

ANIMAL'S SAFE

Juan Ignacio Mac-kinlay Garrido

Jose Ángel Ortiz Pareja

Cádiz, 08/06/2018



Proyecto Integrado del Colegio Salesianos “San Ignacio” de Cádiz
presentado por los alumnos:

Juan Ignacio Mac-kinlay Garrido

Jose Ángel Ortiz Pareja

Título del Proyecto: Animal's Safe.

ÍNDICE

Introducción.	4
Estado actual del tema.	5
Objetivos.	6
Metodología.	7
Recursos necesarios.	9
Plan de trabajo y temporización.	11
Etapa 1 de Animal's Safe.	12
Etapa 2 de Animal's Safe.	13
Etapa 3 de Animal's Safe.	14
Etapa 4 de Animal's Safe.	15
Etapa 5 de Animal's Safe.	16
Resultados y conclusiones.	17
Apéndice. Funcionamiento de la aplicación.	18
Apéndice. Detalles sobre la implementación del proyecto.	25
Trabajo futuro.	27
Webgrafía.	28



Introducción.

La aplicación surge a raíz del alto porcentaje de abandono animal en España, actualmente el país de la Unión Europea donde se abandonan más animales. Más de 137.000 animales son abandonados anualmente siendo un 85% perros y un 14% gatos, según el último estudio de abandono y adopción de la fundación Affinity. Un dato preocupante es que el 89% fueron abandonados en la calle y solo el 11% fueron dejados en los refugios.

Animal's Safe nace para combatir este problema, en un principio, en España, y más tarde, a todo el mundo. Para poder llevar a cabo esta idea, hemos realizado una aplicación, utilizable en todo tipo de dispositivos móviles, desarrollada con ionic y utilizando Firebase como soporte de datos.

Estado actual del tema.

La aplicación que hemos realizado viene precedida por otro proyecto, una página web responsive, adaptada a todo tipo de dispositivos. Esta web fue realizada en el framework de desarrollo web Laravel, herramienta con la cual ya habíamos trabajado con anterioridad. La web tenía una funcionalidad más amplia que la app que hemos desarrollado, pero también conocíamos las herramientas que íbamos a utilizar por eso pudimos ser más ambiciosos.



Objetivos.

El objetivo de Animal's Safe es luchar contra el alto porcentaje de abandono animal, ayudar a las asociaciones que se dedican a la acogida de animales dándole un hogar a estos y tener una aplicación donde acoger los anuncios de animales perdidos de los usuarios.

Todo esto se integra en una interfaz fácil e intuitiva, mediante el uso de mapas para ofrecer al usuario la mejor experiencia posible y la búsqueda rápida de asociaciones, clínicas y animales perdidos.

Uno de los objetivos de la aplicación es intentar recuperar un animal perdido de forma rápida y con ayuda de los usuarios de la aplicación, para esto se integrará un mapa en el cual se plasmarán todos los anuncios de animales perdidos mediante marcadores.

En cuanto a nuestros objetivos personales, con el desarrollo de esta aplicación queremos aprender un nuevo lenguaje de programación y trabajar con nuevas herramientas para potenciar nuestra capacidad de respuesta ante lo desconocido.

Metodología.

Para poder alcanzar los objetivos previamente expuestos, dividimos el desarrollo completo de la aplicación en diversas etapas. Cada etapa la hemos dividido, a su vez, en bloques de trabajo a realizar por cada uno de los integrantes del proyecto. Estos bloques eran definidos al comienzo de cada etapa, en una reunión entre los participantes del proyecto.

Una vez establecido y asignados los bloques, durante su desarrollo, cualquier duda que surgiera era comentada por ambos, existiendo un equilibrio de poder a la hora de tomar decisiones.

Uno de los bloques de desarrollo ha sido la composición de la base de datos. Hemos hecho un esquema entidad-relación (fig.1) para mostrar el funcionamiento de esta, aunque luego tuviéramos que adaptar este esquema a una base de datos orientada a objetos.

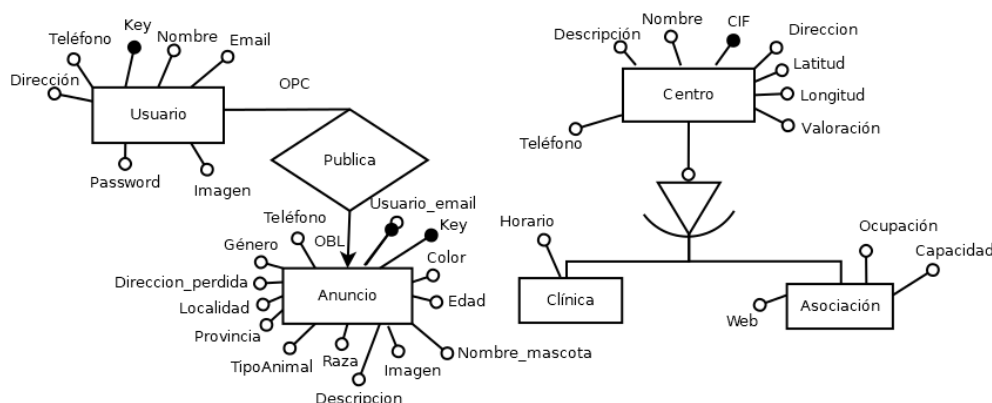


Fig.1: Esquema entidad-relación

Podemos decir que la metodología utilizada ha sido una metodología ágil, puesto que hemos avanzado en el proyecto de una manera iterativa e incremental.

Las etapas han sido cinco:

1. Establecer funcionalidades a realizar y herramientas a utilizar.
2. Estudio de las herramientas y puesta en marcha.
3. Desarrollo completo de la funcionalidad.
4. Pruebas, corrección de errores e implementación de mejoras.
5. Presentación y documentación.

Por último, antes de comenzar el proyecto hicimos un Diagrama de Flujo como guía para el desarrollo de la aplicación (Fig.2).

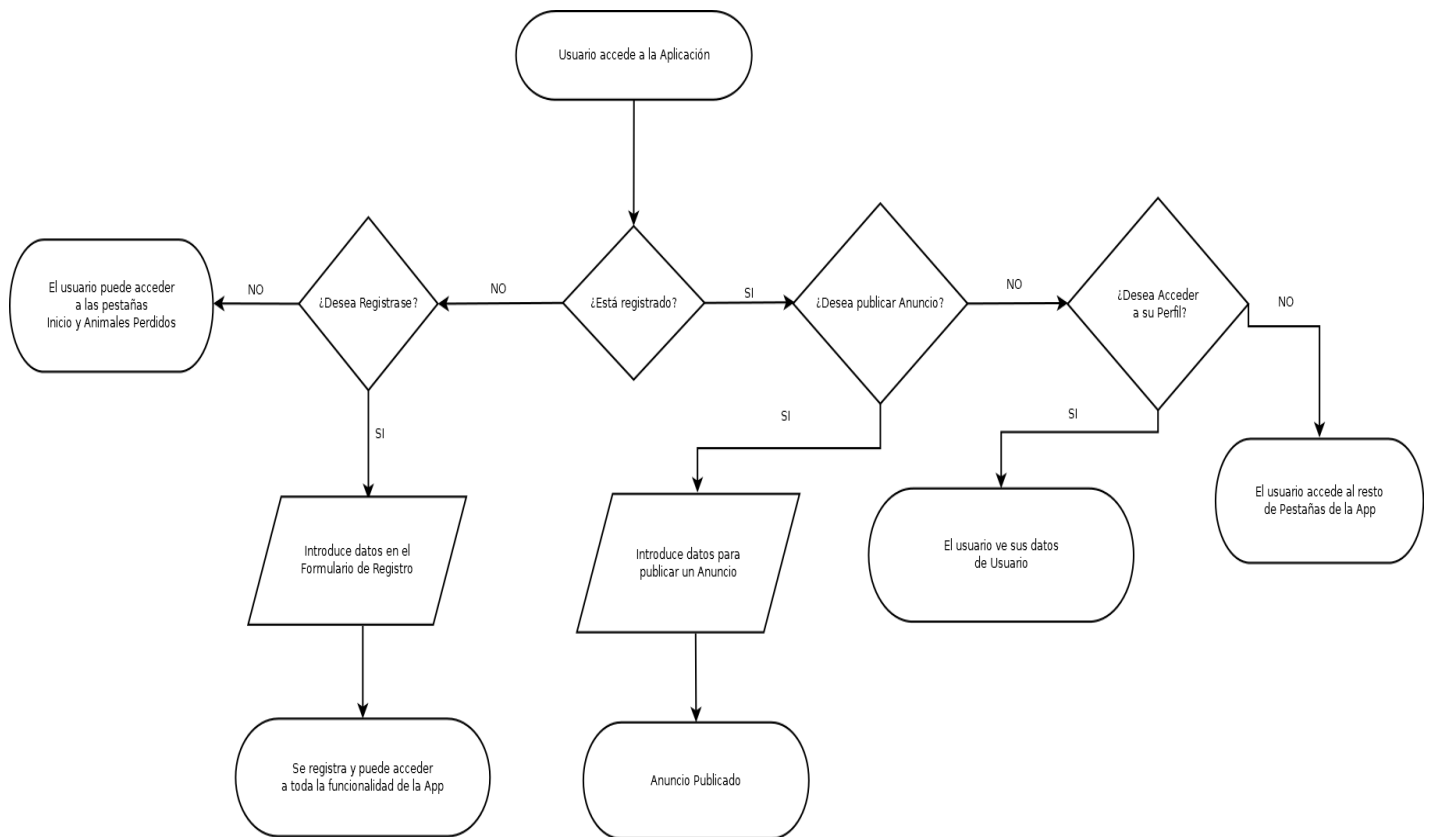


Fig.2 Diagrama de flujo de la Aplicación

Recursos necesarios.

Animal's Safe va dirigida para dispositivos móviles con Sistema Operativo IOS y Android. Para esto se usará Ionic, un framework de código abierto escrito en JavaScript, para el desarrollo híbrido de aplicaciones mediante el uso de HTML, AngularJS, Sass y TypeScript.



HTML5: lenguaje de marcado para la creación de páginas web. En Ionic se usa para crear la interfaz gráfica que ve el usuario final.



Sass: conocido como Super CSS. Tiene el mismo lenguaje que CSS pero utiliza una estructura que agiliza y facilita la llamada de clases, permite crear variables, crear métodos e incluso permite la herencia. Un ejemplo:

-CSS:

```
.clase {}  
.clase ul{}
```

-Sass:

```
.clase {  
  ul{  
  }  
}
```



TypeScript: es un lenguaje de programación libre y de código abierto, es un superconjunto de JavaScript. En Ionic se usa para todo el backend y funcionalidad de la aplicación.



AngularJS: es un framework de JavaScript de código abierto que se utiliza para crear y mantener aplicaciones web de una sola página. En Ionic se usa como FrontEnd de la aplicación.



Para el almacenamiento de los datos se usará **Firestore Realtime Database**, que es una base de datos en tiempo real. Este servicio proporciona a los desarrolladores una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase. De esta misma plataforma usaremos **Firestore Storage** para la carga de imágenes generada por el usuario y **Firestore Cloud Messaging** para las notificaciones junto con el plugin **OneSignal** que permite enviarlas.



Para el uso de los mapas hemos usado la **API de google maps**, más concretamente el **SDK**, que te permite, mediante JavaScript, integrar todas las funcionalidades de los mapas, como por ejemplo la geolocalización, la geocodificación, uso de marcadores, etc.

Plan de trabajo y temporización.

Animal's Safe es una aplicación híbrida para dispositivos móviles, la cual ha sido desarrollada en 5 etapas.

Etapas 1 de Animal's Safe.

Tiempo de Ejecución empleado en esta etapa, 3-4 días.

Etapas 2 de Animal's Safe.

Tiempo de Ejecución empleado en esta etapa, 21-22 días.

Etapas 3 de Animal's Safe.

Tiempo de Ejecución empleado en esta etapa, 53-54 días.

Etapas 4 de Animal's Safe.

Tiempo de Ejecución empleado en esta etapa, 10-11 días.

Etapas 5 de Animal's Safe.

Tiempo de Ejecución empleado en esta etapa, 2 días.

Etapa 1 de Animal's Safe.

En esta primera etapa, además de buscar las herramientas que utilizamos en el desarrollo de nuestra app, analizamos lo realizado con anterioridad en la aplicación Web, intentando clarificar cuáles eran las secciones más importantes y cuales incluiríamos en nuestra aplicación híbrida.

Decidimos descartar las ampliaciones realizadas a la idea original, mostrar las asociaciones cercanas a la persona que se encuentre un animal abandonado y mostrar los anuncios de animales perdidos en el mapa.

Una vez decidido el trabajo que realizaríamos, necesitábamos establecer los recursos a utilizar. Puesto que queríamos realizar una aplicación híbrida, buscamos un lenguaje el cual nos lo permitiera, sin tener que reescribir código dependiendo del sistema operativo al que nos quisiéramos dirigir.

Encontramos **ionic**, un framework para el desarrollo de aplicaciones híbridas, de código abierto. Y en cuanto al almacenamiento de datos, como hemos querido implementar notificaciones push en tiempo real, necesitábamos una base de datos en tiempo real, por eso descartamos la utilizada para la aplicación web, MySQL, y hemos utilizado **FireBase**.

FireBase es un servicio proporcionado por Google, el cual nos brinda la oportunidad de trabajar con una base de datos en tiempo real almacenada en la nube siendo esta gratuita, hasta cierto punto.

Esta primera etapa la concluimos con la realización del anteproyecto y estableciendo las entidades de la base de datos de nuestra aplicación con su correspondiente paso a tablas.

Etapas 2 de Animal's Safe.

Una vez escogidas las herramientas, comenzamos con su aprendizaje. En esta segunda etapa recabamos información sobre los recursos utilizados, y comenzamos la redacción de la documentación, empezando por la **Webgrafía**, anotando todas las páginas de las que nos hemos nutrido para realizar el proyecto.

En la primera etapa del desarrollo, solo buscamos información de manera superficial mientras que en esta segunda etapa nos centramos en conocer las muchas oportunidades que nos brindan las herramientas utilizadas.

Esta etapa es muy importante en el desarrollo del proyecto, debido a que teníamos que aprender las bases de las herramientas que hemos usado. A la hora de enfrentarte a recursos 'desconocidos', puesto que sabemos de su existencia pero no hemos trabajado con ellos anteriormente, hay que aprender los conceptos básicos para poder comenzar el desarrollo de la aplicación.

Y una vez recabada la información necesaria para poder iniciar el proyecto, comenzamos. Establecimos el diseño general de la aplicación para tener una idea algo más concreta de lo que realizaríamos y en cuanto a la base de datos, los distintos modelos a almacenar en nuestra realtime database.

Centrándonos en el desarrollo de la aplicación, tras instalar todo lo necesario para poder comenzar, conectamos la base de datos, alojada ya en FireBase, a nuestra aplicación y realizamos algunas tareas de código, como mostrar un mapa geo-localizando nuestra posición con un marcador.

Terminamos esta etapa ejecutando la aplicación en un emulador proporcionado por **Ionic**, en el cual pudimos ver como lucía la aplicación en ambos tipos de dispositivo, **Android** e **IOS**.

Etapas 3 de Animal's Safe.

Tras pasar la etapa de estudio y formación de las herramientas a utilizar, llega la fase de desarrollo. En esta fase nos ocupamos de establecer todas las funcionalidades de nuestra **aplicación**. Además, comenzamos a profundizar en la memoria del proyecto, pudiendo establecer secciones como los objetivos, los recursos y algunas etapas.

En primer lugar nos centramos en introducir los anuncios en la base de datos, ya que una vez hecho esto, podremos introducir todo tipo de datos desde la aplicación hacia **Firestore**.

En segundo lugar implementamos todo lo referente al mapa y a los servicios proporcionados por **Google Maps** junto con los diferentes tipos de marcadores. Tuvimos algunos problemas a la hora de **geocodificar** las direcciones que introducimos mediante el formulario, puesto que el servicio de google no funcionaba como debiera así que implementamos la **geocodificación** de otra manera.

Hecho lo anterior, nos centramos en la autenticación de usuarios. **Firestore** nos proporciona, junto con **Angular**, una serie de métodos ya implementados para realizar la autenticación de usuarios (registro, login, logout) y también métodos para conocer los datos del usuario con la sesión activa. Aquí se nos planteó, de nuevo, un problema que más tarde sería recurrente.

El siguiente paso fue introducir imágenes en el servicio **Storage** provisto por **Firestore**, para guardar archivos, también en tiempo real. Tanto introducirlas como recuperarlas para después ser mostradas en la aplicación.

Por último, las notificaciones **push** para que los usuarios sean notificados cuando haya novedades o nuevos registros en la **aplicación**.

La fase de desarrollo, que ocupa la mayor parte del tiempo dentro de todo proyecto, es la más difícil, debido a que cada paso adelante que se da en una herramienta desconocida, aunque previamente estudiada, puede ser un error y un paso atrás. Por eso a esta fase se le dedica mucho más tiempo, puesto que necesita avanzar por la vía ensayo-error.

Aunque en esta fase también es donde se puede apreciar el avance real de la aplicación, por lo que es muy buen momento de cara a la motivación para seguir adelante con el proyecto.

Etapas 4 de Animal's Safe.

Terminado el grosso del desarrollo de nuestro código, nos ponemos a realizar pruebas a nuestra aplicación. Pruebas tanto de caja blanca como de caja negra, permitiéndonos conocer una mayor cantidad de errores a corregir.

Solucionamos errores de implementación, capturar excepciones que no hayamos capturado y depurar la aplicación.

Al terminar toda la funcionalidad principal de la aplicación, también comenzamos a implementar mejoras para la aplicación. Cambios para proporcionar una mejor experiencia al usuario.

Además de esto, mejoramos el diseño de la aplicación, puesto que tenemos opiniones externas que nos ayudan a ver la app con mayor perspectiva.

Al terminar esta etapa, la documentación en referencia a la implementación del código y al funcionamiento de la app se termina, para dar comienzo a la última parte de esta.

Etapa 5 de Animal's Safe.

En la etapa final del desarrollo de la app, solo nos queda redactar la documentación no referente al código y preparar la presentación de cara a la exposición.

Terminamos el desarrollo de la aplicación y la documentación a la vez que realizamos la presentación del proyecto. En ella exponemos todos los conocimientos adquiridos durante el desarrollo de la aplicación junto con una explicación de que hace cada bloque de la **App**. Además resaltamos los problemas que hemos tenido durante el desarrollo del proyecto.

Resultados y conclusiones.

El resultado final de la aplicación ha sido satisfactorio. El desarrollo nos ha permitido adquirir conocimientos sobre tecnologías, para nosotros, desconocidas. Consideramos que tras todo lo aprendido somos capaces de desarrollar nuevas aplicaciones de un nivel de dificultad medio utilizando estas herramientas. También nos ha servido como una puesta en práctica real de todo lo aprendido durante todo el ciclo formativo. Hemos integrado elementos de varias materias, como por ejemplo desarrollo de bases de datos, diseño web, programación por eventos... Por otro lado la redacción de la documentación también ha sido un ejercicio muy enriquecedor de cara a futuros proyectos.

La dificultad ha sido elevada puesto que no teníamos ningún conocimiento sobre desarrollo de aplicaciones móviles híbridas, además de que nunca habíamos trabajado con interacción de datos en tiempo real.

La aplicación cumple todos los objetivos marcados y es completamente funcional. Una vez terminado el proyecto y continuemos aprendiendo de estas herramientas, también ampliaremos las distintas funcionalidades de la aplicación, haciendo mejoras y proporcionando así una mejor experiencia de usuario.

El potencial de la aplicación que hemos desarrollado es muy elevado. Actualmente no existe nada parecido en el mercado, y puede tener un gran impacto social ya que la aplicación se basa en una idea solidaria que fomenta la protección de los animales.

Para concluir, la aplicación que hemos desarrollado es una aplicación híbrida, completamente operativa en móviles IOS y Android, que se podrá adquirir en las stores de los dispositivos de manera gratuita.

Apéndice. Funcionamiento de la aplicación.

Al abrir la aplicación, el usuario se situará en la pestaña **Inicio**. En ella (Fig.3, Fig.4) podrá visualizar un mapa donde, mediante marcadores, se situará su posición y las asociaciones y clínicas cercanas a él. Cada marcador tendrá información para que el usuario pueda conocer ciertos datos, como el nombre o el horario de apertura. Cualquier usuario, esté registrado o no, podrá acceder a esta pestaña.

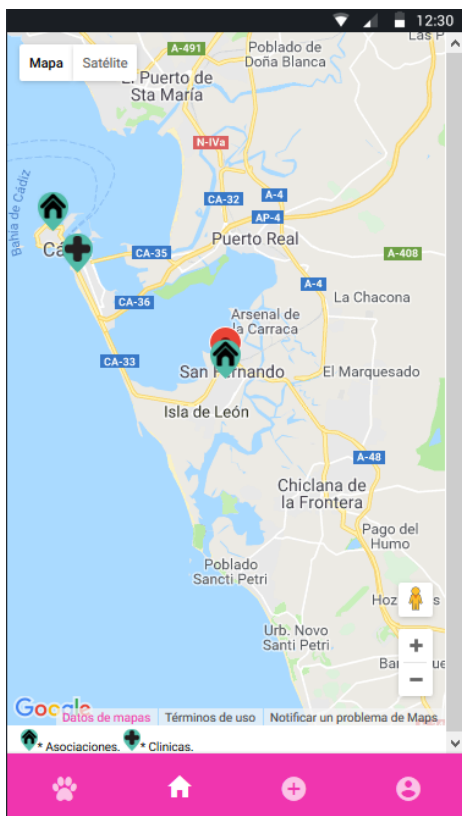


Fig.3 Pestaña Inicio Dispositivo Android

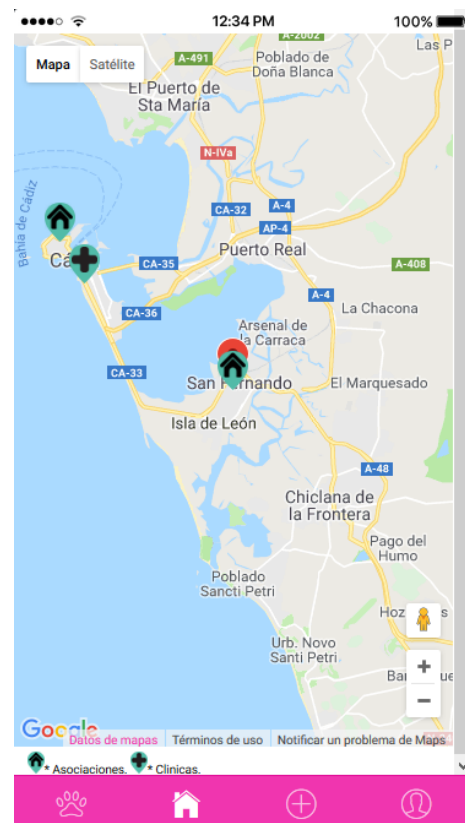


Fig.4 Pestaña Inicio Dispositivo IOS

La pestaña de **Animales Perdidos** (Fig.5, Fig.6) tiene una función parecida a la pestaña **Inicio**. Muestra los anuncios publicados por los usuarios de sus mascotas perdidas. A esta pestaña también puede acceder un usuario que no esté registrado en la aplicación.

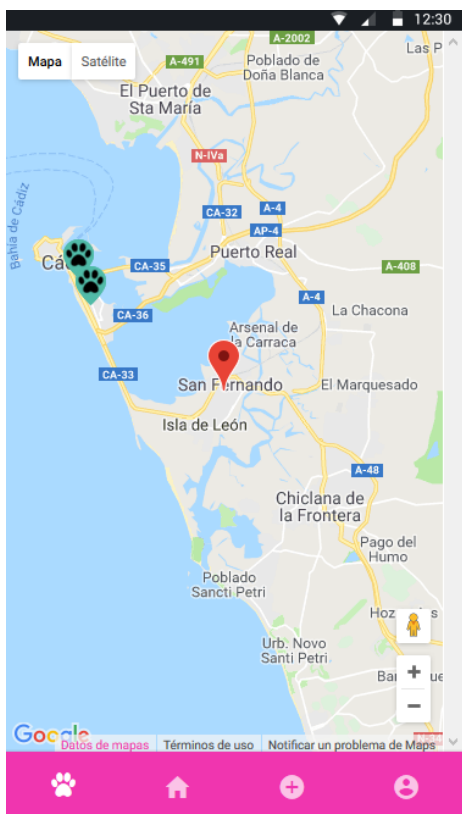


Fig.5 Pestaña Animales Perdidos, Android

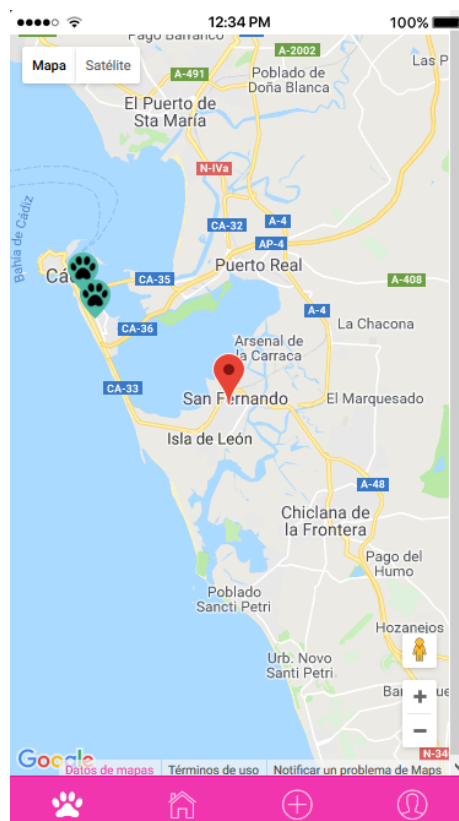


Fig.6 Pestaña Animales Perdidos, IOS

La siguiente pestaña, **Anuncios**, tiene dos plantillas, la primera (Fig.7, Fig.8) muestra un texto cuando un usuario no está autenticado en la aplicación, junto con un botón para ir al login/registro. La segunda (Fig.9, Fig.10) es un formulario que permitirá a los usuarios introducir un anuncio sobre su mascota perdida, incluyendo su localización y una foto. Este anuncio aparecerá en la pestaña **Animales Perdidos**.

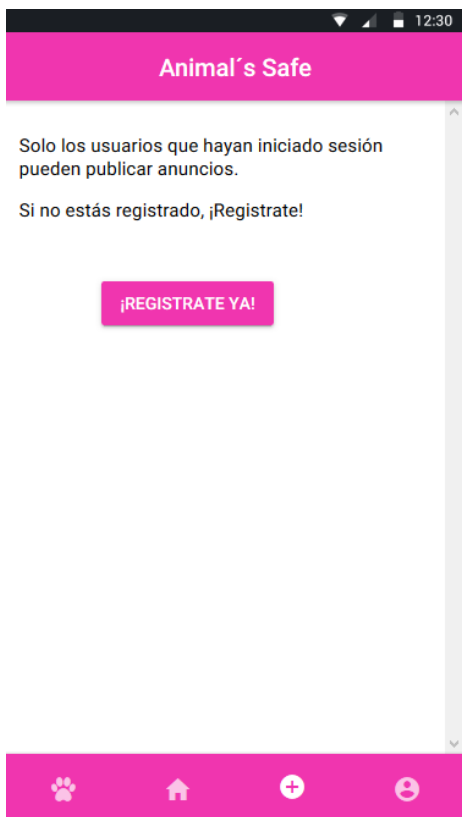


Fig.8 Pestaña Anuncios, Android

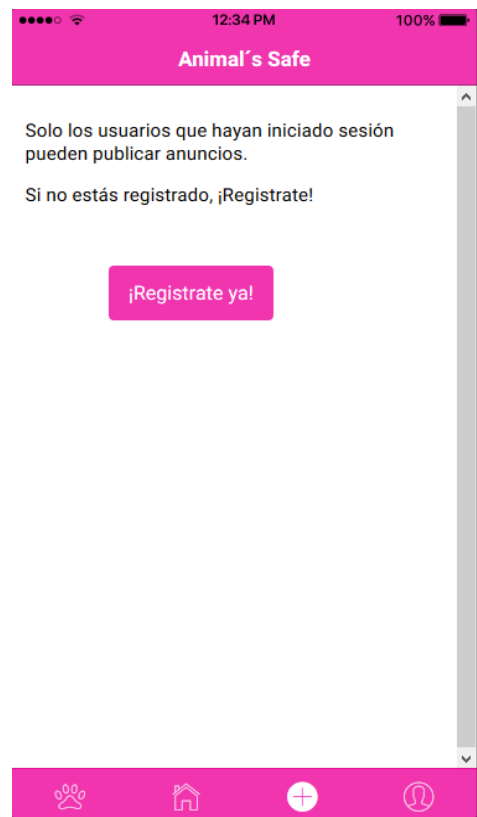


Fig.9 Pestaña Anuncios, IOS

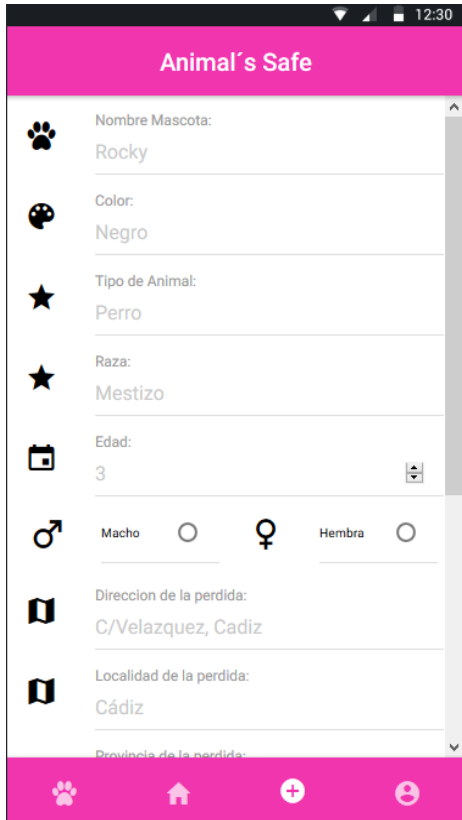


Fig.10 Pestaña Anuncios, Android

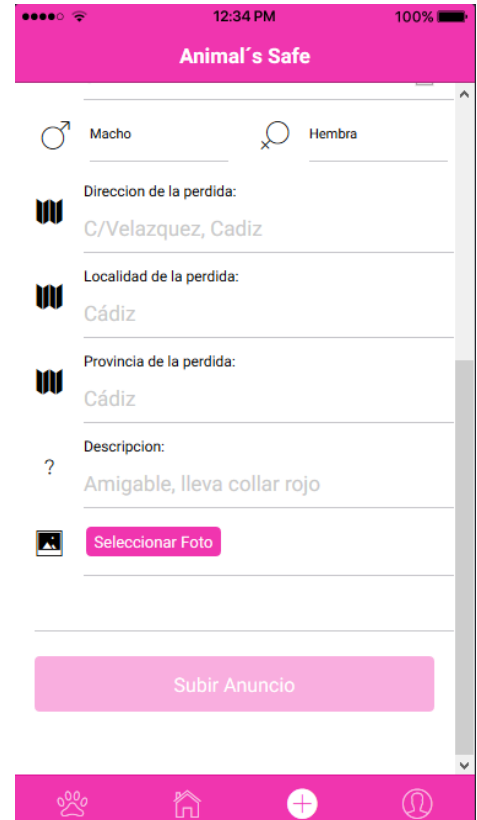


Fig.11 Pestaña Anuncios, IOS

Por último, la pestaña **Perfil de Usuario**. En ella, si el usuario no está autenticado (Fig.12, Fig.13) aparece un formulario de inicio de sesión, y un botón que lleva a un formulario de registro.

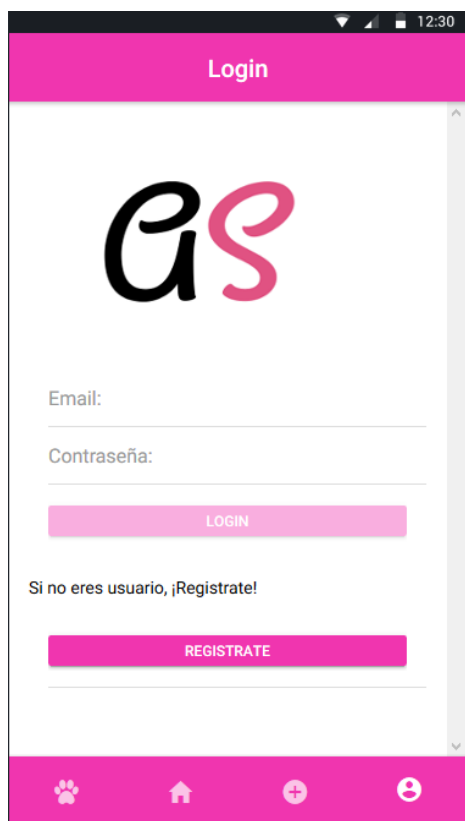


Fig.12 Pestaña Perfil de Usuario, Android

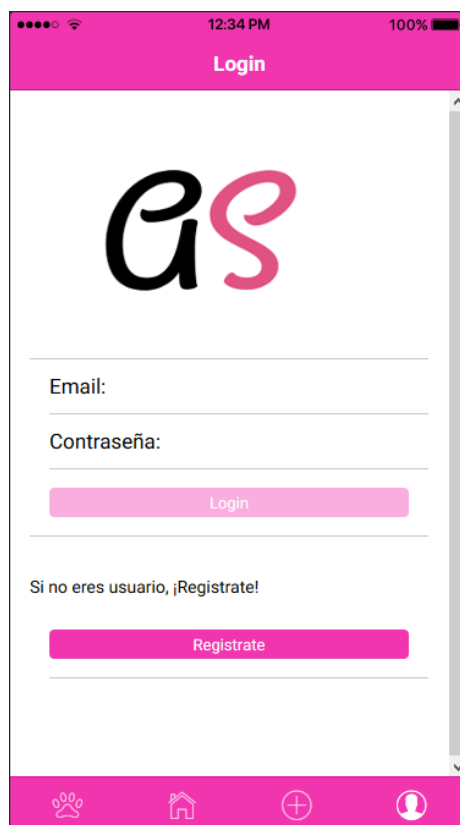


Fig.13 Pestaña Perfil de Usuario, IOS

Si el usuario pulsa sobre el botón de 'Regístrate', le llevará a un formulario (Fig.14, Fig.15) que le pedirá una serie de datos para darle de alta como nuevo usuario.

Fig.14 Pestaña Perfil de Usuario (Registro), Android

Fig.15 Pestaña Perfil de Usuario (Registro), IOS

Si el usuario está autenticado, en la pestaña **Perfil de Usuario** podrá ver sus datos (Fig.16, Fig.17).

Fig.16 Pestaña Perfil de Usuario (Datos), Android

Fig.17 Pestaña Perfil de Usuario (Datos), IOS

Si el Usuario pulsa en el botón situado arriba a la derecha en su perfil (ver Fig.16, Fig.17) podrá acceder a un formulario (Fig.18, Fig.19) con el que editar los datos de su perfil.

Fig.18 Pestaña Perfil de Usuario (Editar), Android

Fig.19 Pestaña Perfil de Usuario (Editar), IOS

Por último, dentro del propio Perfil de Usuario, el Usuario podrá acceder a sus anuncios. Pulsando en el botón Anuncios Publicados (ver Fig.16), obtendrá una lista de sus anuncios (Fig.20, Fig.21). En esta lista el usuario podrá eliminar sus anuncios deslizando estos hacia la izquierda (ver Fig.20).

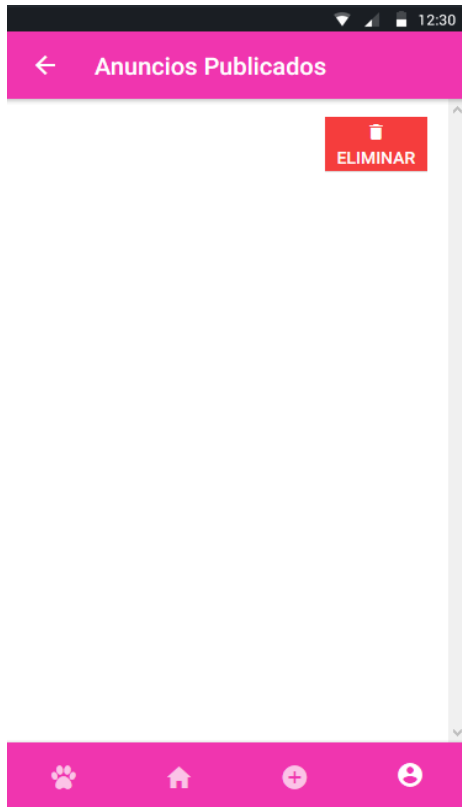


Fig.20 Pestaña Perfil de Usuario (Anuncios), Android

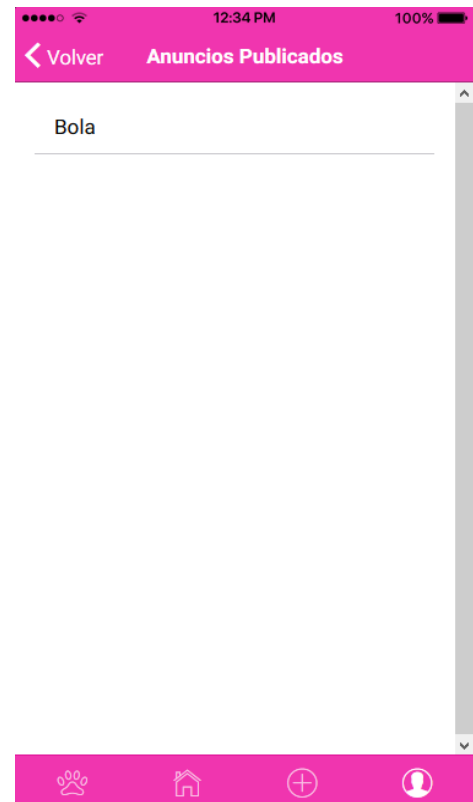


Fig.20 Pestaña Perfil de Usuario (Anuncios), IOS

Y finalizando la pestaña Perfil de Usuario, si se pulsa en el anuncio, el usuario podrá ver los datos de este (Fig.21, Fig.22).

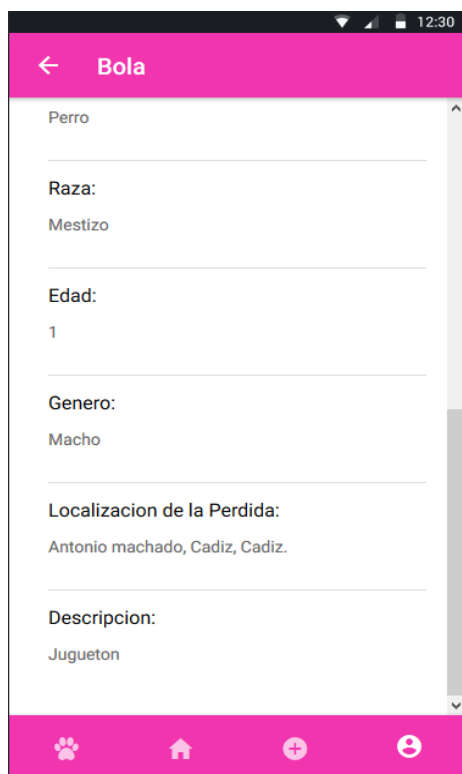


Fig.21 Pestaña Perfil de Usuario (Detalles Anuncio), Android

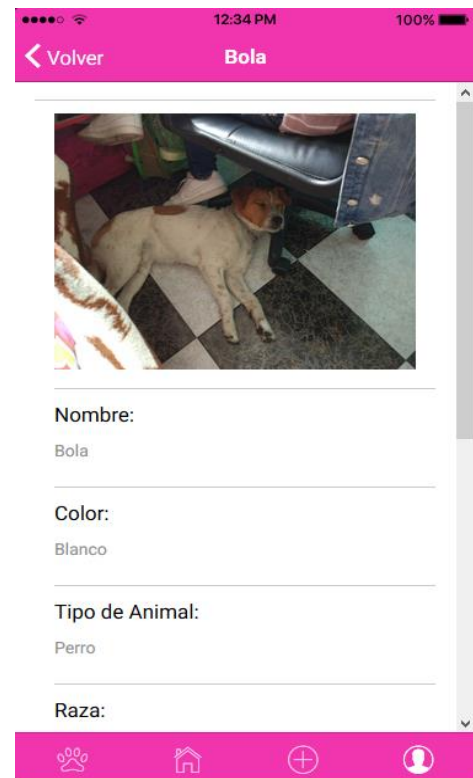


Fig.21 Pestaña Perfil de Usuario (Detalles Anuncio), IOS

Apéndice. Detalles sobre la implementación del proyecto.

En este apéndice explicamos cómo hemos confeccionado cada parte de la aplicación, pasando por los diferentes **tabs** (pestañas) y **la base de datos**.

Una vez creada la aplicación e instalados los módulos **IOS** y **Android**, para que la aplicación pueda ser ejecutada en ambos tipos de dispositivos, establecimos los **tabs**. En principio pensamos en situar 3 **tabs** en la parte inferior de la pantalla, pero luego añadimos una cuarta sección para que los usuarios pudieran gestionar su perfil.

Los dos primeros **tabs**, **Inicio** y **Animales Perdidos**, son dos mapas a pantalla completa que, mediante **marcadores**, muestra distintos lugares de relevancia. En la pestaña **Inicio** mostrará las **asociaciones** y **clínicas** cercanas al usuario, con una pequeña información en cada marcador sobre estas. A su vez, también implementamos el **tab Anuncios**, en el cual, como su nombre indica, el usuario puede introducir un nuevo anuncio de una mascota perdida.

El desarrollo de estas pestañas lo realizamos en paralelo puesto que el mapa necesita recoger registros de la base de datos para poder mostrarlos en forma de marcador.

En este punto nos encontramos la primera dificultad. Los métodos para la **geocodificación** proporcionados por **la API de Google Maps** no funcionaban como debieran, por lo que tuvimos que crear un **provider** propio (**GeocodingProvider**), el cual, luego utilizaremos en las pestañas que muestran los mapas. Este nuevo **provider** recoge la dirección proporcionada por el usuario en el formulario de publicación de un nuevo anuncio, lo pasa por la **API** mediante una petición **GET HTTP** de la cual se obtiene su latitud y longitud, y con eso, ponemos un nuevo marcador en el mapa.

Una vez logramos tanto introducir datos como plasmarlos en forma de marcador en los mapas, pasamos a introducir archivos desde el formulario de anuncios hacia la base de datos.

Aquí surgió un nuevo problema. Los archivos no se pueden almacenar directamente en la base de datos realtime, por lo que tuvimos que usar el servicio proporcionado por **Firestore** llamado **Firestore Storage**.



En él tuvimos que crear una referencia para cada usuario de tal manera que pudiésemos almacenar las distintas imágenes de sus anuncios y tener así el **Storage** organizado.

Además de crear la referencia y almacenarla, también teníamos que guardar el nombre de la imagen junto con los datos del anuncio en la base de datos. Esto lo hicimos implementando un método que buscara por nombre de imagen junto con el email del usuario para así poder obtener la imagen correspondiente al anuncio.

Por último llegamos a la **gestión de Usuarios**. **Firestore** implementa un servicio de autenticación para poder manejar tanto el alta como la baja de usuarios en nuestra App.

Creamos tanto un formulario de registro como un formulario de acceso a la aplicación. Además también creamos una referencia a los usuarios en la base de datos para poder almacenar no solo su email y contraseña, si no también algunos datos como su nombre, dirección o una foto de perfil. Esta última ha sido almacenada y recuperada como hicimos previamente en los anuncios de animales perdidos.

El usuario, una vez autenticado, podrá colgar anuncios en la aplicación y ver su información de perfil junto con los anuncios que haya publicado. Además podrá interactuar con esta información, ya que hemos implementado un formulario por si deseara cambiar algunos datos de su perfil, o borrar alguno de sus anuncios.

Trabajo futuro.

Una vez finalizado el proyecto y realizada su exposición, queremos continuar con el desarrollo de la aplicación. Seguir incorporándole mejoras y publicitándola para que se dé a conocer, y que pueda llegar a ser una aplicación usada mundialmente.

Webgrafía.

Documentación oficial de Ionic:

- <https://ionicframework.com/docs/>

Documentación oficial de TypeScript:

- <https://www.typescriptlang.org/docs/home.html>

Como usar bases de datos en tiempo real:

- https://manuais.iessanclemente.net/index.php/Firebase_-_Base_de_Datos_en_Tiempo_Real

Manual de SASS:

- <https://librosweb.es/libro/sass/>

Cursos de Ionic para crear una aplicación básica:

- <https://www.udemy.com/curso-de-ionic-aprende-a-crear-una-aplicacion-ios-o-android/>

Como introducir mapas usando GoogleMap JS e Ionic:

- <https://blog.ng-classroom.com/blog/ionic2/google-maps-js-and-ionic/>.
- <https://www.joshmorony.com/ionic-2-how-to-use-google-maps-geolocation-video-tutorial/>
- Mapa de calor:
<https://developers.google.com/maps/documentation/javascript/rebase?hl=es>
- Múltiples marcadores:
<https://github.com/angular/angularfire2/issues/558>

Cómo guardar localizaciones de un mapa en una Base de Datos.

- <https://reviblog.net/2017/04/06/tutorial-de-ionic-2-crear-una-aplicacion-para-guardar-nuestros-sitios-geolocalizados-parte-5-guardando-nuestros-sitios-en-una-base-de-datos-local/>

Cómo acceder a la FireBase desde Ionic. Documentación oficial de Firebase:

- <http://masteringionic.com/blog/2017-10-22-using-firebase-cloud-firestore-with-ionic/>
- <https://firebase.google.com/docs/web/setup>
- <http://cookyourweb.com/ionic-y-firebase/#ionic>

Crear formularios en Ionic:

- <https://blog.ng-classroom.com/blog/ionic2/forms-with-ionic/>

Obtener latitud y longitud de una dirección con geocodificación:

- <https://developers.google.com/maps/documentation/geocoding/start?hl=es>

Realizar el login y signUp de un usuario:

- <https://blog.ng-classroom.com/blog/ionic2/clase-3-user-auth/>
- <https://firebase.google.com/docs/auth/admin/manage-users?hl=es-419>

Subir imagen a firebase desde una aplicación desarrollada en Ionic:

- <https://forum.ionicframework.com/t/ion-input-type-file-full-path/101790/8>

Autocompletar:

- https://www.w3schools.com/howto/howto_js_autocomplete.asp