

APLICACIÓN MULTIPLATAFORMA PARA FACILITAR LA ADOPCIÓN DE ANIMALES

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Memoria del Proyecto



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Enero de 2019

Autor

Alejandro Barajas González

Tutores

Gabriel Villarrubia González

Juan Francisco De Paz Santana

D. Juan Francisco De Paz Santana, D. Gabriel Villarrubia González, profesores/as del Departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo titulado “Aplicación multiplataforma para facilitar la adopción de animales” ha sido realizado por D. Alejandro Barajas González, con DNI 70897600-P y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de esta Universidad. Y para que así conste a todos los efectos oportunos.

En Salamanca, a de Enero de 2019

D. Juan Francisco De Paz Santana

D. Gabriel Villarrubia González

Resumen

El proyecto realizado como Trabajo Fin de Grado consiste en una arquitectura cliente-servidor en el que el cliente es una aplicación móvil multiplataforma. La aplicación está pensada para que las protectoras publiquen sus animales y los adoptantes puedan encontrar un animal lo más adecuado para ellos.

En la aplicación aparecen dos tipos de usuarios diferentes, las protectoras que exponen sus animales y los adoptantes que buscan un animal. Los adoptantes pueden ver los animales que publican las protectoras, pudiéndolos filtrar por las características que más les interesen. Las protectoras podrán hacer publicaciones de animales, editarlas y borrarlas. También, podrán ver todos detalles de los animales que han conseguido dar en adopción. Cada protectora tendrá un perfil y unos datos que podrán ser modificados en cualquier momento desde su menú principal. Las protectoras podrán incorporar colaboradores a través de un código que se envía por e-mail o se puede ver desde el propio menú de la aplicación.

Se ha utilizado un Framework centrado en móviles para darle estilo a la aplicación y que la experiencia de usuario sea lo más agradable posible.

En cuanto al servidor, se ha desarrollado un servicio web RESTful para que la comunicación entre cliente y servidor sea fluida. En dicho servidor, se han incluido algunas bibliotecas para poder realizar funcionalidades algo más complejas.

Para conseguir un proyecto escalable, robusto y mantenible se han utilizado todas las prácticas conocidas de Ingeniería del Software.

Palabras clave: Adopción, animales, protectora, multiplataforma, aplicación, RESTful.

Abstract

The project done as an end-of-degree work consists of a client and server architecture in which the client is a multiplatform mobile application. The application is designed to make it easier to the protectors to publish their animals, and at the same time the adopters can find the better option for them.

Two types of adopters can appear in the application, the protectors that expose their animals and the adopters that are looking for an adoption. Adopters can see the animals that the different protectors have in each moment and they can filter between the characteristics they prefer. The protectors can make publications of their animals, edit them and delete them. They can also see all the animals that have been adopted. Each protector will have a profile and data that can be modified at any time from the main menu. The protectors will be able to incorporate collaborators through a code that is sent by e-mail or can be viewed from the application's menu.

A Framework focused on mobile phones has been used to style the application and make the user experience as enjoyable as possible.

As for the server, a RESTful Web service has been developed to achieve a fluent communication between client and server. In this server, some libraries have been included to be able to make some more complex functionalities.

To achieve a scalable, robust and maintainable project, all known Software engineering practices have been used.

Keywords: adoption, animals, protectors, multiplatform, application, RESTful.

Tabla de contenido

| | | |
|-------|---|----|
| 1. | Introducción y antecedentes..... | 15 |
| 1.1 | Introducción | 15 |
| 1.2 | Estudio de mercado | 16 |
| 1.3 | Conclusiones del estudio..... | 17 |
| 1.4 | Estructura del documento..... | 17 |
| 2. | Objetivos | 19 |
| 2.1 | Objetivos del sistema..... | 19 |
| 2.2 | Objetivos personales..... | 20 |
| 3. | Conceptos teóricos | 21 |
| 3.1 | Arquitectura cliente-servidor | 21 |
| 3.2 | Apache Cordova..... | 22 |
| 3.3 | Servicio REST..... | 23 |
| 3.4 | JSON | 24 |
| 4. | Técnicas | 25 |
| 4.1 | Modelo Vista-Controlador (MVC)..... | 25 |
| 4.2 | Metodología de Durán y Bernárdez | 26 |
| 5. | Herramientas | 27 |
| 5.1 | Herramientas utilizadas en el desarrollo de la aplicación | 27 |
| 5.1.1 | Visual Studio | 27 |
| 5.1.2 | Framework7 | 27 |
| 5.1.3 | GitHub | 28 |
| 5.1.4 | Pixlr | 28 |
| 5.1.5 | Chart..... | 28 |
| 5.1.6 | Adobe Color Wheel..... | 29 |
| 5.1.7 | Lenguajes de programación | 29 |
| 5.2 | Herramientas utilizadas en el desarrollo del servidor | 31 |
| 5.2.1 | SLIM 3..... | 31 |
| 5.2.2 | Sublime Text 3 | 31 |
| 5.2.3 | PostMan | 31 |
| 5.2.4 | XAMPP..... | 32 |
| 5.2.5 | Lenguajes de programación | 32 |

| | | |
|-------|---|----|
| 5.3 | Herramientas CASE | 32 |
| 5.3.1 | Visual Paradigm | 32 |
| 5.3.2 | REM | 33 |
| 5.3.3 | EZEstimate | 33 |
| 5.3.4 | Microsoft Project | 33 |
| 5.3.5 | Microsoft Word | 33 |
| 6. | Proceso de desarrollo | 35 |
| 6.1 | Marco de trabajo | 35 |
| 6.2 | Estimación de Coste y Esfuerzo | 36 |
| 6.3 | Planificación temporal | 41 |
| 6.4 | Especificación de requisitos | 42 |
| 6.4.1 | Objetivos del sistema | 43 |
| 6.4.2 | Requisitos de información | 44 |
| 6.4.3 | Requisitos no funcionales | 45 |
| 6.4.4 | Requisitos funcionales | 46 |
| 6.5 | Análisis del sistema | 48 |
| 6.5.1 | Modelo de dominio | 48 |
| 6.5.2 | Vista arquitectónica del modelo de análisis | 49 |
| 6.5.3 | Vista arquitectónica del modelo de análisis | 49 |
| 6.6 | Diseño del sistema | 50 |
| 6.6.1 | Patrones arquitectónicos | 50 |
| 6.6.2 | Subsistemas de diseño | 51 |
| 6.6.3 | Clases de diseño | 53 |
| 6.6.4 | Vista arquitectónica del modelo de diseño | 54 |
| 6.6.5 | Realización de Casos de Uso en el Diseño | 55 |
| 6.6.6 | Modelo de despliegue | 56 |
| 6.7 | Implementación del sistema | 57 |
| 6.8 | Pruebas..... | 57 |
| 6.9 | Funcionalidad de la aplicación | 58 |
| 6.9.1 | Manual para adoptantes..... | 59 |
| 6.9.2 | Manual para protectoras | 65 |
| 6.9.3 | Cerrar sesión..... | 72 |
| 6.9.4 | Recuperar contraseña..... | 72 |

| | | |
|-----|--|----|
| 7. | Conclusiones y Líneas de trabajo futuras | 73 |
| 7.1 | Conclusiones..... | 73 |
| 7.2 | Líneas de trabajo futuras | 73 |
| 8. | Bibliografía..... | 75 |

Índice de figuras

| | |
|--|----|
| Ilustración 1 - Datos Affinity | 15 |
| Ilustración 2 - Miwuki | 16 |
| Ilustración 3 - Chuby | 16 |
| Ilustración 4 - Allnimal | 16 |
| Ilustración 5 - Arquitectura cliente-servidor | 21 |
| Ilustración 6 - Complementos Apache Cordova | 22 |
| Ilustración 7 - Representación arquitectura REST | 23 |
| Ilustración 8 - Modelo Vista-Controlador | 25 |
| Ilustración 9 - Visual Studio + Apache Cordova | 27 |
| Ilustración 10 - Framework7 | 27 |
| Ilustración 11 - GitHub | 28 |
| Ilustración 12 - Pixlr | 28 |
| Ilustración 13 - Chart | 28 |
| Ilustración 14 - Adobe Color Wheel | 29 |
| Ilustración 15 - HTML5 | 29 |
| Ilustración 16 - CSS3 | 29 |
| Ilustración 17 - JavaScript | 30 |
| Ilustración 18 - JQuery | 30 |
| Ilustración 19 - Slim 3 | 31 |
| Ilustración 20 - Sublime Text 3 | 31 |
| Ilustración 21 - Postman | 31 |
| Ilustración 22 - Xampp | 32 |
| Ilustración 23 - Visual Paradigm | 32 |
| Ilustración 24 - REM | 33 |
| Ilustración 25 - EZEstimate | 33 |
| Ilustración 26 - Microsoft Project | 33 |
| Ilustración 27 - Microsoft Word | 33 |
| Ilustración 28 - Proceso Unificado | 36 |
| Ilustración 29 - Resultados de la Estimación de Esfuerzo | 41 |
| Ilustración 30 - Resumen de la planificación temporal | 41 |
| Ilustración 31 - Planificación de tareas | 42 |
| Ilustración 32 - Modelo de dominio | 48 |
| Ilustración 33 - Vista arquitectónica del modelo de análisis | 49 |
| Ilustración 34 - Diagrama de Secuencia "Registrar protectora" | 50 |
| Ilustración 35 - Patrón arquitectónico "MVC" | 51 |
| Ilustración 36 - Subsistemas de diseño | 52 |
| Ilustración 37 - Subsistema "Vistas" | 53 |
| Ilustración 38 - Diagrama de Clases del modelo de Diseño | 53 |
| Ilustración 39 - Vista de la arquitectura | 54 |
| Ilustración 40 - Diagrama de secuencia "Contactar con la protectora" | 55 |
| Ilustración 41 - Diagrama de despliegue | 56 |

| | |
|--|----|
| Ilustración 42 - Pantalla inicial..... | 58 |
| Ilustración 43 - Registro adoptantes..... | 59 |
| Ilustración 44 - Búsqueda de animales | 60 |
| Ilustración 45 - Detalles del animal..... | 60 |
| Ilustración 46 - Contactar | 61 |
| Ilustración 47 - Ver foto | 61 |
| Ilustración 48 - Detalles de la protectora..... | 62 |
| Ilustración 49 - Filtrar | 62 |
| Ilustración 50 - Modificar contraseña | 63 |
| Ilustración 51 - Email "Código de la protectora" | 64 |
| Ilustración 52 - Unirse a una protectora | 64 |
| Ilustración 53 - Registro de protectoras | 65 |
| Ilustración 54 - Añadir animal | 66 |
| Ilustración 55 - Marcar como adoptado | 67 |
| Ilustración 56 - Eliminar | 68 |
| Ilustración 57 - Código de la protectora | 69 |
| Ilustración 58 - Enviar código por email..... | 69 |
| Ilustración 59 - Modificar datos de la protectora..... | 70 |
| Ilustración 60 - Animales adoptados | 70 |
| Ilustración 61 - Datos de la protectora | 71 |
| Ilustración 62 - Cambiar contraseña de la protectora | 71 |
| Ilustración 63 - Recuperar contraseña..... | 72 |

Índice de tablas

| | |
|--|----|
| Tabla 1 – Clasificación de los tipos de actores | 38 |
| Tabla 2 – Clasificación de los tipos de Casos de Uso | 38 |
| Tabla 3 - Factores de complejidad técnica | 40 |
| Tabla 4 - Factores de complejidad de entorno..... | 40 |
| Tabla 5 - Objetivo "Gestión de usuarios" | 44 |
| Tabla 6 - Requisito de Información "Información de protectora" | 45 |
| Tabla 7 - Requisito no funcional "Escalabilidad" | 46 |
| Tabla 8 - Caso de Uso "Registrar Protectora" | 48 |

1. Introducción y antecedentes

1.1 Introducción

Los animales de compañía cada vez forman más parte de los hogares españoles. La decisión de tener un animal en casa debe meditarse concienzudamente. Hay que tener muy en cuenta que un perro o gato puede vivir entre 12 y 15 años en los que precisa cuidados y educación.

Muchas personas no tienen esto en cuenta a la hora de decidir tener un animal en casa, lo que acaba provocando el abandono de los animales.

Según el último estudio de la fundación Affinity, el año pasado fueron recogidos por las protectoras 138.000 perros y gatos en nuestro país. En la siguiente imagen se puede observar un gráfico donde se representa el destino de estos animales:

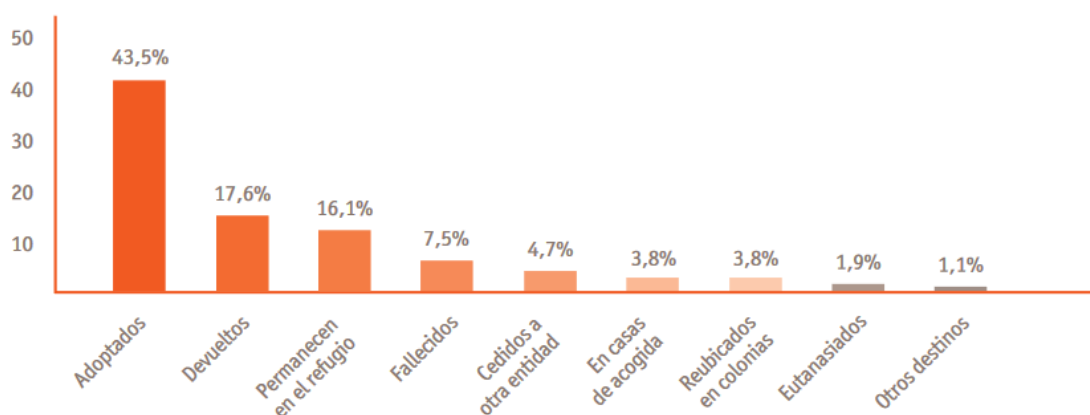


Ilustración 1 - Datos Affinity

Para dar a conocer a todos estos animales las protectoras suelen utilizar las redes sociales, lo que resulta una tarea muy laboriosa. El objetivo principal de este proyecto es la creación de una aplicación que facilite a las protectoras encontrar un hogar para sus animales lo antes posible. Por otro lado, también se ha pensado en los adoptantes, los cuales podrán encontrar un animal que se adapte a sus preferencias fácilmente.

Como objetivo secundario del proyecto surge el estudio y análisis de los datos que se generan en el uso de la aplicación. Estos datos permitirán hacer un estudio más exacto de las tasas de abandono, así como que las protectoras conozcan los resultados de su trabajo en forma de datos.

Antes de comenzar con el proyecto se ha hecho un examen del mercado para ver si existen aplicaciones que participen en el mismo sector y cómo lo hacen.

1.2 Estudio de mercado

Miwuki



Aplicación aparentemente muy completa que permite gestionar la adopción de animales. Tiene buena puntuación en Google Play Store y buenas críticas, sin embargo, hace mucho tiempo que no se actualiza y presenta algunos fallos. Actualmente se encuentra deshabilitada.

Ilustración 2 - Miwuki

Chuby

Aplicación que permite buscar adopciones, poner en adopción animales y colgar anuncios para buscar animales que se han perdido. La aplicación es poco usable a pesar de tener buenas críticas en Play Store. Algunas de sus funcionalidades aparecen descuadradas por la pantalla y los filtros de búsqueda aparentemente no funcionan muy bien.



Ilustración 3 - Chuby

Allnimal



Aplicación móvil con la que se podrán adoptar perros y gatos de Barcelona. También presenta una guía “petfriendly” en la que muestra lugares en los que poder disfrutar con las mascotas. La aplicación en apariencia es demasiado simple, además de ser poco usable. Contiene muchos anuncios que hacen distraer al usuario, lo cual da pie a no querer usar la aplicación.

Ilustración 4 - Allnimal

1.3 Conclusiones del estudio

Una vez analizadas las aplicaciones anteriores y alguna otra, he comprobado que algunas presentan ideas similares a las del proyecto, sin embargo, la mayoría no acaban de cumplir los objetivos para los que está enfocado.

Por tanto, una aplicación futura bien desarrollada del proyecto podría facilitar el trabajo de las protectoras y posiblemente influir en el número de adopciones.

1.4 Estructura del documento

La estructura del proyecto está organizada de la siguiente manera:

1. **Introducción y antecedentes.** Sección actual, destinada a explicar en qué consistirá el trabajo y como se organizará.
2. **Objetivos del proyecto.** Metas que se quieren cumplir con este trabajo, tanto personales como técnicas.
3. **Conceptos teóricos.** En este apartado se explican los conceptos esenciales para comprender el proyecto.
4. **Proceso de desarrollo.** Aquí se mostrará la forma de trabajar que se ha llevado y como se ha ido desarrollando el proyecto paso a paso.
5. **Técnicas y herramientas.** Esta sección incluirá las herramientas y técnicas que se han utilizado para el desarrollo del proyecto, tanto en la parte práctica como la parte documental.
6. **Conclusiones y líneas de trabajo futuras.** Incluye reflexiones personales sobre el proyecto (objetivos conseguidos, problemas encontrados, etc) y posibles ampliaciones y caminos viables de desarrollo de la aplicación en un futuro.
7. **Bibliografía**

Además de esta memoria se incluyen una serie de anexos que componen la documentación técnica. Estos son los anexos:

- **Anexo I. Plan del Proyecto Software**. Contiene la planificación temporal con una distribución de tareas del proyecto de acuerdo con la metodología de trabajo escogida y una estimación aproximada del coste del proyecto en función de los Casos de Uso.
- **Anexo II. Especificación de requisitos del software**.
- **Anexo III. Análisis del sistema**
- **Anexo IV. Diseño del sistema**. Arquitectura, clases de diseño, etc.
- **Anexo V: Documentación técnica**. En este anexo se incluye la forma en que ha sido programada la aplicación con el fin de entender el proceso para posibles ampliaciones futuras.
- **Anexo VI: Manual de usuario**

2. Objetivos

En este apartado se recogen todos los objetivos con los que se ha desarrollado el proyecto. Se desarrollarán tanto los objetivos del sistema software como los objetivos personales a superar.

2.1 Objetivos del sistema

- **Usabilidad:** El principal objetivo es que la aplicación es que sea fácil de utilizar para los usuarios y no sea descartada desde un primer momento. Para ello se presentan las funcionales de manera sencilla y rápida para que sean atractivas para los usuarios.
- **Visibilidad:** La aplicación se desarrollará en formato multiplataforma para llegar al mayor número de usuarios posibles.
- **Seguridad:** La seguridad es clave para que los usuarios tengan confianza al utilizar la aplicación. Este objetivo se aplica principalmente sobre los datos de los usuarios, como por ejemplo sus contraseñas.
- **Velocidad:** Muchos usuarios se cansan de las aplicaciones que son lentas en cargar el contenido. Para ello se ha desarrollado un servidor tipo RESTful para que la respuesta hacia el cliente sea bastante rápida.
- **Escalabilidad:** Se ha diseñado la aplicación con la posibilidad de que pueda crecer. Cuando lleve tiempo utilizándose pueden aparecer nuevas funcionalidades que sugieran los usuarios, por tanto, esta cualidad es muy importante tenerla en cuenta.
- **Comunicación:** Uno de los principales objetivos del proyecto es poder facilitar la comunicación entre las protectoras y los adoptantes.

2.2 Objetivos personales

Estos son los objetivos personales por los que me he decidido a realizar este proyecto:

- **Finalidad de la aplicación:** Me gustan mucho los animales y hacer algo relacionado con ellos me llamaba la atención. Que el objetivo pudiera ayudar a reducir la tasa de abandono me hizo decidirme por este proyecto.
- **Desarrollar una aplicación móvil:** Durante la carrera hemos visto diferentes lenguajes de programación, pero todo lo que hemos hecho estaba centrado en PCs. La tecnología móvil está creciendo muy rápido y aprender sobre esto puede ser una buena salida profesional para el futuro.
- **Aprender:** Nuevos lenguajes de programación, nuevos entornos de desarrollo, nuevas herramientas, nuevos servicios, etc

3. Conceptos teóricos

En este apartado se presentan los conceptos teóricos necesarios para comprender fácilmente el proyecto.

3.1 Arquitectura cliente-servidor

Es la arquitectura que se utiliza en el proyecto. Se trata de un modelo de diseño software en el que las tareas se reparten entre los proveedores o servidores y los clientes.

Por tanto, el cliente es quien necesita servicios y los servidores son los que los proporcionan. El cliente, en nuestro caso la aplicación móvil, realiza peticiones al servidor de quién espera una respuesta. El servidor recibe la petición, la procesa y devuelve una respuesta.

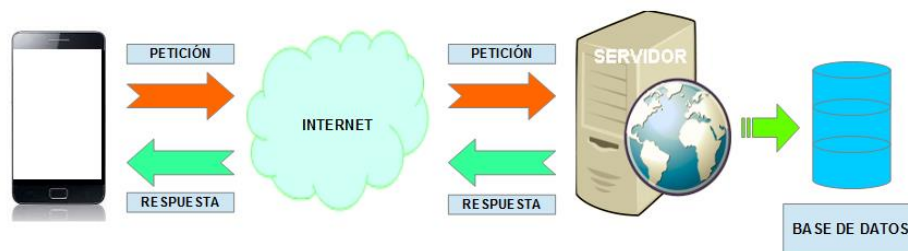


Ilustración 5 - Arquitectura cliente-servidor

Las principales características de estos sistemas son:

- **Comunicación:** La comunicación se realiza a través de intercambio de mensajes, por tanto, cliente y servidor se encuentran débilmente acoplados. En nuestro caso al desarrollar un servidor RESTful las peticiones se basan en Get, Put, Delete y Post a través del protocolo HTTP. Las respuestas del servidor son en formato JSON para facilitar la interpretación de los datos en el cliente.
- **Escalabilidad:** Los sistemas cliente-servidor pueden ser escalados de manera tanto horizontal como vertical. En el caso de este proyecto, el crecimiento horizontal significaría un aumento de usuarios en la aplicación. Este crecimiento conllevaría tener que hacer un crecimiento vertical, es decir, hacer una migración a un servidor más potente o la incorporación de nuevas máquinas.

- **Modularidad:** El diseño modular permite que la aplicación sea tolerante a fallos. En este proyecto podríamos montar más de un servidor para que si alguno fallara los otros continuaran ofreciendo los servicios necesarios.
- **Reutilización:** El código que presenta la parte servidor es reutilizable, pudiéndose utilizar en varios servidores.
- **Compartición de recursos:** El servidor es compartido por los diferentes clientes al mismo tiempo.

Estas son las principales características por las que la arquitectura cliente-servidor está ampliamente extendida. Se decidió esta arquitectura para el proyecto ya que se deseaba cargar lo menos posible la parte del cliente.

3.2 Apache Cordova

Apache Cordova es un marco de desarrollo de código móvil. Este marco permite construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5 y Javascript, evitando utilizar el lenguaje de desarrollo nativo de cada plataforma.

Las aplicaciones resultantes son híbridas, ya que utilizan la representación gráfica vía vista Web en vez del Framework nativo del dispositivo. No se consideran aplicaciones web ya que tienen acceso a las APIs nativas del dispositivo.

Los complementos que proporciona Apache Cordova permiten el acceso al acelerómetro del dispositivo, cámara, sistema de archivos, etc.

Estos son los sistemas y los complementos que soporta actualmente el entorno:

| Funcionalidad | Android ²³ | iPhone / iPhone 3G | iPhone 3GS o más nuevo | Bada | BlackBerry 10 y PlayBook OS | BlackBerry OS 4.6-4.7 | BlackBerry OS 5.0-6.0+ | Firefox OS | Symbian | Tizen | webOS | Ubuntu Touch | Windows Phone |
|---|-----------------------|--------------------|------------------------|------|-----------------------------|-----------------------|------------------------|------------|---------|-------|-------|--------------|---------------|
| Acelerómetro | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | — | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si |
| Cámara | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | — | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si |
| Brújula | ✓ Si | — | ✓ Si | ✓ Si | ✓ Si | — | — | ✓ Si | — | ✓ Si | ✓ Si | ✓ Si | ✓ Si |
| Contactos | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | — | ✓ Si | ✓ Si | ✓ Si | ✓ Si | — | — | ✓ Si |
| Archivos | ✓ Si | ✓ Si | ✓ Si | — | ✓ Si | — | ✓ Si | — | — | ✓ Si | — | ✓ Si | ✓ Si |
| Geolocalización | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si |
| Media | ✓ Si | ✓ Si | ✓ Si | — | ✓ Si | — | — | — | — | ✓ Si | — | ✓ Si | ✓ Si |
| Red | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si |
| Notificaciones (alertas, sonido, vibración) | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si |
| Almacenamiento | ✓ Si | ✓ Si | ✓ Si | — | ✓ Si | — | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si | ✓ Si |

Ilustración 6 - Complementos Apache Cordova

Debido a la cantidad de posibilidades que ofrece este marco fue escogido para realizar el proyecto.

3.3 Servicio REST

REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que dispone de una gran capacidad, pero también mucha complejidad.

En este proyecto se ha optado por la solución REST, ya que facilita la manipulación de datos utilizando el estándar de comunicación JSON.

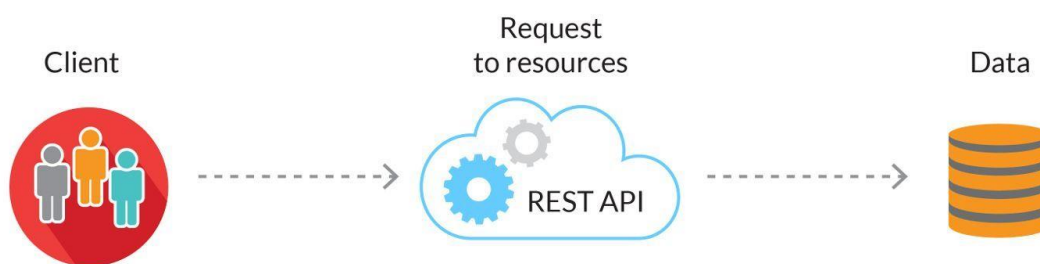


Ilustración 7 - Representación arquitectura REST

Principales características de REST:

- **Operaciones:** Las operaciones más importantes en un sistema REST son: y la especificación HTTP son cuatro: **POST** (crear), **GET** (leer y consultar), **PUT** (editar) y **DELETE** (eliminar).
- **Protocolo sin estado:** Cada petición contiene toda la información necesaria para ejecutarse, por lo que no es necesario almacenar un estado previo ni en el cliente ni en el servidor.
- **URLs:** Identifican los recursos del servidor de forma única. A través de ellas se puede acceder a la información, modificarla o borrarla.
- **Hipermédios:** Consiste en la capacidad de proporcionar una interfaz a los clientes con los enlaces adecuados para ejecutar acciones concretas sobre los datos.

Las razones por las que se ha elegido utilizar una arquitectura REST en este proyecto son las siguientes:

- **Desacoplamiento entre cliente y servidor:** este protocolo separa totalmente al usuario y el almacenamiento de los datos.
- **Visibilidad, fiabilidad y escalabilidad:** la separación de los componentes permite hacer un desarrollo escalar y hacer migraciones o cambios fácilmente.
- **Independiente del tipo de plataforma o lenguajes:** es la principal característica que buscamos en este proyecto, puesto que aplicación del cliente se implementará en diferentes plataformas.

3.4 JSON

Es un formato de texto ligero para el intercambio de datos. Se ha utilizado en este proyecto para la comunicación entre el cliente y el servidor.

La estructura de este formato está construida sobre dos estructuras:

- Colección de pares nombre/valor. En los diferentes lenguajes esto se interpreta como un objeto, registro, estructura, tabla hash, lista o matriz.
- Lista ordenada de valores. En la mayoría de los lenguajes esto se reconoce como matriz, vector o lista

Al utilizar esta estructura, prácticamente todos los lenguajes de programación modernos lo soportan.

En JSON, un objeto comienza con '{' y termina con '}'. Cada nombre va seguido de ':' y los pares nombre/valor están separados por ','.

Esto es un ejemplo de cómo el servidor envía los datos de un usuario a la aplicación:

```
[{"idUsuario":"abarajas","nombre":"Alejandro","apellidos":"Barajas","email":"a_barajas212@usal.es","telefono":"661060956","idProtectora":null}]
```


4. Técnicas

En el desarrollo del proyecto se han utilizado diversas técnicas de diferentes ámbitos, las cuales se expondrán a continuación.

4.1 Modelo Vista-Controlador (MVC)

El modelo Vista-Controlador (MVC) es un patrón arquitectónico que permite realizar una división del sistema en datos, lógica de negocio y representación. Este patrón permite mejorar el mantenimiento y reutilización del sistema.

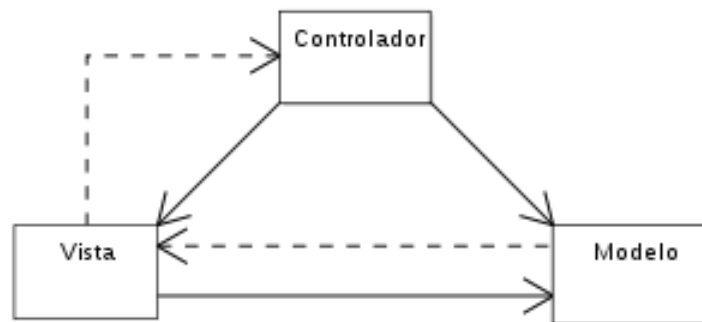


Ilustración 8 - Modelo Vista-Controlador

- **Modelo:** Contiene los datos del sistema.
- **Vista:** Obtiene los datos del modelo y los muestra al usuario. Contiene los componentes encargados de interactuar con el usuario. En este proyecto se encontraría tanto del lado del servidor web como de la vista de la aplicación.
- **Controlador:** Responde a los eventos producidos por la interacción de los usuarios con la vista y a las peticiones del modelo para actualizar la vista. Los controladores de la aplicación se encargarán de realizar las acciones solicitadas por el usuario y redirigirlas al servicio REST, que accederá a los datos.

4.2 Metodología de Durán y Bernárdez

En este proyecto se ha optado por utilizar una metodología del campo de la ingeniería del software propuesto por Amador Durán y Beatriz Bernárdez de la Universidad de Sevilla. Esta metodología propone una manera formal de documentar los objetivos, requisitos, tareas y técnicas del software a implementar.

Será utilizado en las fases de análisis y diseño de la aplicación a través de la herramienta REM.

5. Herramientas

5.1 Herramientas utilizadas en el desarrollo de la aplicación

5.1.1 Visual Studio

Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes, pudiendo ir añadiendo características según se desee.

En este proyecto se ha utilizado la extensión para utilizar Apache Cordova, que añade de manera sencilla los SDK, bibliotecas, herramientas, etc. A través del archivo “config.xml” se pueden añadir las diferentes utilidades para acceder a los recursos de los dispositivos (cámara, sistema de archivos, contactos, etc).

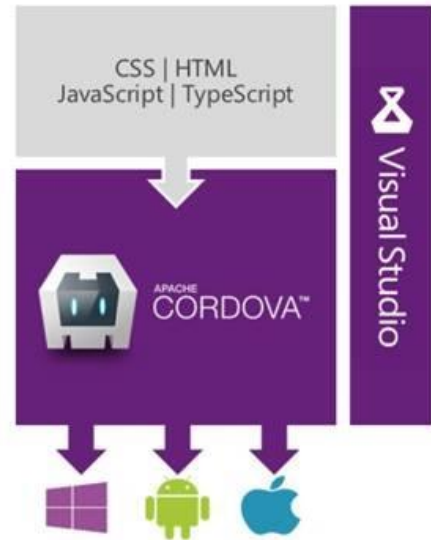


Ilustración 9 - Visual Studio + Apache Cordova

Este IDE ofrece la posibilidad de utilizar emuladores para probar las aplicaciones, además de desplegar la aplicación en cualquier dispositivo.

Independientemente de la plataforma en la que se ejecute la aplicación (iOS, Android o Windows) el depurador se conecta automáticamente. Permite revisar el código con puntos de interrupción e incluso hacer ediciones durante la depuración.

5.1.2 Framework7

El enfoque principal de Framework7 es ofrecer la oportunidad de crear aplicaciones iOS y Android con HTML, CSS y JavaScript. El Framework ofrece muchos elementos como ventanas emergentes, paneles, formularios, etc. Además de todos los componentes incluye los gestos para la interacción, como por ejemplo el barrido de iOS.

Este Framework está enfocado al diseño iOS y Google Material para ofrecer la mejor experiencia, teniendo la posibilidad de hacer aplicaciones híbridas.



Ilustración 10 - Framework7

5.1.3 GitHub



Ilustración 11 - GitHub

Es un servicio de alojamiento basado en web para el control de versiones utilizando Git. Este proyecto se ha desarrollado sobre una rama privada, pero una vez finalizado se cambiará a pública.

Para hacer la sincronización entre el código local y el alojado en Github se ha utilizado un plugin que provee Visual Studio.

En el siguiente enlace se encuentra todo el código de la aplicación:

- <https://github.com/abarajas212/Adoptame-2.0>

5.1.4 Pixlr

En la aplicación se utilizan imágenes en determinadas pantallas, pero sobre todo se utilizan iconos.

Esta herramienta gratuita permite hacer diferentes tratamientos de imágenes a través de su página web. En el proyecto ha sido utilizada para editar los diferentes iconos que aparecen en la aplicación, tanto para hacerlos transparentes, como para editar su color y tamaño.



Ilustración 12 - Pixlr

5.1.5 Chart



Ilustración 13 - Chart

Dentro de la aplicación móvil se ha incluido la biblioteca “Chart”, que permite crear gráficos de manera sencilla utilizando JavaScript.

Esta biblioteca se utilizará para crear diferentes gráficos utilizando los datos que genera el uso de la aplicación por parte de las protectoras.

5.1.6 Adobe Color Wheel

Es una biblioteca que permite utilizar ruedas cromáticas con diferentes configuraciones de colores.

Se ha utilizado en este proyecto para escoger los diferentes colores que se utilizarán en las gráficas. El objetivo era encontrar colores que no fueran demasiado dispares, pero que se apreciaran las diferentes zonas de las que están formadas las gráficas.



Ilustración 14 - Adobe Color Wheel

5.1.7 Lenguajes de programación

5.1.7.1 HTML5



Ilustración 15 - HTML5

Es un lenguaje de programación que se utiliza principalmente para el desarrollo de páginas de Internet. Sus siglas corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto. Esto significa que el lenguaje se basa en etiquetas para crear el contenido de la página.

Este proyecto utiliza Apache Cordova, con lo que la parte visual se desarrolla con HTML5. Al tratarse de una aplicación móvil no se ha utilizado HTML5 puro, sino que se le ha añadido Framework7 para proporcionar una interfaz más amigable al usuario.

5.1.7.2 CSS3

Para dar un estilo personalizado a la aplicación se han utilizado las hojas de estilos. Este componente permite modificar el diseño gráfico que ofrece HTML para hacer la aplicación más atractiva.

Se han utilizado las hojas de estilo que incluye Framework7 para sus componentes, además de diseñar algunas propias para darle un estilo personal a la aplicación.



Ilustración 16 - CSS3

5.1.7.3 JavaScript



JavaScript es un lenguaje de programación orientado a objetos, basado en prototipos, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, permitiendo mejoras en la interfaz de usuario y aportando dinamismo a las páginas web.

Ilustración 17 - JavaScript

Entre sus principales funciones se encuentran:

- Cargar contenido en la página web o enviar datos al servidor sin necesidad de recargar la página.
- Animaciones de los elementos como por ejemplo cambiar su tamaño, color, darle movimiento, etc.
- Validación de la información introducida por el usuario antes de ser enviada al servidor.

5.1.7.4 JQuery

Es una biblioteca multiplataforma de JavaScript que permite interactuar con los elementos HTML de manera más sencilla. En este proyecto se ha utilizado la versión última versión: v3.3.1

Las principales características por las que se escogido son:

- Selección sencilla de elementos del DOM.
- Facilita las modificaciones en el DOM.
- Captura de eventos de manera simplificada.



Ilustración 18 - JQuery

5.2 Herramientas utilizadas en el desarrollo del servidor

5.2.1 SLIM 3

Slim es un microframework para PHP que permite desarrollar APIs de forma rápida y con poco código. Proporciona un enrutador rápido y potente que asigna las devoluciones de llamadas a los métodos de solicitud HTTP y URI específicos.

Para crear la estructura necesaria para utilizar el Framework se ha utilizado Composer. A parte, se ha creado una estructura para las diferentes peticiones, la configuración para necesaria para utilizar la base de datos y una clase para enviar emails.



Ilustración 19 - Slim 3

5.2.2 Sublime Text 3



Es el editor de texto que se ha utilizado en este proyecto para programar la parte del servidor.

Este software fue escogido debido a su facilidad de manejo y su excelente sintaxis remarcada. Además, posee predictor de palabras, detección de cambios y una interfaz muy amigable.

Ilustración 20 - Sublime Text 3

5.2.3 PostMan

Es la herramienta más importante a la hora de probar el funcionamiento de la API. Está compuesto de diferentes herramientas, pero la más utilizada en este proyecto ha sido la creación de peticiones a la API.

Permite definir los tipos de peticiones, cabeceras, parámetros e incluso cargar archivos para comprobar la subida. Estas peticiones pueden ser almacenadas en colecciones estructuradas, lo que permite tener todas las peticiones organizadas para probar la API en cualquier momento.



Ilustración 21 - Postman

5.2.4 XAMPP



Ilustración 22 - Xampp

Como servidor de la aplicación se ha utilizado XAMPP, que incluye Apache y MySQL, necesarios para el funcionamiento del proyecto. Este software permite instalar de manera sencilla todos sus componentes sin importar el sistema operativo.

En este proyecto se han utilizado principalmente las funcionalidades relacionadas con las bases de datos, como por ejemplo “PHPMyAdmin”, que es un gestor de bases de datos gráfico.

5.2.5 Lenguajes de programación

5.2.5.1 PHP + SQL

PHP es un lenguaje de código abierto de lado del servidor originalmente diseñado para el desarrollo web.

SQL es un lenguaje específico para administrar y recuperar información de sistemas de gestión de bases de datos.

Juntos dan la funcionalidad y potencia a la API desarrollada para este proyecto. La consulta se crea utilizando el lenguaje SQL, pero es lanzada desde la API a través de PHP. Esta consulta puede contener parámetros que son cargados desde PHP.

El servidor también posee peticiones puramente PHP, como por ejemplo la de enviar emails. Para esta funcionalidad se ha cargado la biblioteca PHPMailer, descargada directamente desde GitHub e incluida en el servidor.

5.3 Herramientas CASE

5.3.1 Visual Paradigm

Herramienta utilizada en la ingeniería del software para la creación de diagramas de clases, de secuencia, diseño, etc. Utiliza el lenguaje UML.



Ilustración 23 - Visual Paradigm

5.3.2 REM

Programa que proporciona la Universidad de Sevilla utilizado para la captura de requisitos, objetivos y definir casos de uso en la parte de ingeniería del software.



Ilustración 24 - REM

5.3.3 EZEstimate



Es una herramienta que permite estimar el esfuerzo de un proyecto a través de los puntos de Casos de Uso. Nos dará un resultado aproximado del número de horas que conllevará el desarrollo del proyecto.

Ilustración 25 - EZEstimate

5.3.4 Microsoft Project

Software de administración de proyectos que se ha empleado en la gestión del proyecto para la creación de la planificación temporal, la asignación de tareas y el análisis de la carga de trabajo, así como para hacer un cálculo aproximado del presupuesto.



Ilustración 26 - Microsoft Project

5.3.5 Microsoft Word



Para la redacción de la documentación, tanto la memoria como los anexos se ha utilizado esta herramienta. La versión de este software es "Office 365".

Ilustración 27 - Microsoft Word

6. Proceso de desarrollo

En esta sección se especificarán las partes más representativas de las distintas fases del desarrollo del proyecto.

6.1 Marco de trabajo

Se ha utilizado como marco de trabajo el Proceso Unificado, debido a su extensibilidad y flexibilidad en el desarrollo software. El sentido de utilizar este proceso es ir refinando el producto hasta conseguir un sistema final con buena calidad y fácil de mantener.

Las principales características del Proceso Unificado son:

- **Dirigido por Casos de Uso:** los Casos de Uso representarán las funcionalidades del sistema. Los usuarios interactuarán con el sistema y serán los Casos de Uso quien represente estas interacciones. Se utilizan en todas las fases del desarrollo y están ligados a la arquitectura del sistema, por eso se dice que dirigirán el proceso de desarrollo.
- **Centrado en la arquitectura:** la arquitectura muestra la organización del proyecto y está ligada a los Casos de Uso. Se partirá de una arquitectura inicial que irá mejorando a medida que avanza el proyecto.
- **Iterativo e incremental:** el desarrollo del proyecto se dividirá en partes simples, llamadas iteraciones, que se realizarán de forma incremental. Una vez finalizada cada iteración se hace una revisión y se analiza el progreso. Esto se conoce como hito y permite refinar el sistema hasta conseguir un buen producto final.

El Proceso Unificado se compone de diferentes fases: Inicio, Elaboración, Construcción y Transición. Estas fases se dividen en iteraciones para distribuir el esfuerzo entre las diferentes etapas.

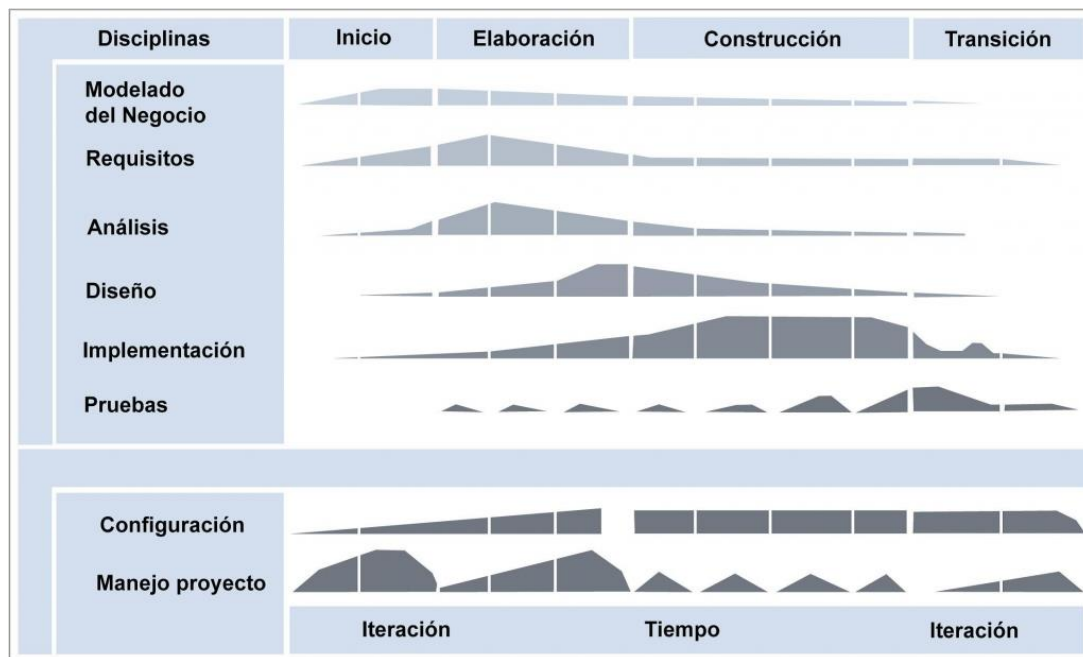


Ilustración 28 - Proceso Unificado

- **Inicio**: Se define el alcance del proyecto y se desarrollan los casos de negocio.
- **Elaboración**: Se planifica el proyecto, se especifica la funcionalidad a través de los casos de uso y se diseña la arquitectura del sistema.
- **Construcción**: Se construye el producto.
- **Transición**: Se obtiene una versión inicial del producto, se corrigen problemas y se incorporan mejoras sugeridas en la revisión.

6.2 Estimación de Coste y Esfuerzo

La estimación de coste y esfuerzo es una técnica que permite conocer de manera aproximada tanto el tamaño del software, cómo el coste de su implementación. Se ha utilizado el método de Puntos de Caso de Uso, que se basa en hacer la medición a través de la funcionalidad que representa cada Caso de Uso. Esta medición nos proporcionará una estimación del número de horas por Caso de Uso.

A continuación, se muestra un resumen de la estimación realizada. Para obtener una información más detallada se debe acceder al “Anexo I – Plan del Proyecto Software”.

Para hacer la estimación del esfuerzo se deberá aplicar la siguiente fórmula:

$$\text{Esfuerzo} = \text{UCP} * \text{F}$$

Siendo:

- **UCP:** Puntos de Caso de Uso
- **F:** Factor de conversión, que consiste en el número de horas de persona por UCP. Por defecto este factor tiene el valor 20.

Para calcular los Puntos de Caso de Uso hay que considerar: actores, escenarios, factores técnicos y factores de entorno.

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF}$$

Siendo:

- **UCP:** Puntos de Caso de Uso
- **UUCP:** Puntos de Caso de Uso sin Ajustar
- **TCF:** Factores de Complejidad Técnica
- **ECF:** Factores de Complejidad del Entorno

Para obtener los Puntos de Caso de Uso sin Ajustar es necesario considerar todos los actores y casos de uso que componen cada módulo del sistema.

$$\text{UUCP} = \text{UAW} * \text{UUCW}$$

Siendo:

- **UUCP:** Puntos de Caso de Uso sin Ajustar
- **UAW:** Factores de peso de los Actores sin Ajustar
- **UUCW:** Factores de peso de los Caso de Uso sin ajustar

Donde:

- **UAW** = n° Actores Simple * Factor de peso Simple + n° Actores Medio * Factor de peso Medio + n° Actores Complejos * Factor de peso Complejo.

| Tipo de Actor | Descripción | Factor de peso | Nº Actores |
|---------------|---|----------------|------------|
| Simple | Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API). | 1 | 0 |
| Medio | Otro sistema interactuando a través de un protocolo o una persona interactuando a través de una interfaz en modo texto. | 2 | 0 |
| Complejo | Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI). | 3 | 3 |

Tabla 1 – Clasificación de los tipos de actores

- **UUCW** = nº Casos de Uso Simple * Factor de peso Simple + nº Casos de Uso Medio * Factor de peso medio + nº Casos de Uso Complejo * Factor de peso Complejo

| Tipo de Caso de Uso | Descripción | Factor de peso | Nº Casos de Uso |
|---------------------|------------------------|----------------|-----------------|
| Simple | De 1 a 3 transacciones | 1 | 18 |
| Medio | De 4 a 7 transacciones | 2 | 9 |
| Complejo | Más de 7 transacciones | 3 | 0 |

Tabla 2 – Clasificación de los tipos de Casos de Uso

Los Factores de Complejidad Técnica requieren considerar el peso de cada factor y la complejidad en cada uno de ellos, basándose en una escala del 0 al 5.

$$\text{TCF} = 0.6 + (0.01 * \text{TF})$$

Siendo:

- **TF:** Sumatorio de la multiplicación de cada peso por el factor de complejidad percibido.

| Factor | Descripción | Peso | Factor de complejidad |
|--------|--|------|-----------------------|
| T1 | Sistema distribuido | 2 | 5 |
| T2 | Objetivos de performance o tiempo de respuesta | 1 | 4 |
| T3 | Eficiencia del usuario final | 1 | 4 |
| T4 | Procesamiento interno complejo | 1 | 2 |
| T5 | El código debe ser reutilizable | 1 | 4 |
| T6 | Facilidad de instalación | 0,5 | 2 |
| T7 | Facilidad de uso | 0,5 | 5 |
| T8 | Portabilidad | 2 | 5 |
| T9 | Facilidad de cambio | 1 | 4 |
| T10 | Concurrencia | 1 | 0 |
| T11 | Incluye objetivos especiales de seguridad | 1 | 3 |
| T12 | Provee acceso directo a terceras partes | 1 | 0 |

| | | | |
|-----|---|---|---|
| T13 | Se requiere facilidades especiales de entrenamiento a usuario | 1 | 0 |
|-----|---|---|---|

Tabla 3 - Factores de complejidad técnica

Los Factores de Complejidad del Entorno se calculan considerando el peso de cada factor y su complejidad basándose en una escala del 0 al 5.

$$ECF = 1.4 + (-0.03 * EF)$$

Siendo:

- **EF:** Sumatorio de la multiplicación de cada peso por el factor de complejidad percibido.

| Factor | Descripción | Peso | Factor de complejidad |
|--------|---|------|-----------------------|
| E1 | Facilidad con UML | 1.5 | 3 |
| E2 | Experiencia en la aplicación | 0.5 | 0 |
| E3 | Experiencia en orientación a objetos | 1 | 5 |
| E4 | Capacidad de los analistas | 0.5 | 3 |
| E5 | Motivación | 1 | 5 |
| E6 | Estabilidad de los requisitos | 2 | 4 |
| E7 | Trabajadores a tiempo parcial | -1 | 1 |
| E8 | Dificultad del lenguaje de programación | -1 | 5 |

Tabla 4 - Factores de complejidad de entorno

El resultado de la estimación son 2592 horas por punto de caso de uso, es decir, aproximadamente 108 días para realizar el desarrollo del sistema.

Estimation Summary

UAW

10

UUCW

160

UUPC = UAW + UUCW

170

TFactor

39

EFactor

21

TCF = 0.6 + (.01*TFactor)

0,99

EF = 1.4 + (-0.03*EFactor)

0,77

UCP = UUCP*TCT*EF

129,591

Total Effort@ 20 Hrs/UCP

2591,82

Ilustración 29 - Resultados de la Estimación de Esfuerzo

6.3 Planificación temporal

La planificación temporal permite organizar las tareas del proyecto con el objetivo de obtener una idea aproximada del orden y duración de las mismas. En este proyecto se ha seguido la metodología del Proceso Unificado, por tanto, las tareas deben de seguir el orden de dicha metodología.

Para obtener toda la información sobre la planificación temporal acceder al “Anexo I – Plan del Proyecto Software”.

En la siguiente imagen aparece la planificación resumida del proyecto, así como la distribución de las mimas mediante un diagrama de Gantt:

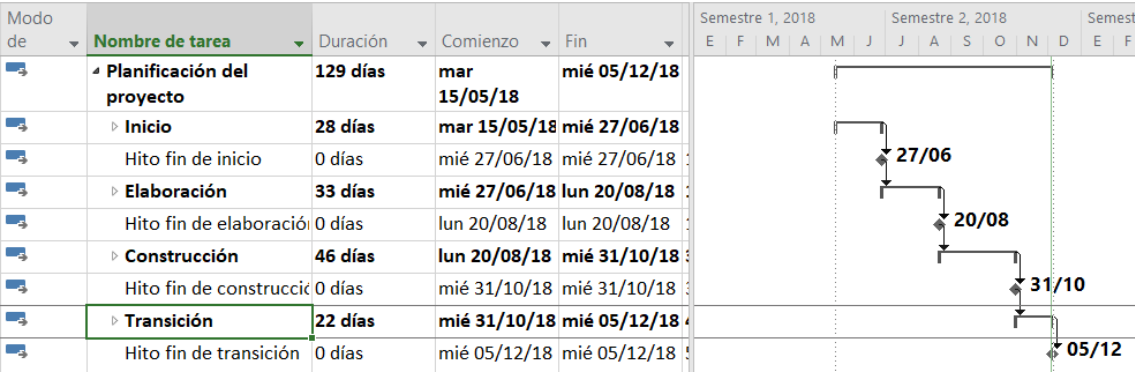


Ilustración 30 - Resumen de la planificación temporal

En cada fase del Proceso Unificado se detallan las diferentes tareas de las que se compone:

| | | | | | | Semestre 1, 2018 | | | | | Semestre 2, 2018 | | | | |
|---|------------|--------------|--------------|-------|--|------------------|---|---|---|---|------------------|---|---|---|---|
| Nombre de tarea ▼ | Duración ▼ | Comienzo ▼ | Fin ▼ | Prede | | E | F | M | A | M | J | J | A | S | O |
| ■ Elaboración | 33 días | mié 27/06/18 | lun 20/08/18 | 18 | | | | | | | | | | | |
| ■ Requisitos | 3 días | mié 27/06/18 | lun 02/07/18 | | | | | | | | | | | | |
| Revisión de requisitos | 3 días | mié 27/06/18 | lun 02/07/18 | | | | | | | | | | | | |
| ■ Análisis | 7 días | lun 02/07/18 | jue 12/07/18 | | | | | | | | | | | | |
| Finalización de la arquitectura | 4 días | lun 02/07/18 | lun 09/07/18 | 21 | | | | | | | | | | | |
| Diagramas de secuencia | 7 días | lun 02/07/18 | jue 12/07/18 | 21 | | | | | | | | | | | |
| ■ Diseño | 9 días | jue 12/07/18 | jue 26/07/18 | 22 | | | | | | | | | | | |
| Refinamiento de la interfaz | 7 días | jue 12/07/18 | mar 24/07/18 | | | | | | | | | | | | |
| Realización de diagramas | 7 días | jue 12/07/18 | mar 24/07/18 | | | | | | | | | | | | |
| Revisión del diseño | 2 días | mar 24/07/18 | jue 26/07/18 | 27 | | | | | | | | | | | |
| ■ Implementación | 9 días | jue 26/07/18 | vie 10/08/18 | 25 | | | | | | | | | | | |
| Primeras funciones REST | 8 días | jue 26/07/18 | jue 09/08/18 | | | | | | | | | | | | |
| Primeras interacciones cliente-servidor | 9 días | jue 26/07/18 | vie 10/08/18 | | | | | | | | | | | | |
| ■ Pruebas | 5 días | lun 13/08/18 | lun 20/08/18 | 29 | | | | | | | | | | | |
| Corrección de errores | 5 días | lun 13/08/18 | lun 20/08/18 | | | | | | | | | | | | |
| Hito fin de elaboración | 0 días | lun 20/08/18 | lun 20/08/18 | 19 | | | | | | | | | | | |

Ilustración 31 - Planificación de tareas

6.4 Especificación de requisitos

La especificación de requisitos recoge los objetivos, requisitos funcionales y no funcionales del sistema. El objetivo es especificar las necesidades que debe cumplir el producto software a desarrollar, por lo que esta fase fija las bases del proyecto.

En el “Anexo II – Especificación de Requisitos del Software” se detalla con profundidad la captura de estos requisitos.

Para comenzar con la Especificación de requisitos se parte de un análisis textual, que permite identificar los elementos de los que se compondrá el sistema. A través del análisis se obtendrán los principales objetivos del sistema, los actores que intervendrán y requisitos que el sistema debe cumplir.

Se ha tomado como análisis textual la propuesta del Trabajo de Fin de Grado.

Título:

Plataforma para el cuidado y la adopción de animales

Descripción:

Desarrollo de una aplicación multiplataforma para facilitar la adopción de animales abandonados. El objetivo principal es intentar facilitar el proceso de adopción para las protectoras de animales mediante un aplicativo móvil. Esta aplicación servirá de unión entre protectoras y usuarios que buscan una adopción. Esta unión permitirá agilizar los trámites de la adopción a través de la aplicación. Se dispondrá de una base de datos donde los interesados pueden ver las características de los diferentes animales, ayudando a elegir cual animal es mejor para un posible dueño.

Objetivos funcionales:

Aplicación de gestión para las protectoras de animales. Aplicativo móvil que permita la adopción de una mascota por parte de la persona interesada. Diseño de un dashboard para mostrar estadísticas: Comunidades Autónomas que más adoptan, Protectoras que tienen más animales en adopción, Etc

A continuación, se muestran las partes más relevantes de la Especificación de requisitos.

6.4.1 Objetivos del sistema

Los principales objetivos que tendrá que cumplir el sistema son:

- Gestión de usuarios
- Gestión de animales
- Comunicación con el servidor
- Gestión de la interfaz de usuario
- Gestión de protectoras

A continuación, se muestra un ejemplo de cómo se han documentado los objetivos del sistema:

| | |
|-----------------|--|
| OBJ-0001 | Gestión de usuarios |
| Versión | 1.0 (20/11/2018) |
| Autores | Alejandro Barajas González |

| | |
|---------------------|--|
| Fuentes | |
| Descripción | El sistema deberá <i>gestionar la información de los usuarios que acceden al mismo</i> |
| Subobjetivos | Ninguno |
| Importancia | vital |
| Urgencia | inmediatamente |
| Estado | validado |
| Estabilidad | alta |
| Comentarios | Ninguno |

Tabla 5 - Objetivo "Gestión de usuarios"

6.4.2 Requisitos de información

Los requisitos de información representan los datos que serán almacenados y gestionados en el sistema.

Los principales requisitos de información capturados son:

- Información de usuario
- Información de protectoras
- Información de animales

Este es un ejemplo de la estructura que se ha utilizado para documentar los requisitos de información:

| | |
|---------------------|---|
| IRQ-0002 | Información de protectora |
| Versión | 1.0 (20/11/2018) |
| Autores | Alejandro Barajas González |
| Fuentes | |
| Dependencias | • [OBJ-0005] Gestión de protectoras |

| | | |
|--------------------------------|---|---------------|
| Descripción | <i>El sistema deberá almacenar la información correspondiente a los datos de las protectoras de la aplicación. En concreto:</i> | |
| Datos específicos | <ul style="list-style-type: none"> ▪ Identificador ▪ Nombre ▪ Descripción ▪ Email ▪ Teléfono ▪ Ciudad | |
| Tiempo de vida | Medio | Máximo |
| | | |
| Ocurrencias simultáneas | Medio | Máximo |
| | | |
| Importancia | vital | |
| Urgencia | inmediatamente | |
| Estado | validado | |
| Estabilidad | alta | |
| Comentarios | Ninguno | |

Tabla 6 - Requisito de Información "Información de protectora"

6.4.3 Requisitos no funcionales

Los requisitos no funcionales hacen referencia a aspectos no relacionados con la funcionalidad a cumplir por el sistema. Los requisitos no funcionales identificados son:

- Escalabilidad
- Usabilidad
- Seguridad
- Ligereza
- Estabilidad
- Accesibilidad

La plantilla utilizada para documentar los requisitos no funcionales presenta la siguiente estructura:

| | |
|---------------------|---|
| NFR-0001 | Escalabilidad |
| Versión | 1.0 (20/11/2018) |
| Autores | Alejandro Barajas González |
| Fuentes | |
| Dependencias | Ninguno |
| Descripción | El sistema deberá <i>ser escalable, es decir, que sea eficiente ante un cambio.</i> |
| Importancia | importante |
| Urgencia | inmediatamente |
| Estado | validado |
| Estabilidad | alta |
| Comentarios | Ninguno |

Tabla 7 - Requisito no funcional "Escalabilidad"

6.4.4 Requisitos funcionales

Los requisitos funcionales se han recogido en forma de Casos de Uso, los cuales representan las diferentes funcionalidades de las que se compone el sistema. Para conocer detalladamente todos los Casos de Uso acceder al anexo citado anteriormente.

Para recoger la funcionalidad de los diferentes Casos de Uso se ha utilizado la siguiente estructura:

| | |
|----------------|-----------------------------|
| UC-0010 | Registrar protectora |
| Versión | 1.0 (21/11/2018) |

| | | |
|----------------------------|---|--|
| Autores | Alejandro Barajas González | |
| Fuentes | ? | |
| Dependencias | <ul style="list-style-type: none"> • [OBJ-0005] Gestión de protectoras • [IRQ-0002] Información de protectoras • [CRQ-0002] Protectora | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>una nueva protectora necesite registrarse</i> | |
| Precondición | La protectora no está resgistrada en el sistema | |
| Secuencia normal | Paso | Acción |
| | 1 | El actor Protectora (ACT-0002) <i>solita a la aplicación registrarse</i> |
| | 2 | El actor Protectora (ACT-0002) <i>Introduce los datos para el registro</i> |
| | 3 | El sistema <i>valida los datos introducidos por la protectora</i> |
| | 4 | El sistema <i>comunica al servidor que almacene los datos de la nueva protectora</i> |
| Postcondición | La protectora queda registrada en la aplicación | |
| Excepciones | Paso | Acción |
| | 3 | Si <i>Los datos de la protectora no cumplen las condiciones que impone el sistema</i> , el sistema <i>notifica al usuario que los datos introducidos son incorrectos</i> , a continuación este caso de uso <i>queda sin efecto</i> |
| Rendimiento | Paso | Tiempo máximo |
| | - | - |
| Frecuencia esperada | PD | |
| Importancia | vital | |
| Urgencia | inmediatamente | |
| Estado | validado | |

| | |
|--------------------|---------|
| Estabilidad | alta |
| Comentarios | Ninguno |

Tabla 8 - Caso de Uso "Registrar Protectora"

6.5 Análisis del sistema

Una vez recogidos los objetivos y los requisitos que debe cumplir el sistema, se empezará el análisis de estos. A través del análisis de los requisitos se identificarán los objetos que intervienen en el sistema y la interacción entre ellos. También se realizará una vista arquitectónica inicial que mostrará la estructura y organización.

En esta memoria se mostrarán las partes más relevantes de esta fase, para una información más detallada es necesario acceder al “Anexo III – Análisis del sistema”.

6.5.1 Modelo de dominio

El modelo de dominio es una representación de los objetos conceptuales del mundo real. A través de él también se pueden identificar las relaciones entre las entidades del dominio, así como conocer sus atributos.

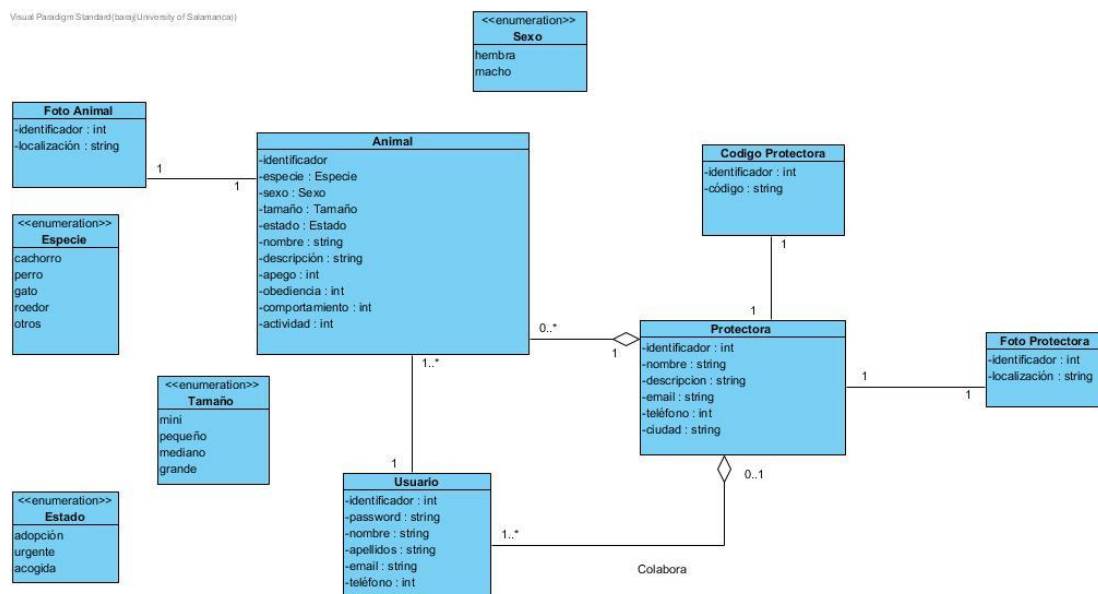


Ilustración 32 - Modelo de dominio

6.5.2 Vista arquitectónica del modelo de análisis

Una vez terminado el modelo de dominio y haber documentado los diferentes paquetes que surgen en el análisis, se realiza la vista arquitectónica inicial. Esta vista representa de manera inicial los diferentes componentes del sistema y su relación entre ellos. A medida que se vaya avanzando en el diseño del sistema, esta vista se irá refinando.

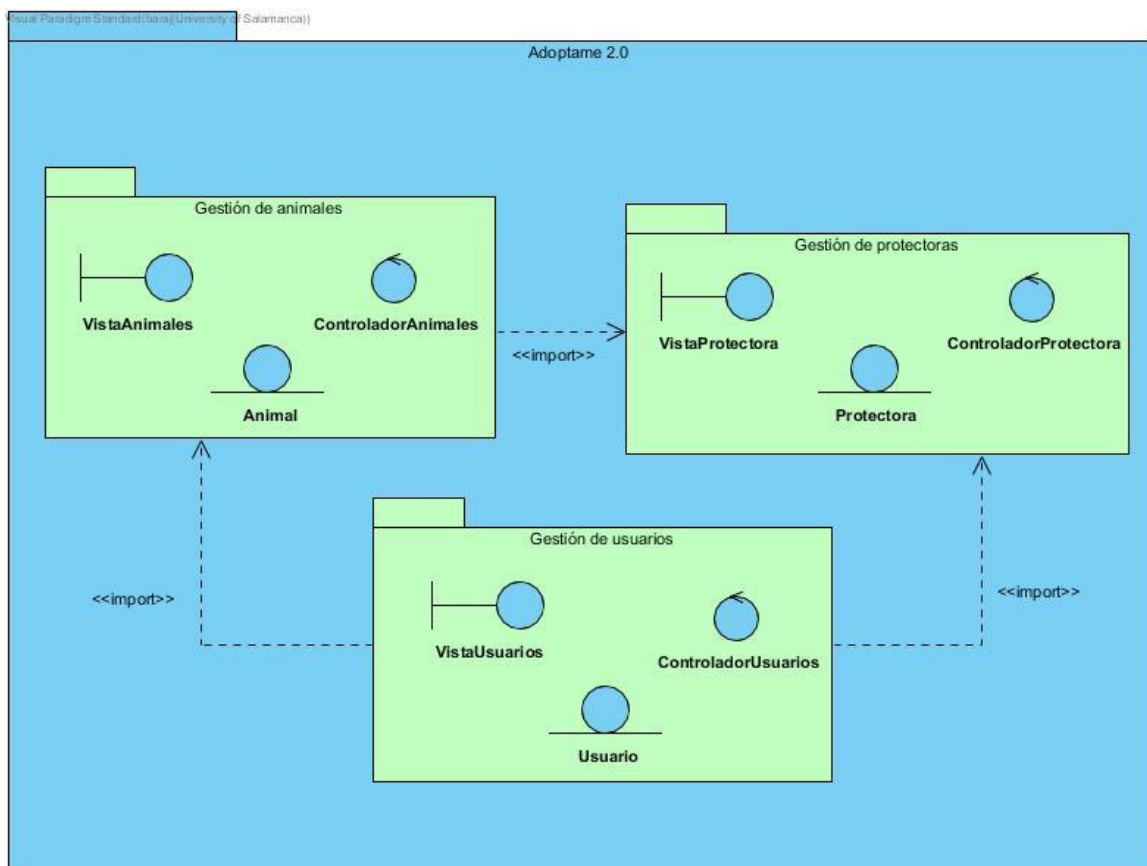


Ilustración 33 - Vista arquitectónica del modelo de análisis

6.5.3 Vista arquitectónica del modelo de análisis

Utilizando los diagramas de secuencia se puede observar la interacción y los mensajes intercambiados entre las diferentes clases y actores identificados en el sistema. En esta fase todavía existe un alto nivel de abstracción, por tanto, los mensajes no son exactamente los utilizados.

En la siguiente ilustración se muestra uno de los diagramas de secuencia que representa la realización de uno de los Casos de Uso.

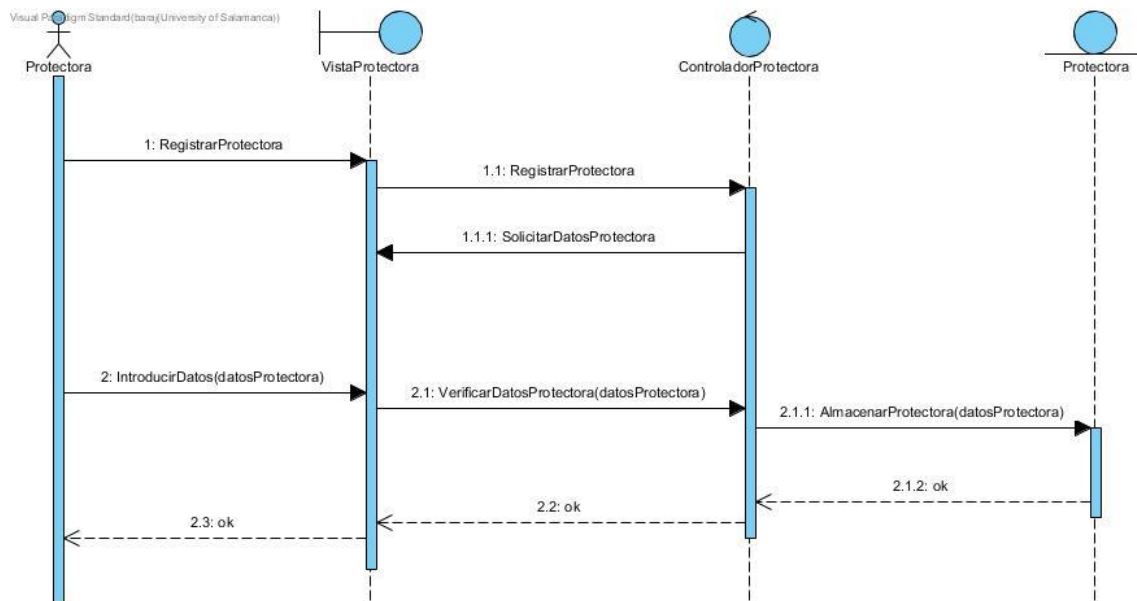


Ilustración 34 - Diagrama de Secuencia "Registrar protectora"

6.6 Diseño del sistema

La fase de Diseño del sistema está centrada en el dominio de la solución, basándose en el dominio del problema. El objetivo de esta fase es detallar los requisitos y componentes que se han analizado anteriormente. Se parte del dominio del problema y se va refinando poco a poco hasta obtener todos los detalles de los componentes. Una vez finalizada la fase del diseño se realizará la implementación del sistema.

Para obtener más detalles de la fase de Diseño accede al *"Anexo IV – Diseño del Sistema"*.

6.6.1 Patrones arquitectónicos

El patrón arquitectónico principal que se ha utilizado en este proyecto es el patrón "Modelo-Vista-Controlador".

El patrón MVC separa los datos, la lógica y las interfaces de usuario. Debido a su separación en tres capas bien diferenciadas se consigue una alta cohesión y un bajo acoplamiento entre las mismas.

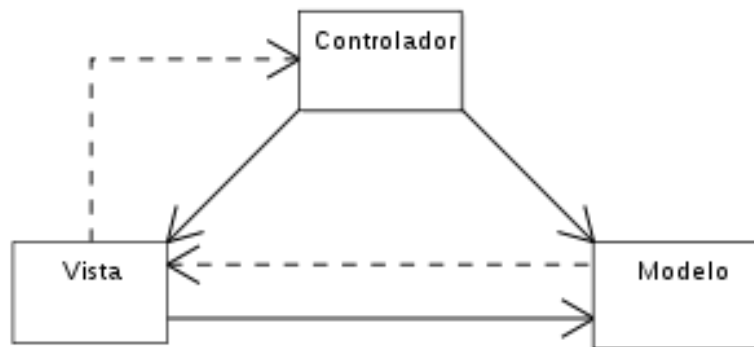


Ilustración 35 - Patrón arquitectónico "MVC"

- **Modelo:** Es la capa encargada de los datos, es decir, la que se encarga de hacer peticiones a la base de datos para enviar y recibir información.
- **Vista:** Se trata de la capa que obtiene los datos del modelo y los muestra al usuario. Es la capa encargada de interactuar con el usuario y presentarle la información. Este componente se encuentra tanto del lado de la aplicación móvil como de la del servidor.
- **Controlador:** Es la capa que sirve de enlace entre la vista y el modelo. Se encarga de responder a los eventos producidos que surgen de la interacción de los usuarios con la vista o responder a las peticiones del modelo para actualizar la vista. En esta capa aparecerán funcionalidades ofrecidas tanto por el servicio web como por la aplicación móvil.
Los controladores de la aplicación móvil se encargarán de gestionar las peticiones realizadas por los usuarios y redirigirlas a los controladores del servicio REST, que accederá a la base de datos.

6.6.2 Subsistemas de diseño

Los subsistemas de diseño permiten ver los distintos módulos de los que se compone la aplicación y sus relaciones.

Cada subsistema recoge las clases que tienen una funcionalidad similar para que el sistema quede bien estructurado.

En este proyecto hay dos partes bien diferenciadas, la aplicación móvil y el servicio REST. Dentro de cada uno de ellos surgen diferentes subsistemas como se puede observar en la imagen.

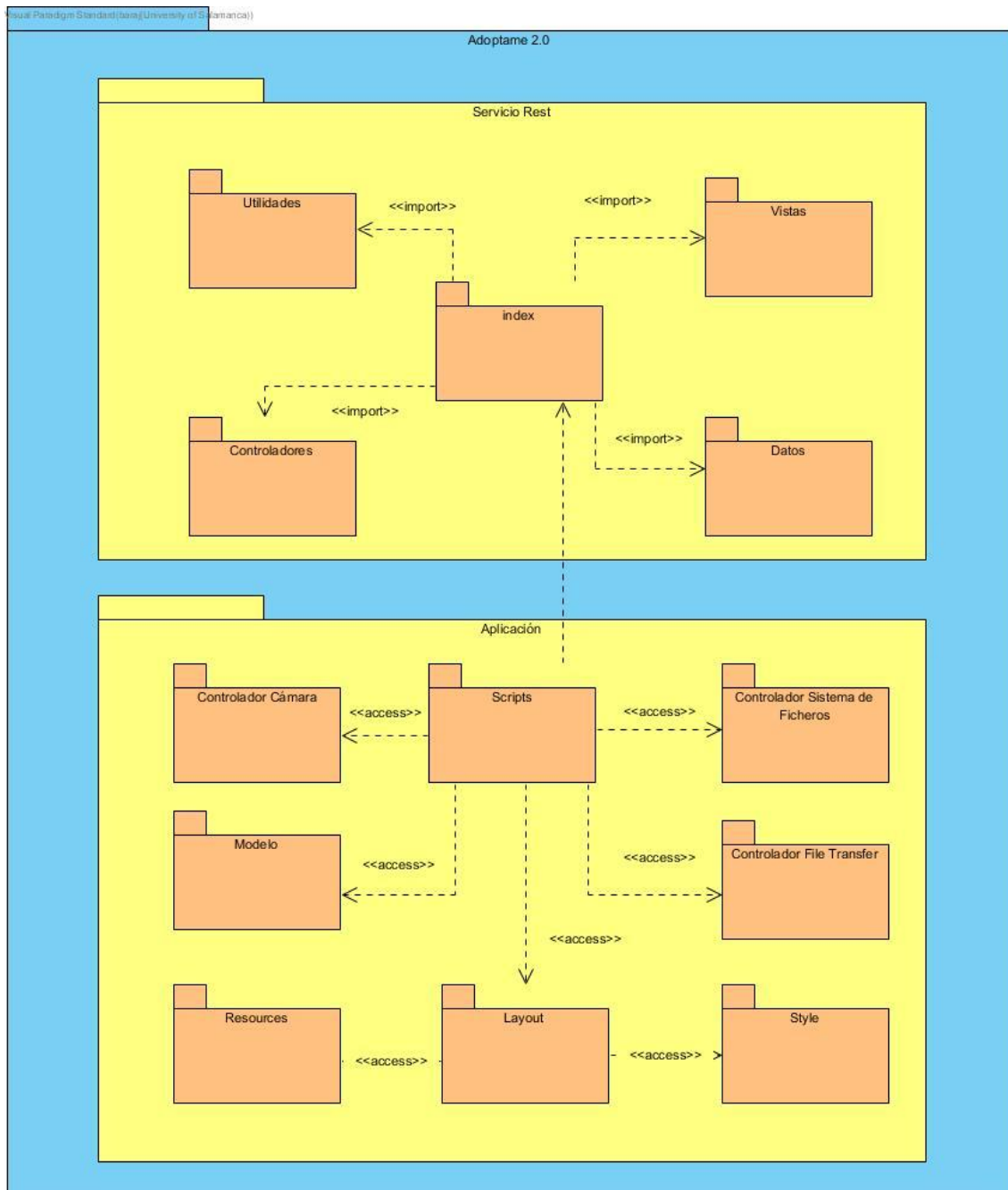


Ilustración 36 - Subsistemas de diseño

En el “Anexo IV – Diseño del Sistema” se detallan los diferentes subsistemas. En la imagen de a continuación se puede observar el subsistema “Vistas”, que pertenece a la parte del servicio REST.

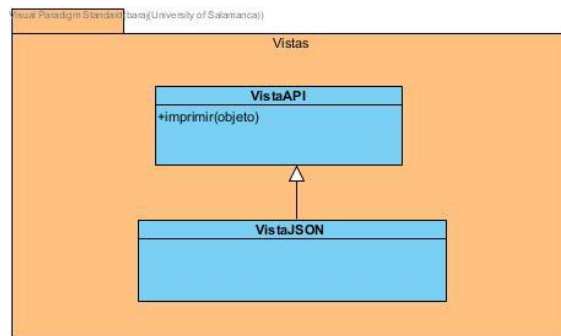


Ilustración 37 - Subsistema "Vistas"

6.6.3 Clases de diseño

En este diagrama se describen las clases del modelo de diseño, que parte del diagrama de clases del modelo de análisis. En él se observan las diferentes entidades que aparecen en el sistema, sus atributos, operaciones y relaciones.

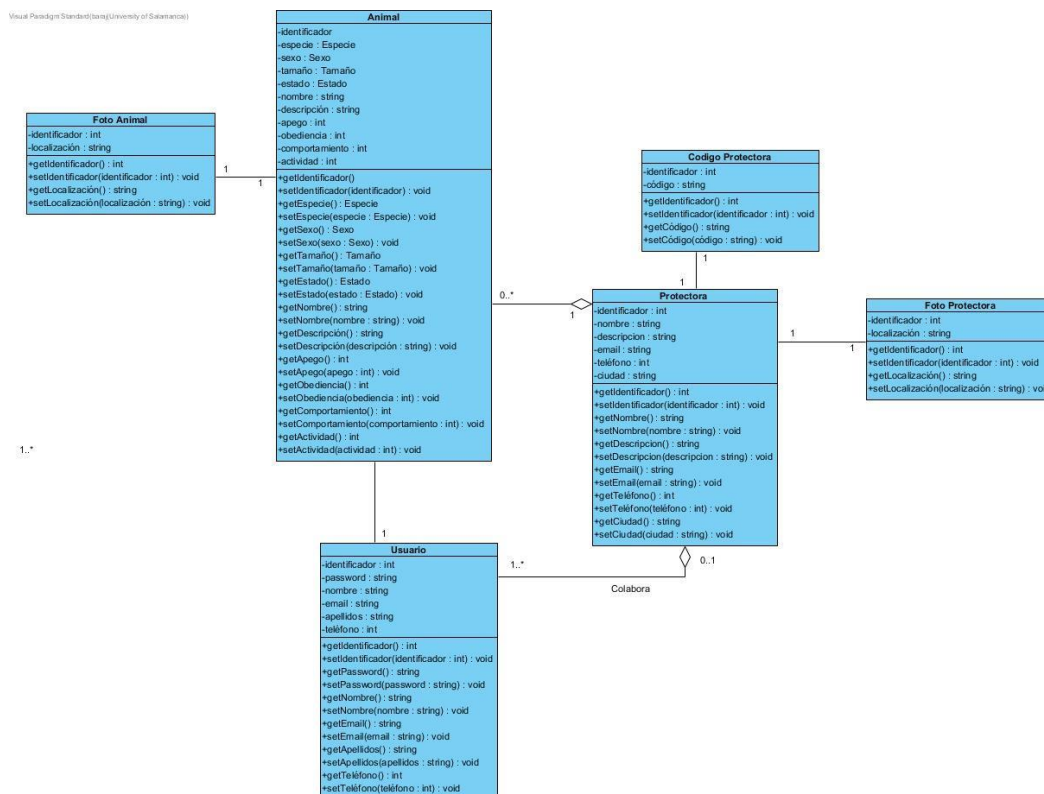


Ilustración 38 - Diagrama de Clases del modelo de Diseño

6.6.4 Vista arquitectónica del modelo de diseño

En este diagrama se muestran todos los subsistemas que componen el sistema organizados según el patrón MVC.

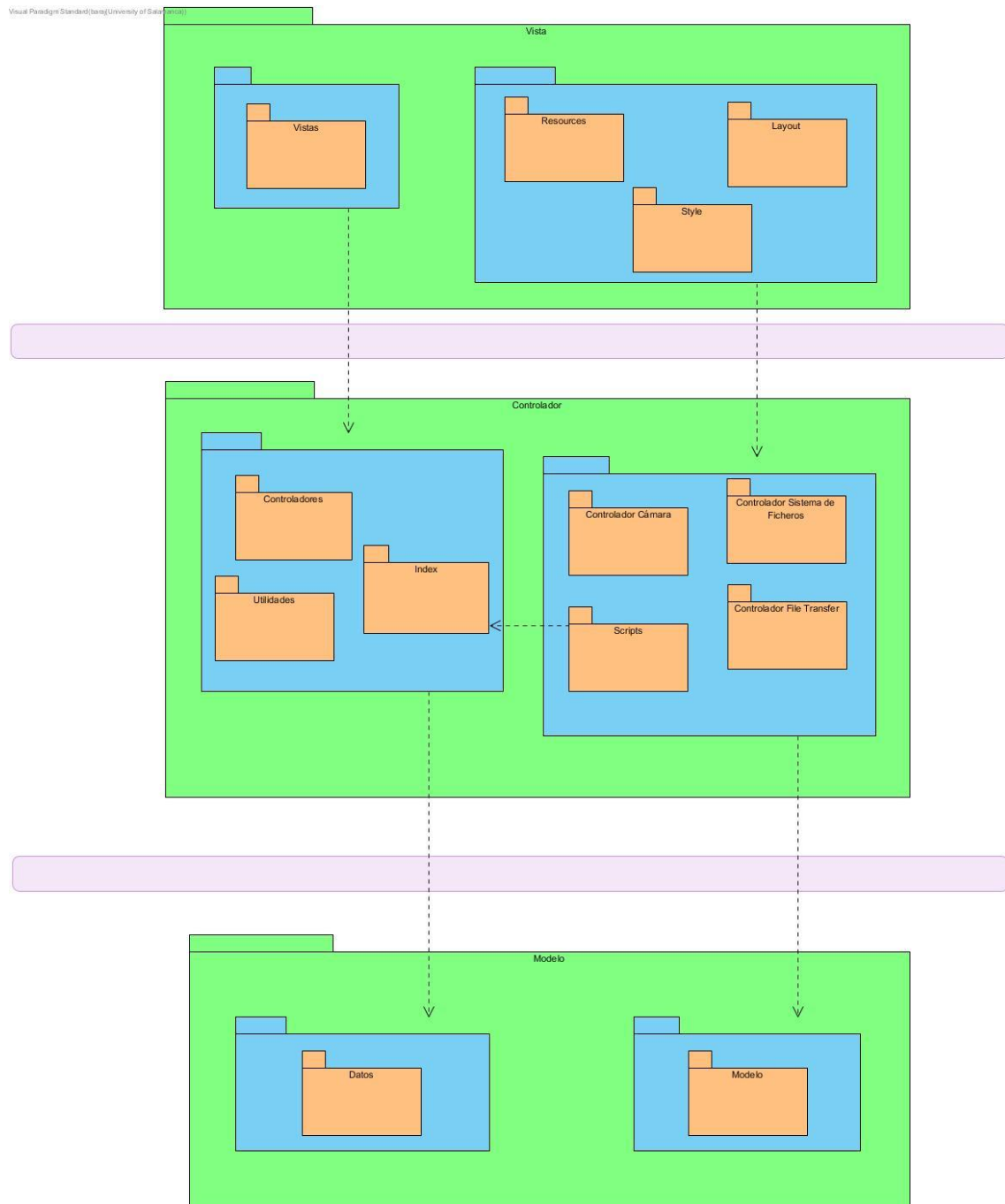


Ilustración 39 - Vista de la arquitectura

6.6.5 Realización de Casos de Uso en el Diseño

La realización de Casos de Uso en esta fase permite mostrar los mensajes intercambiados entre los diferentes componentes que componen el sistema de una manera más detallada que en la fase de Análisis.

A continuación, se muestra un diagrama de secuencia obtenido en la fase de diseño.

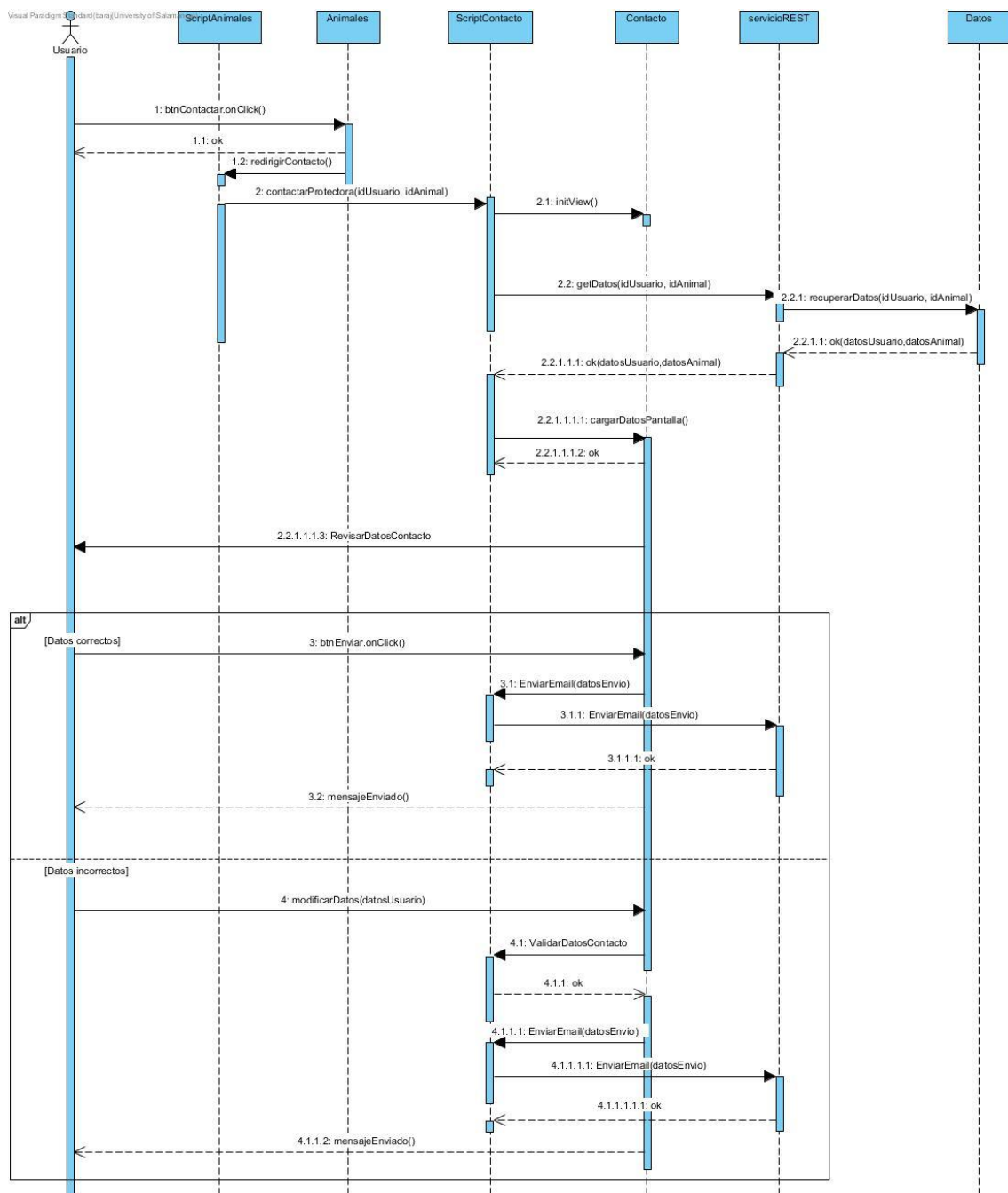


Ilustración 40 - Diagrama de secuencia "Contactar con la protectora"

6.6.6 Modelo de despliegue

La vista de despliegues permite mostrar las diferentes partes de las que se compone el sistema y como se distribuye la funcionalidad dentro del sistema.

Los dos nodos principales que aparecen en este proyecto son la aplicación y el servicio REST. Ambos nodos se comunican uno con el otro a través de Internet. El servicio REST se comunica a su vez con el servidor de base de datos para obtener y manejar la información.

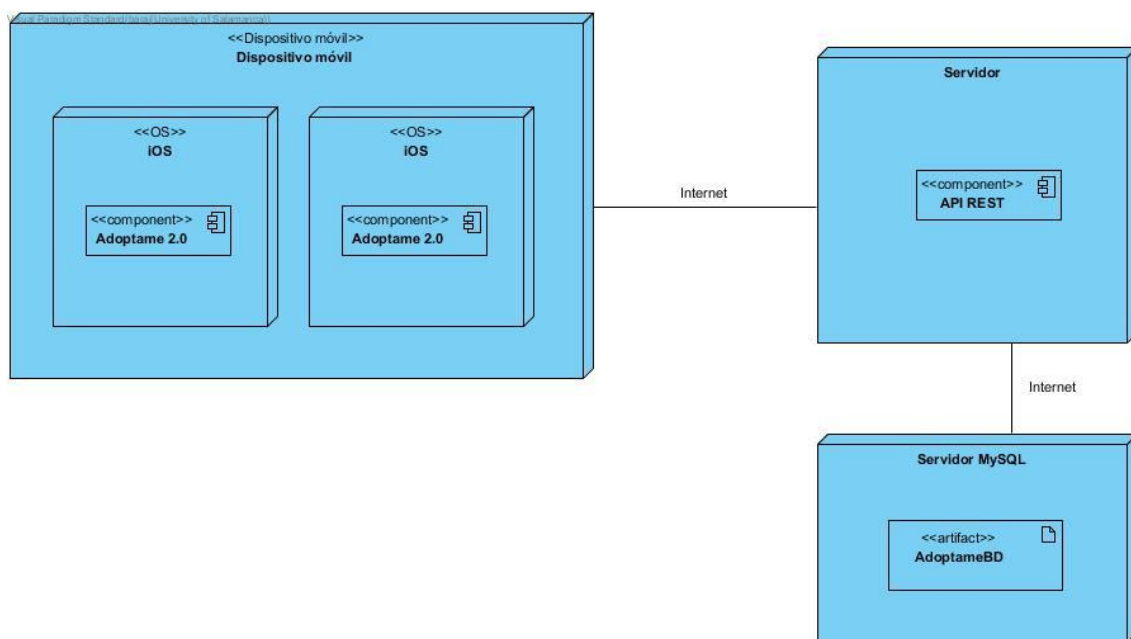


Ilustración 41 - Diagrama de despliegue

6.7 Implementación del sistema

La implementación del sistema se ha realizado siguiendo las especificaciones obtenidas en la fase de Diseño. En esta fase se incluye la implementación de la aplicación móvil y el servicio REST.

Esta fase ha sido la más larga del proyecto puesto que no se conocían la mayoría de las herramientas a utilizar. Además, era necesaria la instalación de todos los entornos de desarrollo y la integración de sus diferentes componentes.

Para el desarrollo de la aplicación ha sido clave toda la documentación aportada por Apache Cordova y Framework 7 a través de su página web, ya que no se encuentra gran volumen de información sobre aplicaciones con estas características.

En cuanto al desarrollo del servicio REST ha sido más sencillo, ya que una vez creada la estructura y haber configurado los accesos, solo faltaba implementar las diferentes funciones. Para crear la estructura inicial se ha utilizado “Composer” que instala automáticamente las librerías que son necesarias para el funcionamiento del servicio.

En la parte del servidor cabe destacar que se ha utilizado la librería “PHPMailer” que permite configurar el servidor para enviar emails de manera sencilla.

Para obtener toda la información técnica del proyecto se debe acceder al “*Anexo V – Documentación técnica*”, donde se detallan los archivos y configuraciones más importantes de todas las partes del proyecto.

6.8 Pruebas

Durante la implementación del sistema se han ido realizando pruebas para comprobar el correcto funcionamiento de los objetivos que se estaban desarrollando.

Utilizando pruebas unitarias se puede asegurar que cada unidad desarrollada funciona de manera correcta y eficiente. Estas pruebas han permitido detectar errores durante el desarrollo, que se han ido corrigiendo y optimizando para que las funcionalidades ofrecidas sean óptimas.

A medida que se iban añadiendo componentes al sistema se han realizado pruebas de integración. Estas pruebas permiten controlar que el sistema sea compacto y que se cumplan los objetivos de las diferentes funcionalidades.

6.9 Funcionalidad de la aplicación

La aplicación está diseñada para dos tipos de usuarios, las protectoras y los adoptantes. Para cada tipo de usuario se presentan funcionalidades adaptadas a sus necesidades.

Ambos tipos de usuarios deberán registrarse en la pantalla inicial de la aplicación. En esta pantalla también deberán loguearse y podrán recuperar su contraseña.

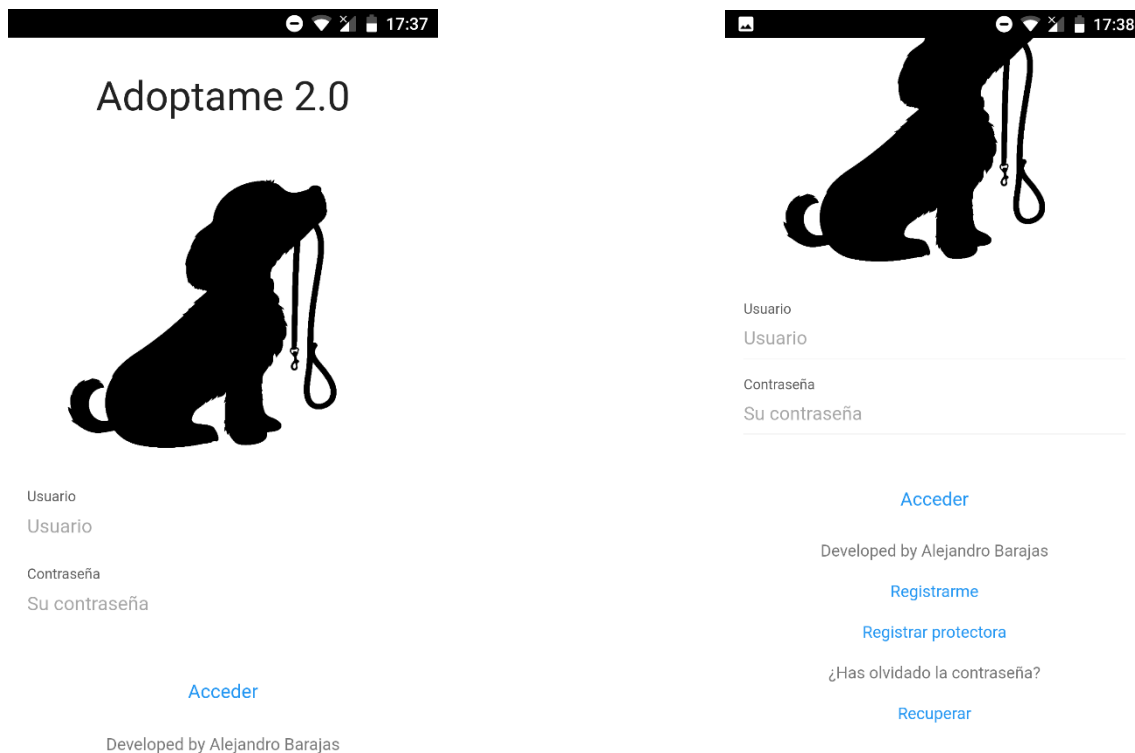


Ilustración 42 - Pantalla inicial

A continuación, se describe brevemente las funcionalidades que podrán realizar cada tipo de usuario. Para obtener más información sobre el funcionamiento de la aplicación, se debe acceder al “Anexo VI – Manual de usuario”.

6.9.1 Manual para adoptantes

6.9.1.1 Registro

Para poder acceder, todos los adoptantes deben registrarse a través del enlace de registro, que se encuentra en la parte inferior de la pantalla de inicio de la aplicación.

A través de la siguiente pantalla se deberán introducir todos los datos de usuario:

The image displays two screenshots of the 'Registro' (Registration) screen in an Android application. Both screenshots show a blue header bar with a back arrow and the title 'Registro'. The status bar at the top indicates the time as 17:38.

Left Screenshot: Shows the initial registration form. The fields are: 'Usuario' (Username) with 'Identificador usuario' (Username identifier) as a hint; 'Contraseña' (Password) with 'Contraseña' as a hint; 'Confirmacion contraseña' (Confirm password) with 'Repita la contraseña' (Repeat the password) as a hint; 'Mostrar contraseñas' (Show passwords) toggle switch; 'Nombre' (Name) with 'Nombre' as a hint; 'Apellidos' (Surname) with 'Apellidos' as a hint; 'E-mail' with 'E-mail' as a hint; 'Teléfono' (Phone) with 'Número de teléfono' (Phone number) as a hint; and 'Colaboro con una protectora' (I collaborate with a protector) toggle switch.

Right Screenshot: Shows the same form, but the 'Confirmacion contraseña' field is highlighted. The 'Mostrar contraseñas' toggle is now turned on. The 'Colaboro con una protectora' toggle is also turned on. A blue 'REGISTRAR' (Register) button is visible at the bottom right.

Ilustración 43 - Registro adoptantes

Si un usuario colabora con una protectora podrá unirse a ella directamente desde el registro, marcando la opción “Colaboro con una protectora”.

6.9.1.2 Búsqueda de animales

La pantalla principal de los adoptantes es la de “Búsqueda de animales”. Esta pantalla muestra la lista de todos los animales pendientes de adoptar:

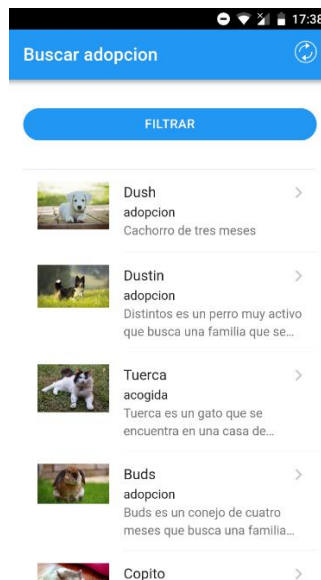


Ilustración 44 - Búsqueda de animales

6.9.1.3 Detalles del animal seleccionado

Desde la pantalla de búsqueda, el adoptante puede pulsar sobre cualquier animal para ver en detalle sus características.

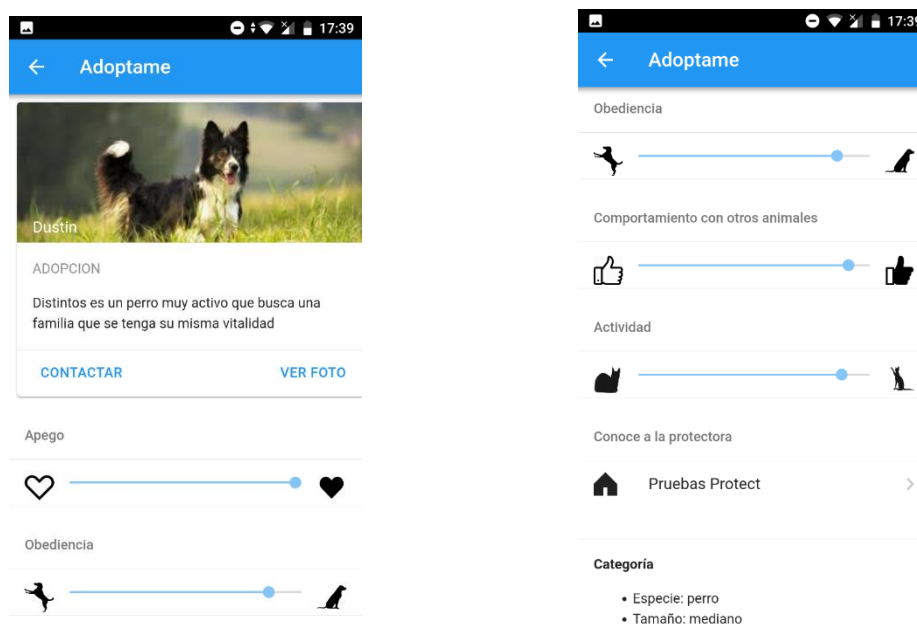
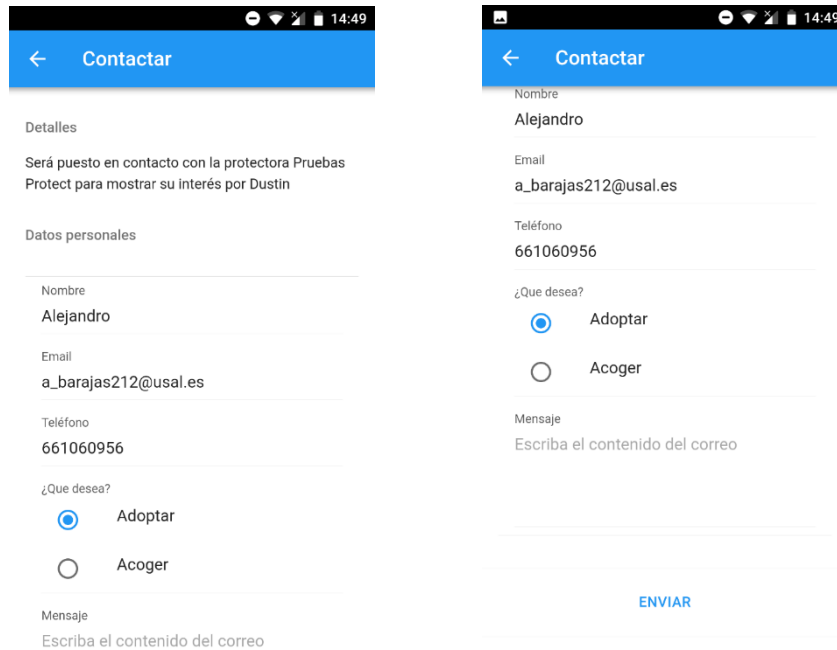


Ilustración 45 - Detalles del animal

6.9.1.4 Contactar

Si un usuario está interesado en un animal debe utilizar la opción “Contactar”. Esta opción enviará un email a la protectora encargada del animal.



← Contactar

Detalles

Será puesto en contacto con la protectora Pruebas Protect para mostrar su interés por Dustin

Datos personales

Nombre
Alejandro

Email
a_barajas212@usal.es

Teléfono
661060956

¿Que desea?

☒ Adoptar

☐ Acoger

Mensaje
Escriba el contenido del correo

ENVIAR

Ilustración 46 - Contactar

6.9.1.5 Ver foto

Esta opción permite ver la foto del animal seleccionado en formato galería

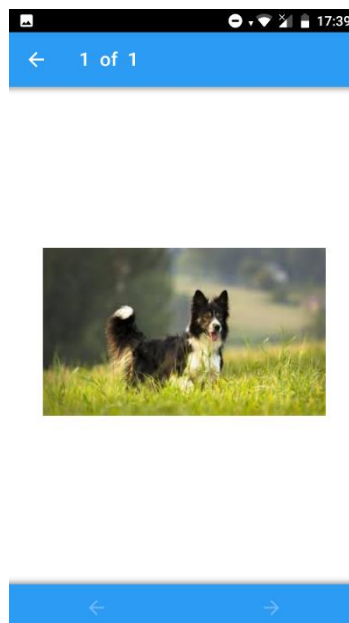


Ilustración 47 - Ver foto

6.9.1.6 Detalles de la protectora

Desde el apartado “Conoce a la protectora” se cargará una página con los detalles de la protectora que se encarga del animal seleccionado.

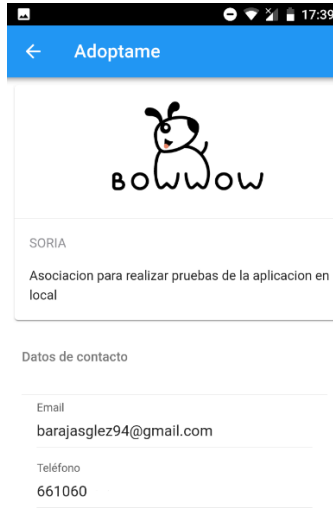


Ilustración 48 - Detalles de la protectora

6.9.1.7 Filtrado en la búsqueda

Para realizar un filtrado de los animales desde la pantalla principal de búsqueda, se debe utilizar el botón “Filtrar” que mostrará la siguiente pantalla:

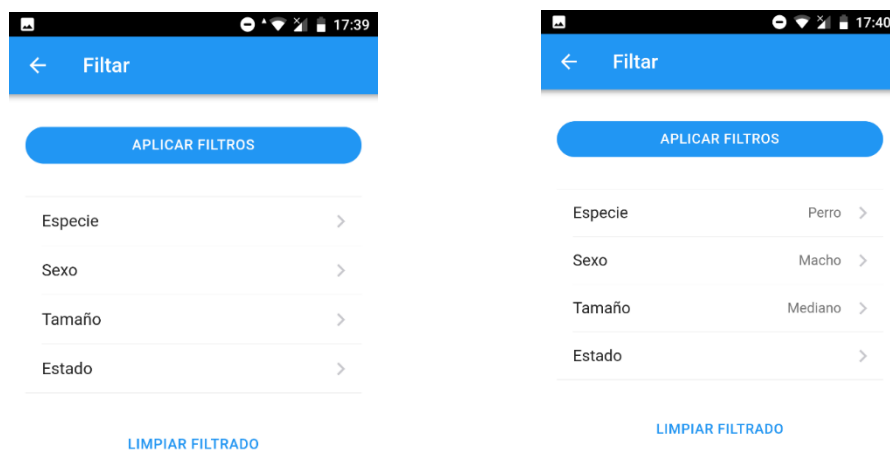


Ilustración 49 - Filtrar

En esta pantalla se podrán seleccionar las características de los animales que los adoptantes estén buscando.

Cuando se apliquen los filtros, en la pantalla de búsqueda aparecerán únicamente los animales que cumplan los requisitos seleccionados.

Estos filtros podrán ser eliminados pulsando sobre el botón “Eliminar filtrado”.

6.9.1.8 Cambiar contraseña

Para realizar las siguientes funcionalidades el usuario debe desplegar la pestaña izquierda de la aplicación desde la pantalla principal de búsqueda.

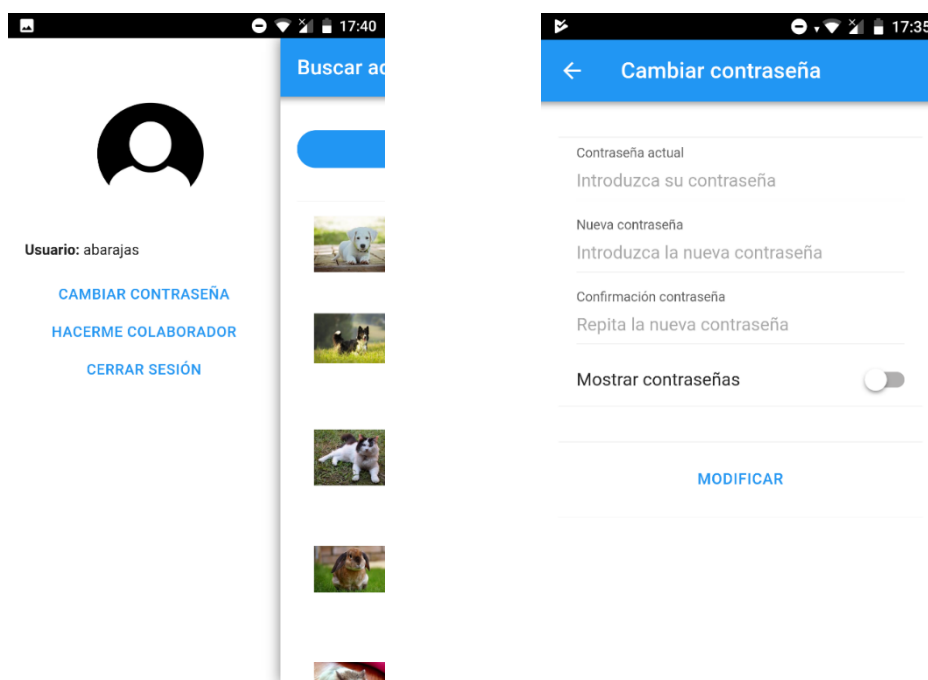


Ilustración 50 - Modificar contraseña

6.9.1.9 Unirse a una protectora

Las protectoras pueden contar con voluntarios para realizar sus tareas. Para que un usuario pueda unirse a una protectora debe recibir el identificador de la protectora y su código.

Las protectoras podrán enviar dicha información a través de un email como el de la siguiente imagen.

Hola!

La protectora **Pruebas Protect** ha decidido enviarte sus datos para que puedas unirte a través de la aplicación y colaborar en sus tareas.

Desde adoptame te damos las gracias por invertir tu tiempo en ayudar a los peludos

Aquí tienes los datos de la protectora: **Pruebas Protect**

Identificador de la protectora:

PruebasProtect

Código:

ybKvL6kU80CJ6IPBw83V18jV5

Gracias!



Fecha y Hora: 11-12-2018 04:57:39

Ilustración 51 - Email "Código de la protectora"

Los usuarios podrán introducir estos datos desde la pestaña desplegable en la opción "Hacerme colaborador".

Una vez los usuarios introduzcan los datos correctamente pasarán a participar en todas las labores de la protectora. Esto implica que la aplicación presentará otras opciones diferentes a las explicadas en el manual de usuario, ya que ahora el usuario pertenece a la parte de las protectoras.

Una vez terminado los trámites para convertirse en colaborador se cerrará la sesión del usuario y será enviado a la pantalla de login. La próxima vez que se acceda a la aplicación tendrá las funcionalidades de la protectora a la que se unió.

← Unirse a una protectora

¿Cómo unirse?

Para unirse a una protectora primero tienes que contactar con ella. Si ellos deciden que puedes unirse para colaborar deberán suministrarte su id de protectora y un código. Estos datos los deberás introducir a continuación y podrás formar ayudar en las tareas de la protectora

Id de la protectora

Codigo protectora
Introduzca el código que le han enviado

UNIRSE

Ilustración 52 - Unirse a una protectora

6.9.2 Manual para protectoras

6.9.2.1 Registro

Todas las protectoras que utilicen la aplicación deben registrarse desde la pantalla de inicio de la aplicación a través del enlace “Registrar protectora”.

Los datos que se introducen en el registro podrán ser modificados posteriormente por si se ha cometido un error o hay algún cambio en la información.

Registro de protectoras

Identificador
Identificador para login

Contraseña
Contraseña

Confirmación contraseña
Repita la contraseña

Mostrar contraseñas

Nombre
Nombre de la protectora

Descripción
Escriba una descripción para la protectora

E-mail
E-mail de la protectora

Teléfono
Teléfono de contacto de la protectora

Registro de protectoras

E-mail
E-mail de la protectora

Teléfono
Teléfono de contacto de la protectora

Pulsar para seleccionar una ciudad

AÑADIR FOTO

REGISTRAR

Ilustración 53 - Registro de protectoras

6.9.2.2 Añadir animal

Esta opción permite añadir animales a la plataforma para que los adoptantes puedan encontrarlos desde su panel de búsqueda.

Todos los animales deberán tener información sobre su especie, sexo, tamaño y estado. Esta estructuración de la información permitirá que los adoptantes encuentren los animales con las características que buscan rápidamente.

Se podrá añadir más información sobre el animal, incluyendo fotos que se podrán cargar desde la cámara del dispositivo o desde la galería de imágenes.

Añadir animal

Sexo

☒ Macho

☐ Hembra

Tamaño

☐ Mini

☐ Pequeño

☒ Mediano

☐ Grande

Estado

☒ Adopción

☐ Urgente

Añadir animal

Comportamiento con otros animales

Actividad

AÑADIR DESDE GALERÍA

AÑADIR

66

6.9.2.3 Editar animal

En el caso de que al publicar un animal se hayan producido errores existe la posibilidad de modificar los datos.

Desde la opción “Editar” se mostrará una lista con todos los animales que tiene pendientes de adoptar la protectora.

Cuando se pulsa sobre uno de ellos se cargarán todos sus datos en una pantalla similar a la de añadir animal.

La protectora modificará los datos que sean necesarios y una vez pulse sobre modificar se publicarán los cambios.

6.9.2.4 Marcar como adoptado

Cuando una protectora consigue una familia para un animal debe eliminarlo de la lista de adopciones para que no aparezca en las búsquedas.

Esto se hace a través de la opción “Marcar como adoptado”, desde la cual se despliega una lista con todos los animales de la protectora.

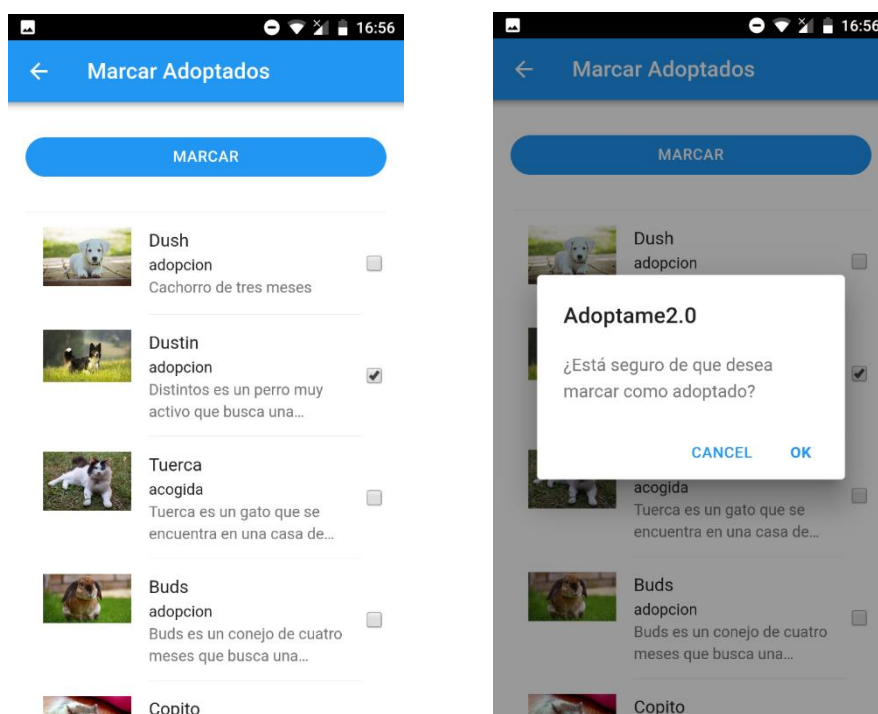


Ilustración 55 - Marcar como adoptado

6.9.2.5 Eliminar

La opción de “Eliminar animal” permite a las protectoras eliminar todos los datos de los animales que seleccionen para eliminar.

Esta opción eliminará las fotos y datos, por lo que no aparecerán datos en estadísticas. Solo se debe usar esta opción en caso de error al añadir algún animal.

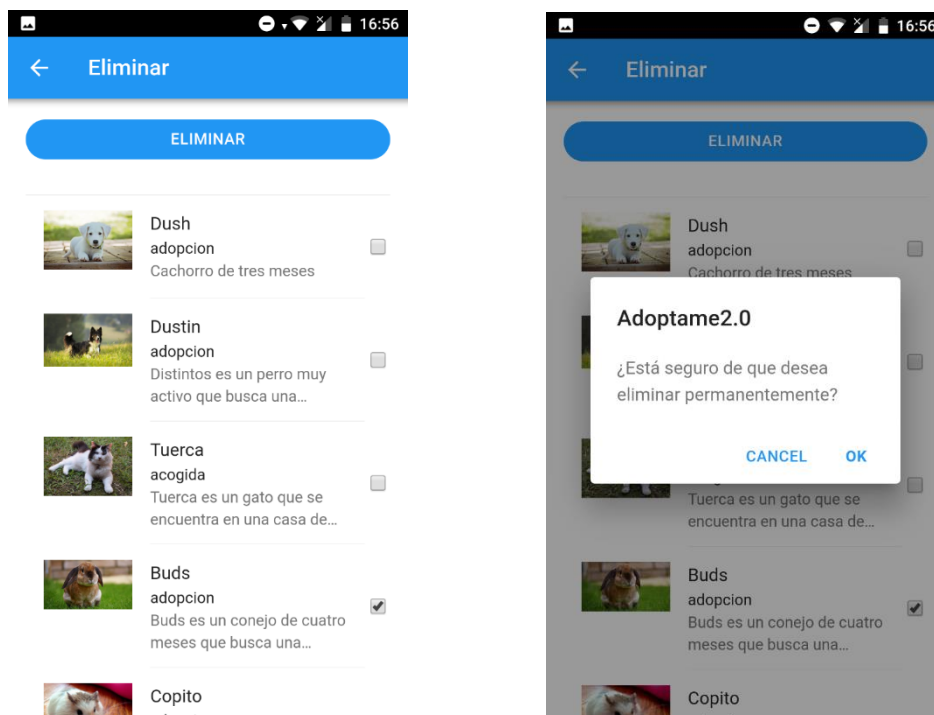


Ilustración 56 - Eliminar

6.9.2.6 Ver código

Las protectoras pueden incluir colaboradores para realizar las tareas. En la aplicación los usuarios que se unan a las protectoras podrán realizar las mismas tareas por ellas para así repartir el trabajo.

A través de la opción “Ver código” la protectora puede ver su identificador y código de protectora que debe suministrar a los usuarios para unirse. Este código se modificará cada vez que se una un usuario, además de poder modificarse desde este apartado.

Esta pantalla tiene habilitadas la opción de copiar, por tanto, será sencillo coger el código y enviarlo.

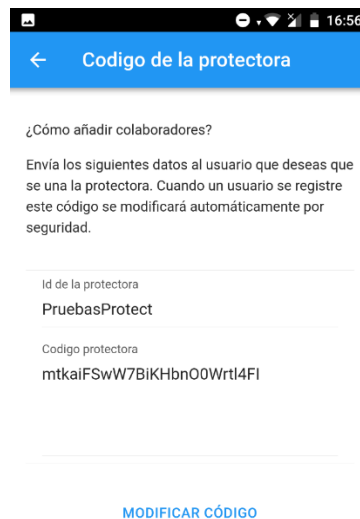


Ilustración 57 - Código de la protectora

6.9.2.7 Enviar código por email

Para hacer más sencillo el envío de los datos de la protectora para que los usuarios puedan unirse se ha implementado la opción de enviarlos a través de email.

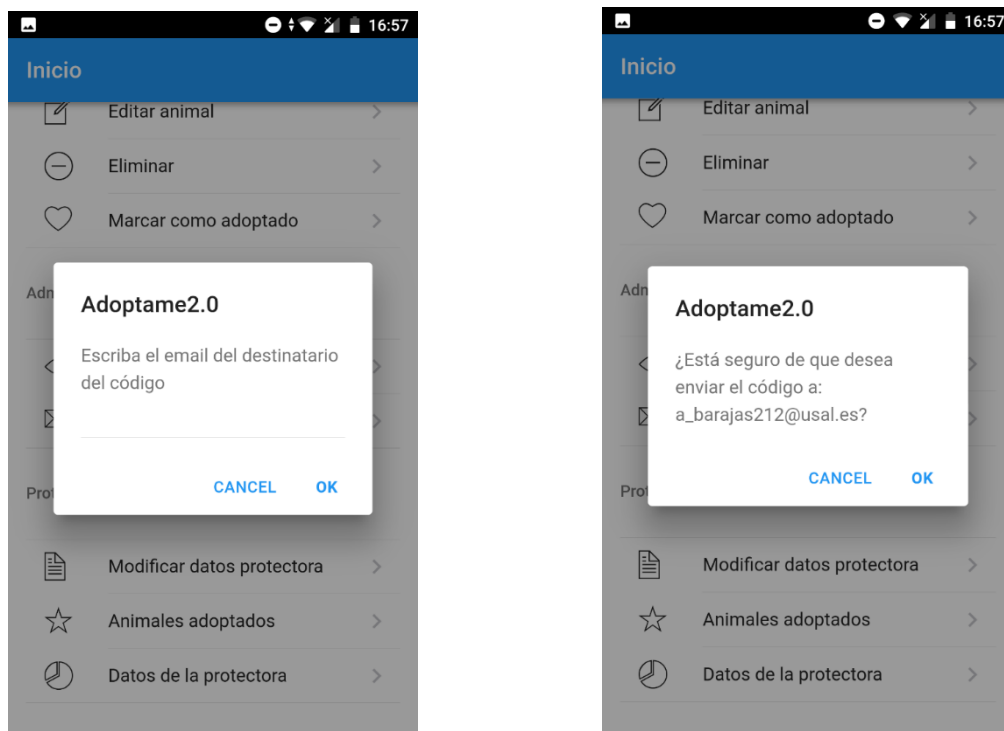


Ilustración 58 - Enviar código por email

6.9.2.8 Modificar datos de la protectora

Esta opción permite modificar los datos relacionados con la protectora que han sido introducidos en el registro.

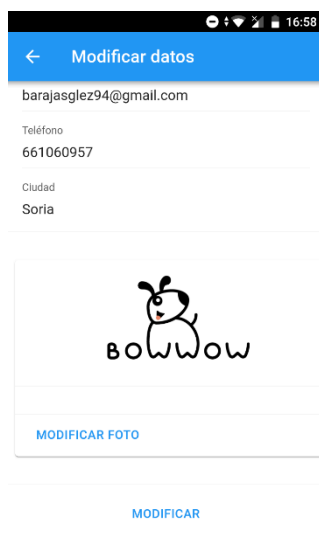


Ilustración 59 - Modificar datos de la protectora

6.9.2.9 Animales adoptados

Dentro de “Animales adoptados” encontraremos una lista con todos los animales que han sido marcados como adoptados por la protectora.

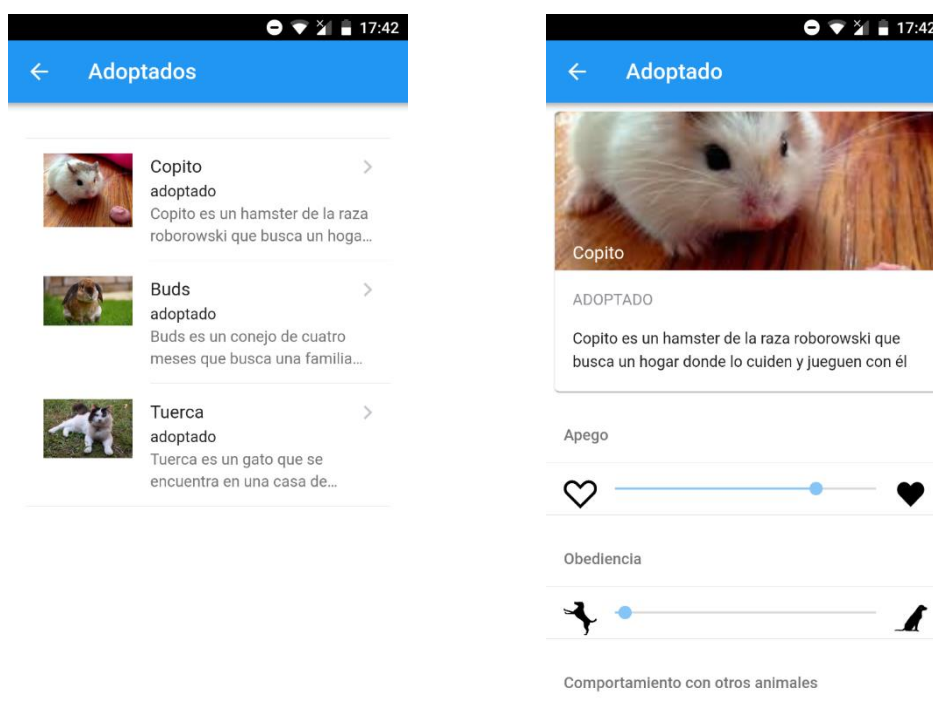


Ilustración 60 - Animales adoptados

6.9.2.10 Datos de la protectora

En el apartado “Datos de la protectora” las protectoras podrán ver gráficos relacionados con sus animales. El primero de ellos muestra la información del número de animales que están pendientes para adoptar y su especie.



Ilustración 61 - Datos de la protectora

6.9.2.11 Cambiar contraseña

Para realizar las siguientes funcionalidades la protectora debe desplegar la pestaña izquierda de la aplicación desde la pantalla principal de búsqueda.

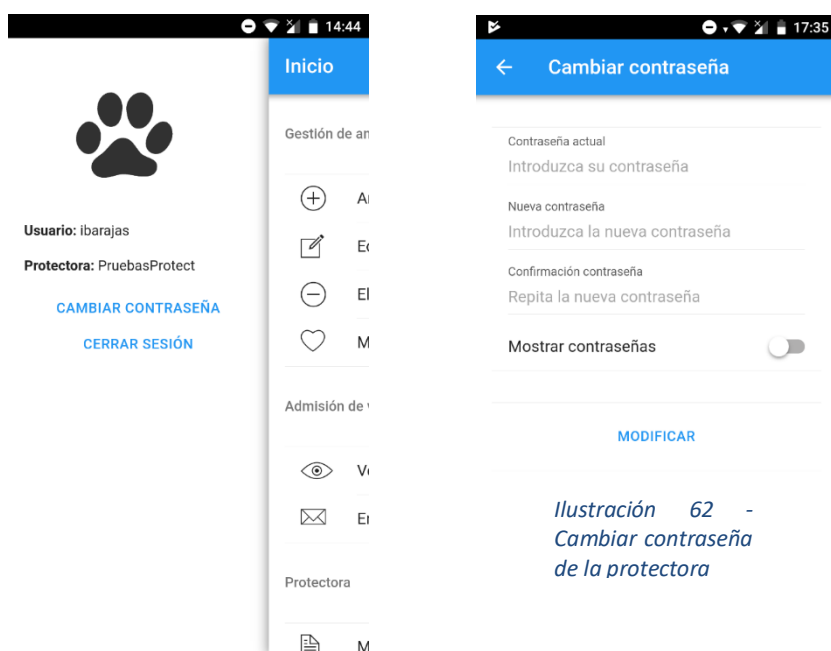


Ilustración 62 - Cambiar contraseña de la protectora

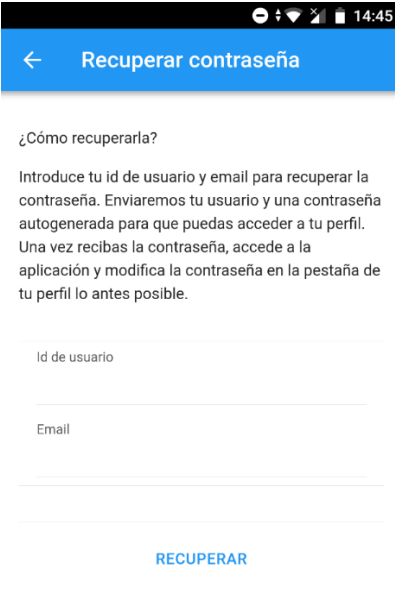
6.9.3 Cerrar sesión

Tanto protectoras como adoptantes pueden cerrar sesión para volver a la pantalla de login.

La aplicación mostrará un pop up antes de salir y los datos de sesión serán eliminados, por lo que la próxima vez que se acceda a la aplicación, no se logueará automáticamente al usuario.

6.9.4 Recuperar contraseña

Esta opción permite tanto a protectoras como adoptantes recuperar su contraseña. Se deberá introducir el id de usuario y el email para que la aplicación envíe un email con una contraseña autogenerada que servirá al usuario para acceder y cambiar la contraseña.



14:45

← Recuperar contraseña

¿Cómo recuperarla?

Introduce tu id de usuario y email para recuperar la contraseña. Enviaremos tu usuario y una contraseña autogenerada para que puedas acceder a tu perfil. Una vez recibas la contraseña, accede a la aplicación y modifica la contraseña en la pestaña de tu perfil lo antes posible.

Id de usuario

Email

RECUPERAR

Ilustración 63 - Recuperar contraseña

7. Conclusiones y Líneas de trabajo futuras

7.1 Conclusiones

Con la finalización y entrega de este proyecto se puede decir que se han cumplido los objetivos principales que se fijaron al inicio.

Entre los objetivos principales se pretendía conseguir una aplicación móvil multiplataforma que permitiese a las protectoras publicar sus animales. Además, se pensó en los adoptantes con el fin de que encontraran un animal con las características que buscan.

La aplicación implementada cumple con los requisitos de funcionalidad y usabilidad, lo que me ha permitido entender dichos conceptos para conseguir una interfaz sencilla y amigable que determinará el éxito de la aplicación.

A través de la implementación del proyecto he obtenido experiencia en el desarrollo de aplicaciones multiplataforma, servicios REST y el uso de herramientas Open Source, tanto del lado de la aplicación como del servidor.

Por último, me gustaría destacar el cumplimiento de los objetivos personales, tanto el poder crear un proyecto software completo, como el haberme enfrentado a tecnologías que no conocía. Esto me permitirá enfrentarme a nuevos proyectos con la tranquilidad y seguridad de haber realizado un proyecto completo.

7.2 Líneas de trabajo futuras

El proyecto realizado tiene muchas líneas de trabajo futuras en las que se podrá ir añadiendo funcionalidad o complementando la ya existente.

A continuación, se plantean diversas ampliaciones y modificaciones:

- **Donaciones:** La primera ampliación que se plantea es la de poder realizar donaciones a las protectoras. Esta funcionalidad se incluirá para los adoptantes, que podrán donar a cualquier protectora que tenga activada esta opción. Las donaciones se podrían realizarse a través de “Paypal” para que los pagos se realicen de forma segura. Las protectoras deberán incluir sus datos bancarios al realizar el registro para que esta opción esté activada.

- **Agenda de colaboradores:** La segunda idea es incluir una opción en el menú de la protectora en la que aparezca una agenda con los datos de los colaboradores de la protectora. De esta manera todos los colaboradores tendrán los datos para contactar unos con otros si fuese necesario.
- **Fotos de animales:** La funcionalidad que presenta en este momento la aplicación en cuanto a las imágenes de los animales es que solo se permite añadir una foto de cada uno de ellos. Una ampliación sencilla sería que se pudieran añadir más fotos de cada animal. Si el uso de la aplicación creciera mucho habría que tener en cuenta en ampliar y redistribuir el espacio de almacenamiento en el servidor para que la carga de imágenes no se ralentice.
- **Marcado de animales:** Como ampliación para los adoptantes se podría incluir el marcado de animales que le gustan. Debería crearse una opción dentro de su menú en la que se mostrara la lista de los animales que ha marcado. Esto permitiría a los adoptantes ver de manera resumida todos los animales en los que está interesado.
- **Notificaciones:** La comunicación entre adoptantes y protectoras en la aplicación se realiza a través de email. Una modificación importante sería que estos mensajes se guardaran dentro de la aplicación y se mostraran en un apartado de la protectora. Esta ampliación es relativamente sencilla, ya que los datos de la comunicación ya se utilizan en la aplicación.
- **Tareas:** En el menú de las protectoras se podría incluir una opción en la que se programen tareas, como por ejemplo pasear a los perros. Los colaboradores podrían escoger las tareas en las que puedan participar y así repartir el trabajo. Esta ampliación puede ser la más costosa, ya que se deben incluir todas las opciones para el manejo de las tareas, además de la relación entre tareas y colaboradores.

8. Bibliografía

- Datos de adopción y abandono de animales
<https://www.fundacion-affinity.org/estudios-abandono-y-adopcion>
- Arquitectura cliente-servidor
<https://es.wikipedia.org/wiki/Cliente-servidor>
- Detalles de Apache Cordova
https://es.wikipedia.org/wiki/Apache_Cordova
- Definición de REST
<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- Modelo vista-controlador
<https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- Metodología de Durán y Bernárdez
<http://www.lsi.us.es/docs/informes/lsi-2000-10.pdf>
- Visual Studio
<https://visualstudio.microsoft.com/es/vs/features/cordova/>
- Framework 7
<https://framework7.io/>
- JSON
<https://www.json.org/>
- HTML5
<https://codigofacilito.com/articulos/que-es-html>
- Slim 3
<https://www.slimframework.com/>

- Detalles de Sublime Text 3
<https://www.genbeta.com/herramientas/despues-de-un-ano-de-esperalllega-sublime-text-3-0-para-windows-linux-y-macos>
- Descripción de la herramienta Postman
<https://www.paradigmadigital.com/dev/postman-gestiona-construyetus-apis-rapidamente/>
- Página oficial de Postman
<https://www.getpostman.com/>
- Biblioteca PHP mailer
<https://github.com/PHPMailer/PHPMailer>
- Alojamiento en Github del Proyecto
<https://github.com/abarajas212/Adoptame-2.0>