

# Find My Paw

Arguelles Macosay Mariana, Guillén López Alam, Romero Andrade Cristian

## Índice

<b>I.</b>	<b>Descripción</b>	1
<b>II.</b>	<b>Estado de la Técnica</b>	1
II-A.	Distribución del proyecto . . . .	1
II-A1.	assets . . . . .	1
II-A2.	model . . . . .	1
II-A3.	page . . . . .	1
II-A4.	tools . . . . .	1
II-B.	Herramientas que se usaron . . .	1
II-B1.	Paquetes . . . . .	1

### I. Descripción

Este proyecto es la base para enviar y recibir ubicación de cualquier dispositivo que reciba SMS. La idea en general consiste en atrapar los SMS que la aplicación reciba e interpretarla en un mapa que utiliza la API de Google Maps

### II. Estado de la Técnica

#### II-A. Distribución del proyecto

Flutter a ser de un framework para aplicaciones móviles multiplataforma, nos da la libertad de crear nuestra estructura como es conveniente. En este proyecto se uso una estructura similar a MVC donde los directorios están distribuidas de la siguiente manera

Donde, en `./find_my_paw/lib`, flutter compila el proyecto que estamos desarrollando

*II-A1. assets:* Se ubica la configuración global del proyecto, como son los colores y las llaves de las API para el uso de distintos módulos externos<sup>1</sup>.

*II-A2. model:* Aquí se hacen las operaciones para recuperar la información de una base de datos o bien la implementación de una API REST.

*II-A3. page:* En esta ubicación se guardan los Widgets a mandar al usuario (que aquí sería el controlador y vista).

*II-A4. tools:* Aquí se ubica diferentes herramientas que se pueden implementar tanto en el directorio `./find_my_paw/lib/model` y `./find_my_paw/lib/pages`. Como en el caso de las solicitudes http.

#### II-B. Herramientas que se usaron

- Se uso Mockoon y en parte Flask para obtener el API para el desarrollo de este proyecto
- Se compilaba en un emulador Android Pixel 3A con la versión SDK 30 con una arquitectura x86.

##### II-B1. Paquetes:

- `sms` Este paquete esta a la escucha de la respuesta del dispositivo que envía la ubicación para su compilación necesaria, se necesita pasar el argumento `--no-sound-null-safety`. Se eligió este paquete por ser de fácil implementación para ambos dispositivos (según los repositorios de los paquetes de flutter/dart).
- `google_maps_flutter` Usamos la Api de Google, específicamente Google Maps para la ubicación.
- `flutter_polylinepoints` Este paquete nos ayuda a pintar los puntos de la ruta, la cual se calcula.
- `geocoding`
- `geolocator` Los paquetes anteriores **geocoding** y **geolocator** nos ayudan a obtener nuestra ubicación en tiempo real
- `http` Para recuperar información usando método GET.

<sup>1</sup>En este caso para la API de Google Maps