

Tindog

Aplicación híbrida para la adopción de animales de protectoras y asociaciones.

David Ponce Segura

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a Eduard Martín Lineros

Nombre Profesor/a responsable de la asignatura Carles Garrigues Olivella

Junio 2019



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Aplicación híbrida para la adopción de animales de protectoras/asociaciones.
Nombre del autor:	David Ponce Segura
Nombre del consultor/a:	Eduard Martín Lineros
Nombre del PRA:	Carles Garrigues Olivella
Fecha de entrega:	13/3//2019
Titulación::	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	Castellano
Palabras clave	<i>aplicación, híbrida, animales</i>

Resumen del Trabajo:

En el presente trabajo se pretende abordar el desarrollo de una aplicación móvil con el objetivo de facilitar la búsqueda y adopción de animales de compañía que se encuentren en protectoras o a cargo de asociaciones especializadas.

La aplicación se diferenciará de su competencia más directa intentando otorgarle una experiencia de usuario cercana a las aplicaciones de búsqueda de pareja, o de contactos personales, dándole así, por un lado, un tono más simpático y, por otro, distanciándola lo máximo posible del concepto de tienda o mercado.

Se utilizarán tecnologías de desarrollo híbridas, ya que las especificaciones técnicas así lo permiten y esto facilitará su portabilidad a los principales sistemas móviles: Android e iOS.

Como complemento a la aplicación, sería deseable el desarrollo de una plataforma web para la gestión de la información, fichas, adopciones y contacto por parte de las protectoras o asociaciones.

Como producto final se obtendrá una aplicación en la que los usuarios podrán explorar perfiles de animales disponibles para su adopción, según una serie de preferencias establecidas por ellos mismos y, una vez seleccionado uno (o varios), se pondrán en contacto con la entidad responsable para completar el trámite.

Abstract:

The present work tries to approach the development of a mobile application with the aim of facilitating the search and adoption of companion animals that are in charge of specialized associations.

The application will be differentiated from its more direct competition by providing an user experience closer to the dating applications, giving it, on one hand, a more sympathetic tone and, on the other, distancing it as much as possible from the concept of a store or a market.

Hybrid development technologies will be used, since the technical specifications allow it and this will facilitate their portability to the main mobile systems: Android and iOS.

As a complement to the application, it would be desirable to develop a web platform for the management of information, cards, adoptions and contact by the associations.

As an end product, an application will be obtained in which users will be able to explore available animals for adoption, according to a series of preferences established by themselves and, once one (or several) is selected, they will contact the responsible entity to complete the procedure.

Índice

1.Introducción	8
1.1 Contexto y justificación del Trabajo	8
1.2 Objetivos del Trabajo	8
1.2.1 Objetivos principales	8
1.2.1 Objetivos secundarios	9
1.2.2 Breve resumen de productos obtenidos	9
1.2.3 Enfoque y método seguido	9
2. Planificación	10
3. Diseño	12
3.1 Benchmarking	12
Animalia	12
Miwuki	13
Adoptarperro.es	14
3.2 Análisis de requisitos	14
3.2.1 Usuarios y contexto de uso	14
Asociación	15
Usuario	15
Personas	16
3.3 Diseño conceptual	17
3.3.1 Casos de uso	17
3.3.2 UML base de datos	20
3.3.3 Arquitectura del sistema	21
3.3.4 Prototipado	22
Prototipo baja fidelidad	22
3.3.5 Elementos gráficos	23
Gama cromática	23
Tipografías	23
Iconos y elementos gráficos	24
Logotipo	24
3.3.6 Prototipo alta fidelidad	25
4. Desarrollo	28
4.1 Tecnologías y herramientas para el desarrollo	28
4.1.1 Aplicación móvil	29
Librerías	29
4.1.2 Servidor	31
4.1.3 Gestión	31

4.1.4 API REST	36
5 Conclusiones	38
6 Posibles ampliaciones	38
7 Bibliografía	38
Anexo I	38

Índice de figuras

Fig 1. Reparto temporal de tareas.	12
Fig 2.Capturas pantalla Animalia.	12
Fig 3.Capturas pantalla Miwuki.	13
Fig 4.Adoptaperro.es.	
14	
Fig 5. Perfil persona.	16
Fig 6.Caso de uso: registro de usuario.	17
Fig 7.Caso de uso: configuración y búsqueda.	18
Fig 8.Caso de uso: revisión de solicitudes/contacto.	19
Fig 9.Diagrama UML base de datos.	20
Fig 10.Arquitectura del sistema.	21
Fig 11.Prototipo baja fidelidad.	22
Fig 12.Prototipo alta fidelidad.	27
Fig 13. Arquitectura de aplicación Cordova.	28
Fig 14.Intranet-Listado de asociaciones.	32
Fig 15.Intranet-Gestión de asociaciones.	32
Fig 16.Intranet-Gestión de fichas.	33

Índice de tablas

Tabla 1. Planificación por bloques.	10
-------------------------------------	----

1.Introducción

1.1 Contexto y justificación del Trabajo

El abandono de perros y gatos en España es un problema social real y del que en la actualidad se están ocupando, mayoritariamente, asociaciones privadas o grupos de vecinos, sin el apoyo institucional adecuado.

Según los resultados del estudio *Él nunca lo haría* [Affinity 2018] de la asociación Affinity sobre el abandono, más de 138.000 perros y gatos fueron recogidos a lo largo de 2018.

Muchas de estas asociaciones, dedicadas a la recogida y adopción de animales abandonados, se ven desbordadas por la cantidad de perros y gatos que deben atender y mantener y les resulta difícil poner en conocimiento de los posibles adoptantes las características y disponibilidad de adopción de los animales.

Aunque existen plataformas con fines similares, e incluso estas protectoras/asociaciones realizan esta misma tarea de difusión de una forma u otra en las redes sociales, es un trabajo laborioso para sus empleados y no siempre obtienen el alcance deseado.

1.2 Objetivos del Trabajo

El presente trabajo abarca el desarrollo de una aplicación móvil destinada a poner en contacto protectoras de animales/asociaciones con posibles adoptantes.

Con el desarrollo de esta aplicación se pretende facilitar el trabajo de alta y mantenimiento de nuevos animales en el sistema, así como los primeros pasos en el contacto entre asociación y adoptante, aunque no es objetivo de este trabajo el que la adopción se formalice completamente a través de la aplicación, debido a las diferencias de protocolo entre las diferentes entidades.

Por otra parte, y de cara al usuario final, se presentará con el aspecto y funcionalidad ya conocidos por muchos usuarios de las aplicaciones para encontrar pareja o entablar relaciones de persona a persona, dotándola así de un tono simpático y transmitiendo la idea de compromiso, en contraposición al de adquisición.

1.2.1 Objetivos principales

- Implementar una aplicación móvil que cubra la necesidad de difusión de animales en adopción por parte de las asociaciones y contacto entre las mismas y los posibles adoptantes.
- Detectar las posibles aplicaciones o sistemas que presten un servicio similar y analizar sus fortalezas y debilidades para aplicar los datos obtenidos en el desarrollo actual.
- Crear un diseño y un conjunto de elementos visuales propios para el desarrollo visual de la aplicación.
- Desarrollar una interfaz de usuario actual, amigable e intuitiva, basada en las aplicaciones de relaciones personales ya conocidas por muchos usuarios.

1.2.1 Objetivos secundarios

- Obtener un producto gratuito y de código abierto.
- Realizar todo el desarrollo utilizando únicamente herramientas de código abierto.

1.2.2 Breve resumen de productos obtenidos

- Aplicación compilada para Android: Una aplicación nativa compilada para Android.
- Aplicación compilada para iOS: Una aplicación nativa compilada para iOS.
- Base de datos común MySQL: Base de datos MySQL con estructura y datos preparada y alojada en un servidor de bases de datos, que servirá de nexo entre las aplicaciones cliente y la interfaz web.
- API de comunicación entre aplicaciones y base de datos: Sistema (posiblemente PHP) de comunicación basado en el estándar REST para la comunicación entre las aplicaciones y la base de datos.
- Intranet web para gestión de base de datos (opcional): Aplicación web de gestión para las asociaciones.

1.2.3 Enfoque y método seguido

Se ha optado por el uso de tecnologías de desarrollo híbridas, que permitirán adaptar la aplicación a los sistemas móviles mayoritarios en el mercado (Android e iOS) con mínimo esfuerzo, aunque exigirá prestar especial atención al rendimiento y al diseño de la interfaz.

Como complemento a la aplicación, sería deseable el desarrollo de una plataforma web para la gestión de la información, fichas, adopciones y contacto por parte de las protectoras o asociaciones.

2. Planificación

Para el desarrollo de la aplicación se emplearán tecnologías híbridas, por lo que será necesario un editor adecuado y las librerías de compilación para cada sistema operativo (Android e iOS). Adicionalmente sería deseable contar con, al menos, un dispositivo de cada tipo para realizar pruebas reales. Las herramientas empleadas se detallan en el Anexo I.

El trabajo se desarrollará dividido cuatro bloques:

Planteamiento y planificación. Fecha: 10/03/2019	<ul style="list-style-type: none">● Selección de tema. (3d)● Análisis de necesidades y objetivos.(6d)● Análisis de mercado actual y competencia.(4d)● Elección de metodología y plan de trabajo.(3d)
Diseño. Fecha: 03/04/2019	<ul style="list-style-type: none">● Estructuración de datos. (3d)● Sketch y pruebas. (2d)● Prototipo de baja fidelidad. (4d)● Prototipo de alta fidelidad. (6d)
Implementación. Fecha: 15/05/2019	<ul style="list-style-type: none">● Diseño de base de datos. (5d)● Programación de la interfaz. (10d)● Programación de API REST. (5d)● Desarrollo de plataforma web/intranet (opcional). (10d)
Finalización. Fecha: 05/06/2019	<ul style="list-style-type: none">● Revisiones y pruebas. (4d)● Análisis de objetivos cumplidos. (2d)● Presentación y memoria. (6d)

Tabla 1. Planificación por bloques.

Distribución de tareas por días y dependencias.

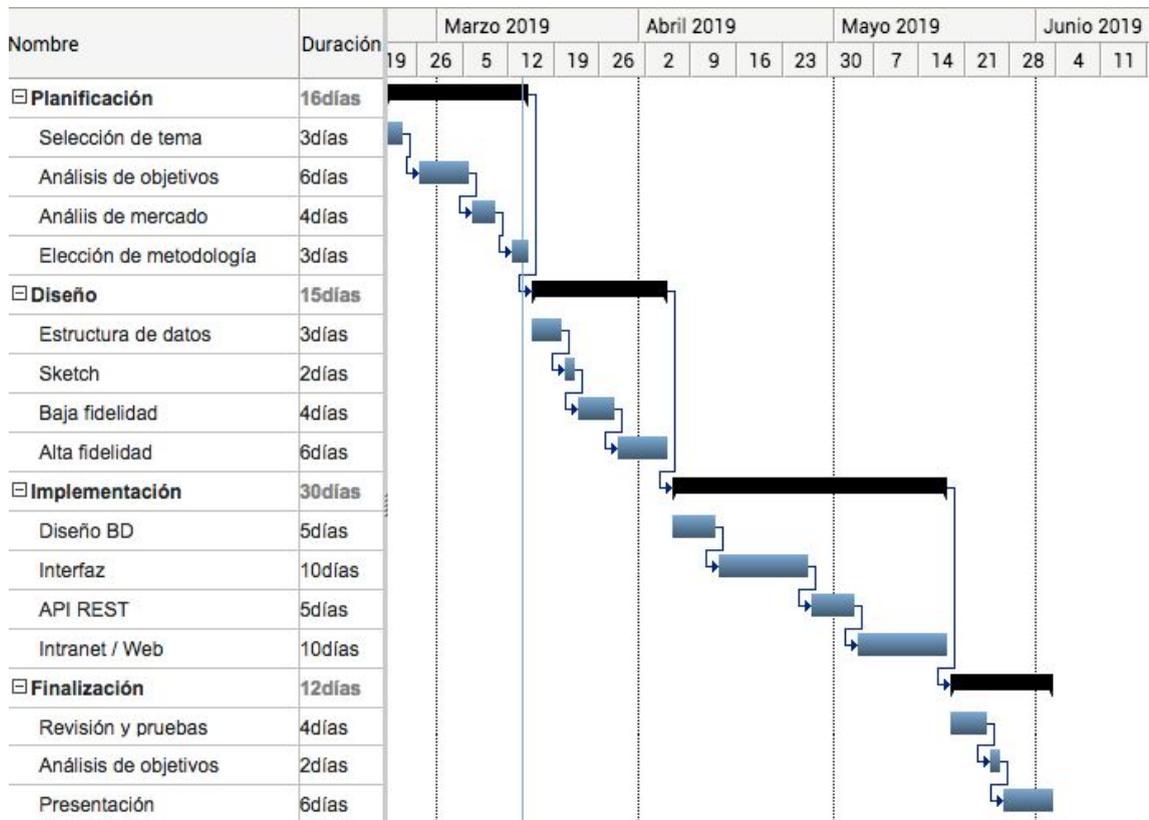


Fig 1. Reparto temporal de tareas.

3. Diseño

3.1 Benchmarking

En esta sección se analizarán tres de las principales aplicaciones o servicios con una funcionalidad y objetivos similares a las deseadas en el presente desarrollo.

Se destacan los principales puntos fuertes y debilidades de cada una de las aplicaciones analizadas.

Animalia

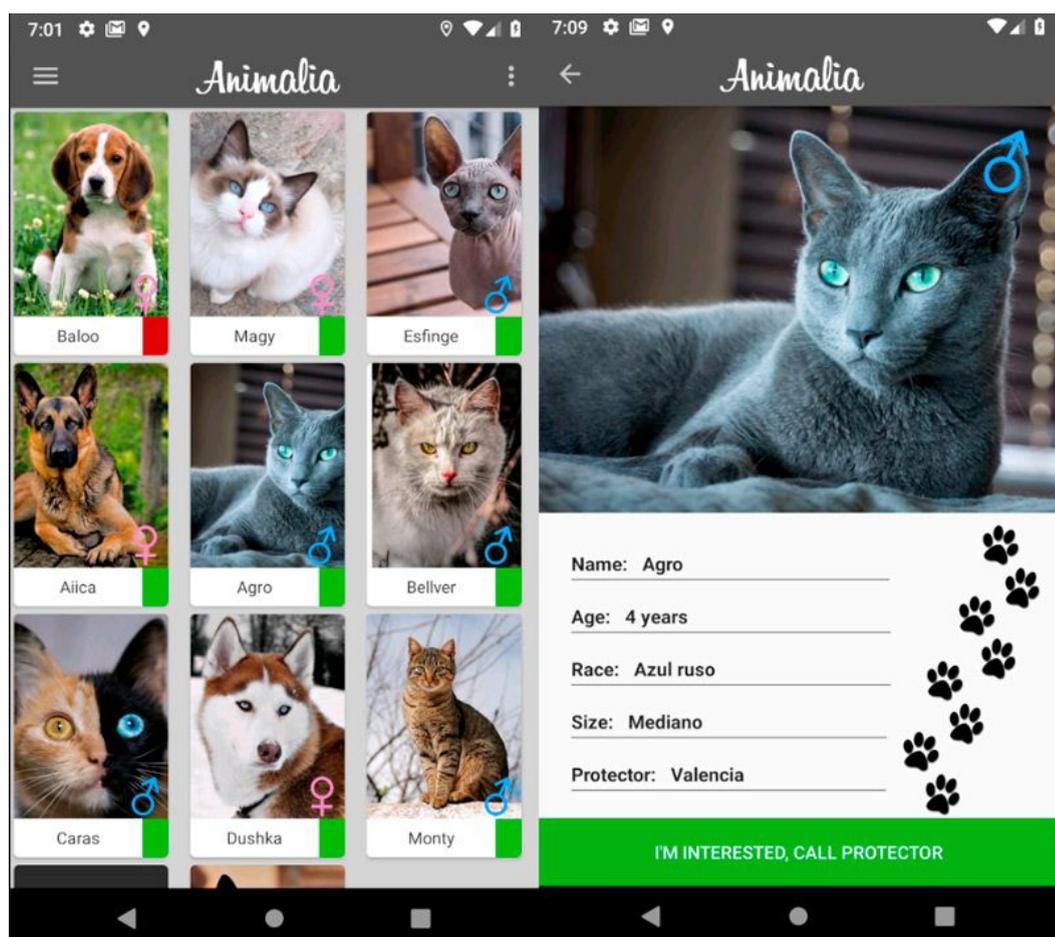


Fig 2.Capturas pantalla Animalia.

La más parecida en cuanto a funcionalidades respecto a nuestro objetivo. Funciona bien y tiene una interfaz clara.

Fortalezas: Es clara, con una interfaz limpia y actual. Sencilla y potente. Versión de escritorio disponible.

Debilidades: No hay asociaciones españolas afiliadas.

Miwuki

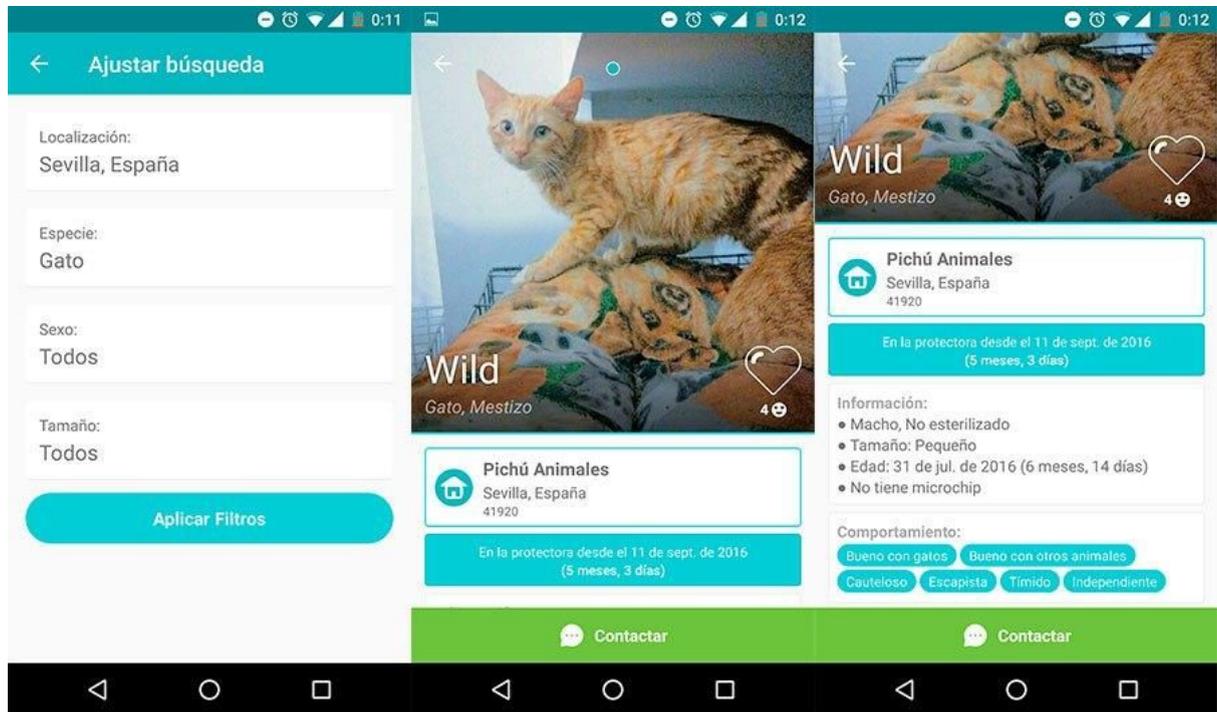


Fig 3. Capturas pantalla Miwuki.

También mantiene la sencillez y claridad de interfaz, aunque algo más compleja que Animalia. Es la más descargada en España.

Fortalezas: Interfaz limpia y actual. Dispone de versión web para uso más cómodo desde un PC. Base de usuarios buena.

Debilidades: La principal debilidad es la complicación que supone para las asociaciones el adaptar su sistema de gestión para encajar con el diseñado por la aplicación.

Adoptarperro.es



Fig 4. Adoptarperro.es.

Interfaz pobre e incómoda. Funcionalidad correcta, pero algo limitada.

Fortalezas: Es fácil de utilizar. Tiene una buena base de asociaciones Españolas participantes.

Debilidades: Falta de funcionalidades para búsquedas precisas. Interfaz pobre. Sin versión app (sólo web).

3.2 Análisis de requisitos

3.2.1 Usuarios y contexto de uso

En el siguiente apartado se analiza el tipo y las necesidades de los dos perfiles principales de usuarios implicados en el desarrollo de la aplicación: La asociación y el usuario final.

Por un lado, las asociaciones serán las encargadas de gestionar las altas, bajas y modificaciones de los perfiles de los animales, así como de seleccionar a los candidatos para la adopción, por lo que resulta importante tener en cuenta que datos les resultarán más útiles para este fin.

Los usuarios accederán a los perfiles en sus dispositivos y decidirán si están o no interesados en la adopción. Además, deberán rellenar un primer formulario de contacto con datos personales que las asociaciones necesitan para evaluar su idoneidad.

Asociación

Del lado de las asociaciones necesitamos conocer qué datos requieren para formalizar una adopción. En este caso nos encontramos con que cada asociación utiliza ya unos formularios en papel y protocolos, diferentes de unas a otras.

Se ha optado por contactar con varias asociaciones y generar un formulario de tipo encuesta para obtener un listado de los datos mínimos de cada usuario que puedan servir como base para generar un esquema común a todas ellas. (Formulario en el Anexo II, Resumen de respuestas en el Anexo III)..

Como resultado de la síntesis de los datos proporcionados por las asociaciones, obtenemos estos datos como imprescindibles para solicitar a los usuarios:

PERSONALES

- Nombre
- Dirección
- Ocupación
- Salario anual

VIVIENDA

- Tipo de residencia: piso, casa aislada, adosado.
- Miembros de la familia: adultos / niños
- Otros animales: perros / gatos / otros

Usuario

Aprovechando que, de una manera u otra, las asociaciones ya disponen de bases de datos de sus usuarios, las tomaremos como punto de partida para desarrollar los perfiles de usuario medios.

Para obtener un mejor perfil del usuario final, se ha aprovechado la experiencia de las asociaciones contactadas para extraer datos, aunque con dos precauciones:

- Los datos obtenidos pueden tener un margen de error estadístico a considerar, ya que las asociaciones nos los proporcionan de forma informal y sin referencias.
- El uso de la aplicación objeto del presente trabajo podría excluir algunos de los perfiles de usuarios presenciales en las asociaciones y/o llegar a perfiles no habituales actualmente en estos centros..

Otras fuentes:

Según el Informe Ditrendia: Mobile [Mobile, 2018] en España y en el mundo 2018, alrededor del 65% de usuarios de aplicaciones móviles se encuentra en el rango de edad de entre los 22 y los 45 años. De este informe extraemos también un uso de 2 horas al día.

Según el informe anual de la fundación Affinity [Affinity 2018], los 3 motivos más importante para optar por la adopción son: la sensibilidad frente al problema del abandono (40,2%), el deseo de colaborar con una entidad de protección animal (22,5%) y la recomendación de amigos o familiares..

Datos resumen para la generación del perfil persona:

54% mujeres y 46% son hombres.

Franja de edad de 22 a 45.

50% de estudios universitarios.

Personas

PROJECT: untitled



Background

Gloria acaba de terminar la carrera de diseño y ha comenzado a trabajar como diseñadora gráfica por cuenta propia. Trabaja desde casa y se reúne con sus clientes en una oficina de co-working.
Vive con su pareja en un piso céntrico de alquiler. No es muy grande, pero le gusa la zona. Tiene un gato desde hace 3 años, aunque su casera no lo sabe.
Le gustaría vivir en una casa unifamiliar con jardín para tener más animales.

Demographic

Female 26 years
Valencia
Single
Diseñadora gráfica

Motivations

Ganar más dinero.
Conseguir mejores clientes.
Adoptar otro animal de compañía.

Frustrations

Demasiado trabajo.
Casa muy pequeña.

Skills

Internet: 50%
Aplicaciones móviles: 50%
Redes sociales: 75%

Scenarios

Gloria termina tarde su jornada laboral tarde. Trabajar en casa a veces hace que se alargue demasiado.
Se sienta en el sofá junto a su gato y piensa que siempre ha querido tener un perro, aunque le resultaría difícil por espacio. Tal vez otro gato?
Mientras ve la televisión, abre su su aplicación de búsqueda de mascotas y empieza a visualizar perfiles hasta que encuentra al gato perfecto para ella. Lo agrega a la lista de favoritos y espera que la protectora la elija como candidata.

Fig 5. Perfil persona.

3.3 Diseño conceptual

3.3.1 Casos de uso

Registro de usuario

Descripción: El usuario accede a la pantalla principal de identificación. Si es la primera vez que accede, no tendrá credenciales por lo que accederá al proceso de registro. Deberá rellenar los datos solicitados con los que será dado de alta en el sistema. Una vez registrado, podrá acceder con sus credenciales desde la pantalla de identificación.

Actores: usuario.

Prerrequisitos: (ninguno).

Post Requisitos: Nuevo usuario registrado.

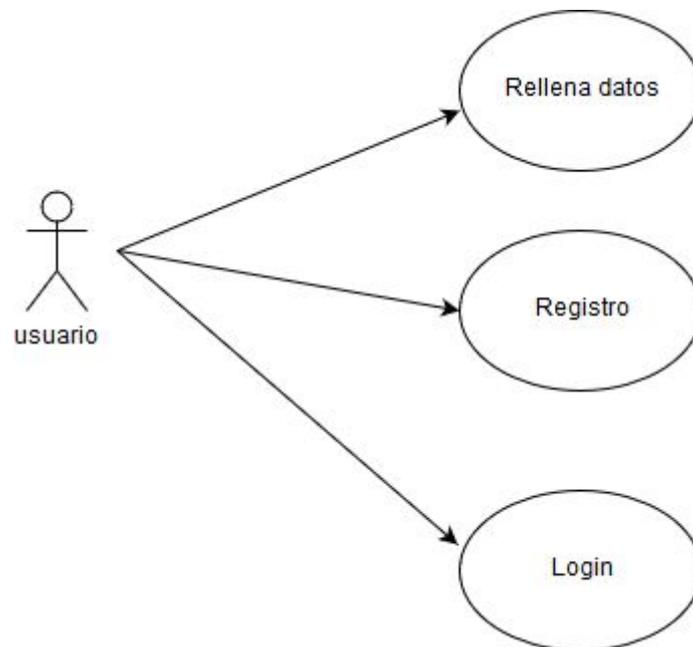


Fig 6.Caso de uso: registro de usuario.

Configuración y búsqueda

Descripción: El usuario accede a la aplicación con sus credenciales. Puede configurar sus preferencias de búsqueda y visualizar fichas que coincidan con esta configuración.

Si alguna ficha es de su interés, la marcará como favorita.

Esta información será accesible por el administrador.

Actores: usuario, administrador.

Prerrequisitos: Usuario registrado.

Post Requisitos: Nueva solicitud de adopción.

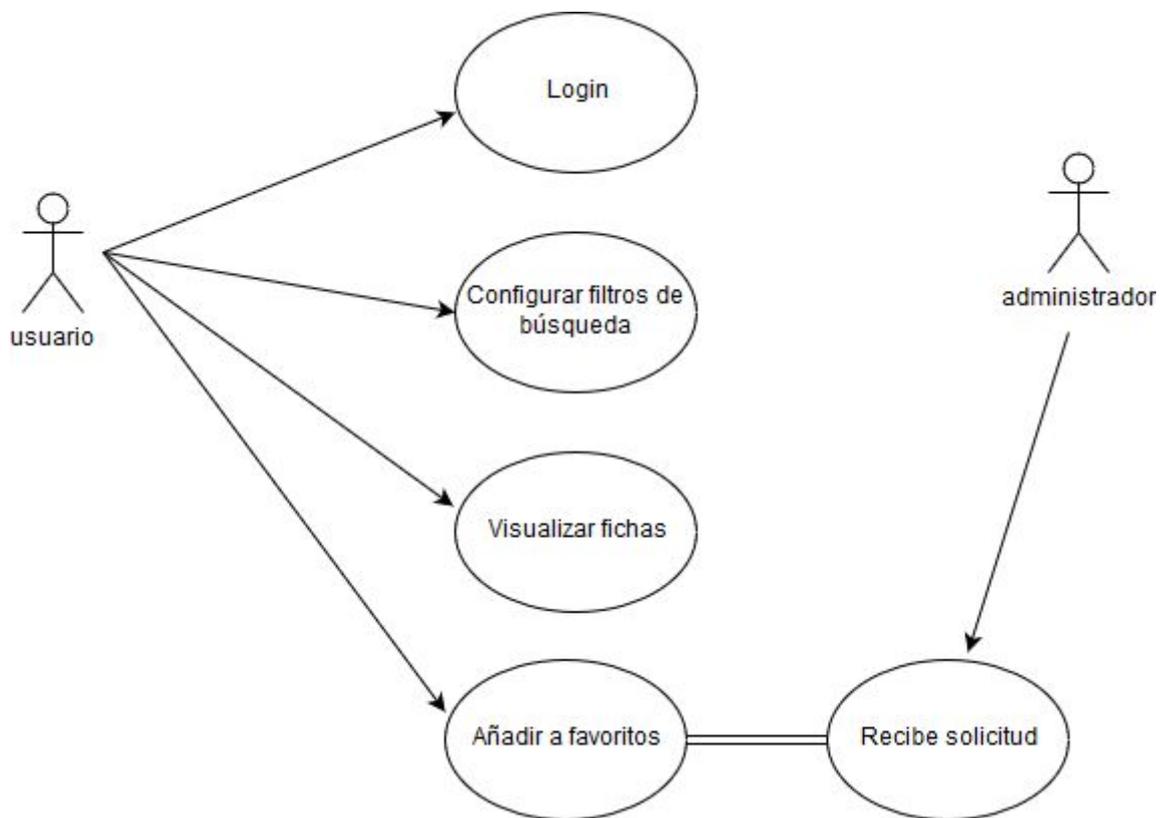


Fig 7.Caso de uso: configuración y búsqueda.

Revisión de solicitudes/contacto

Descripción: El usuario puede acceder a un listado de sus solicitudes pendientes y comprobar si están o no aprobadas. En el caso de que un administrador haya aprobado una solicitud, usuario y administrador establecerán contacto por medio ajeno a la aplicación.

Actores: usuario, administrador.

Prerrequisitos: Usuario registrado.

Post Requisitos: Contacto por vía externa a la aplicación.

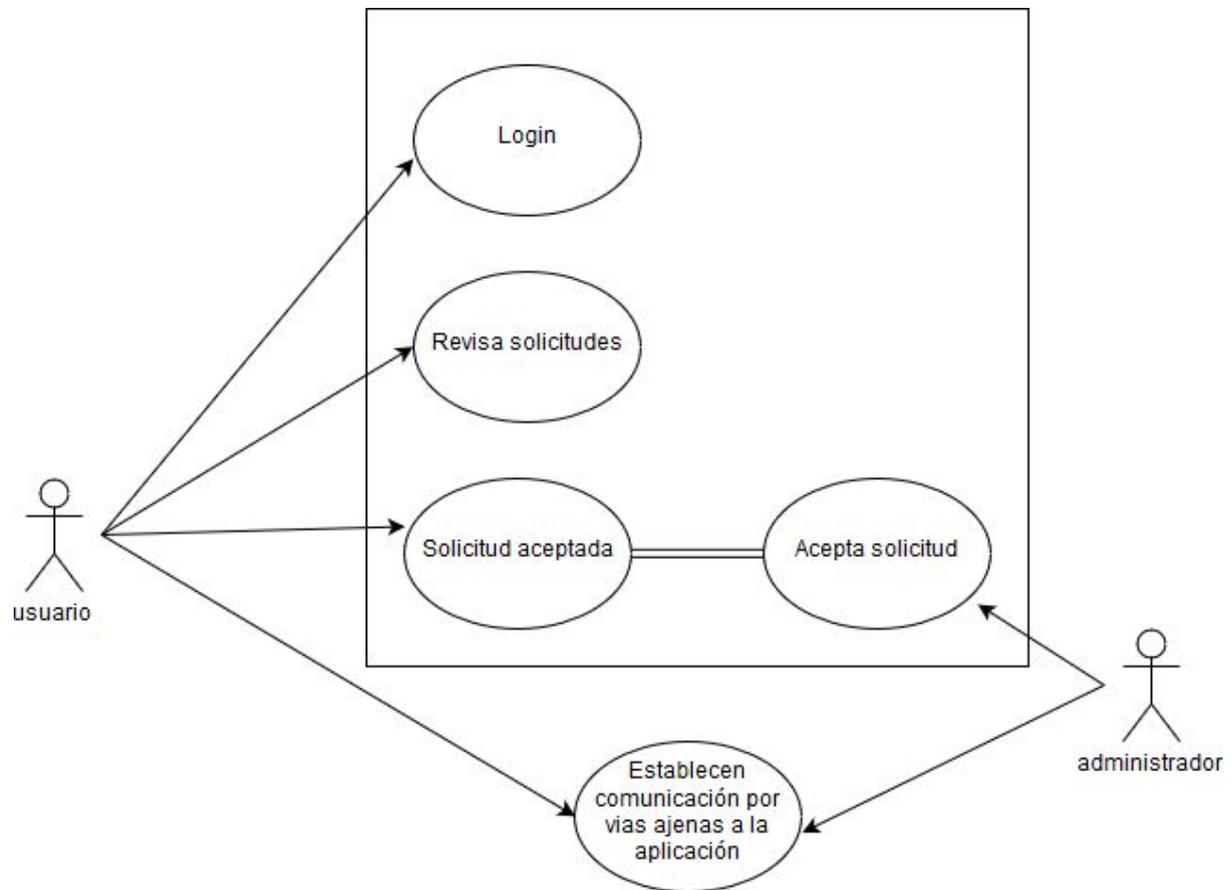


Fig 8.Caso de uso: revisión de solicitudes/contacto.

3.3.2 UML base de datos

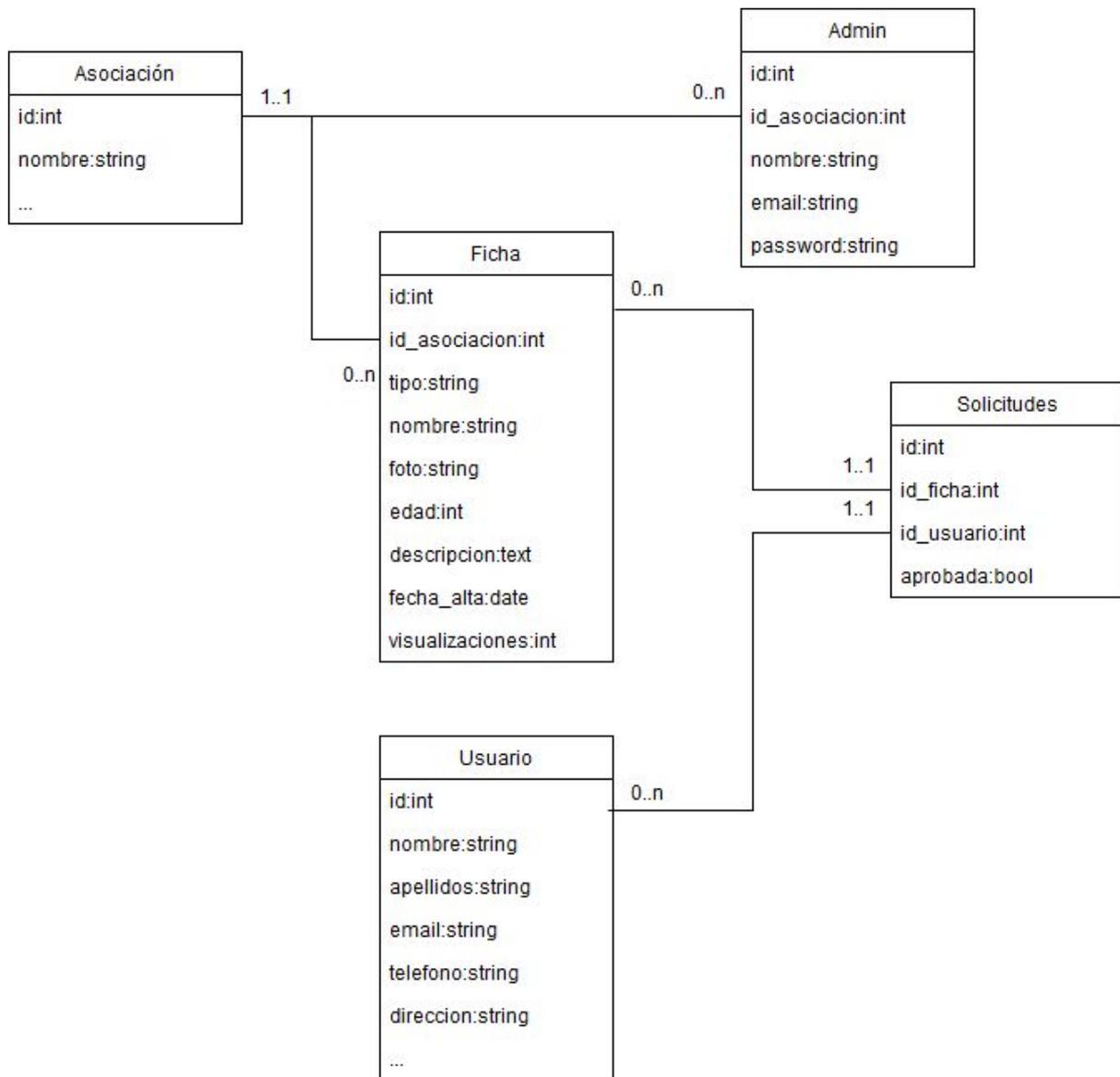


Fig 9. Diagrama UML base de datos.

3.3.3 Arquitectura del sistema

Se plantea la aplicación como una aplicación híbrida que nos permita su compilación en los principales sistemas operativos móviles (Android e iOS), con las mínimas modificaciones. Esto es posible gracias a que la aplicación no requiere de grandes recursos gráficos o de hardware por parte del dispositivo y de que se utilizará una base de datos remota común.

Del lado del servidor se plantea una base de datos MySQL, controlada por una aplicación PHP que haga las veces de controlador y de servidor REST para las aplicaciones cliente.

Se ha optado por implementar una arquitectura modelo-vista-controlador, utilizando una capa REST entre la capa de datos (servidor) y la aplicación.

En la parte cliente, se ha utilizado el framework Vue.js como nexa entre la vista y el controlador y Cordova como framework que hará las veces de comunicación entre nuestra aplicación, los recursos del dispositivo y la interfaz web.

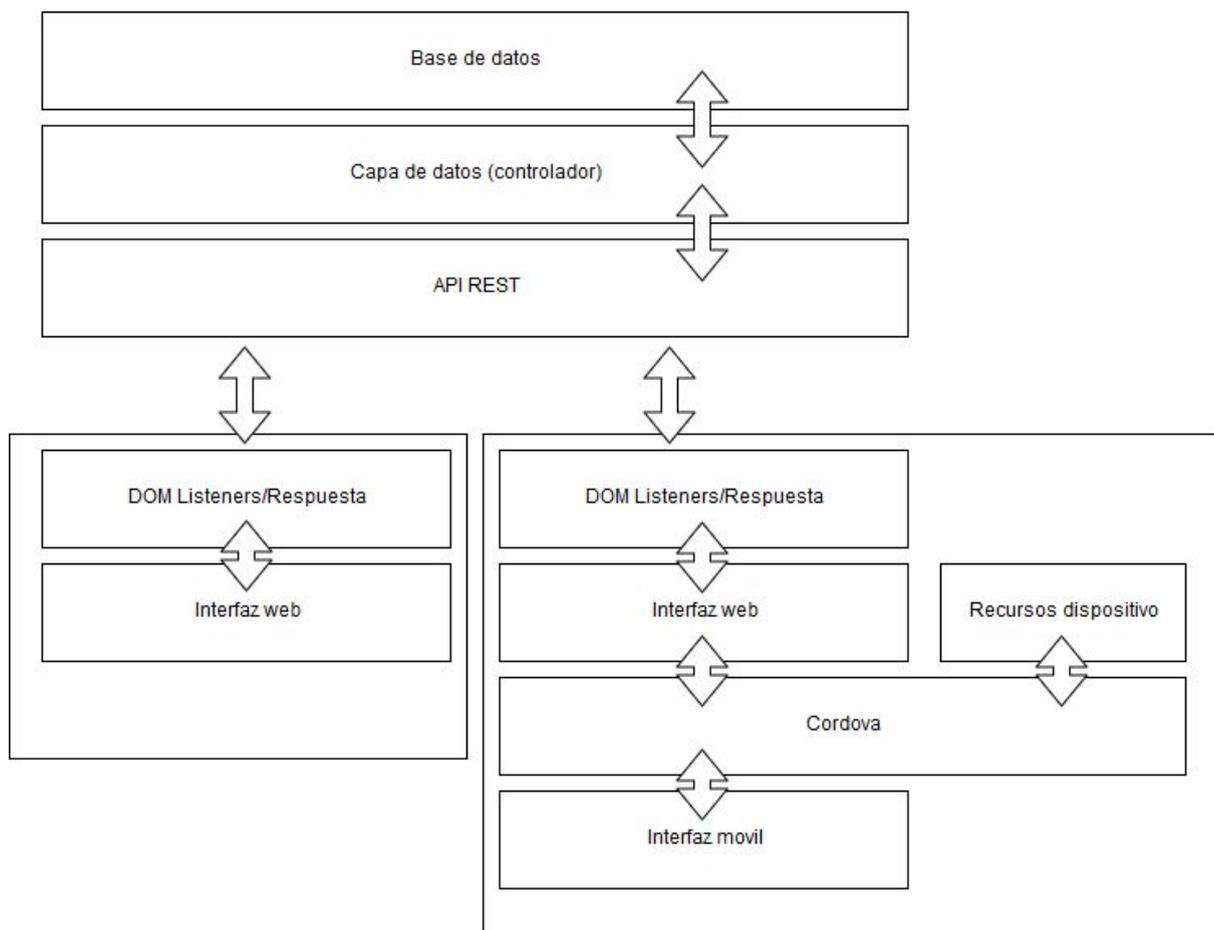


Fig 10.Arquitectura del sistema.

3.3.4 Prototipado

Prototipo baja fidelidad

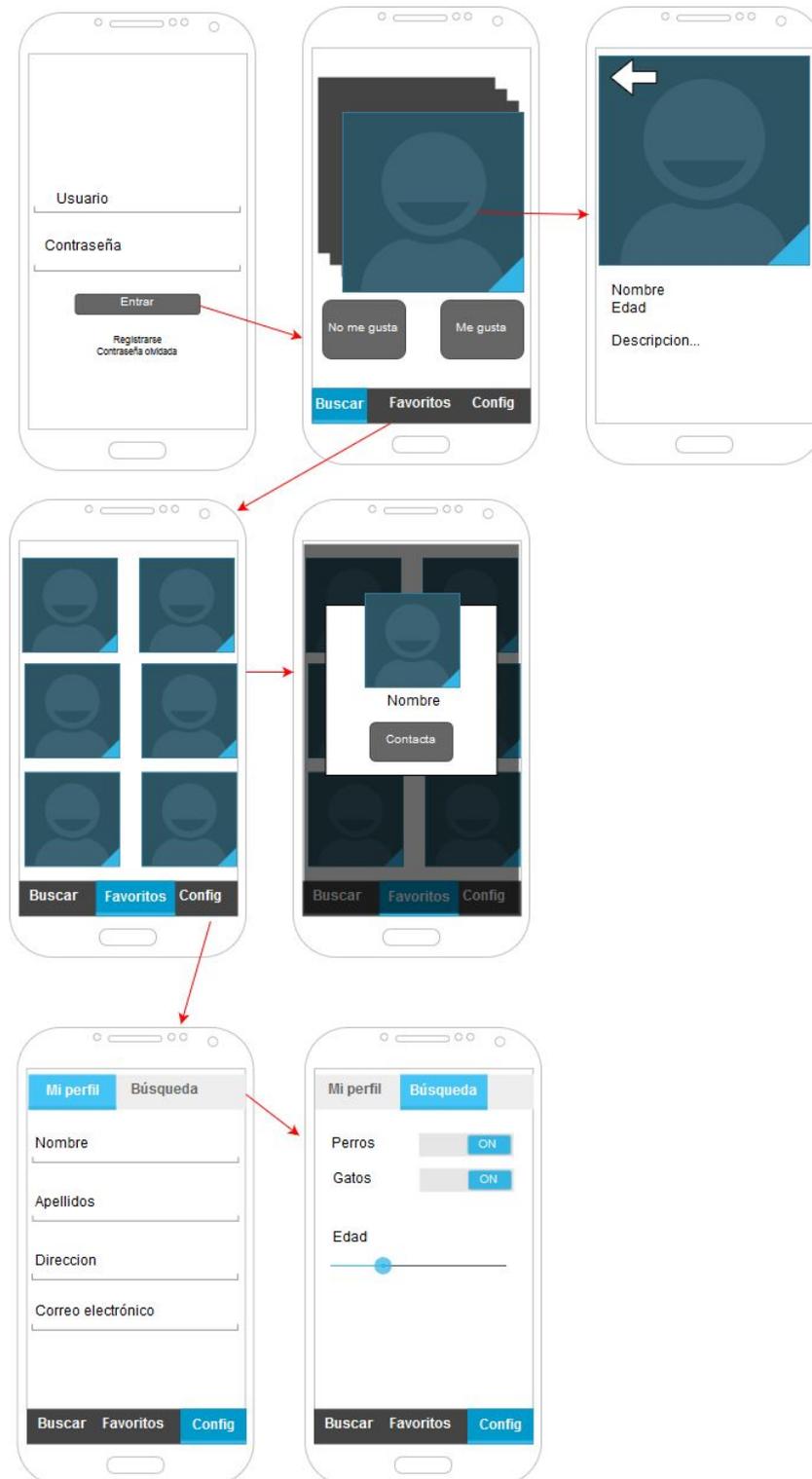


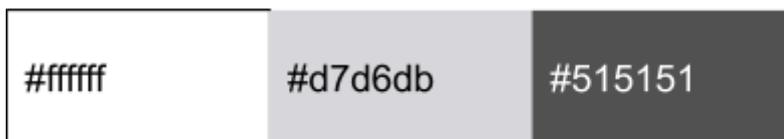
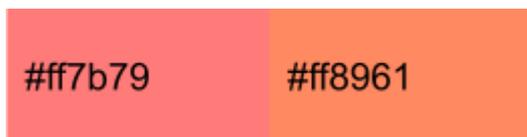
Fig 11. Prototipo baja fidelidad.

3.3.5 Elementos gráficos

Gama cromática

Se ha decidido el uso de una gama cromática cálida, con tonos cercanos al rojo o rosa y una estética general similar a las de las aplicaciones de citas como Tinder o Badoo destinadas a las relaciones personales.

Con esto se pretende dar al usuario una experiencia que recuerde más a la búsqueda de una relación personal con la mascota.



Tipografías

Las tipografías son gratuitas y disponibles en el repositorio de Google Fonts.

Para los títulos se ha elegido la fuente **Abril Fatface**, fuente display con serifa muy decorativa.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Para el texto corrido se ha utilizado **Actor**, una fuente de palo, limpia y de fácil lectura en pantalla.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Iconos y elementos gráficos

Para los iconos se ha utilizado el repositorio de iconos *Fontawesome*, el cual proporciona una gran cantidad de iconos de uso común.

Estos iconos se adjuntan como una tipografía más en el proyecto, por lo que pueden ser escalados y coloreados sin perder calidad y con un impacto mínimo en el peso de la aplicación.

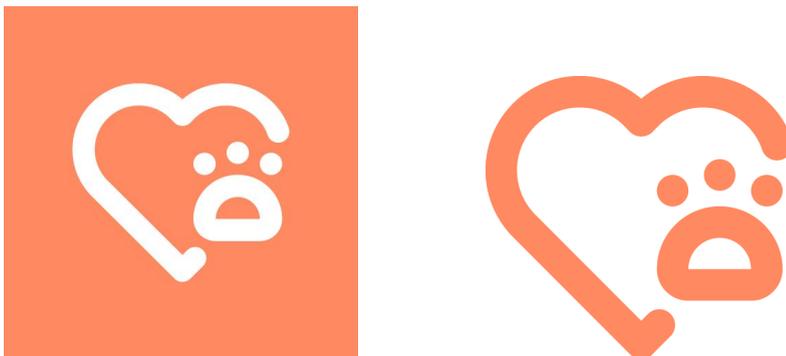
Por contra, nos limita al uso de los iconos disponibles en el repositorio sin poder personalizar el aspecto de los mismos más allá del color.

Logotipo

El logotipo está compuesto por un corazón y una huella de animal, representando la idea general de la aplicación de ofrecer una experiencia de relación con la mascota.

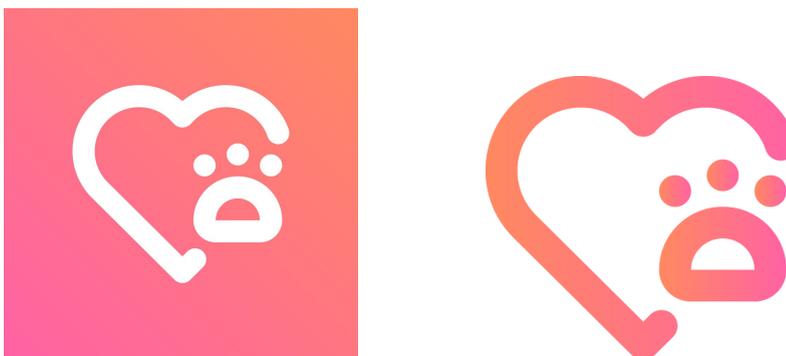
Se plantea el uso del logotipo en blanco, sobre fondo de color plano o en el color #ff8961 sobre fondo blanco.

Colores planos:

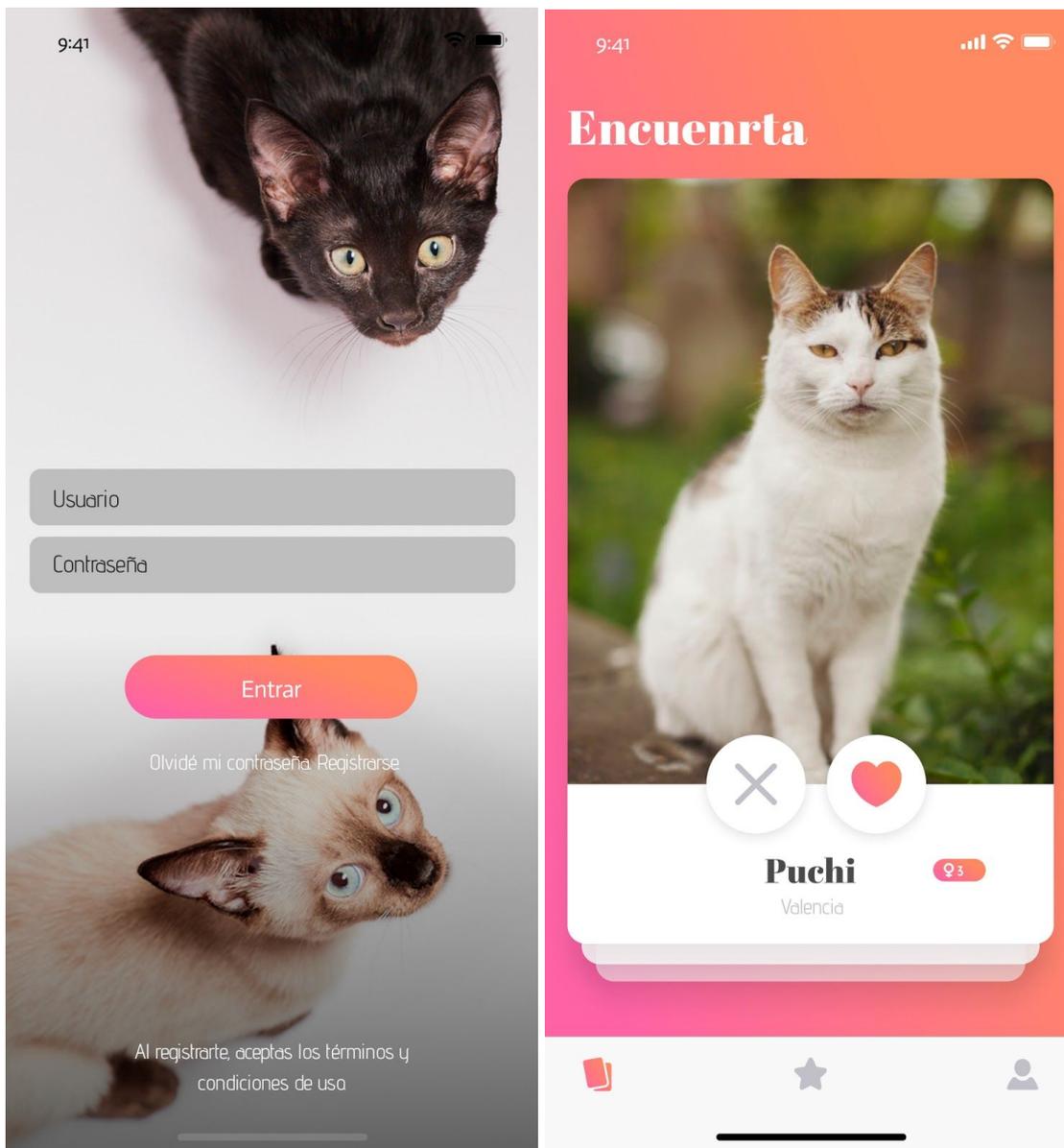


Como representación alternativa se plantea el uso del degradado de los dos colores principales, tanto como fondo como para relleno sobre fondo blanco.

Degradado:



3.3.6 Prototipo alta fidelidad





Puchi ♀ 3

Valencia

Puchi es cariñosa y tranquila. Se lleva bien con perros, pero no es apta para vivir con otros gatos.

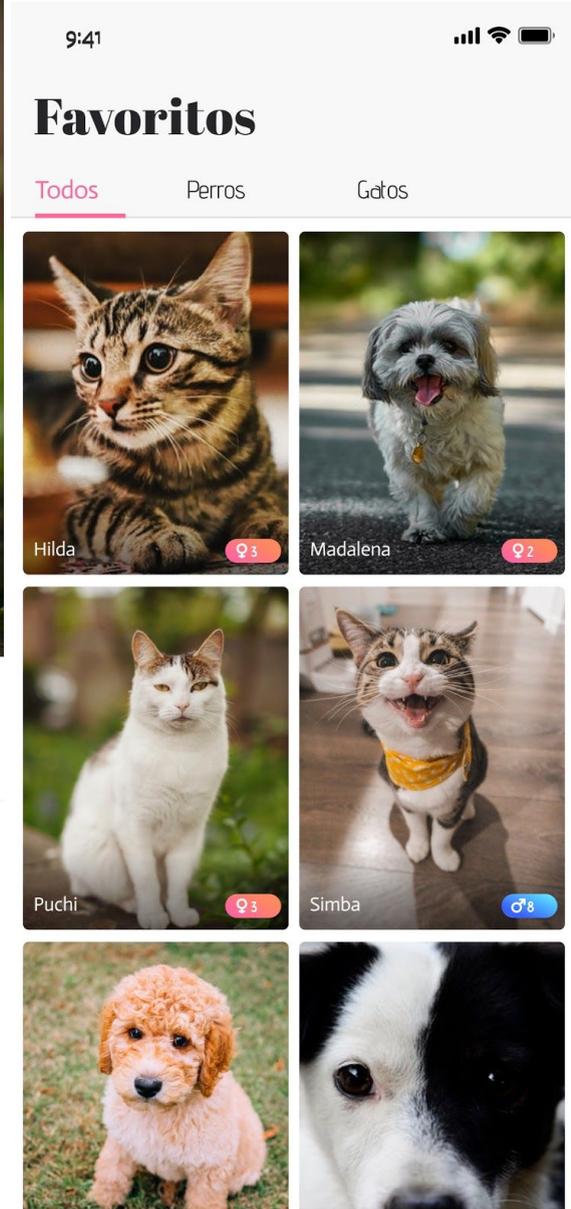
Datos

Raza: Común europeo

Carácter: Tranquilo

Test inmuno: Negativo

Leucemia: Negativo



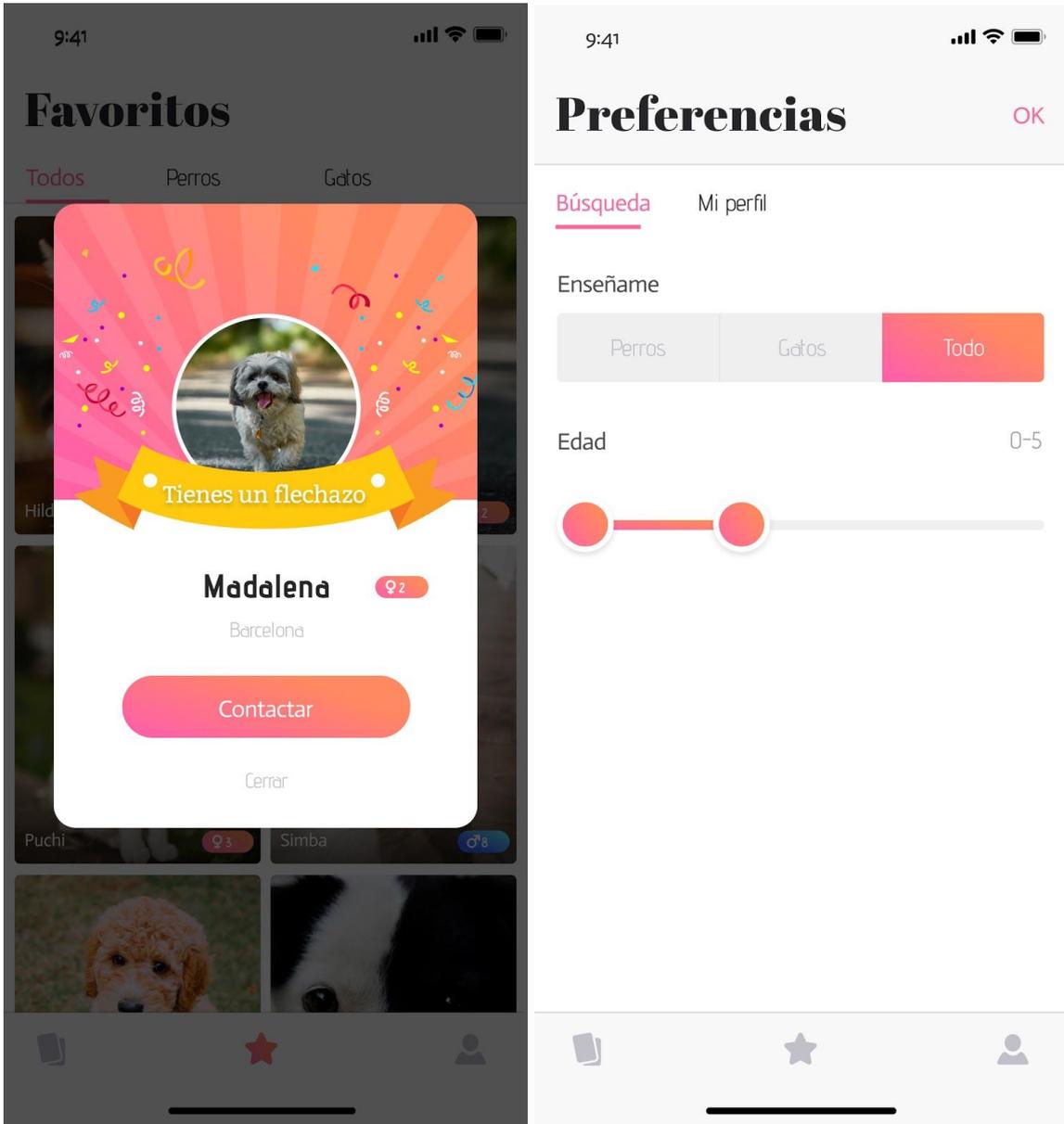


Fig 12. Prototipo alta fidelidad.

4. Desarrollo

4.1 Tecnologías y herramientas para el desarrollo

Con el fin de desarrollar una aplicación que pueda compilarse para todos los sistemas operativos móviles actuales (Android e iOS), se ha optado por utilizar técnicas de desarrollo híbridas.

Este paradigma de desarrollo se basa en la utilización de lenguajes web (HTML/CSS/JS) para el *frontend*.

El *backend* se desarrollará con lenguajes de servidor. En concreto una base de datos MySQL y un sistema de acceso basado en el estándar REST implementado en PHP.

Para compilar la aplicación desarrollada con lenguajes web se hará uso del *framework* de código abierto Cordova, desarrollado por Apache.

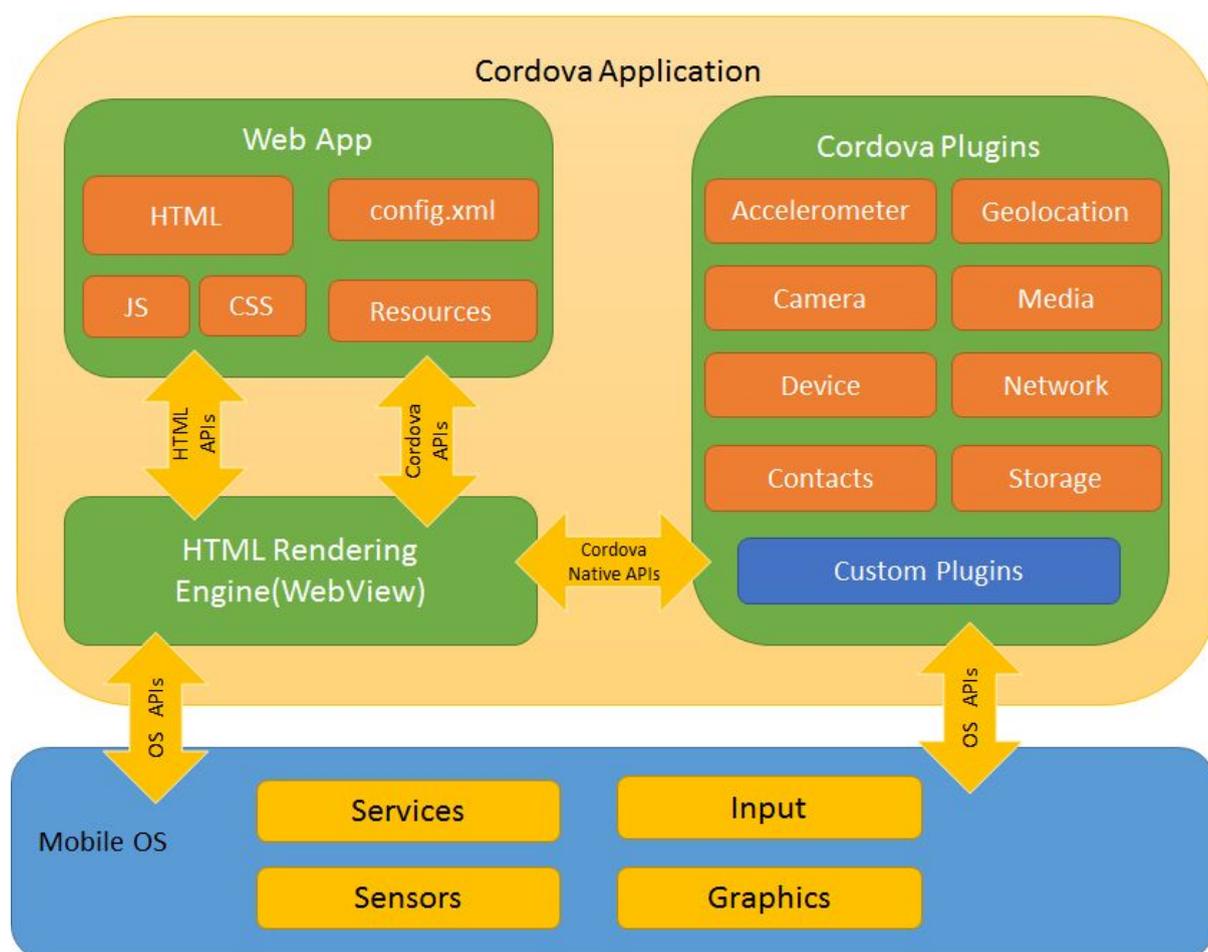
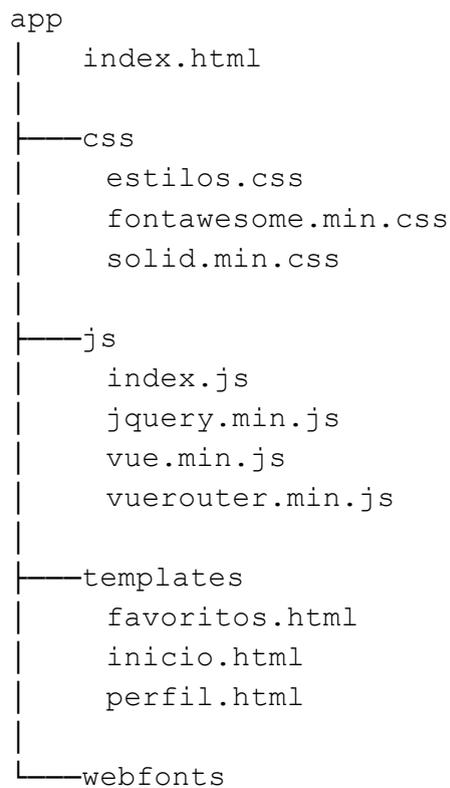


Fig 13. Arquitectura de aplicación Cordova.

4.1.1 Aplicación móvil



Librerías

Vue.js - Se ha optado por el uso de la librería Vue.js, frente a otras opciones similares como Angular, React, etc. por ser ligera y estar exclusivamente diseñada para gestionar el enlace de datos entre el DOM y el código Javascript, dejando plena libertad para maquetar el diseño y el comportamiento de la aplicación de forma “manual”.

El objeto principal creado por Vue se encarga de enlazar con el DOM y define las estructuras de datos y las funciones necesarias para gestionar los eventos del sistema.

```

var app = new Vue({
  el: '#app',
  router,
  data: {
    variable: 'valor',
  },
  methods: {
    funcion1:function(){
    }
  }
});

```

Vuerouter - La librería Vuerouter, complementaria de Vue.js, permite crear interfaces multipágina aprovechando el sistema de enlaces propio de HTML.

```

const Listado = { template: '<div>...</div>' }
const Favoritos = { template: '<div>...</div>' }
const Perfil = { template: '<div>...</div>' }

const routes = [
  { path: '/listado, component: Listado },
  { path: '/favoritos, component: Favoritos }
  { path: '/perfil, component: Perfil }
];

const router = new VueRouter({
  routes
});

```

Al incluir Vuerouter, podemos definir rutas URL y asignarlas a objetos Template de Vue. Así, al tocar sobre un enlace, podemos controlar qué elementos de la pantalla se actualizarán y cuales no, podemos recargar sólo una parte del documento y podemos añadir animaciones, haciendo la interacción más fluida y consistente con el comportamiento típico en las aplicaciones móviles.

jQuery - Aunque la sincronización entre interfaz y datos y el control de eventos puede implementarse completamente con Vue, se ha optado por añadir la librería jQuery, más potente a la hora de gestionar la interactividad con el usuario (animaciones, transiciones, efectos visuales, etc).

FontAwesome - Es un paquete de iconos distribuido como una tipografía web. Puede adjuntarse en páginas web y aplicaciones como una tipografía más y provee de una colección de iconos prediseñados, vectoriales y de poco peso.

Cordova - El proyecto original desarrollado por Apache que permite utilizar tecnologías web para desarrollar aplicaciones móviles. Desde su aparición han surgido varios proyectos derivados mantenidos por diferentes empresa, desde gratuitos hasta modelos de suscripción y con diferentes grados de modificaciones, añadidos y plugins. En este caso hemos optado por utilizar la librería original, gratuita y de código abierto para evitar la dependencia de empresas privadas en el desarrollo.

4.1.2 Servidor

Todas las instancias de la aplicación funcionan mediante una base de datos centralizada en un servidor MySQL + PHP.

La comunicación entre la aplicación y el servidor se realiza mediante una interfaz REST implementada con la ayuda de las librerías `php-crud-api` y `php-api-auth`.

php-crud-api - Librería PHP de código abierto que implementa un sistema básico de acceso mediante el estándar REST a una base de datos MySQL.

php-api-auth - Extensión para `php-crud-api` que permite añadir un sistema de autenticación para el uso del servicio REST.

4.1.3 Gestión

Trasladando la idea del presente trabajo a un proyecto real, sería deseable disponer de un sistema de gestión completo para facilitar las operaciones de alta, baja y modificación de datos por parte de las asociaciones.

Como prueba de concepto y para gestionar la parte de datos del servidor que debe atender a las solicitudes de la aplicación, se ha implementado una intranet básica utilizando el framework *Intraneteador*, desarrollado con anterioridad por el autor del presente trabajo.

Intraneteador - Se trata de un framework de propósito general para la gestión CRUD de bases de datos MySQL, creado con PHP.

El sistema se configura mediante un archivo de datos, también en PHP, en el que se especifica que tablas y campos de la base de datos deben editarse y con qué formato.

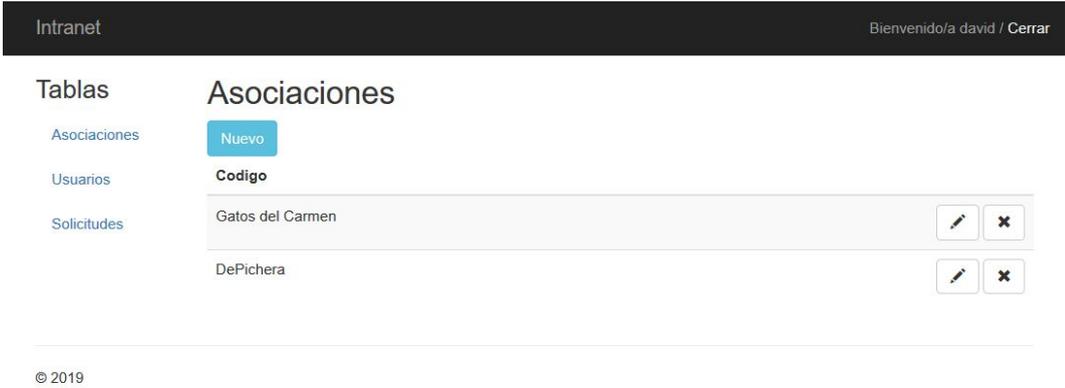


Fig 14.Intranet-Listado de asociaciones.

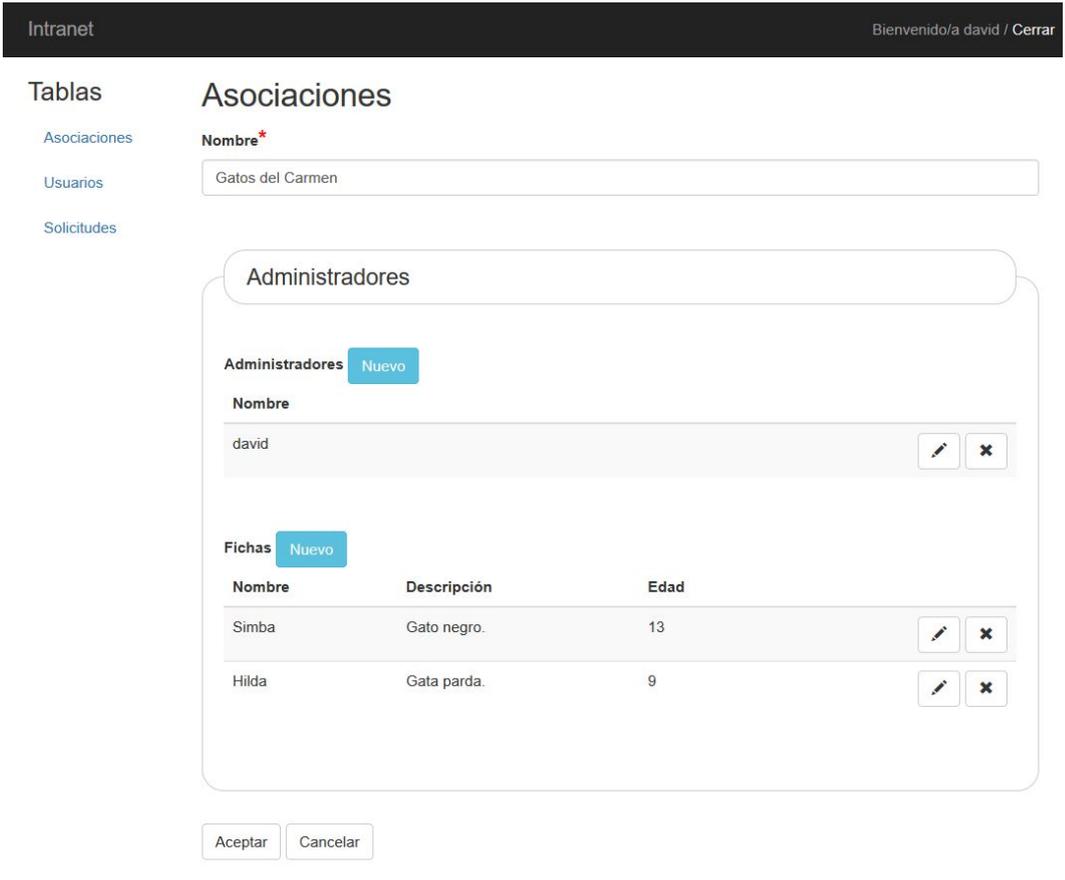


Fig 15.Intranet-Gestión de asociaciones.

Tablas

[Asociaciones](#)[Usuarios](#)[Solicitudes](#)

Fichas

Nombre**Foto** Eliminar actual

No se ha seleccionado ningún archivo.

Edad**Descripción****Fecha alta****Visualizaciones**

Fig 16.Intranet-Gestión de fichas

Configuración del proyecto `inc/estructura.php`:

```
$config['thumbnails']=array(
    'thumbnail'=>array(100,90,true),
    'medium'=>array(400,300,true),
    'big'=>array(1280,900,true),
);

$config['menu']=array(
    array('tipo'=>'tit', 'texto'=>'Tablas'),
    array('tipo'=>'tbl', 'tabla'=>'tbl_asociacion'),
    array('tipo'=>'tbl', 'tabla'=>'tbl_usuario'),
    array('tipo'=>'tbl', 'tabla'=>'tbl_solicitud')
);

$config['tablas']=array(
    'tbl_administrador'=>array(
        'tabla'=>'administrador',
        'indice'=>'id',
        'etiqueta'=>'Administradores',
        'menu'=>'normal',
        'columnas'=>array('nombre'=>'Nombre'),
        'campos'=>array(

'id_asociacion'=>array('etiqueta'=>'Asociacion', 'tipo'=>'relacion', 'rel_id'=>'asociacion.id', 'rel_nombre'=>'nombre', 'oculto'=>true),
        'nombre'=>array('etiqueta'=>'Nombre', 'tipo'=>'text'),
        'email'=>array('etiqueta'=>'Email', 'tipo'=>'text'),
        'password'=>array('etiqueta'=>'Password', 'tipo'=>'text'),
        )
    ),

    'tbl_ficha'=>array(
        'tabla'=>'ficha',
        'indice'=>'id',
        'etiqueta'=>'Fichas',
        'menu'=>'normal',

'columnas'=>array('nombre'=>'Nombre', 'descripcion'=>'Descripción', 'edad'=>'Edad'),
        'campos'=>array(

'id_asociacion'=>array('etiqueta'=>'Asociacion', 'tipo'=>'relacion', 'rel_id'=>'asociacion.id', 'rel_nombre'=>'nombre', 'oculto'=>true),
        'nombre'=>array('etiqueta'=>'Nombre', 'tipo'=>'text'),
        'foto'=>array('etiqueta'=>'Foto', 'tipo'=>'imagen'),
        'edad'=>array('etiqueta'=>'Edad', 'tipo'=>'text'),
        'descripcion'=>array('etiqueta'=>'Descripción', 'tipo'=>'textarea'),
        'fecha_alta'=>array('etiqueta'=>'Fecha alta', 'tipo'=>'fecha'),

'visualizaciones'=>array('etiqueta'=>'Visualizaciones', 'tipo'=>'text'),
        )
    ),

    'tbl_asociacion'=>array(
```

```

        'tabla'=>'asociacion',
        'indice'=>'id',
        'etiqueta'=>'Asociaciones',
        'menu'=>'normal',
        'columnas'=>array('nombre'=>'Codigo'),
        'campos'=>array(
'nombre'=>array('etiqueta'=>'Nombre','tipo'=>'text','obligatorio'=>true),
        '-1'=>array('etiqueta'=>'Administradores'),

'administrador'=>array('etiqueta'=>'Administradores','tipo'=>'relacion_inversa','rel_id'=>'administrador.id_asociacion','rel_this'=>'id'),

'ficha'=>array('etiqueta'=>'Fichas','tipo'=>'relacion_inversa','rel_id'=>'ficha.id_asociacion','rel_this'=>'id'),
        )
    ),
    'tbl_usuario'=>array(
        'tabla'=>'usuario',
        'indice'=>'id',
        'etiqueta'=>'Usuarios',
        'menu'=>'normal',
        'columnas'=>array('nombre'=>'Nombre'),
        'campos'=>array(
            'nombre'=>array('etiqueta'=>'Nombre','tipo'=>'text'),
            'apellidos'=>array('etiqueta'=>'Apellidos','tipo'=>'text'),
            'email'=>array('etiqueta'=>'Email','tipo'=>'text'),
            'telefono'=>array('etiqueta'=>'Teléfono','tipo'=>'text'),
            'direccion'=>array('etiqueta'=>'Dirección','tipo'=>'text'),
            'password'=>array('etiqueta'=>'Password','tipo'=>'text'),
        )
    ),
    'tbl_solicitud'=>array(
        'tabla'=>'solicitud',
        'indice'=>'id',
        'etiqueta'=>'Solicitudes',
        'menu'=>'normal',
        'columnas'=>array('>id_ficha'=>'Nombre Ficha','>id_usuario'=>'Nombre
Usuario'),
        'campos'=>array(

'id_ficha'=>array('etiqueta'=>'Ficha','tipo'=>'relacion','rel_id'=>'ficha.id','rel_nombre'=>'nombre'),

'id_usuario'=>array('etiqueta'=>'Usuario','tipo'=>'relacion','rel_id'=>'usuario.id','rel_nombre'=>'nombre'),
            'aprobada'=>array('etiqueta'=>'Aprobada','tipo'=>'check')
        )
    ),
);

```

4.1.4 API REST

La aplicación se nutre de los datos proporcionados por el servidor mediante el estándar REST. Para ello se ha utilizado la librería de código abierto **php-crud-api**, la cual acepta peticiones HTML y devuelve los datos en formato JSON para ser procesados y visualizados por la aplicación.

El servidor admite operaciones GET, POST, PUT y DELETE

GET: Devuelve uno o varios registros de la base de datos.

POST: Inserta un nuevo registro en la base de datos. Los datos del registro deben ir adjuntos a la petición mediante el bloque POST.

PUT: Modifica un registro existente en la base de datos. Los datos del registro deben ir adjuntos en la petición mediante el bloque POST.

DELETE: Elimina un registro de la base de datos.

Nota: En el estado actual del sistema, se requiere autenticación para acceder a la API, pero, una vez identificado, la aplicación tiene capacidades de acceso total a la base de datos. Como requisito para trasladar el presente trabajo a un proyecto real, sería indispensable controlar qué acciones puede realizar la aplicación sobre la base de datos y cuáles no.

Formato de las peticiones:

```
GET http://localhost/tindog/tindogapi.php/records/ficha
```

Respuesta del servidor:

```
{
  "records": [
    {
      "id":1,
      "id_asociacion":1,
      "tipo":0,
      "nombre":"Simba",
      "foto":"1558031316.jpg",
      "edad":13,
      "descripcion":"Gato negro.",
      "fecha_alta":"2019-05-12",
      "visualizaciones":1
    },
    {
      "id":2,
      "id_asociacion":1,
      "tipo":0,
      "nombre":"Hilda",
      "foto":"1558031326.jpg",
```

```
"edad":9,
"descripcion":"Gata parda.",
"fecha_alta":"2019-05-01",
"visualizaciones":0
},
{
"id":3,
"id_asociacion":2,
"tipo":0,
"nombre":"Firulais",
"foto":"1558031345.jpg",
"edad":23,
"descripcion":"Perro grande.",
"fecha_alta":"2019-04-25",
"visualizaciones":1
}
]
}
```

5 Conclusiones

La aplicación resultante funciona correctamente en un entorno de pruebas, con una base de datos reducida.

Las pruebas realizadas con usuarios indican que es fácil de utilizar y familiar en su comportamiento.

El aspecto visual y la usabilidad animan al usuario a establecer una relación de vínculo más personal con los animales que con otras aplicaciones similares.

En el apartado técnico, el uso de Cordova + Vue.js y REST han permitido el despliegue de la aplicación en los sistemas operativos móviles dominantes (Android e iOS) y un desarrollo ágil.

Con respecto a los objetivos secundarios deseables según el punto 1.2.1, se han utilizado únicamente herramientas y librerías de código abierto. La única excepción es la compilación para iOS, que debe realizarse con Xcode, que no es de código abierto, pero sí gratuito. Por este motivo, el producto resultante puede ser distribuido con este modelo de código abierto.

6 Posibles ampliaciones

Aunque la aplicación puede funcionar como mínimo producto viable en el estado descrito en el presente trabajo, se plantean algunas ampliaciones que podrían aplicarse de forma más inmediata:

- Más pasos para finalizar adopción: Aunque las asociaciones consultadas se muestran reticentes a formalizar las adopciones de forma totalmente digital, sí que podrían añadirse en un futuro algunos pasos más en el proceso, como el primero contacto por mensaje de texto, la presentación del contrato de adopción o el envío de documentación.
- Seguimiento adopciones: Una de las prácticas seguidas por la mayoría de protectoras y asociaciones es el seguimiento del bienestar del animal durante los primeros meses o años de la adopción. Esto podría plantearse como una ampliación de la aplicación permitiendo el envío de fotografías o informes veterinarios.
- Búsqueda geográfica: En la versión actual la búsqueda se realiza el nombre de la población en la que se encuentra la protectora. Sería más adecuado incluir coordenadas y geolocalización para buscar por proximidad.
- Seguridad: Aunque la librería REST utilizada incorpora un módulo de autenticación que se presume seguro, no se ha prestado especial atención a la seguridad en el desarrollo del presente prototipo. Sería necesaria una auditoría completa si la aplicación se convirtiese en un producto real.

7 Bibliografía

Affinity, 2018 - (2018, junio 26). Resultados del Estudio del abandono en España 2018 | Fundación Se recuperó el mayo 15, 2019 de <https://www.fundacion-affinity.org/blog/resultados-del-estudio-del-abandono-en-espana-2018>

Mobile, 2018 - (2018, julio 11). Informe Mobile en España y en Mundo 2018 - DITRENDIA. Se recuperó el abril 31, 2019 de <https://ditrendia.es/informe-mobile-2018/>

Freeman, A. (2018). Pro Vue.js 2. New York: Apress.

Halliday, P. (2018). Vue.js 2 design patterns and best practices : build enterprise-ready, modular Vue.js applications with Vuex and Nuxt. Birmingham, UK: Packt Publishing.

Krug, S. (2015). No me hagas pensar, actualización : una aproximación a la usabilidad en la Web y los móviles. Madrid: Anaya Multimedia.

Richardson, L. & Amundsen, M. (2013). RESTful Web APIs. Sebastopol, Calif: O'Reilly.

Anexo I

Entorno software empleado en el desarrollo.

Servidor web:

S.O: Linux Raspbian

Apache 2.4.10 + PHP 7.2.11

MySQL 5.0.12-dev + PhpMyAdmin 2.7

Desarrollo:

Windows 10 ver 10.0.17

Editor de código: Atom 1.35

Android Studio 2.7.0

MacOS Mojave 10.14.5

Xcode 4

Entorno hardware empleado en el desarrollo.

Servidor web:

Raspberry Pi 3 (ARM Cortex-A53, 1.2GHz, 1Gb RAM)

Desarrollo:

Equipo principal:

Intel Core i5, 8Gb RAM

Equipo para compilar iOS:

Macbook Pro 13" (Core i7, 8Gb RAM)

Terminales de prueba:

Xiaomi MI A2 - Android 9.0

iPhone 5S - iOS 12