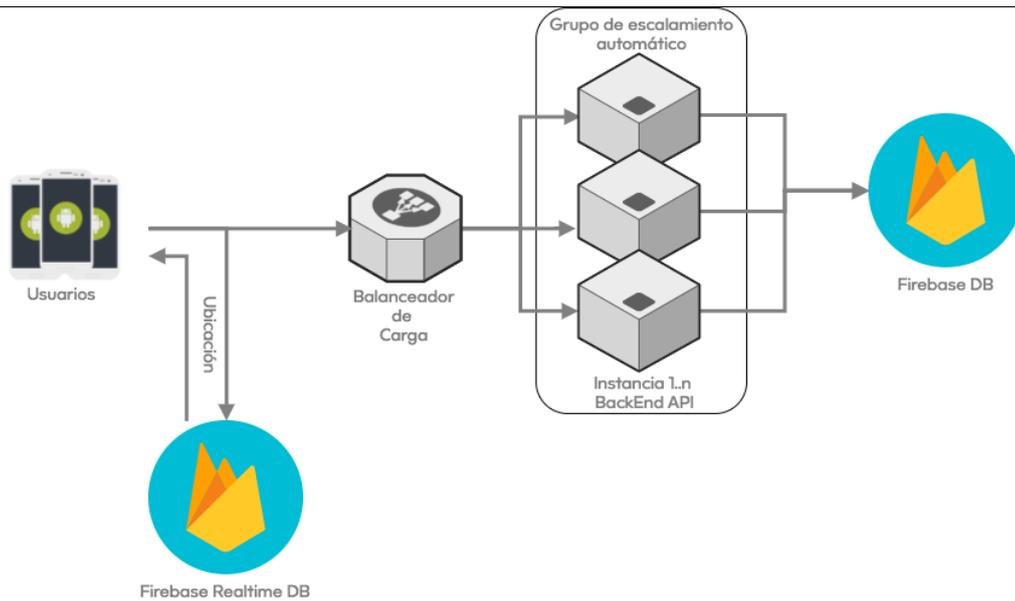
Un componente fundamental de este proyecto es aquel que permite al Dueño saber la ubicación de un Paseador camino a su casa y de su mascota a lo largo de su paseo. Debe cumplir con dos características:

- **Frecuencia de recolección apropiada:** Permite al Dueño reconocer que la información es actual y verídica (mínimo cada 5 metros).
- **Baja latencia entre recolección y actualización:** El Dueño podrá confiar que en la ubicación del Paseador ya sea cuando se realiza el retiro de la mascota como cuando es devuelta a su dueño después del paseo

### Estructura de Datos



### Vista de Despliegue



## Aplicación Back-end

Para soportar el funcionamiento de la aplicación mobile necesitamos el apoyo de una aplicación back-end accesible desde cualquier lugar mediante el uso de internet. La misma va a proveer el manejo de la autenticación del usuario, la base de datos, y el envío de notificaciones push a los dispositivos. Las tecnologías utilizadas para realizar la misma se eligieron en base a los siguientes criterios:

- **Facilidad de uso:** Herramientas y lenguajes de programación conocidos o con una curva de aprendizaje baja.
- **Asincronismo:** Soportar y servir múltiples pedidos de clientes al mismo tiempo.
- **Disponibilidad:** Utilizaremos proveedores reconocidos en el ambiente a nivel global por su calidad y servicio, como Amazon, Google o Heroku. El porcentaje de disponibilidad que normalmente ofrecen en sus planes es del 99%.
- **Escalabilidad:** En caso de aumentar la demanda de la aplicación Back-end, el servidor hospedado en Heroku puede incrementar el soporte replicando la aplicación en nuevas instancias (horizontal) o aumentando el procesamiento de las instancias existentes (vertical). Las consultas son distribuidas de manera transparente para el usuario a las instancias menos ocupadas gracias al Balanceador de Carga que provee Heroku.

- **Seguridad y mantenimiento:** Tecnologías nuevas o actuales, con actualizaciones frecuentes ante problemas de seguridad, buena documentación y soporte.

En base a lo anterior, elegimos utilizar NodeJS con Express para manejar el enrutamiento de la API (*Application Programming Interface*) y Firebase con sus módulos de autenticación, mensajería y base de datos en tiempo real para soportarla:

NodeJS es una tecnología basada en JavaScript y el modelo asíncrono de “funciones callback” en donde una consulta no bloquea el hilo de ejecución sino que continúa con la ejecución del resto del código hasta que se reciba el resultado.

Express es un framework para NodeJS que provee el fácil manejo del enrutamiento en una aplicación web. Incluye el concepto de “middleware” o filtro a utilizar antes de que una ruta sea alcanzada, como en el caso de requerir que el usuario esté autenticado y autorizado antes de utilizar una ruta protegida.

Firebase empezó siendo una solución simple a las bases de datos utilizadas en las aplicaciones modernas. Integraba una base de datos No-SQL presente “en la nube” con el “modelo de eventos” en donde si un cliente realiza cambios, los mismos se reflejarían en el resto de los clientes en cuestión de segundos. En el año 2016 fue adquirida por Google y se la mejoró ofreciéndola como un paquete de soluciones para el desarrollador. Hoy buscan unificar las herramientas requeridas por el software moderno y las ofrecen para plataformas web y mobile agrupadas en los siguientes módulos:

- **Realtime Database:** Provee una base de datos no relacional, con actualizaciones en tiempo real a los clientes subscriptos a eventos.
- **Authentication:** Provee de una manera simple la autenticación del usuario utilizando email y contraseña o los principales proveedores de identidad como son Google, Facebook y Twitter.
- **Cloud Messaging:** Provee el envío de mensajes o notificaciones push a los clientes.
- **Storage:** Provee el almacenamiento de archivos en la nube.
- **Hosting:** Provee el hosting de aplicaciones web.
- **Crash Reporting:** Provee el reporte automático de cualquier error o excepción ocurrida y no capturada por los clientes, con el fin de identificar su causa y mejorar el código.

- **Remote Config:** Provee el cambio a distancia de valores simples de configuración de la aplicación.
- **Test Lab:** Provee una “granja” de dispositivos reales en donde se pueden ejecutar pruebas de UI y obtener los reportes.
- **Ad Mob:** Provee anuncios para monetizar el uso de la aplicación.
- **App Indexing:** Provee facilidades para los equipos de marketing al momento de anunciar novedades o lanzamientos de las aplicaciones.
- **Invites:** Provee enlaces dirigidos a una pantalla particular de la aplicación con un estado dado. Utilizado en conjunto con App Indexing logra aumentar el porcentaje de convertibilidad de la aplicación.

### Especificación de la API

La aplicación back-end define un conjunto de rutas utilizadas por los clientes móviles:

<b>POST /login</b>			
<b>Descripción - CU001 - CU002</b>			
Registra los datos (identificador, email, nombre, imagen de perfil) del usuario en el sistema.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Cuerpo	-	Información extra del usuario.	{"fcm_token": "CO6FDhO8..."}
	fcm_token	Token para Notificaciones Push	CO6FDhO8...
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Paseadores Cercanos	{"id":"qeCEZVUaC7QnWbUQnG0JbF1Fw032","name":"owner@walkerr","picture_url":"http://placekitten.com/160/139","role":"owner"}

			, "email": "owner@walkers.com" }
	email	Email del Paseador	owner@walkers.com
	name	Nombre del Paseador	owner@walkerr
	picture_url	Imagen Del Paseador	http://placekitten.com/160/139
	id	Identificador	qeCEZVUaC7QnWbUQnG0Jb...
	role	Rol del Usuario [Owner Walker]	owner
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{ "status": 500, "message": "DB Connection Error." }
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>GET /walkers</b>			
<b>Descripción - CU005</b>			
Busca los paseadores cercanos al usuario.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Parámetro	range	Radio de Búsqueda (Kilómetros)	10
Parámetro	lat	Latitud (grados decimales)	-34.1672
Parámetro	lng	Longitud (grados decimales)	-58.9588983
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJIUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo

200	-	Paseadores Cercanos	[{"email":"walker@walkers.com", "name":"Walker","picture_url":"http://placekitten.com/150/155","rating":"3.29","id":"9eZ8FYIre3NntxDTCO6FDhO8bLe2","distance":0, "location":[-34.1672,-58.9588983]}]
	email	Email del Paseador	walker@walkers.com
	name	Nombre del Paseador	Walker
	picture_url	Imagen Del Paseador	http://placekitten.com/150/155
	id	Identificador	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	distance	Distancia al Usuario (Kilómetros)	0
	location	Arreglo [Latitud, Longitud]	[-34.1672,-58.9588983]
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>POST /walks/request</b>			
<b>Descripción - CU006</b>			
Un usuario “Dueño” solicita a un “Paseador” el paseo de su mascota.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Cuerpo	-	Información del paseador.	{"walker_id": "9eZ8FYIre3NntxDTCO6FDhO8bLe2"}

	walker_id	ID de paseador solicitado	9eZ8FYIre3NntxDTCO6FDhO8bLe2
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Solicitud Creada	{ }
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>POST /walks/confirm</b>			
<b>Descripción - CU007</b>			
En base a una "Solicitud de Paseo" el "Paseador" decide aceptarla o no.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Cuerpo	-	Respuesta de la solicitud.	{"owner_id":"qeCEZVUaC7QnWbUQnG0JbF1Fw032","accepted":true"}
	owner_id	ID del dueño solicitante.	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	accepted	Respuesta de la solicitud.	true
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo

200	-	Nuevo Paseo	{"walker_id":"9eZ8FYIre3NntxDTCO6FDhO8bLe2","dogs":{"qeCEZVUaC7QnWbUQnG0JbF1Fw032":{"pick_up_location":[-34.1694115,-58.9642273],"name":"owner@walkerr","created":1488376493}},"started":1488376493,"id":"-Ke98jS89KsgaRhJPCAx"}
	id	ID del Paseo	-Ke98jS89KsgaRhJPCAx
	walker_id	ID del Paseador	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	name	Nombre del Paseador	owner@walkerr
	started	Fecha de inicio del Paseo (Unix Timestamp)	1488376493
	dogs	Owners {owner_id} -> Información del Owner	"dogs":{"qeCEZVUaC7QnWbUQnG0JbF1Fw032":{"pick_up_location":[-34.1694115,-58.9642273],"name":"owner@walkerr","created":1488376493}}
	dogs.{owner_id}.pick_up_location	Punto de encuentro [Latitud, Longitud]	[-34.1694115,-58.9642273]
	dogs.{owner_id}.name	Nombre del Dueño	owner@walkerr
	created	Fecha de creación del Paseo (Unix Timestamp)	1488376493
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>POST /schedule/request</b>	
-------------------------------	--

<b>Descripción - CU009</b>			
Un usuario “Dueño” envía una Propuesta de paseo programado a un usuario “Paseador”.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Cuerpo	-	Información del paseador.	<code>{"walker_id": "9eZ8FYIre3NntxDTCO6FDhO8bLe2", "date": 1493722801}</code>
	walker_id	ID de paseador solicitado	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	date	Hora elegida (Unix Timestamp)	1493722801
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Solicitud Creada	<code>{"owner_id": "qeCEZVUaC7QnWbUQnG0JbF1Fw032", "walker_id": "9eZ8FYIre3NntxDTCO6FDhO8bLe2", "created": 1493868871, "distance": 67.37, "starts": 1493722800, "id": "e3a469379fb85520ebacbe6def71f2c19bbc9823"}</code>
	id	ID del Paseo	e3a469379fb85520ebacbe6def71f2c19bbc9823
	owner_id	ID de dueño	qeCEZVUaC7QnWbUQnG0JbF1Fw032
	walker_id	ID de paseador solicitado	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	created	Fecha de creación del Paseo (Unix Timestamp)	1488376493
	distance	Distancia Paseador-Paseo	67.37

		(metros)	
	starts	Hora elegida (Unix Timestamp)	1493722800
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>POST /schedule/confirm</b>			
<b>Descripción - CU010</b>			
Aceptación de propuesta de paseo programado			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Cuerpo	-	Información del paseador.	{"scheduled_invitation_id": "e3a469379fb85520ebacbe6def71f2c19bbc9823", "accepted": true}
	scheduled_invitation_id	ID de paseador solicitado	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	accepted	True si acepta el paseo.	True
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Solicitud Creada	{"owner_id": "qeCEZVUaC7QnWbUQnG0JbF1Fw032", "walker_id": "9eZ8FYIre3NntxDTCO6FDhO8bLe2", "created": 149386}

			8871,"distance":"67.37","starts":1493722800,"accepted":true,"id":"e3a469379fb85520ebacbe6def71f2c19bbc9823"}
	id	ID del Paseo	e3a469379fb85520ebacbe6def71f2c19bbc9823
	owner_id	ID de dueño	qeCEZVUaC7QnWbUQnG0JbF1Fw032
	walker_id	ID de paseador solicitado	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	created	Fecha de creación del Paseo (Unix Timestamp)	1488376493
	distance	Distancia Paseador-Paseo (metros)	67.37
	accepted	True si el paseo fue aceptado	true
	starts	Hora elegida (Unix Timestamp)	1493722800
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>POST /schedule/start</b>	
<b>Descripción - CU011</b>	
Inicio de un paseo programado	
<b>Petición</b>	

Tipo	Nombre	Descripción	Ejemplo
Cuerpo	-	Información del paseador.	<code>{"scheduled_invitation_id": "e3a469379fb85520ebacbe6def71f2c19bbc9823"}</code>
	<code>scheduled_invitation_id</code>	ID de paseador solicitado	<code>9eZ8FYIre3NntxDTCO6FDhO8bLe2</code>
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Nuevo Paseo	<code>{"walker_id": "9eZ8FYIre3NntxDTCO6FDhO8bLe2", "dogs": {"qeCEZVUaC7QnWbUQnG0JbF1Fw032": {"pick_up_location": [-34.1694115, -58.9642273], "name": "owner@walkerr", "created": 1488376493}}, "started": 1488376493, "id": "-Ke98jS89KsgaRhJPCAx"}</code>
	<code>id</code>	ID del Paseo	<code>-Ke98jS89KsgaRhJPCAx</code>
	<code>walker_id</code>	ID del Paseador	<code>9eZ8FYIre3NntxDTCO6FDhO8bLe2</code>
	<code>name</code>	Nombre del Paseador	<code>owner@walkerr</code>
	<code>started</code>	Fecha de inicio del Paseo (Unix Timestamp)	<code>1488376493</code>
	<code>dogs</code>	Owners {owner_id} -> Información del Owner	<code>"dogs": {"qeCEZVUaC7QnWbUQnG0JbF1Fw032": {"pick_up_location": [-34.1694115, -58.9642273], "name": "owner@walkerr", "created": 1488376493}}</code>
	<code>dogs.{owner_id}.pick_up_location</code>	Punto de encuentro [Latitud, Longitud]	<code>[-34.1694115, -58.9642273]</code>
	<code>dogs.{owner_id}.name</code>	Nombre del Dueño	<code>owner@walkerr</code>

	created	Fecha de creación del Paseo (Unix Timestamp)	1488376493
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>DELETE /schedule/{scheduled_invitation_id}</b>			
<b>Descripción - CU010</b>			
Un usuario “Dueño” cancela una Propuesta de paseo programado previamente hecha a un usuario “Paseador”.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Solicitud Eliminada Correctamente	{}
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>GET /walks/{walk_id}</b>
-----------------------------

<b>Descripción - CU014 - CU016</b>			
Busca un paseo determinado.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Paseo	<pre>{ "dogs": { "D6KFBd4HfZdCbTU4X3J2mGggkit1": { "created": 1488651230, "ended": 1488651247, "name": "User Owner 2", "pick_up_location": [-34.1694115, -58.9642273], "started": 1488651245 }, "qeCEZVUaC7QnWbUQnG0JbF1Fw032": { "created": 1488651226, "ended": 1488651250, "information": { "text": "Text Owner 1", "when": 1488651242 } }, "name": "owner@walkerr", "pick_up_location": [-34.1694115, -58.9642273], "started": 1488651240 } }, "ended": 1488651250, "started": 1488649788, "walker_id": "9eZ8FYlre3NntxDTCO6FDhO8bLe2", "id": "-KePRGdiGcwX0n21GGPB" }</pre>
	id	ID del Paseo	-KePRGdiGcwX0n21GGPB
	walker_id	ID del Paseador	9eZ8FYlre3NntxDTCO6FDhO8bLe2
	name	Nombre del Paseador	owner@walkerr
	started	Fecha de inicio del Paseo (Unix Timestamp)	1488649788
	ended	Fecha de fin del Paseo (Unix Timestamp)	1488651250

	dogs	Owners {owner_id} -> Información de los paseos por Dueño	"dogs":{"qeCEZVUaC7QnWbUQnG0JbF1Fw032":{...},"D6KFBd4HfZdCbTU4X3J2mGggkit1":{...}}
	dogs.{owner_id}	Información del paseo	{"created":1488651226,"ended":1488651250,"information":[{"text":"Text Owner 1","when":1488651242}], "name":"owner@walkerr","pick_up_location":[-34.1694115,-58.9642273],"rating":3,"started":1488651240}
	dogs.{owner_id}.pick_up_location	Punto de encuentro [Latitud, Longitud]	[-34.1694115,-58.9642273]
	dogs.{owner_id}.name	Nombre del Dueño	owner@walkerr
	dogs.{owner_id}.created	Fecha de creación del Paseo (Unix Timestamp)	1488651226
	dogs.{owner_id}.started	Fecha de Inicio del Paseo (Unix Timestamp)	1488651240
	dogs.{owner_id}.ended	Fecha de fin del Paseo (Unix Timestamp)	1488651250
	dogs.{owner_id}.information	Información del paseo.	[{"text":"Text Owner 1","when":1488651242}]
	dogs.{owner_id}.information[0].text	Texto Enviado	Text Owner 1
	dogs.{owner_id}.information[0].when	Fecha de Envío	1488651242
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500

	message	Mensaje de Error	DB Connection Error.
--	---------	------------------	----------------------

<b>POST /walks/{walk_id}/dogs/{owner_id}/start</b>	
--	--

**Descripción - CU015**

Inicia el paseo de una mascota determinada.

**Petición**

Tipo	Nombre	Descripción	Ejemplo
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...

**Respuesta**

Código	Nombre	Descripción	Ejemplo
200	-	Paseo Actualizado Ver GET /walks/{walk_id}	{"dogs":{"D6KFBd4HfZdCbTU4X3J2mGggit1":{"... "started":1488651240}}...}
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

**POST /walks/{walk\_id}/dogs/{owner\_id}/information**

**Descripción - CU016**

Envía información relativa a un paseo en curso.

**Petición**

Tipo	Nombre	Descripción	Ejemplo
Cuerpo	-	Información del paseo.	{"text": "Text Owner 1", "when": 1486937357}
	text	Texto Enviado	Text Owner 1
	when	Fecha de Envío	1488651242
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Paseo Actualizado Ver GET /walks/{walk_id}	{"dogs":{"D6KFBd4HfZdCbTU4X3J2mGgkit1":{"...": "ended": 1488651247}}...": "ended": 1488651247}
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>POST /walks/{walk_id}/dogs/{owner_id}/end</b>			
<b>Descripción - CU018 - CU019</b>			
Finaliza el paseo de una mascota determinada. Si es el último paseo del paseador finaliza el paseo.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			

Código	Nombre	Descripción	Ejemplo
200	-	Paseo Actualizado Ver GET /walks/{walk_id}	{"dogs":{"D6KFBd4HfZdCbTU4X3J2mGggkit1":{"...ended":1488651247}}...ended":1488651247}
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>PUT /walks/{walk_id}</b>			
<b>Descripción - CU020</b>			
Valoración de un paseo.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
Cuerpo	-	Valoración del paseo	{"rating":2}
	rating	Valoración entre [1-5]	2
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Paseo Valorado	
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB

			Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>GET /walks/{walk_id}/report</b>			
<b>Descripción - CU019</b>			
Obtiene el reporte de un paseo dado.			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Paseo	{ "walker_name": "walker@walk7e", "walk_id": "-KePRGdiGcwX0n21GGPB", "walker_id": "9eZ8FYlre3NntxDTCO6FDhO8bLe2", "points": [{"date": 1488651240, "lat": -34.1672983, "lng": -58.9588983}, {"date": 1488651245, "lat": -34.1672983, "lng": -58.9588983}, {"date": 1488651247, "lat": -34.1672983, "lng": -58.9588983}, {"date": 1488651250, "lat": -34.1672983, "lng": -58.9588983}], "started": 1488649788, "ended": 1488651250, "information": [{"text": "Text Owner 1", "when": 1488651242}], "rating": 3, "dogs": ["User Owner 2", "owner@walkerr"], "dogs": ["owner@walkerr"]} }
	walk_id	ID del Paseo	-KePRGdiGcwX0n21GGPB

	walker_id	ID del Paseador	9eZ8FYIre3NntxDTCO6FDhO8bLe2
	walker_name	Nombre del Paseador	owner@walkerr
	started	Fecha de inicio del Paseo (Unix Timestamp)	1488649788
	ended	Fecha de fin del Paseo (Unix Timestamp)	1488651250
	rating	Calificación del paseo	3
	dogs	Nombre de los Dueños Paseados (solo Paseadores)	"dogs": ["User Owner 2", "owner@walkerr"]
	information	Información del paseo. (Solo Dueños)	{"text": "Text Owner 1", "when": 1486937357}
	text	Texto Enviado	Text Owner 1
	when	Fecha de Envío	1488651242
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

<b>GET /walks</b>			
<b>Descripción - CU021</b>			
Obtiene los paseos realizados por el Usuario			
<b>Petición</b>			
Tipo	Nombre	Descripción	Ejemplo

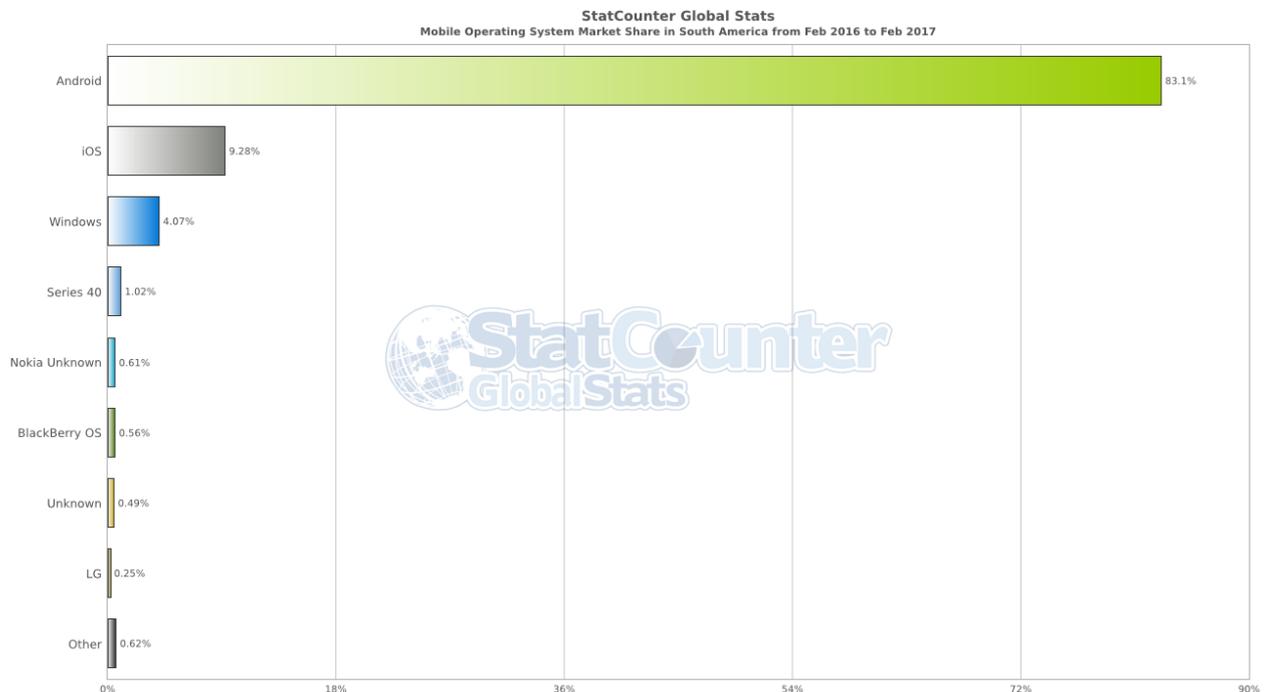
Encabezado	Authorization	Sesión de Usuario	Bearer eyJhbGciOiJSUzI1Ni...
<b>Respuesta</b>			
Código	Nombre	Descripción	Ejemplo
200	-	Paseo	<pre>[{"walker_name":"walker@walk7e","walk_id":-KePRGdiGcwX0n21GGPB","walker_id":"9eZ8FYlre3NntxDTCO6FDhO8bLe2","points":[{"date":1488651240,"lat":-34.1672983,"lng":-58.9588983},{"date":1488651245,"lat":-34.1672983,"lng":-58.9588983},{"date":1488651247,"lat":-34.1672983,"lng":-58.9588983},{"date":1488651250,"lat":-34.1672983,"lng":-58.9588983}], "started":1488649788,"ended":1488651250,"information": [{"text":"Text Owner 1","when":1488651242}], "rating": 3,"dogs":["User Owner 2","owner@walkerr"],"dogs":["owner@walkerr"]}]</pre>
	walk_id	ID del Paseo	-KePRGdiGcwX0n21GGPB
	walker_id	ID del Paseador	9eZ8FYlre3NntxDTCO6FDhO8bLe2
	walker_name	Nombre del Paseador	owner@walkerr
	started	Fecha de inicio del Paseo (Unix Timestamp)	1488649788
	ended	Fecha de fin del Paseo (Unix Timestamp)	1488651250
	rating	Calificación del paseo	3
	dogs	Nombre de los Dueños Paseados (solo Paseadores)	"dogs": ["User Owner 2", "owner@walkerr"]
	information	Información del paseo. (Solo Dueños)	{"text": "Text Owner 1", "when": 1486937357}

	text	Texto Enviado	Text Owner 1
	when	Fecha de Envío	1488651242
401	-	Error de Autenticación	Unauthorized
501	-	Error Desconocido	{"status":500,"message":"DB Connection Error."}
	status	Código de Error	500
	message	Mensaje de Error	DB Connection Error.

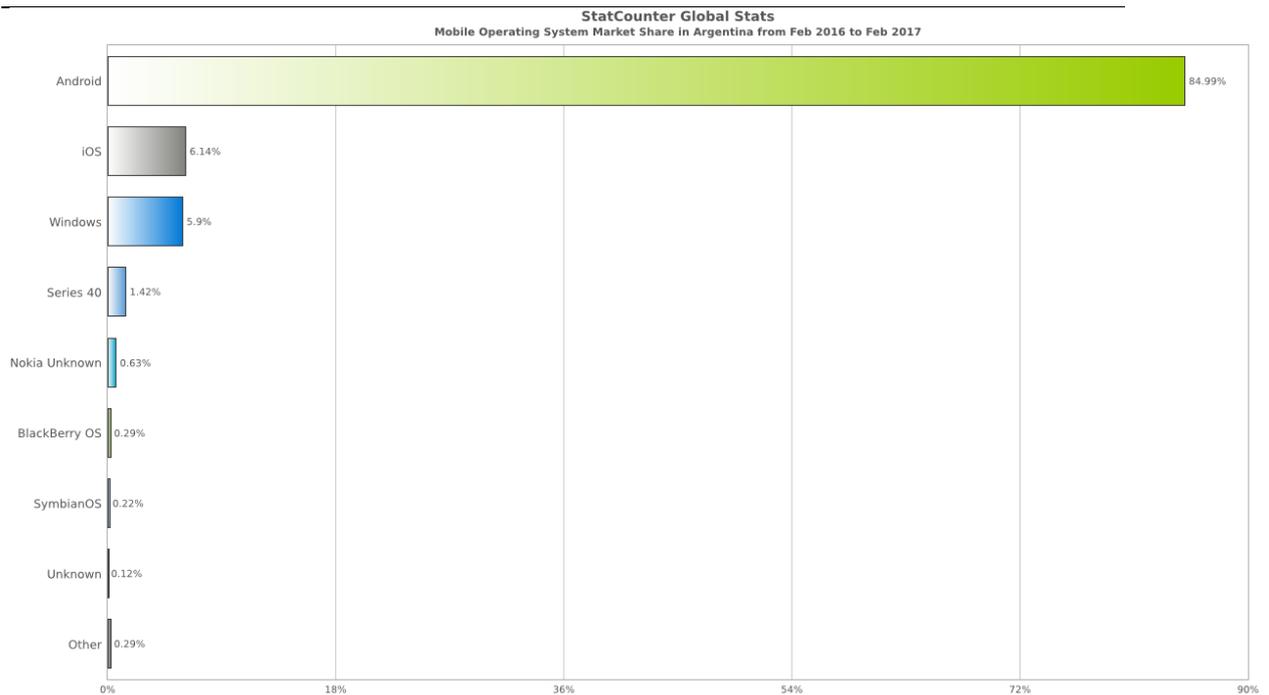
### Aplicación Mobile

Al momento de comenzar el desarrollo de la aplicación mobile, la primera incógnita que surgió fue a qué sistema operativo o plataforma orientarla. Para eso acudimos a las estadísticas de uso de cada plataforma mobile utilizada en los últimos 12 meses y estos fueron los resultados:

Para América del Sur:



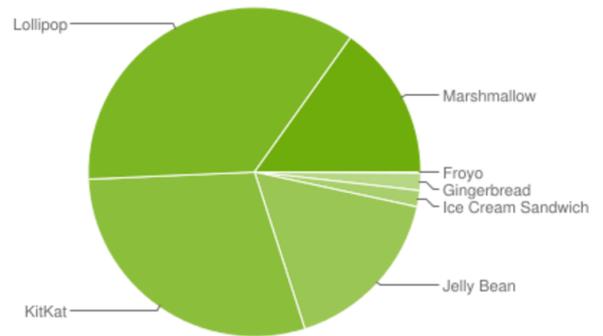
Para Argentina:



Analizando los gráficos descubrimos que más del 80% del mercado utiliza teléfonos con la plataforma Android, mientras que menos del 10% utiliza iOS. Dado que queremos que la aplicación sea utilizada principalmente en Argentina, la misma será realizada en la plataforma Android con el fin de obtener la mayor audiencia posible.

El siguiente paso fue averiguar qué versiones del sistema operativo anteriores a la actual todavía son utilizadas por las personas. Esto nos permitió utilizar un SDK (*Software Development Kit*) más completo y seguro sin dejar de lado a aquellas personas con teléfonos de una generación pasada. Los datos consultados corresponden a los publicados por Google y representan la fragmentación de versiones del sistema operativo Android a nivel mundial.

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.6%
4.1.x	Jelly Bean	16	6.0%
4.2.x		17	8.3%
4.3		18	2.4%
4.4	KitKat	19	29.2%
5.0	Lollipop	21	14.1%
5.1		22	21.4%
6.0	Marshmallow	23	15.2%



Datos recopilados durante un período de 7 días hasta el 1 de agosto de 2016.  
No se muestran versiones con una distribución inferior al 0,1%.

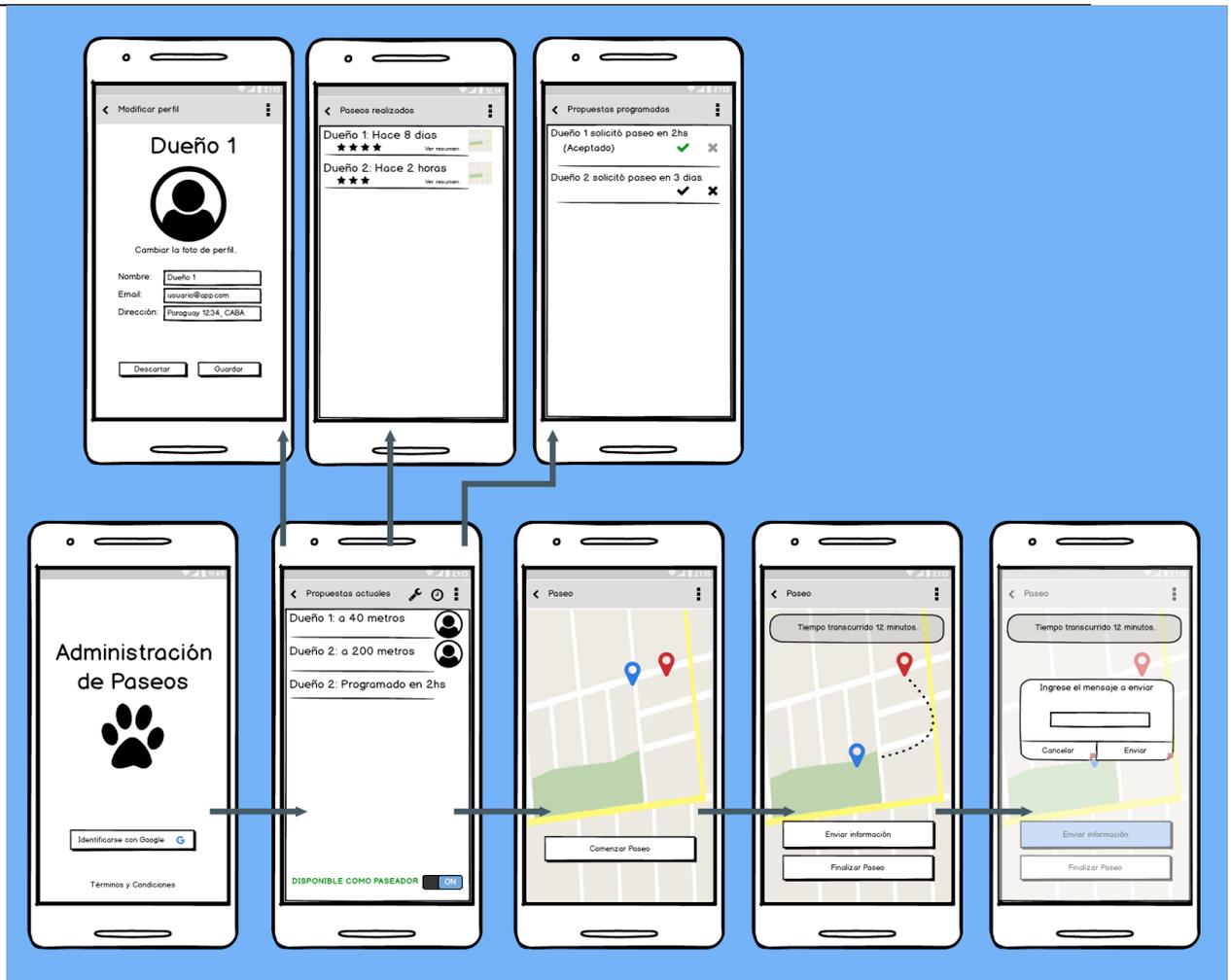
Analizando la tabla decidimos soportar el sistema operativo Android en sus versiones Jelly Bean (API 16) y superiores, lo que se traduce en alcanzar más del 96% de los dispositivos en el caso de que la aplicación sea distribuida mundialmente.

Una vez establecida la plataforma y su versión mínima el siguiente paso fue pensar las pantallas que requería la aplicación, para lo cual tomamos el listado de requerimientos y realizamos estos mockups:

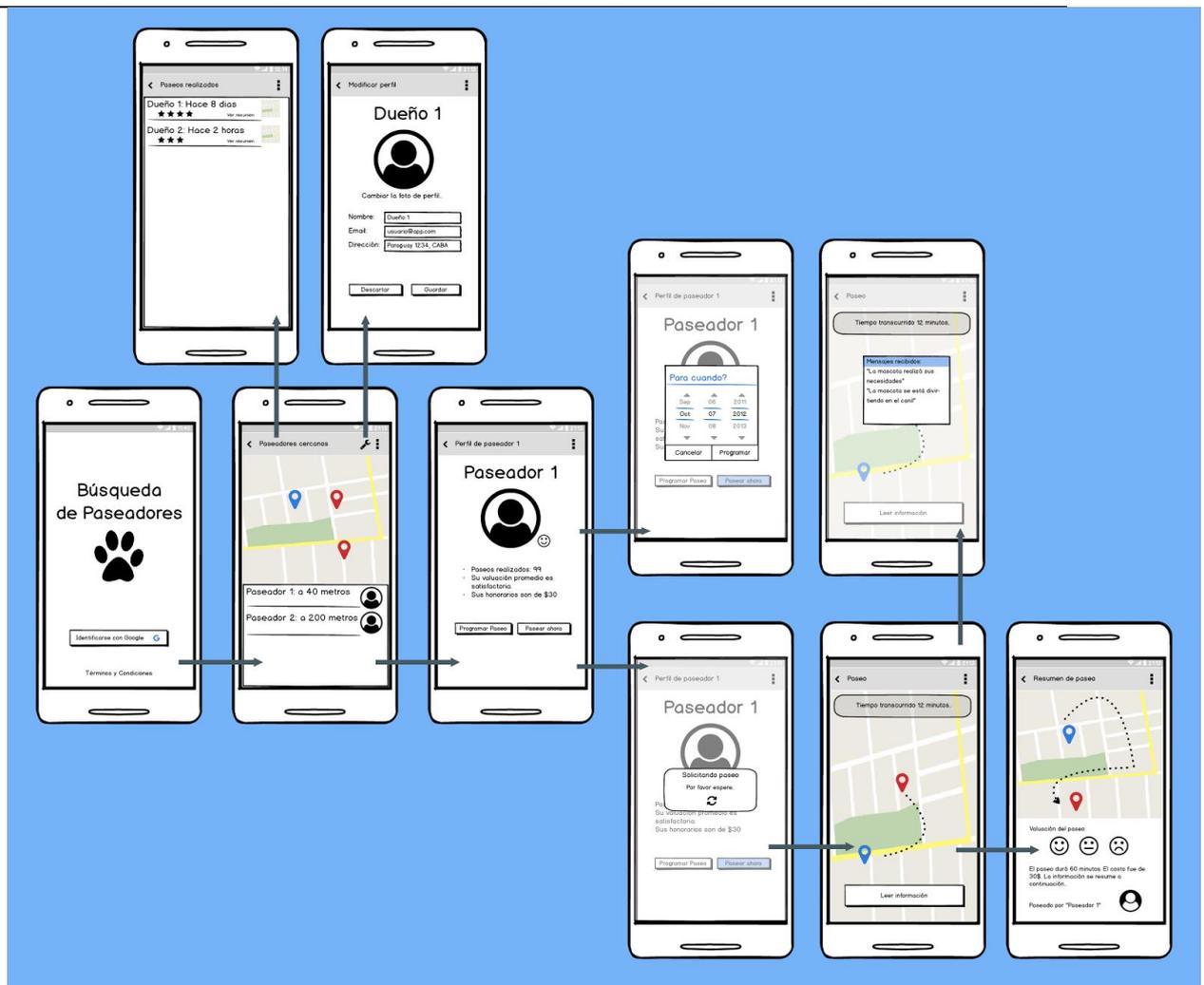
- Pantalla de bienvenida con inicio de sesión de usuario.
- Pantalla de configuración de perfil del usuario.
- Pantalla principal con lista de usuarios Paseadores de mascotas cercanos.
- Pantalla principal con lista de usuarios Dueños de mascotas cercanos, con deseo de encontrar un paseador de mascota.
- Pantalla con lista de propuestas de paseo programadas.
- Pantalla de paseo con visualización de los datos del mismo en un mapa.
- Pantalla de paseos realizados.
- Pantalla de reporte de paseo realizado.

## Diagrama de Flujo de UX

### Variante “Paseadores”



**Variante “Dueños”**



Analizado cuidadosamente cada caso, encontramos que mucha funcionalidad de la aplicación es compartida entre el rol del paseador y del dueño de la mascota. Decidimos entonces utilizar “Android Flavors” para tener variantes de una misma aplicación con pequeñas diferencias de código o de componentes gráficos entre sí. De esta forma, obtenemos dos aplicaciones listas para instalar y particularmente enfocadas en los distintos roles, sin necesidad de repetir código ni de tener todo el negocio en una única aplicación.

### Lista de requerimientos encontrados

Los siguientes requerimientos surgieron luego de aplicar la técnica Design Thinking:

- **Autenticación de usuarios:** El sistema debe permitir la identificación de usuarios utilizando un email y una contraseña como medida de seguridad.
- **Administración de usuarios:** El sistema debe permitir la creación y modificación de los datos del perfil asociado a un usuario. El sistema deberá tener en cuenta que no exista ya un usuario creado con el mismo email

ingresado. En el caso de la modificación de los mismos, el sistema deberá comprobar primero su existencia.

- **Búsqueda de paseadores a demanda:** El sistema debe permitir la búsqueda de paseadores de mascotas a demanda, y el envío y la aceptación de propuestas de paseo.
- **Programación de paseos:** El sistema debe permitir la programación de paseos a futuro con un paseador y la aceptación de la propuesta realizada al mismo.
- **Valoración de paseos realizados:** El sistema debe permitir la valoración de los paseos realizados con motivo de mejorar continuamente la calidad del servicio ofrecida a los usuarios.
- **Consulta de paseos realizados:** El sistema debe permitir la consulta del reporte de todos los paseos realizados en el pasado.

## Identificación de los casos de uso

### Usuarios

1. Inicio de sesión de Usuario
2. Alta de Usuario “Dueño”
3. Modificación del perfil de Usuario
4. Consulta de perfil de paseador

### Servicio a demanda

5. Búsqueda de paseadores a demanda
6. Envío de propuesta de paseo a demanda
7. Aceptación de propuesta de paseo a demanda

### Servicio programado

8. Búsqueda de Propuestas de paseo programadas
9. Envío de propuesta de paseo programado
10. Aceptación de propuesta de paseo programado
11. Inicio de paseo programado

### Paseos

12. Comienzo del paseo
13. Actualización de la posición del Paseador
14. Consulta de la posición del Paseador
15. Retiro de la mascota

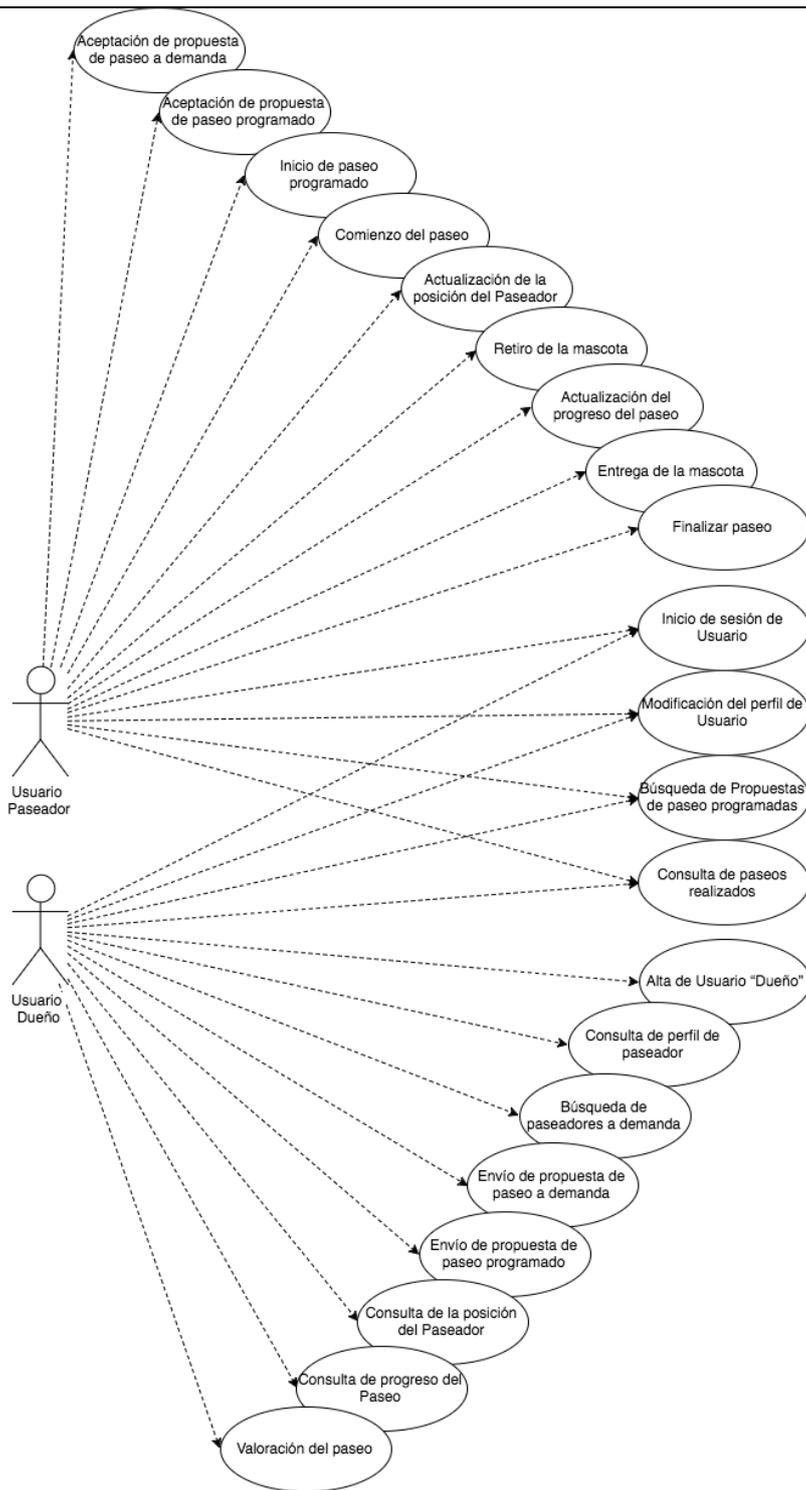
16. Actualización del progreso del paseo.
17. Consulta de progreso del Paseo.
18. Entrega de la mascota
19. Finalizar paseo
20. Valoración del paseo
21. Consulta de paseos realizados

**Trazabilidad entre requerimientos y casos de uso**

Requerimiento	Caso de Uso
Autenticación de usuarios	Inicio de sesión de Usuario
Administración de usuarios	Alta de Usuario “Dueño”
	Modificación del perfil de Usuario
	Consulta de perfil de paseador
Búsqueda de paseadores a demanda	Búsqueda de paseadores a demanda
	Envío de propuesta de paseo a demanda
	Aceptación de propuesta de paseo a demanda
Programación de paseo	Búsqueda de Propuestas de paseo programadas
	Envío de propuesta de paseo programado
	Aceptación de propuesta de paseo programado
	Inicio de paseo programado
Gestión del paseo	Comienzo del paseo
	Actualización de la posición del Paseador
	Consulta de la posición del Paseador
	Retiro de la mascota

	Actualización del progreso del paseo
	Consulta de progreso del Paseo
	Entrega de la mascota
	Finalizar paseo
Valoración del paseo	Valoración del paseo
Consulta de paseos realizados	Consulta de paseos realizados

**Diagrama de casos de uso**



**Listado de casos de uso detallados**

<b>Caso de Uso ID:</b>	CU001
<b>Caso de Uso Nombre:</b>	Inicio de sesión de Usuario

<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	20/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario “Dueño” o Usuario “Paseador”		
<b>Descripción:</b>	Un usuario se identifica dentro de la aplicación para poder utilizarla.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario no se encuentra autenticado en la aplicación.</li> <li>• Existe una instancia de usuario con el email dado en el sistema.</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario se encuentra autenticado en la aplicación.</li> <li>• El usuario cambia su estado a “Disponible”.</li> </ul>		
<b>Prioridad:</b>			
<b>Frecuencia de uso:</b>	Baja		
<b>Flujo Normal:</b>	Actor	Sistema	
	1. El usuario presiona el botón “Iniciar sesión”.	2. El sistema solicita el ingreso del email.	
	3. El usuario ingresa su email y presiona el botón “Enviar”.	4. El sistema busca los datos del usuario asociado al email recibido.	
		5. El sistema comprueba que el email se corresponde a un usuario “Dueño”.	
		6. El sistema devuelve el perfil del usuario “Dueño”, muestra la pantalla principal de la “aplicación dueño” y cambia el estado del usuario a “Disponible”.	
		7. FIN CU	
<b>Flujos Alternativos:</b>	Actor	Sistema	
Usuario “Paseador”		5. El sistema comprueba que el email se corresponde a un usuario “Paseador”.	
		6. El sistema devuelve el perfil del usuario “Paseador”,	

		muestra la pantalla principal de la “aplicación paseador” y cambia el estado del usuario a “Disponible”.
		7. FIN CU
<b>Pantallas:</b>		

<b>Caso de Uso ID:</b>	CU002		
<b>Caso de Uso Nombre:</b>	Alta de Usuario “Dueño”		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	20/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario “Dueño”		
<b>Descripción:</b>	Un usuario no existente se registra en la aplicación para poder utilizarla.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario no se encuentra autenticado en la aplicación.</li> <li>• No existe un usuario con el email dado en el sistema.</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se crea una nueva instancia de usuario “Dueño” con el email dado.</li> </ul>		

	<ul style="list-style-type: none"> <li>El usuario “Dueño” se encuentra autenticado en la aplicación.</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario presiona el botón “Iniciar sesión”.	2. El sistema solicita el ingreso del email.
	3. El usuario ingresa su email y presiona el botón “Enviar”.	4. El sistema busca los datos del usuario asociado al email recibido.
		5. El sistema verifica que el email no se corresponde a ningún usuario registrado.
		6. El sistema crea un nuevo usuario “Dueño” con el email dado y muestra la pantalla principal de la aplicación.
		7. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
--		



<b>Caso de Uso ID:</b>	CU003		
<b>Caso de Uso Nombre:</b>	Modificación del perfil de Usuario		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Dueño" o Usuario "Paseador"		
<b>Descripción:</b>	Un usuario modifica los datos asociados a su perfil.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario existe en el sistema.</li> <li>• El usuario se encuentra autenticado en la aplicación.</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Los datos asociados al perfil del usuario son modificados.</li> </ul>		
<b>Prioridad:</b>			
<b>Frecuencia de uso:</b>	Baja		
<b>Flujo Normal:</b>	Actor	Sistema	

	1. El usuario presiona el botón “Ver Perfil” en la pantalla principal de la aplicación.	2. El sistema muestra la pantalla de Perfil del Usuario.
	3. El usuario presiona el botón “Modificar”.	4. El sistema muestra la pantalla de modificación del perfil del usuario y espera el ingreso de nuevos datos.
	5. El usuario ingresa los nuevos valores en el campo de texto y presiona el botón “Guardar”.	6. El sistema valida los datos ingresados y actualiza el perfil del usuario.
		7. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Sin modificaciones	5. El usuario no ingresa datos nuevos y presiona el botón “Guardar”.	6. El sistema no realiza modificaciones y muestra la pantalla principal de la aplicación.
		7. FIN CU
<b>Pantallas:</b>		

<b>Caso de Uso ID:</b>	CU004		
<b>Caso de Uso Nombre:</b>	Consulta de perfil de paseador		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Dueño"		
<b>Descripción:</b>	Un usuario "Dueño" consulta la información asociada al perfil de un usuario "Paseador".		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Dueño" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Dueño" realizó una búsqueda de paseadores cercanos o solicitó ser contactado por algún paseador mediante el servicio programado.</li> </ul>		
<b>Post-condiciones:</b>			
<b>Prioridad:</b>			
<b>Frecuencia de uso:</b>	Baja		
<b>Flujo Normal:</b>	Actor	Sistema	
		1. El sistema informa que hay paseadores disponibles para tomar paseos.	
	2. El usuario selecciona un paseador y presiona el botón "Ver Perfil del Paseador".	3. El sistema muestra la pantalla de perfil del paseador, con datos como el nombre, puntaje o rating, y precio de sus paseos.	
		4. FIN CU	
<b>Flujos Alternativos:</b>	Actor	Sistema	
--			



<b>Caso de Uso ID:</b>	CU005		
<b>Caso de Uso Nombre:</b>	Búsqueda de paseadores a demanda		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Dueño"		
<b>Descripción:</b>	Un usuario "Dueño" consulta los paseadores de mascotas disponibles cercanos.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Dueño" tiene el estado "Disponible".</li> </ul>		
<b>Post-condiciones:</b>			
<b>Prioridad:</b>			

<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario presiona el botón “Buscar paseador”.	2. El sistema verifica que el usuario “Dueño” tiene el estado “Disponible”.
		3. El sistema muestra en pantalla los resultados de la búsqueda en un mapa junto a la ubicación física actual del usuario.
		4. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Paseo en marcha		1. El sistema verifica que el usuario “Dueño” tiene el estado “Paseo en marcha”.
		2. El sistema informa que en este momento no se puede realizar la búsqueda.
		3. FIN CU
<b>Pantallas:</b>		

<b>Caso de Uso ID:</b>	CU006		
<b>Caso de Uso Nombre:</b>	Envío de propuesta de paseo a demanda		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Dueño"		
<b>Descripción:</b>	Un usuario "Dueño" envía una propuesta de paseo a un usuario "Paseador".		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" realizó previamente una búsqueda de paseadores cercanos a su ubicación.</li> <li>• El usuario "Dueño" tiene el estado "Disponible".</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Paseador" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Paseador" tiene el estado "Disponible" o "Paseo en marcha".</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se crea una nueva instancia de Propuesta de paseo a demanda en el sistema, relacionando ambos usuarios.</li> <li>• El usuario "Paseador" recibe la Propuesta de paseo a demanda.</li> </ul>		
<b>Prioridad:</b>			
<b>Frecuencia de uso:</b>	Baja		
<b>Flujo Normal:</b>	Actor	Sistema	
	1. El usuario selecciona un usuario "Paseador" del listado para ver su perfil.		
	2. El usuario presiona el botón "Solicitar Paseo".	3. El sistema genera una nueva instancia de Propuesta de paseo a demanda, relacionando el usuario "Dueño", el usuario "Paseador".	
		4. El Sistema notifica al usuario "Paseador" que una nueva	

		Propuesta de paseo a demanda se encuentra disponible.
		5. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Cancelar propuesta	1. El usuario presiona el botón "Regresar".	2. El sistema muestra la pantalla principal.
		3. FIN CU
<b>Pantallas:</b>		

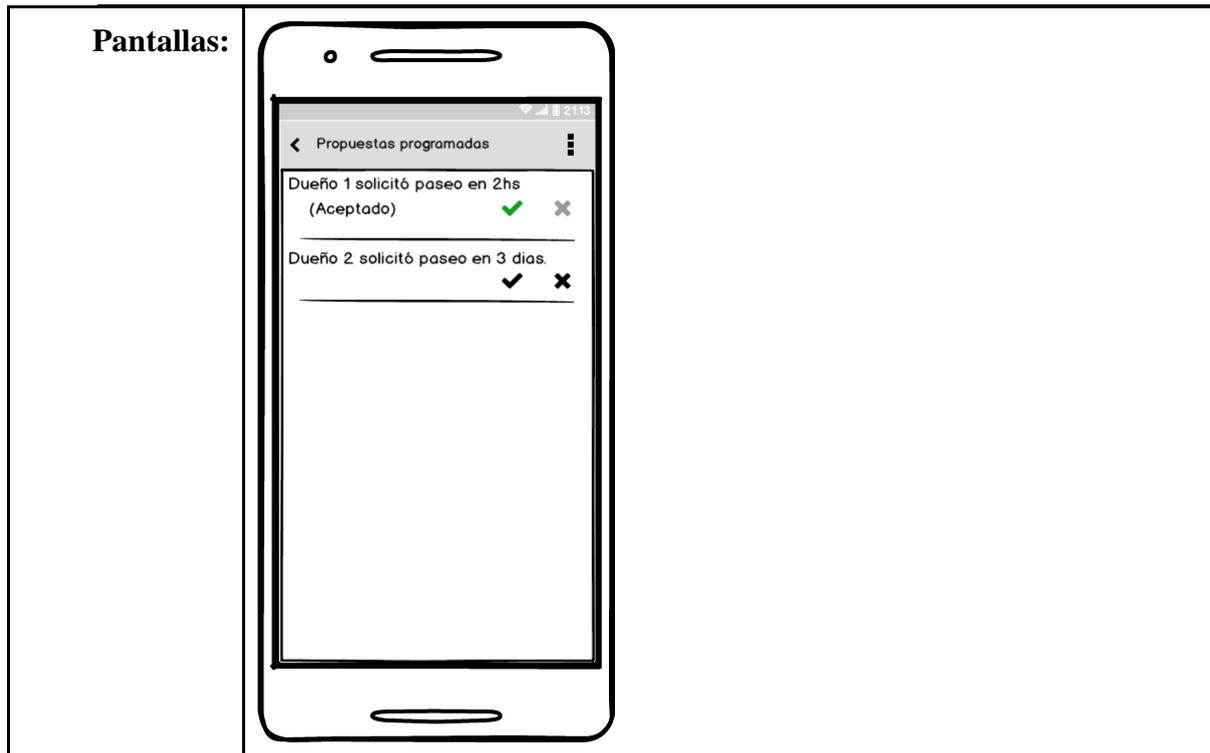
<b>Caso de Uso ID:</b>	CU007		
<b>Caso de Uso Nombre:</b>	Aceptación de propuesta de paseo a demanda		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		

<b>Descripción:</b>	Un usuario “Paseador” recibe una Propuesta de paseo a demanda por parte de un usuario “Dueño” y la acepta.	
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario “Dueño” existe en el sistema.</li> <li>• El usuario “Dueño” se encuentra autenticado en la aplicación.</li> <li>• El usuario “Dueño” tiene el estado “Disponible”.</li> <li>• El usuario “Paseador” existe en el sistema.</li> <li>• Hay instancias de Propuesta de paseo a demanda en el sistema, vinculando a ambos usuarios.</li> <li>• El usuario “Paseador” tiene el estado “Disponible” o “Paseo en marcha”.</li> </ul>	
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se crea una nueva instancia de Paseo con el estado “En marcha”.</li> <li>• La Propuesta de paseo a demanda cambia su estado a “Aceptada”.</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
		1. El sistema notifica al usuario la recepción de una nueva Propuesta de paseo a demanda, mostrando el nombre del dueño y la distancia a la que se encuentra del paseador.
	2. El usuario presiona el botón “Aceptar propuesta”.	3. El sistema cambia el estado de la Propuesta de paseo a demanda a “Aceptada”.
		4. El sistema verifica que el usuario paseador tiene el estado “Disponible”.
		5. El sistema crea una nueva instancia de Paseo utilizando los datos del usuario “Paseador” y actualiza su estado a “En marcha”
		6. Continúa en el CU “Comienzo del paseo”.
		7. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema

Negación de propuesta	2. El usuario presiona el botón “Negar propuesta”.	3. El sistema cambia el estado de la Propuesta de paseo a demanda a “Negada”.
		4. El sistema muestra la pantalla principal de la aplicación.
		5. FIN CU
Paseo en marcha existente		4. El sistema verifica que el usuario paseador tiene el estado “Paseo en marcha”.
		5. El sistema busca la instancia de Paseo asociada al Paseador.
		6. Continúa en el CU “Comienzo del paseo”.
		7. FIN CU
<b>Pantallas:</b>		

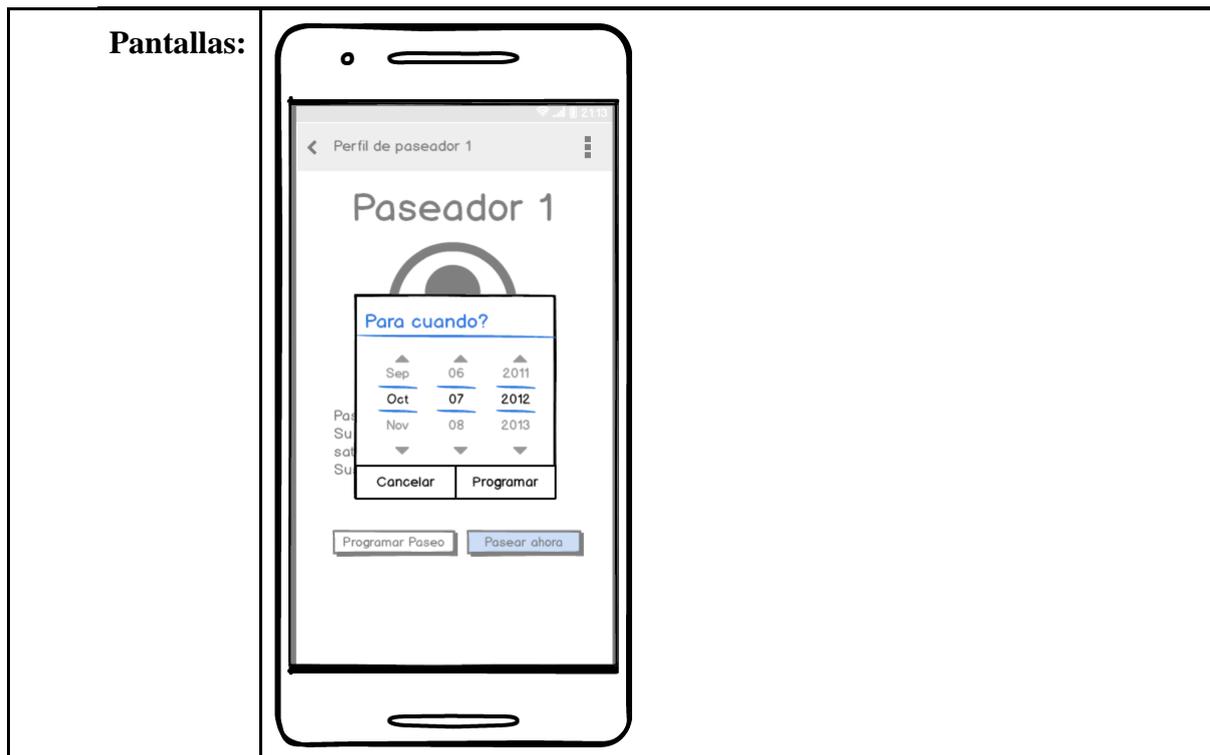
<b>Caso de Uso ID:</b>	CU008
<b>Caso de Uso Nombre:</b>	Búsqueda de Propuestas de paseo programadas

<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario “Dueño” o Usuario “Paseador”		
<b>Descripción:</b>	Un usuario consulta las Propuestas de paseo programadas disponibles		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario existe en el sistema.</li> <li>• El usuario se encuentra autenticado en la aplicación.</li> </ul>		
<b>Post-condiciones:</b>			
<b>Prioridad:</b>			
<b>Frecuencia de uso:</b>	Baja		
<b>Flujo Normal:</b>	Actor	Sistema	
	1. El usuario presiona el botón “Paseos programados”.	2. El sistema muestra en pantalla los resultados de la búsqueda en una lista junto al estado de cada Propuesta.	
		3. FIN CU	
<b>Flujos Alternativos:</b>	Actor	Sistema	
---			



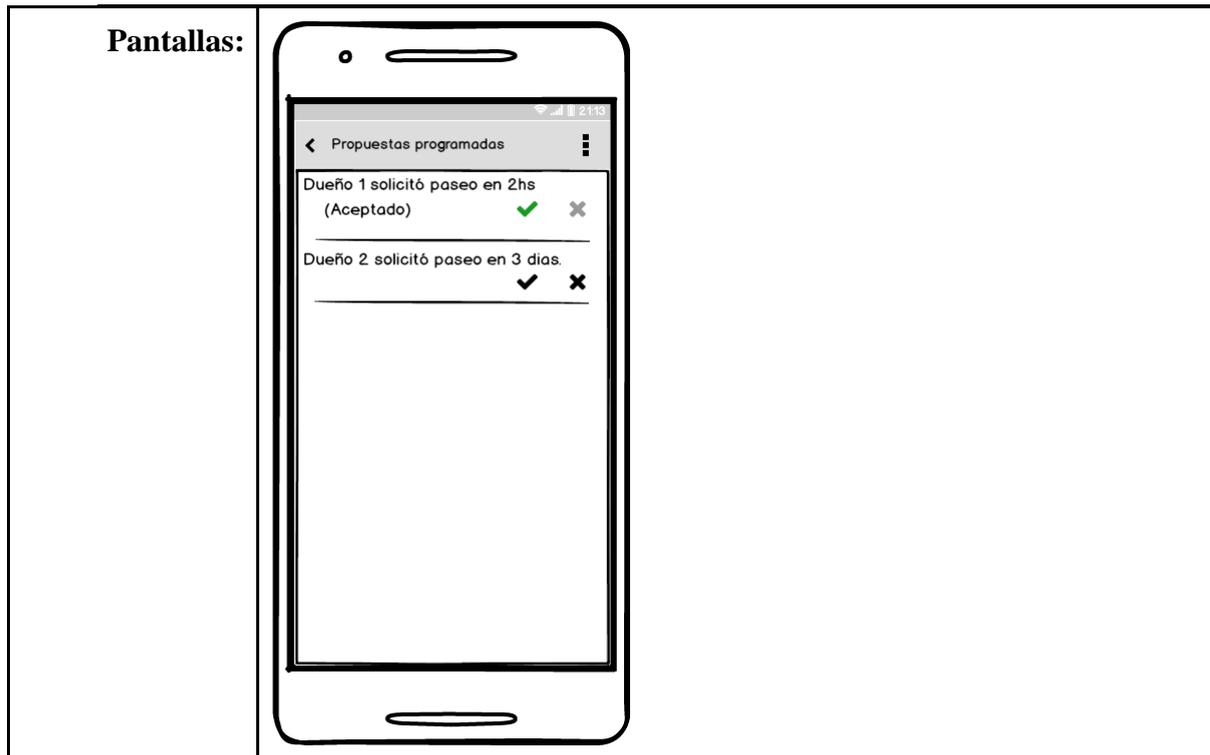
<b>Caso de Uso ID:</b>	CU009		
<b>Caso de Uso Nombre:</b>	Envío de propuesta de paseo programado		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Dueño"		
<b>Descripción:</b>	Un usuario "Dueño" envía una Propuesta de paseo programado a un usuario "Paseador".		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" realizó previamente una búsqueda de paseadores cercanos a su ubicación.</li> <li>• El usuario "Dueño" tiene el estado "Disponible".</li> <li>• El usuario "Paseador" existe en el sistema.</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se crea una nueva instancia de Propuesta de paseo programado en el sistema, relacionando ambos usuarios.</li> <li>• El usuario "Paseador" recibe la Propuesta de paseo programado.</li> </ul>		
<b>Prioridad:</b>			

<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario selecciona un usuario "Paseador" del listado y presiona el botón "Programar Paseo".	2. El sistema solicita el día y horario en que el usuario "Dueño" desea programar el paseo.
	3. El usuario ingresa la fecha y hora para programar el paseo y presiona el botón "Programar".	4. El sistema genera una nueva instancia de Propuesta de paseo programado, relacionando el usuario "Dueño", el usuario "Paseador".
		5. El Sistema notifica al usuario "Paseador" que una nueva Propuesta de paseo programado se encuentra disponible.
		6. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Cancelación del envío de la propuesta	3. El usuario presiona el botón "Cancelar".	4. El sistema muestra la pantalla de perfil del paseador.
		5. FIN CU



<b>Caso de Uso ID:</b>	CU010		
<b>Caso de Uso Nombre:</b>	Aceptación de propuesta de paseo programado		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		
<b>Descripción:</b>	Un usuario "Paseador" acepta una Propuesta de paseo programado recibida por parte de un usuario "Dueño" para un día y horario previsto.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" no tiene asignado un Paseo en el día y horario previsto.</li> <li>• El usuario "Paseador" realizó previamente una búsqueda de Propuestas de paseo programados.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Paseador" se encuentra autenticado en la aplicación.</li> <li>• Hay instancias de Propuesta de paseo programado en el sistema, vinculando a ambos usuarios.</li> </ul>		

<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>La Propuesta de paseo a programado cambia su estado a "Aceptada".</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario selecciona una Propuesta de paseo programado de la lista y presiona en el botón "Aceptar propuesta".	2. El sistema verifica que no existe una instancia de Propuesta de paseo programado con estado "Aceptada" para ese usuario Dueño en ese día y horario.
		3. El sistema modifica el estado de la Propuesta de paseo programado a "Aceptada".
		4. El sistema notifica al usuario Dueño que la Propuesta de paseo programado fue aceptada.
		5. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Negación de propuesta de paseo programado	1. El usuario presiona el botón "Negar propuesta".	2. El sistema cambia el estado de la Propuesta de paseo programado a "Negada" y notifica al usuario Dueño.
		3. FIN CU
Propuesta de paseo programado existente		2. El sistema verifica que ya existe una instancia de Propuesta de paseo programado con estado "Aceptada" para ese usuario Dueño en ese día y horario y le informa al usuario que ya no puede aceptarla.
		3. El sistema elimina la instancia de Propuesta de paseo programado del sistema.
		4. FIN CU



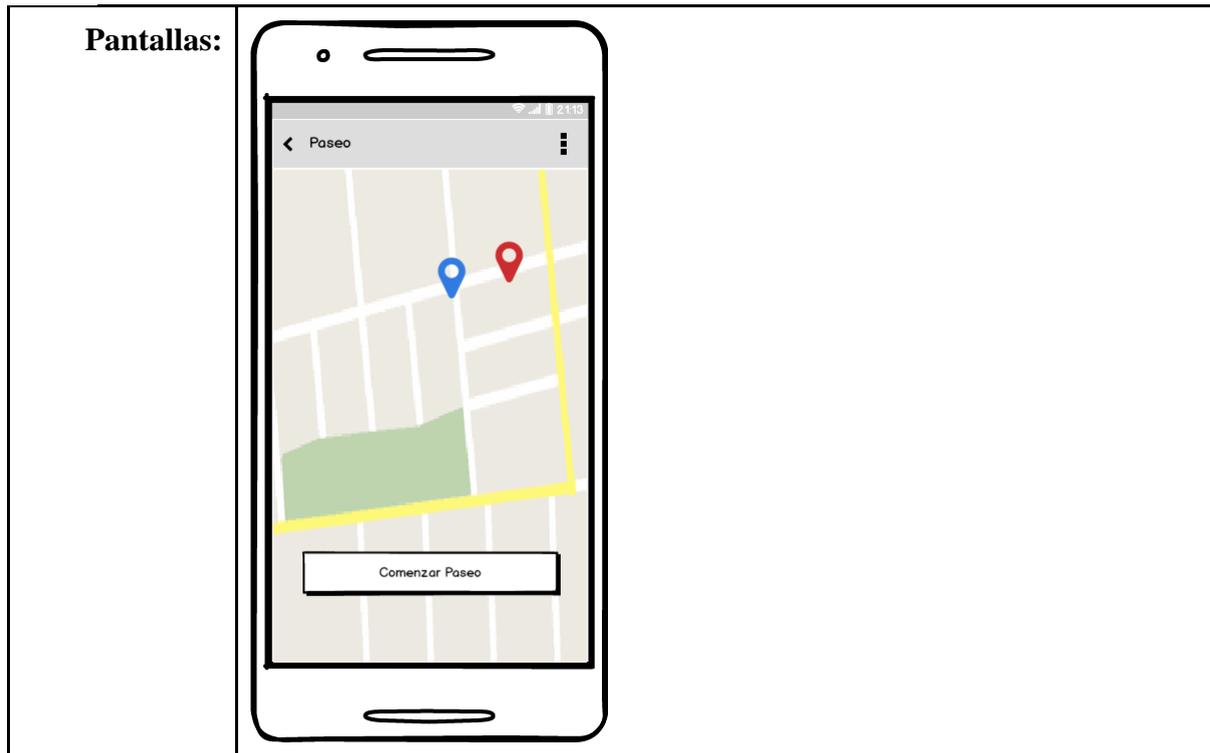
<b>Caso de Uso ID:</b>	CU011		
<b>Caso de Uso Nombre:</b>	Inicio de paseo programado		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		
<b>Descripción:</b>	Un usuario "Paseador" inicia un paseo programado.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Paseador" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" tiene el estado "Disponible".</li> <li>• Hay instancias de Propuesta de paseo programado en el sistema, vinculando a ambos usuarios con el estado "Aceptada".</li> <li>• El usuario "Paseador" tiene el estado "Disponible" o "Paseo en marcha"</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se crea una nueva instancia de Paseo con el estado "En marcha".</li> <li>• La Propuesta de paseo programado se elimina del sistema.</li> </ul>		

<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
		1. El sistema notifica al usuario de un Paseo programado próximo a ocurrir.
	2. El usuario presiona el botón "Iniciar paseo".	3. El sistema elimina la instancia de Propuesta de paseo programado del sistema.
		4. El sistema verifica que el usuario paseador tiene el estado "Disponible".
		5. Continúa en el CU "Comienzo del paseo".
		6. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Paseo en marcha existente		4. El sistema verifica que el usuario paseador tiene el estado "Paseo en marcha".
		5. El sistema busca la instancia de Paseo asociada al Paseador.
		6. Continúa en el CU "Comienzo del paseo".
		7. FIN CU



<b>Caso de Uso ID:</b>	CU012		
<b>Caso de Uso Nombre:</b>	Comienzo del paseo		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	26/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		
<b>Descripción:</b>	El usuario Paseador comienza un Paseo		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Dueño" tiene el estado "Disponible".</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• Existe una instancia de paseo con estado "En marcha" vinculando a ambos usuarios.</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se crea una nueva instancia de Mascota con los datos del usuario dueño y estado "Paseo creado" y se la incluye como participante del Paseo.</li> </ul>		

	<ul style="list-style-type: none"> <li>• El usuario “Paseador” cambia su estado a “Paseo en marcha”.</li> <li>• El usuario “Dueño” cambia su estado a “Paseo en marcha”.</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
		1. El sistema crea una nueva instancia de Mascota con los datos del usuario “Dueño” y actualiza su estado a “Paseo creado”.
		2. El sistema agrega la instancia de Mascota al Paseo.
		3. El sistema actualiza el estado de ambos usuarios a “Paseo en marcha” y les notifica que el paseo está en marcha.
		4. El sistema muestra la pantalla de “Paseo en marcha” con la ubicación actual del usuario “Dueño” y del usuario “Paseador”.
		5. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
---		



<b>Caso de Uso ID:</b>	CU013		
<b>Caso de Uso Nombre:</b>	Actualización de la posición del Paseador		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		
<b>Descripción:</b>	Un usuario "Paseador" informa su posición actual, mientras se lleva a cabo el paseo.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Paseador" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Paseador" tiene el estado "Paseo en marcha".</li> <li>• Hay un Paseo asociado al usuario "Paseador" con estado "En marcha".</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se crea una nueva instancia de Ubicación y se la asocia con el usuario "Paseador" y el Paseo actual.</li> </ul>		
<b>Prioridad:</b>			

<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
		1. El sistema de manera periódica y mientras se lleva a cabo el paseo, actualiza la posición física del usuario y mueve el marcador en el mapa a la última posición conocida.
		2. El sistema crea una nueva instancia de Ubicación con los datos actualizados de la posición y la asocia al usuario y al Paseo actual.
		3. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
--		
<b>Pantallas:</b>		

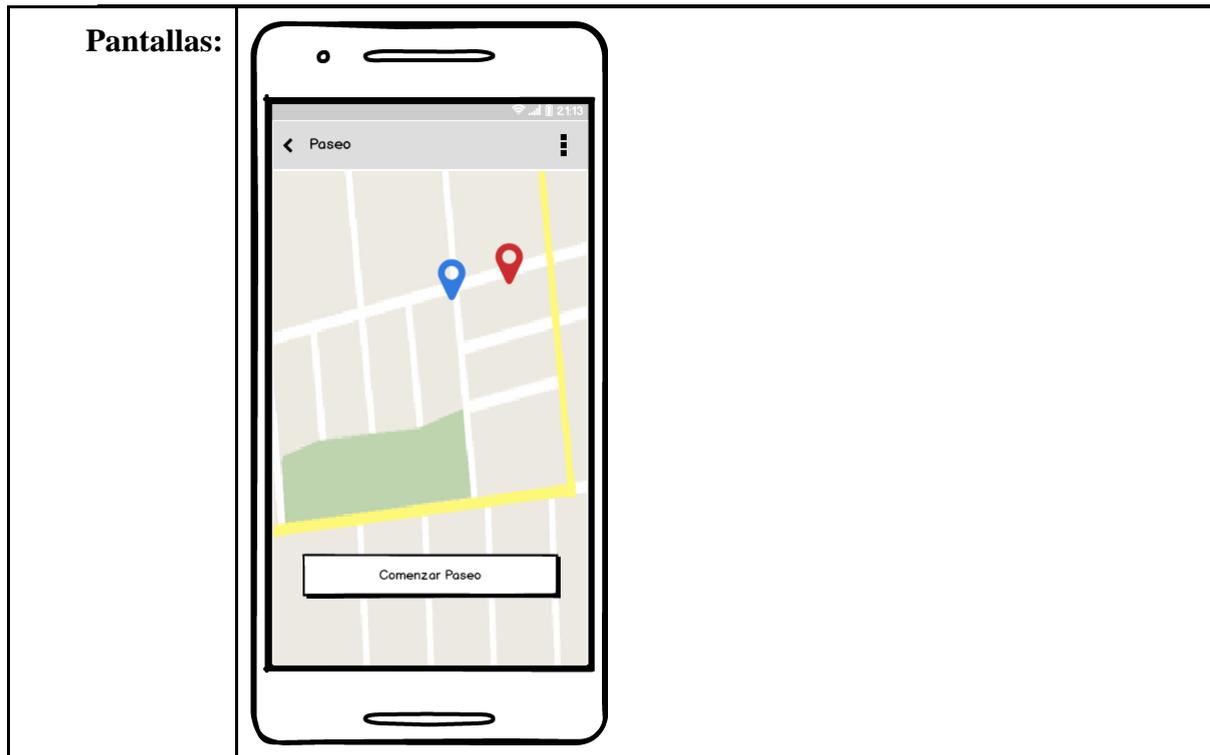
<b>Caso de Uso ID:</b>	CU014
------------------------	-------

<b>Caso de Uso Nombre:</b>	Consulta de la posición del Paseador	
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>
<b>Actor:</b>	Usuario "Dueño"	
<b>Descripción:</b>	Un usuario "Dueño" consulta la posición actual de un usuario "Paseador", mientras se lleva a cabo un Paseo.	
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Paseador" tiene el estado "Paseo en marcha".</li> <li>• El usuario "Dueño" tiene el estado "Paseo en marcha".</li> <li>• Hay instancias de Mascota en el sistema.</li> <li>• Hay un Paseo asociado al usuario "Paseador" y al usuario "Dueño" con estado "En marcha".</li> </ul>	
<b>Post-condiciones:</b>		
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
		1. El sistema consulta periódicamente para el Paseo actualmente en marcha, la instancia de Ubicación más reciente asociada al usuario "Paseador".
		2. El sistema mueve el marcador en el mapa a la última ubicación informada del usuario "Paseador", dejando visible el recorrido realizado por el mismo hasta el momento.
		3. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema

--	
<b>Pantallas:</b>	

<b>Caso de Uso ID:</b>	CU015		
<b>Caso de Uso Nombre:</b>	Retiro de la mascota		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		
<b>Descripción:</b>	Mientras se lleva a cabo un Paseo, el usuario "Paseador" se reúne físicamente con el usuario "Dueño" y retira la mascota.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Paseador" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Paseador" tiene el estado "Paseo en marcha".</li> <li>• El usuario "Dueño" tiene el estado "Paseo en marcha".</li> <li>• Hay una Mascota asociada al usuario "Dueño" en el Paseo con el estado "Paseo en marcha".</li> <li>• Hay un Paseo asociado a ambos usuarios con estado "En marcha".</li> </ul>		

<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>La Mascota del Paseo cambia su estado a “Paseo en marcha”.</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario selecciona en la pantalla el usuario dueño con el que se encontró físicamente y presiona el botón “Comenzar paseo”.	2. El sistema verifica que la distancia física que separa ambos usuarios sea menor a 15 metros.
		3. El sistema cambia el estado de la Mascota del Paseo a “Paseo en marcha”.
		4. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Usuarios alejados		2. El sistema verifica que la distancia física que separa ambos usuarios es mayor a 15 metros.
		3. El sistema le informa al usuario que todavía no puede comenzar el Paseo.
		4. FIN CU



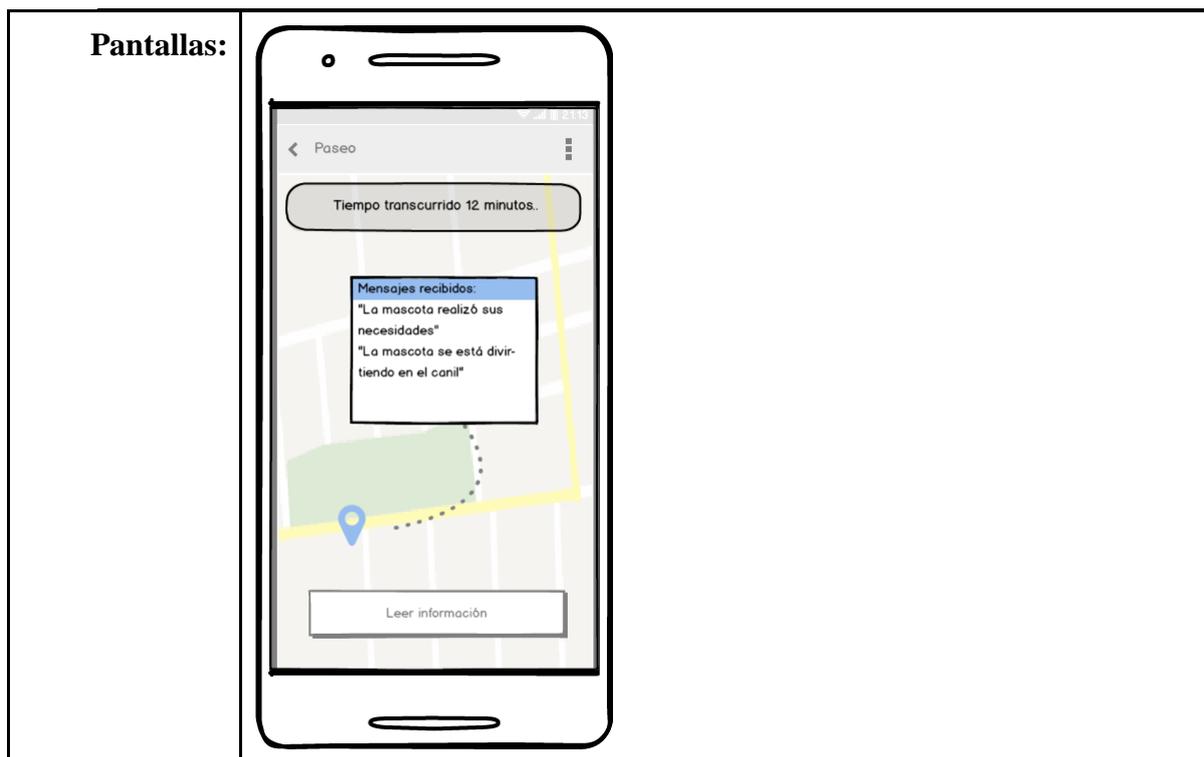
<b>Caso de Uso ID:</b>	CU016		
<b>Caso de Uso Nombre:</b>	Actualización del progreso del paseo		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		
<b>Descripción:</b>	Un usuario "Paseador" informa al usuario "Dueño" una novedad relacionada al Paseo que actualmente se está llevando a cabo.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Paseador" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Paseador" tiene el estado "Paseo en marcha".</li> <li>• El usuario "Dueño" tiene el estado "Paseo en marcha".</li> <li>• Hay una Mascota asociada al usuario "Dueño" en el Paseo con el estado "Paseo en marcha".</li> <li>• Hay un Paseo asociado a ambos usuarios con estado "En marcha".</li> </ul>		

<b>Post- condiciones:</b>	<ul style="list-style-type: none"> <li>Se crea una nueva instancia de Novedad y se la asocia a la Mascota del Paseo que actualmente se está llevando a cabo.</li> <li>Se informa al usuario “Dueño” de la novedad.</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario presiona el botón “Informar novedad”.	2. El sistema solicita el ingreso del tipo de novedad.
	3. El usuario selecciona el tipo de novedad “Necesidades”.	4. El sistema crea una nueva instancia de Novedad con el tipo “Necesidades” y la asocia a la Mascota del Paseo que actualmente se está llevando a cabo.
		5. El Sistema informa al usuario “Dueño” asociado al Paseo que actualmente se está llevando a cabo, que hay novedades.
		6. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Selecciona “Texto”	3. El usuario selecciona el tipo de novedad “Texto”.	4. El sistema solicita el ingreso de un mensaje de texto.
	5. El usuario ingresa un mensaje de texto y presiona “Enviar”.	6. El sistema crea una nueva instancia de Novedad con el tipo “Texto” y el mensaje ingresado por el usuario y la asocia a la Mascota del Paseo que actualmente se está llevando a cabo.
		7. El Sistema informa al usuario “Dueño” asociado al Paseo que actualmente se está llevando a cabo, que hay novedades.

		8. FIN CU
Selecciona "Imagen"	3. El usuario selecciona el tipo de novedad "Imagen".	4. El sistema muestra la pantalla de captura de fotos y solicita la captura de una imagen.
	5. El usuario captura una imagen y presiona "Enviar".	6. El sistema crea una nueva instancia de Novedad con el tipo "Imagen" y la imagen ingresada por el usuario y la asocia a la Mascota del Paseo que actualmente se está llevando a cabo.
		7. El Sistema informa al usuario "Dueño" asociado al Paseo que actualmente se está llevando a cabo, que hay novedades.
		8. FIN CU
<b>Pantallas:</b>		

<b>Caso de Uso ID:</b>	CU017
<b>Caso de Uso Nombre:</b>	Consulta de progreso del Paseo

<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Dueño"		
<b>Descripción:</b>	Un usuario "Dueño" consulta las novedades relacionadas a un Paseo, mientras se lleva a cabo el mismo.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" tiene el estado "Paseo en marcha".</li> <li>• Hay un Paseo asociado a ambos usuarios con el estado "En marcha".</li> <li>• Hay una Mascota asociada al usuario "Dueño" en el Paseo con el estado "Paseo en marcha".</li> </ul>		
<b>Post-condiciones:</b>			
<b>Prioridad:</b>			
<b>Frecuencia de uso:</b>	Baja		
<b>Flujo Normal:</b>	Actor	Sistema	
		1. El sistema notifica al usuario que una nueva instancia de Novedad se encuentra disponible para el Paseo en marcha.	
	2. El usuario presiona el botón "Ver novedades".	3. El sistema muestra un mensaje con el tipo de novedad y el dato asociado.	
		4. FIN CU	
<b>Flujos Alternativos:</b>	Actor	Sistema	
--			



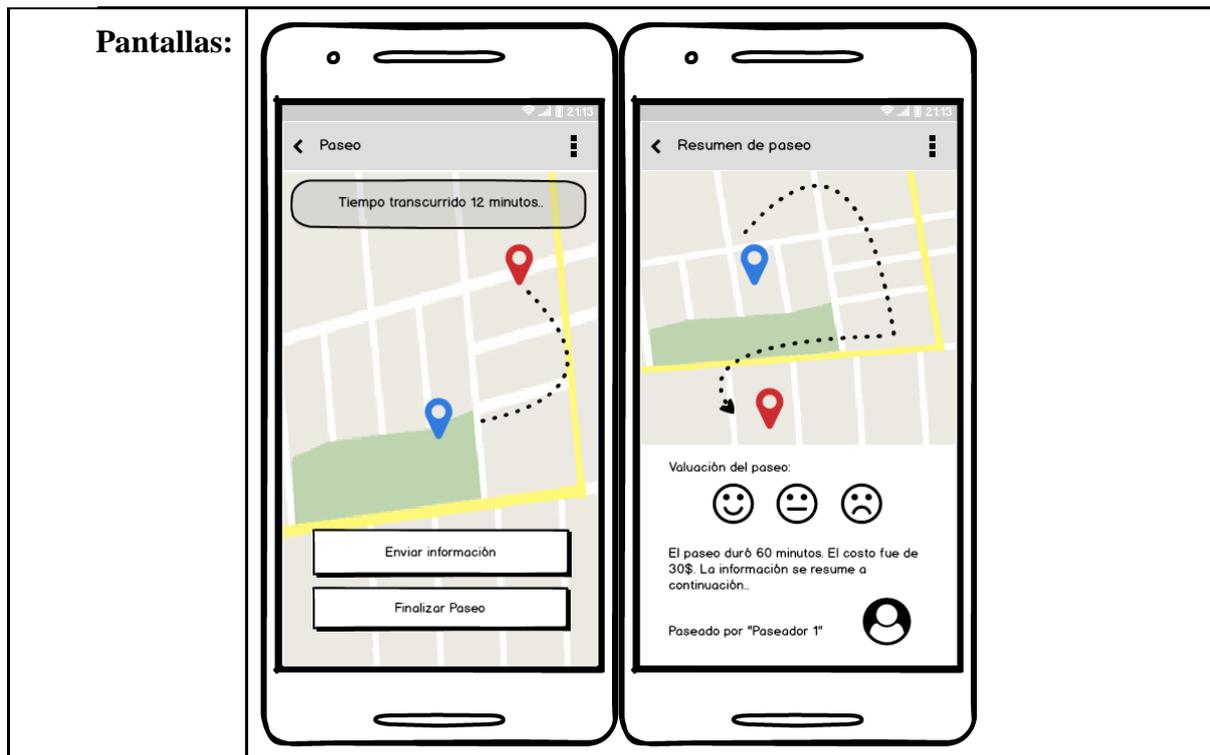
<b>Caso de Uso ID:</b>	CU018		
<b>Caso de Uso Nombre:</b>	Entrega de la mascota		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Paseador"		
<b>Descripción:</b>	Mientras se lleva a cabo un Paseo, el usuario "Paseador" se reúne físicamente con el usuario "Dueño" y entrega la mascota.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Paseador" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Paseador" tiene el estado "Paseo en marcha".</li> <li>• El usuario "Dueño" tiene el estado "Paseo en marcha".</li> <li>• Hay un Paseo asociado a ambos usuarios con el estado "En marcha".</li> <li>• Hay una Mascota asociada al usuario "Dueño" en el Paseo con el estado "Paseo en marcha".</li> </ul>		

<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• La Mascota del Paseo cambia el estado a “Paseo finalizado”.</li> <li>• El usuario “Dueño” cambia su estado a “Disponible”.</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario selecciona en la pantalla el usuario dueño con el que se encontró físicamente y presiona el botón “Finalizar paseo”.	2. El sistema verifica que la distancia física que separa ambos usuarios sea menor a 15 metros.
		3. El sistema cambia el estado de la Mascota del Paseo asociada a ambos usuarios a “Paseo finalizado”.
		4. El sistema verifica que todavía restan Mascotas en el Paseo con estado “Paseo creado” o “Paseo en marcha”.
		5. El sistema actualiza el estado del usuario “Dueño” a “Disponible”.
		6. El sistema envía al usuario “Dueño” un resumen del Paseo realizado junto a una solicitud de Valoración y muestra la pantalla principal.
		7. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Usuarios alejados		2. El sistema verifica que la distancia física que separa ambos usuarios es mayor a 15 metros.
		3. El sistema le informa al usuario que todavía no puede finalizar el Paseo.

		4. FIN CU
Sin mascotas pendientes		4. El sistema verifica que todas las Mascotas del Paseo tienen el estado “Paseo finalizado”.
		5. Continúa en el CU “Finalizar paseo”.
		6. FIN CU
<b>Pantallas:</b>		

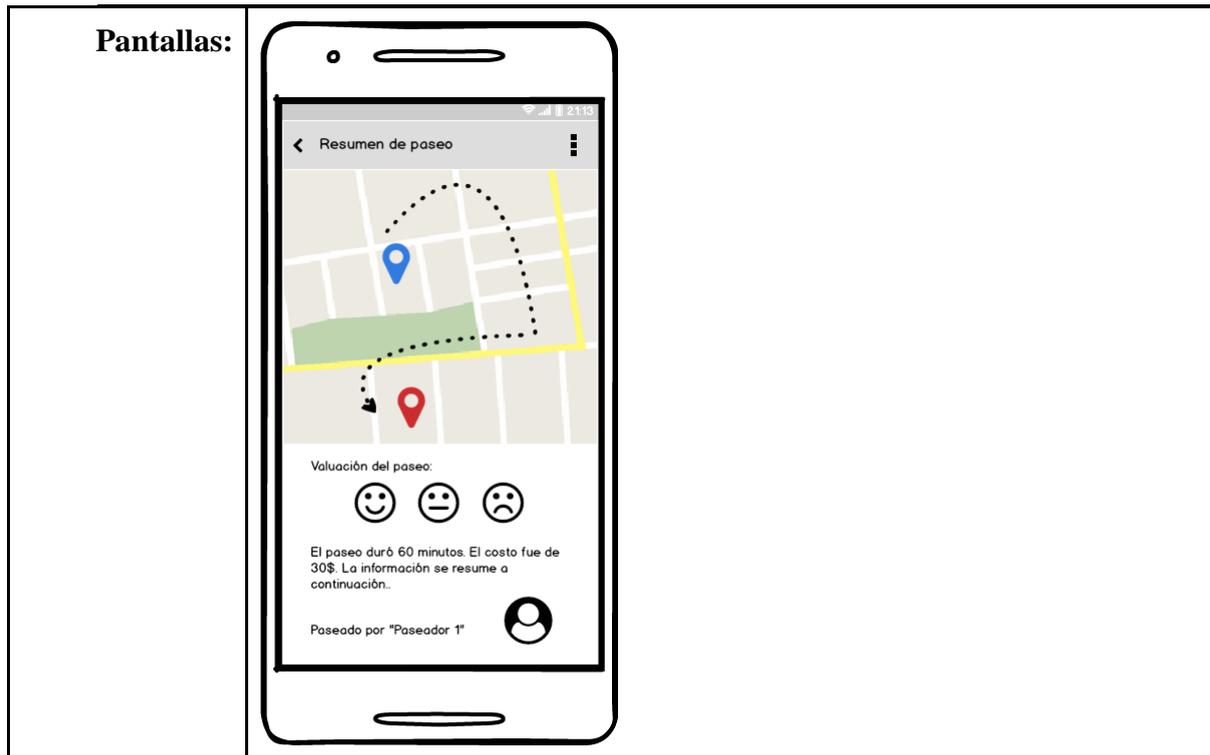
<b>Caso de Uso ID:</b>	CU019		
<b>Caso de Uso Nombre:</b>	Finalizar paseo		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario “Paseador”		
<b>Descripción:</b>	El usuario “Paseador” entregó todas las Mascotas del Paseo y lo da por finalizado.		

<b>Pre- condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario “Paseador” se encuentra autenticado en la aplicación.</li> <li>• El usuario “Paseador” existe en el sistema.</li> <li>• El usuario “Paseador” tiene el estado “Paseo en marcha”.</li> <li>• Hay un Paseo asociado a ambos usuarios con el estado “En marcha”.</li> <li>• Todas las Mascotas del Paseo tienen el estado “Paseo finalizado”.</li> </ul>	
<b>Post- condiciones:</b>	<ul style="list-style-type: none"> <li>• El Paseo cambia su estado a “Finalizado”.</li> <li>• El usuario “Paseador” cambia su estado a “Disponible”.</li> </ul>	
<b>Prioridad:</b>		
<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
		1. El sistema verifica que todas las Mascotas del Paseo tienen el estado “Paseo finalizado”.
		2. El sistema cambia el estado del Paseo a “Finalizado”.
		3. El sistema envía a los usuarios un resumen del Paseo realizado y muestra la pantalla principal.
		4. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
--		



<b>Caso de Uso ID:</b>	CU020		
<b>Caso de Uso Nombre:</b>	Valoración del paseo.		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario "Dueño"		
<b>Descripción:</b>	El usuario "Dueño" valora un Paseo finalizado.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario "Dueño" se encuentra autenticado en la aplicación.</li> <li>• El usuario "Paseador" existe en el sistema.</li> <li>• El usuario "Dueño" existe en el sistema.</li> <li>• El usuario "Dueño" tiene el estado "Disponible".</li> <li>• Hay un Paseo asociado a ambos usuarios con estado "Finalizado" y sin valoración.</li> </ul>		
<b>Post-condiciones:</b>	<ul style="list-style-type: none"> <li>• Se actualiza la valoración del Paseo.</li> </ul>		
<b>Prioridad:</b>			

<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
		1. El sistema solicita el ingreso de un valor del 1 al 5 para valorar el Paseo realizado.
	2. El usuario ingresa un valor del 1 al 5 y presiona el botón “Enviar”.	3. El sistema actualiza la valoración del Paseo.
		4. El sistema notifica al usuario “Dueño” que el Paseo fue valorado.
		5. El sistema muestra la pantalla principal de la aplicación.
		6. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
Cancela valoración	2. El usuario presiona el botón “No valorar Paseo”.	3. El sistema muestra la pantalla principal de la aplicación.
		4. FIN CU



<b>Caso de Uso ID:</b>	CU021		
<b>Caso de Uso Nombre:</b>	Consulta de paseos realizados.		
<b>Creado por:</b>	Luciano Balmaceda	<b>Última actualización por:</b>	
<b>Fecha Creación:</b>	27/11/2016	<b>Fecha última actualización:</b>	
<b>Actor:</b>	Usuario		
<b>Descripción:</b>	El usuario consulta la lista de paseos realizados en el pasado.		
<b>Pre-condiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario se encuentra autenticado en la aplicación.</li> <li>• El usuario existe en el sistema.</li> <li>• El usuario tiene el estado "Disponible".</li> <li>• Hay instancias de Paseo en el sistema.</li> </ul>		
<b>Post-condiciones:</b>			
<b>Prioridad:</b>			

<b>Frecuencia de uso:</b>	Baja	
<b>Flujo Normal:</b>	Actor	Sistema
	1. El usuario presiona el botón “Listar paseos realizados”.	2. El sistema busca las instancias de Paseo con estado “Finalizado” relacionadas con el usuario actualmente autenticado.
		3. El sistema muestra la pantalla de Lista de Paseos realizados con los resultados obtenidos.
		4. FIN CU
<b>Flujos Alternativos:</b>	Actor	Sistema
--		
<b>Pantallas:</b>		

### Diagrama de clases

Dado que la aplicación mobile tiene 2 variaciones según el rol del usuario a la que está destinada, decidimos mostrar 2 diagramas de clases separados. Esto permite apreciar las clases compartidas entre ambas versiones.

### Variante “Paseadores”





En cuanto a las mejoras posibles, el proyecto ya incluía la comunicación en un sentido, paseador → dueño, pero no permitía la comunicación en el sentido inverso. Ejemplos de dicho comportamiento serían requerir que el paseo sea finalizado de inmediato o que la entrega del animal se demore unos minutos por la ausencia temporal del dueño. Otra posible mejora es aprovechar los datos recolectados en el seguimiento de cada mascota y analizarlos para incluir en el reporte el estado general del bienestar del animal incluyendo atributos como la cantidad de pasos dados y número de paseos semanales realizados. Los resultados luego podrán utilizarse para motivar la participación continua del dueño y sus amigos en el servicio, aplicando el concepto de ludificación (“gamification”).

## Bibliografía

- Android. *Android* [en línea], [sin fecha]. [en línea]. [Consulta: Marzo 4 2017]. Disponible en: <https://www.android.com/>.
- BASS, L., CLEMENTS, P. and KAZMAN, R., 2013. *Software architecture in practice*. Upper Saddle River, NJ: Addison-Wesley.
- CLEMENTS, P., 2014. *Documenting software architectures views and beyond*. Upper Saddle River, NJ: Addison-Wesley.
- Configure Build Variants. *Android Developers* [en línea], [sin fecha]. [en línea]. [Consulta: Marzo 4 2017]. Disponible en: <https://developer.android.com/studio/build/build-variants.html>.
- Inscripción al Registro de Paseadores de Perros. *Buenos Aires Ciudad - Gobierno de la Ciudad Autónoma de Buenos Aires* [en línea], [sin fecha]. [en línea]. [Consulta: Marzo 15 2017]. Disponible en: <http://www.buenosaires.gob.ar/tramites/inscripcion-al-registro-de-paseadores-de-perros>.
- Mobile operating system market share in Argentina. *StatCounter Global Stats* [en línea], [sin fecha]. [en línea]. [Consulta: Marzo 4 2017]. Disponible en: <http://gs.statcounter.com/os-market-share/mobile/argentina#monthly-201602-201702-bar>.
- Mobile operating system market share in South America. *StatCounter Global Stats* [en línea], [sin fecha]. [en línea]. [Consulta: Marzo 4 2017]. Disponible en: <http://gs.statcounter.com/os-market-share/mobile/south-america#monthly-201602-201702-bar>.
- Netverify Identity Verification Service | Trusted Identity as a Service. *Jumio* [en línea], [sin fecha]. [en línea]. [Consulta: Marzo 15 2017]. Disponible en: <https://www.jumio.com/netverify/>.
- Real-time identity verification and business verification. *Intelligent Identity Verification - BlockScore* [en línea], [sin fecha]. [en línea]. [Consulta: Marzo 15 2017]. Disponible en: <https://blockscore.com/>.
- A Virtual Crash Course in Design Thinking. *Stanford d.school* [en línea], [sin fecha]. [en línea]. [Consulta: Mayo 7 2017]. Disponible en: <https://dschool.stanford.edu/resources-collections/a-virtual-crash-course-in-design-thinking>