

Grado en Ingeniería Informática de Gestión y Sistemas de Información

APP PARA PROTECTORA DE ANIMALES

Memoria

ISMAEL MARÍN GALLEGO
TUTORA: CATALINA JUAN NADAL

CURSO ACADÉMICO 2022-2023

Agradecimientos

Quiero agradecer a mis padres, ya que sin ellos no habría llegado a donde me encuentro ahora ni a ser quien soy, y a mi pareja, la cual ha sido uno de mis mayores apoyos durante este tiempo y me ha ayudado a avanzar en las diferentes pruebas de este producto.

Abstract

This project aims to design and develop a mobile and web application aimed at the adoption service of an Animal Protector. The application must be able to allow users to view the wide range of animals that are available, see information about them and schedule previous appointments. It must also allow the entity to publish, modify or delete announcements or events, animals and manage the previous appointments of the users.

Resum

Aquest projecte té com a objectiu dissenyar i desenvolupar un aplicatiu mòbil i web dirigit al servei d'adopció d'una Protectora d'Animals. L'aplicació ha de poder permetre als usuaris visualitzar l'ampli ventall d'animals que es disposa, veure informació sobre aquests i agenda cites prèvies. També ha de permetre a l'entitat publicar, modificar o eliminar anuncis o esdeveniments, animals y administrar les cites prèvies dels usuaris.

Resumen

Este proyecto tiene como objetivo diseñar y desarrollar una aplicación móvil y web dirigida al servicio de adopción de una Protectora de Animales. La aplicación debe poder permitir a los usuarios visualizar el amplio abanico de animales que se dispone, ver información sobre éstos y agendar citas previas. También debe permitir a la entidad el publicar, modificar o eliminar anuncios o eventos, animales y administrar las citas previas de los usuarios.

Índice

| | |
|--|-----|
| Índice de figuras..... | V |
| Índice de tablas | VII |
| Glosario de términos | IX |
| 1. Objeto del proyecto | 1 |
| 2. Estudio previo | 3 |
| 2.1. Contexto..... | 3 |
| 2.1.1. Perreras Municipales | 3 |
| 2.1.2. Protectora de Animales | 3 |
| 2.2. Antecedentes | 4 |
| 2.2.1. AdoptaMe..... | 4 |
| 2.2.2. Klygo..... | 5 |
| 2.3. Conclusiones..... | 5 |
| 3. Objetivos y alcance | 7 |
| 4. Metodología | 9 |
| 4.1. Etapa de Inicio | 9 |
| 4.2. Etapa de Planificación y estimación | 9 |
| 4.3. Etapa de Implementación..... | 10 |
| 4.4. Etapa de Revisión y retrospectiva..... | 10 |
| 4.5. Etapa de Lanzamiento..... | 10 |
| 5. Definición de requerimientos funcionales y tecnológicos | 11 |
| 5.1. Plataforma APP..... | 11 |
| 5.1.1. Requerimientos funcionales | 11 |
| 5.1.2. Requerimientos tecnológicos | 11 |
| 5.2. Plataforma Web | 12 |
| 5.2.1. Requerimientos funcionales | 12 |
| 5.2.2. Requerimientos tecnológicos | 12 |
| 5.3. Justificación de las tecnologías escogidas | 12 |
| 5.3.1. Plataforma de Backend | 12 |
| 5.3.2. Plataforma Web..... | 13 |
| 5.3.3. Plataforma APP..... | 13 |

| | |
|--|----|
| 5.3.4. Posibles alternativas | 14 |
| 6. Desarrollo del proyecto | 17 |
| 6.1. Requerimientos para el desarrollo de las plataformas | 17 |
| 6.1.1. Requerimientos Plataforma Backend..... | 17 |
| 6.1.2. Requerimientos Plataforma Web | 17 |
| 6.1.3. Requerimientos Plataforma APP | 18 |
| 6.2. Base de datos | 19 |
| 6.2.1. Estructura de la Base de Datos..... | 19 |
| 6.3. Diseño de interfaces..... | 22 |
| 6.3.1. Interfaz de plataforma APP..... | 23 |
| 6.3.2. Interfaz de plataforma Web..... | 24 |
| 6.4. Definición de casos de uso | 25 |
| 6.4.1. Registrar un usuario | 25 |
| 6.4.2. Iniciar sesión | 26 |
| 6.4.3. Cerrar sesión | 27 |
| 6.4.4. Visualizar un animal | 28 |
| 6.4.5. Visualizar un anuncio..... | 29 |
| 6.5. Problemáticas encontradas durante el desarrollo..... | 29 |
| 6.5.1. Problemas con versiones desfasadas de módulos | 29 |
| 6.5.2. Problemas de conexión con la Base de Datos en MongoDB Atlas | 30 |
| 6.6. Entrevistas con el cliente | 30 |
| 6.6.1. Primera entrevista con el cliente | 30 |
| 6.6.2. Segunda entrevista con el cliente y usuarios..... | 30 |
| 6.6.3. Última entrevista con el cliente y usuarios | 31 |
| 7. Análisis y conclusiones | 33 |
| 7.1. Análisis del producto | 33 |
| 7.2. Conclusiones del proyecto..... | 33 |
| 8. Posibles ampliaciones..... | 35 |
| 8.1. Filtrado de listas..... | 35 |
| 8.2. Etiquetado y categorización..... | 35 |
| 8.3. Generación de citas automáticas..... | 35 |
| 8.4. Autenticación mediante redes sociales | 35 |

7. Bibliografía 37

Índice de figuras

| | |
|--|----|
| Figura 1.1 Porcentaje de hogares equipados con teléfono móvil España de 2005 a 2021..... | 1 |
| Figura 2.2.1 Pantallas de app AdoptaMe..... | 4 |
| Figura 2.2.2 Pantallas de app Klygo..... | 5 |
| Figura 6.2 Selección de tipo, proveedor y región del Clúster..... | 19 |
| Figura 6.2.1 Diagrama de Base de Datos..... | 20 |
| Figura 6.3.1 Galería de secciones Plataforma APP..... | 23 |
| Figura 6.3.2 Galería de secciones Plataforma Web..... | 24 |

Índice de tablas

| | |
|---|----|
| Tabla 6.4.1 Caso de uso: Registrar un usuario..... | 25 |
| Tabla 6.4.2 Caso de uso: Iniciar sesión..... | 26 |
| Tabla 6.4.3 Caso de uso: Cerrar sesión..... | 27 |
| Tabla 6.4.4 Caso de uso: Visualizar un animal..... | 28 |
| Tabla 6.4.5 Caso de uso: Visualizar un anuncio..... | 29 |

Glosario de términos

| | |
|---------------------|--|
| TFG | Trabajo de Final de Grado |
| JSX | JavaScript eXtension |
| <i>React JS</i> | Biblioteca de JavaScript |
| <i>React Native</i> | Framework de interfaces |
| <i>SCRUM</i> | Metodología basada en sprints por tarea |
| <i>Framework</i> | Conjunto de conceptos, prácticas y criterios para resolver problemáticas |
| <i>Stakeholder</i> | Individuo u organización involucrada en el proyecto |
| <i>Backlog</i> | Lista de tareas pendientes en cada sprint |
| APP | Application |
| <i>Gantt</i> | Herramienta gráfica para exponer tareas y su estimación de tiempo |
| <i>Hosting</i> | Servicio de alojamiento web |
| <i>Backend</i> | Servicio que brinda un servidor para la entrega de datos al cliente. |
| DOM | Document Object Model |
| <i>TypeScript</i> | Lenguaje de programación en superconjunto de JavaScript |
| <i>NativeScript</i> | Lenguaje de programación en superconjunto de JavaScript |
| SDK | Software Development Kit |
| SEO | Search Engine Optimization |
| CSS | Cascading Style Sheets |
| HTML | HyperText Markup Language |

1. Objeto del proyecto

Con el aumento de dispositivos móviles en la sociedad cada día más presente en la vida de las personas, volviéndose casi esenciales, se ha creado una tendencia alrededor del desarrollo de aplicaciones móviles para abastecer la demanda del acceso rápido, sencillo y visual al contenido y la información.

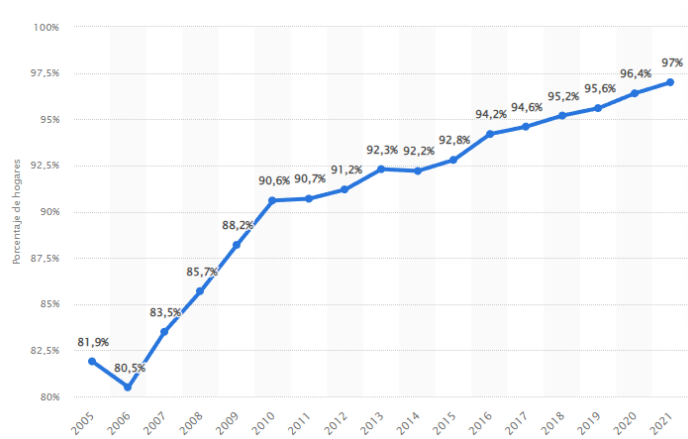


Figura 1.1 Porcentaje de hogares equipados con teléfono móvil en España de 2005 a 2021 - Fuente: Statista.com

Por lo tanto, podemos concluir que se ha creado una necesidad por parte de las entidades como empresas, asociaciones... de dar visibilidad a sus servicios a través de estas plataformas.

Así pues, partiendo de esta idea, se puede potenciar el alcance de un servicio como el de una protectora de animales, desarrollando una aplicación que facilite la comunicación entre los posibles usuarios y la entidad, además de contar con un portal web que posibilite a la entidad la administración y gestión del contenido en publicaciones y/o anuncios.

2. Estudio previo

2.1. Contexto

Primeramente, es importante explicar ciertos conceptos que son causantes de confusión ya que son términos que usualmente se utilizan de forma errónea y por lo tanto es esencial diferenciarlos. Por un lado, existen las perreras municipales y por otro la protectora de animales. Ambas comparten aspectos en común pero su finalidad, así como su forma de funcionar son bastante diferentes [1].

2.1.1. Perreras Municipales

Las Perreras Municipales por lo general suelen tratarse de empresas privadas con ánimo de lucro que están financiadas por la administración o administraciones públicas de la zona donde residen. Su función es la prestación de un servicio que se basa en la recogida de animales rechazados o abandonados, pasando por el alojamiento de éstos en centros propios y finalmente dándolos en adopción en el mejor de los casos, o sacrificándolos si se cumple el plazo máximo de estadía [2].

Este plazo máximo está dictado por la administración pertinente de la zona y permite a las perreras municipales el sacrificar a un animal si no se consigue dar en adopción una vez alcanzado ese plazo, consiguiendo de esta forma un equilibrio entre el espacio limitado del que se dispone y el gasto en alimentación y material.

2.1.2. Protectora de Animales

Una Protectora de Animales, a diferencia de una Perrera Municipal, es una organización sin ánimo de lucro que normalmente está financiada por capital privado y particular, gestionadas por sus propios socios y con personal cualificado y con vocación para dar un bienestar a los animales.

Las Protectoras no suelen prestar servicio de recogida de animales, dado que de eso se encarga la perrera del municipio, así pues, esta entidad acoge animales directamente de personas particulares o de empresas.

Otra de las diferencias claves es el camino que sigue el animal una vez entra en los registros, puesto que las Protectoras de Animales normalmente no disponen de un centro exclusivo para el alojamiento de éstos, teniendo que depender de terceras personas u organizaciones para los cuidados y hospedajes durante su transición. En este punto, se termina desembocando en dos diferentes casos:

1. El animal puede llegar a entrar en adopción, lo cual significaría la salida de la posada donde se encuentra y preparación del entorno al que se dispone a entrar, para su posterior acogida definitiva.

2. El animal no es acogido durante su transición, por lo que se seguirá hospedando hasta el final de sus días, pero sobre ningún caso se ofrecerá a sacrificio.

En el segundo caso podemos ver otra de las grandes diferencias, puesto que el animal nunca será sacrificado y la Protectora siempre abogará por el bienestar de éste, sin importar las circunstancias [2].

2.2. Antecedentes

En el territorio español, existen diversas aplicaciones que parten de la misma idea de este proyecto, facilitando la adopción del amplio abanico de animales y maximizando el alcance. Aun así, cada una guarda sus propias cualidades, diferenciándose así las unas de las otras.

2.2.1. AdoptaMe

Esta aplicación española fue lanzada el 8 de abril de 2021 y está disponible únicamente en plataforma Android. Cuenta actualmente con más de 50.000 descargas y ha aparecido en varios medios de difusión españoles como pueden ser Los40[3] y Xataka Android [4].

Esta aplicación creada por David un, hasta ese momento, estudiante de ingeniería que culminó su trabajo de fin de grado con la creación de AdoptaMe, aportando su granito de arena a esta finalidad, tal y como mencionó en una entrevista:

"Está hecha con la idea de generar un punto único de adopciones en la sociedad (como un trivago para adopción animal), así será mucho más fácil dar visibilidad y encontrar un nuevo hogar para muchos animales"

Esta aplicación está dirigida tanto a usuarios que buscan adoptar como a protectoras que publicitan sus animales disponibles para la acogida.

A continuación, una muestra de sus pantallas principales:

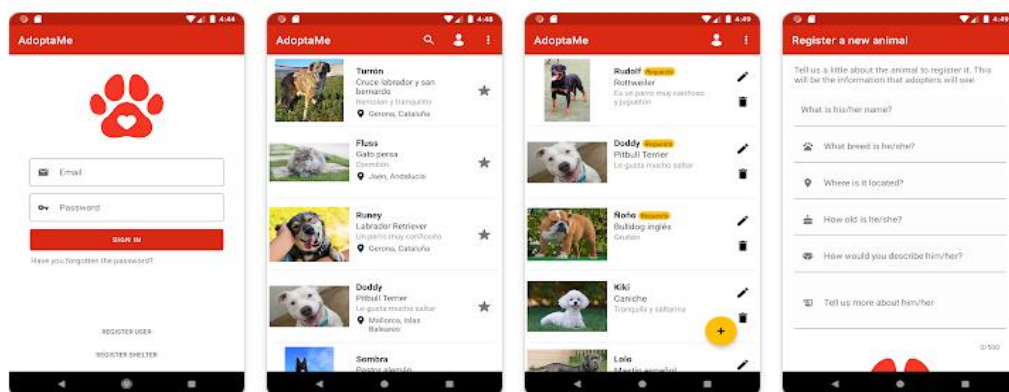


Figura 2.2.1 Pantallas de app AdoptaMe - Fuente: play.google.com

2.2.2. Klygo

Esta aplicación fue lanzada en febrero de 2021 y está disponible tanto para plataforma Android como iOS. Cuenta actualmente con más de 10.000 descargas en Android. Al igual que el caso anterior, ha salido en diferentes medios como AppMarketingNews [5] y WWWhatsNew [6].

Esta aplicación fue creada por la pareja Yasmin y Kilian, cuyo objetivo era dar lugar a una plataforma fácil de manejar y concisa para la adopción únicamente de gatos y perros, no solo por parte de entidades como en el caso anterior, sino también por parte de usuarios.

A diferencia del caso anterior, añade una funcionalidad de búsqueda por mapa, lo que facilita la localización y permite marcar una zona en concreto, además de contar con un chat para facilitar la comunicación.

A continuación, una muestra de sus pantallas principales:

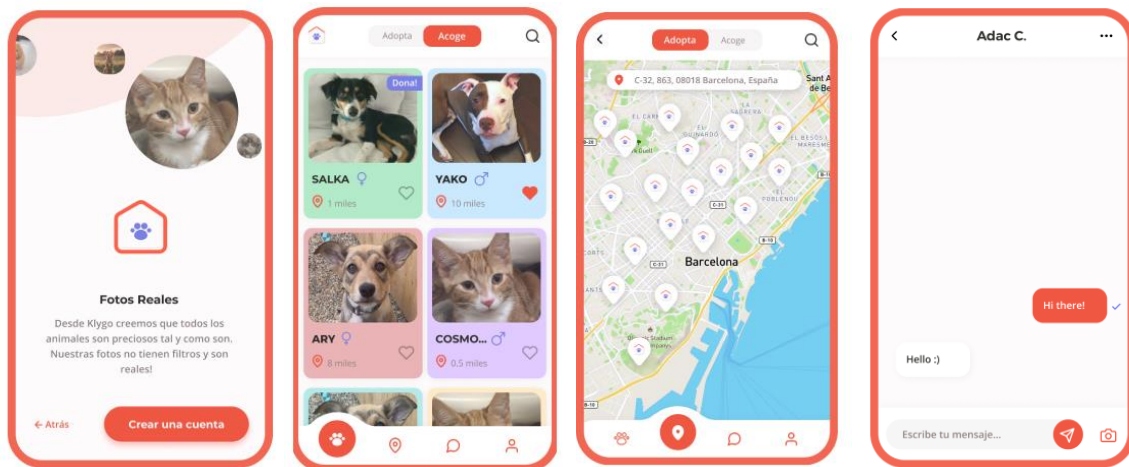


Figura 2.2.2 Pantallas de app Klygo - Fuente: play.google.com

2.3. Conclusiones

En base a estas dos aplicaciones mencionadas, se obtienen varias conclusiones respecto a las funcionalidades que entregan cada una y que se pueden extrapolar a este proyecto.

En el caso de AdoptaMe, como punto diferenciador se observa que permite añadir cualquier tipo de animal en la plataforma y la publicación está limitada únicamente a protectoras de animales registradas previamente, manteniendo siempre datos reales y no posibles manipulaciones.

Por otro lado, la app Klygo muestra un diseño más moderno y adaptado al periodo actual, además de añadir ciertas funcionalidades interesantes como un chat y la búsqueda por mapa, pero, a diferencia de la anterior, permite publicar tanto a usuarios particulares como a entidades de protección, y como punto negativo se toma en consideración la limitación de añadir únicamente gatos y perros, reduciendo así el mercado potencial de ésta.

Tomando estas dos aplicaciones como referencia, este proyecto une varias de las ventajas entre ellas, como puede ser el permitir añadir cualquier tipo de animales además de la limitación de publicaciones exclusivamente por entidades (como en el caso de AdoptaMe), y una interfaz más fresca y fluida (como en el caso de Klygo).

3. Objetivos y alcance

Este proyecto aspira a conseguir una aplicación funcional que permita el intercambio de información entre la protectora de animales y el usuario final, proporcionando información útil y necesaria sobre las mascotas disponibles, como edad, sexo, raza, entre más cosas, así como también facilitando la posibilidad de concretar una cita entre usuario y protectora. Se pretende también añadir complementos como favoritos o herramientas de filtrado que facilite la búsqueda y proporcione dinamismo al mismo tiempo.

Para cumplir con este fin, se desarrollan dos interfaces diferentes; una aplicación destinada a la visualización de las diferentes publicaciones, así como la interacción por parte del usuario mediante dispositivos móviles y un portal web destinado a la administración del contenido por parte de la protectora, además de servir como portal de visualización de publicaciones, al igual que en su versión móvil.

Como objetivos por parte de la Protectora de Animales durante este proyecto, se han especificado los siguientes:

- Cubrir la necesidad de la protectora a la hora de publicar y compartir información sobre animales o anuncios.
- Mejorar y agilizar el sistema de citas y horas concertadas actual mediante el registro en la aplicación o web.
- Disponer de un portal web que permita la visualización por parte del usuario de la información basada en la aplicación.
- Disponer de un portal web que permita a la entidad administrar fácilmente el contenido y citas.

Como objetivos desde el punto de vista de un usuario durante este proyecto, se han especificado los siguientes:

- Disponer de un acceso rápido al contenido con una fácil visualización y entendimiento.
- Disponer de un sistema ágil para solicitar cita previa sobre un animal seleccionado.
- Disponer de la posibilidad de añadir un elemento como favorito para su posterior visualización.
- Obtener un control total sobre la cuenta del usuario para permitir su modificación o eliminación.

4. Metodología

Con la finalidad de llevar este proyecto a cabo, se recurre a la metodología AGILE, centrándose en la técnica de SCRUM, que mediante un sistema de sprints permiten visualizar fácilmente nuestro progreso a lo largo del tiempo.

Esta metodología se divide en 5 etapas.

4.1. Etapa de Inicio

En esta primera etapa, se deben identificar las necesidades básicas de cada sprint que ayudan a completar los objetivos generales y específicos marcados.

Las preguntas que se deben realizar son:

- **¿Qué quiero conseguir?:** Se debe especificar cual será el producto final, aquello que se quiere conseguir una vez finalizado todo el proceso.
- **¿Cómo lo quiero conseguir?:** ¿Qué métodos se utilizarán? ¿Cuáles son las herramientas usadas para tal finalidad?...
- **¿Cuándo lo quiero conseguir?:** Se debe especificar una franja de tiempo aceptable para finalizar el proyecto con éxito.

En base a las respuestas dadas, se logra crear una visión del proyecto, identificando quiénes son los Scrum Master y los Stakeholders, saber cómo se desarrolla y obtener la planificación del lanzamiento.

4.2. Etapa de Planificación y estimación

Esta segunda etapa se centrará en crear, estimar y establecer las historias de usuario, crear y repartir las tareas necesarias, así como planificar las iteraciones de éstas.

Lo más importante en este paso será tener clara la planificación de cada sprint para conseguir obtener los plazos y objetivos a tiempo, evitando comprometer el resto del proyecto.

Desde la perspectiva del Master Scrum, se deben crear las correspondientes tareas del proyecto y permitir la asignación, sin solapamiento, a los diferentes equipos, añadiendo a su vez una lista que permita clasificarlas con un orden de prioridad.

4.3. Etapa de Implementación

Esta tercera etapa tiene como finalidad el desarrollo del proyecto, incluyendo una parte fundamental en esta metodología; la comunicación.

A medida que cada grupo de trabajo complete los sprints, se deben crear entregables que permitan avanzar en la implementación, realizar una reunión diaria denominada “daily stand-up”, que vendrían siendo pequeñas reuniones en las que cada equipo debe dar sus actualizaciones y problemas que hayan surgido durante el día y la revisión de la lista de tareas aún pendientes, denominada Backlog, para tener siempre una visualización clara de qué se ha realizado y qué queda por hacer.

A estas alturas del desarrollo no se debe realizar ningún cambio, ya que podría comprometer los plazos de entrega del proyecto, aunque existe una pequeña excepción, y es que, si el cambio es clave para el éxito de éste, si se podrá realizar, siempre y cuando se haya consultado con el resto de equipos y Stakeholders.

4.4. Etapa de Revisión y retrospectiva

Esta cuarta y penúltima etapa se enfoca en la revisión del proceso de implementación una vez que haya finalizado, realizando una autocrítica y evaluación interna de los grupos.

Durante las reuniones de esta etapa, se deben obtener opiniones constructivas y soluciones viables, teniendo en consideración la demostración y validez de cada sprint realizado y la retrospectiva que se tiene de éste.

4.5. Etapa de Lanzamiento

Esta quinta y última etapa se centra en el final del proyecto y la entrega del producto, añadiendo los entregables realizados durante la tercera etapa y la retrospectiva realizada durante la cuarta etapa.

5. Definición de requerimientos funcionales y tecnológicos

A continuación, se especificarán los requerimientos tanto de la plataforma Web como de la plataforma APP, especificando en el caso de los requerimientos funcionales, si afecta al cliente (marcado como “ADMIN” al inicio) o si es genérico (usuario y cliente).

5.1. Plataforma APP

5.1.1. Requerimientos funcionales

- Registrar usuario
- Identificar usuario
- Modificar datos de usuario
- Dar de baja usuario
- Mostrar tablón de anuncios y eventos
- Visualizar el contenido del anuncio o evento
- Mostrar lista de registro de animales en el sistema
- Visualizar la información introducida de un animal de la lista
- Marcar un animal como favorito
- Agendar una cita previa sobre un animal de la lista
- Cancelar una cita previa sobre un animal de la lista
- Visualizar la lista de favoritos marcados
- Eliminar registro de la lista de favoritos marcados

5.1.2. Requerimientos tecnológicos

- La aplicación debe funcionar de forma fluida.
- La aplicación debe cambiar de pantalla en un tiempo lo suficientemente corto para no afectar al usuario.
- Aprender y entender cómo funciona el Framework React Native.
- Aprender y entender cómo funciona el entorno NodeJS.
- La aplicación debe estar adaptada a diferentes tamaños de pantalla.

5.2. Plataforma Web

5.2.1. Requerimientos funcionales

- Registrar usuario
- Identificar usuario
- Modificar datos de usuario
- Dar de baja usuario
- Mostrar tablón de anuncios y eventos
- Visualizar el contenido del anuncio o evento
- Mostrar lista de registro de animales en el sistema
- Visualizar la información introducida de un animal de la lista
- Marcar un animal como favorito
- Agendar una cita previa sobre un animal de la lista
- Visualizar la lista de favoritos marcados
- Eliminar registro de la lista de favoritos marcados
- (ADMIN) Visualizar, modificar o eliminar una cita previa registrada en el sistema
- (ADMIN) Eliminar o bloquear acceso a un usuario
- (ADMIN) Aceptar o rechazar una cita previa registrada en el sistema
- (ADMIN) Añadir, modificar o eliminar un registro de animal en el sistema

5.2.2. Requerimientos tecnológicos

- El portal web debe funcionar de forma fluida.
- El tiempo de respuesta al cambiar de página debe ser lo suficientemente rápida para no afectar al usuario.
- Aprender y entender cómo funciona el Framework React JS.
- Aprender y entender cómo funciona el entorno NodeJS.
- El portal web debe estar adaptado a diferentes tamaños de pantalla.

5.3. Justificación de las tecnologías escogidas

5.3.1. Plataforma de Backend

En el apartado de Backend, se opta por el desarrollo de una Rest API con base en NodeJS.

Este entorno permite la integración de varios módulos que pueden ser útiles para la creación del Backend, como por ejemplo las conexiones con la base de datos o la transformación de los datos o peticiones antes de ser procesados.

Además de esta ventaja, encontramos una compatibilidad total con los Framework ReactJS y React Native (utilizados para el portal web y la aplicación móvil respectivamente), por lo que se garantiza una funcionalidad completa, permitiendo utilizar un mismo sistema para todo.

También contaremos con una base de datos creada en MongoDB [7], que permitirá una alta disponibilidad (imprescindible en este tipo de proyectos) y una gran escalabilidad y flexibilidad gracias a su sistema de ficheros que no fija un único tipo de modelamiento de datos.

5.3.2. Plataforma Web

En el apartado de Plataforma Web, se opta por el desarrollo mediante el Framework ReactJS[8].

Este Framework, al igual que en la Plataforma de Backend, permite integrar varios módulos que faciliten la creación de este entorno, además de, por su naturaleza, brindar la posibilidad de un diseño modular, ofreciendo funciones o porciones de código en forma de bloque, alojados (o no) dentro del mismo fichero, lo que permitirá reaprovechar código sin necesidad de repetir constantemente porciones del diseño.

Además de estas ventajas, al compartir base con React Native (Framework utilizado para el desarrollo de la Plataforma APP) posibilita el trasladar parte del código de una plataforma u otra, escribiendo el código una vez y utilizándolo en dos plataformas.

Este Framework está soportado por Facebook, que garantiza su correcto funcionamiento y constantes actualizaciones y mantenimientos. Como ejemplos de proyectos podemos encontrar Facebook, Instagram, WhatsApp web, New York Times o Netflix [9].

5.3.3. Plataforma APP

En el apartado de Plataforma APP, se opta por el desarrollo mediante el Framework React Native [10].

Este Framework, al igual que en la Plataforma de Backend, permite integrar varios módulos que faciliten la creación de esta plataforma, además de, por su naturaleza, brindar la posibilidad de un diseño modular, ofreciendo funciones o porciones de código en forma de bloque, alojados (o no) dentro del mismo fichero, lo que permitirá reaprovechar código sin necesidad de repetir constantemente porciones del diseño.

A estas ventajas se le suman una facilidad de gestión de pantallas que permite navegar fácilmente entre ellas desde un dispositivo móvil, y ofrecer una transición fluida entre éstas.

Además de estas ventajas, al compartir base con ReactJS (Framework utilizado para el desarrollo de la Plataforma Web) posibilita el trasladar parte del código de una plataforma u otra, escribiendo el código una vez y utilizándolo en dos plataformas.

Este Framework está soportado por Facebook, que garantiza su correcto funcionamiento y constantes actualizaciones y mantenimientos. Como ejemplos de proyectos podemos encontrar Discord, Uber Eats, Pinterest o Coinbase [11].

5.3.4. Posibles alternativas

Como posibles alternativas al desarrollo de la Plataforma APP se encuentra Android SDK y Angular (proporcionado en ambos casos por Google) [12].

- **Angular** es un Framework que permite la creación de aplicaciones móviles, web y de escritorio (al igual que todo el marco React) desarrollado y apoyado por Google.
 - Como **puntos positivos** se observa que permite crear una aplicación por pantallas de forma sencilla y fácil de mantener, además de normalmente no requerir librerías adicionales para expresar su uso.
 - Como **puntos negativos** se observa que en su versión 2.0 impusieron el uso del lenguaje TypeScript, forzando el aprendizaje de este si se quería utilizar, además de apoyarse en exceso sobre Ionic (otro Framework) en el desarrollo de APPs.

- **Android SDK** es un kit de desarrollo de software lleno de herramientas proporcionadas por Google para crear una aplicación de móvil Android.
 - Como **puntos positivos** se observa que permite crear una aplicación completamente nativa e incluye todas las herramientas necesarias para la creación de ésta.
 - Como **puntos negativos** se observa que debe actualizarse el SDK para cada una de las versiones de Android que Google lanza, forzándonos a estar actualizando constantemente nuestra aplicación.

En base a estos puntos, se decide escoger React Native, puesto que es un framework más sencillo de utilizar y aprender, además de permitir su integración con el lenguaje NativeScript (muy parecido a JavaScript y HTML/CSS), su fácil mantenimiento y la posibilidad de integrar nuevos módulos y librerías creadas constantemente por la comunidad para facilitar algunas funcionalidades. Además de, a diferencia de lo que se piensa, permitir crear una aplicación completamente nativa y de la integración posterior con la Plataforma Web [13].

Como posibles alternativas al desarrollo de la Plataforma Web se encuentra nuevamente AngularJS, VueJS y Svelte [14].

- **AngularJS** es un Framework que permite la creación de aplicaciones móviles, web y de escritorio (al igual que todo el marco React) desarrollado y apoyado por Google.
 - Como **puntos positivos** se observa que tiene una menor curva de aprendizaje, además de permitir realizar casi cualquier funcionalidad necesaria de base (sin librerías externas). También se encuentra que está diseñado para ser fácilmente testeable.
 - Como **puntos negativos** se observa que en su versión 2.0 impusieron el uso del lenguaje TypeScript, forzando el aprendizaje de este si se quería utilizar, además de imposibilitar su migración hacia otro framework.

- **VueJS** es un Framework progresivo que permite la creación de aplicaciones de una sola página e interfaces de usuario, inspirado en AngularJS.
 - Como **puntos positivos** se observa que es de fácil aprendizaje y utilización, permitiendo usar los lenguajes JavaScript y TypeScript, además de su pequeño tamaño que afecta de forma positiva al SEO y la interfaz en sí.
 - Como **puntos negativos** se observa que no cuenta con el apoyo suficiente para grandes proyectos al ser un framework relativamente joven, estando limitado en varias funcionalidades, además de una documentación bastante reducida al provenir la mayoría de comunidad de China (estando ésta en su lenguaje nativo).

- **Svelte** es un compilador de JavaScript diseñado para crear interfaces de usuario de alto rendimiento, usando una máquina virtual de JavaScript personalizada en lugar del DOM.
 - Como **puntos positivos** se observa que al no utilizar DOM las aplicaciones se vuelven más rápidas y confiables, al mismo tiempo que se utiliza el estilo de alcance con JavaScript en lugar de CSS, permitiendo incluir los estilos en medio de los documentos. También permite transformar cualquier variable en global añadiendo simplemente un '\$:' al inicio de ésta.
 - Como **puntos negativos** se observa que hay una gran falta de soporte IDE y apoyo por parte de la comunidad al ser relativamente pequeña, además de existir muy pocos kits de herramientas disponibles para este desarrollo.

En base a estos puntos, se decide escoger React JS, puesto que es un framework más sencillo de utilizar y aprender, además de permitir su integración con el lenguaje que se desee utilizar (TypeScript o JavaScript), su fácil mantenimiento y la posibilidad de integrar nuevos módulos y librerías creadas constantemente por la comunidad para facilitar algunas funcionalidades. Además de la integración posterior con la Plataforma APP y la base creada en componentes renderizados con la biblioteca DOM virtual, lo que la hace más ligera de cargar [15].

6. Desarrollo del proyecto

6.1. Requerimientos para el desarrollo de las plataformas

Para completar satisfactoriamente este proyecto se deben tener en consideración ciertos requerimientos previos para la correcta implementación y asegurar todas las funcionalidades necesarias. Es por ello que se requieren ciertas librerías a instalar para cada plataforma.

6.1.1. Requerimientos Plataforma Backend

Para esta plataforma es necesario tener instalado NodeJS desde su página oficial en la versión 17.0 para garantizar la completa compatibilidad con los módulos a instalar posteriormente.

Adicionalmente y después de obtener NodeJS, se deberán utilizar los siguientes módulos en esta plataforma:

- **Express (ver. 4.18.1):** Es un Framework minimalista y flexible que provee un amplio set de funcionalidades tanto para portal web como aplicativos móviles. En este caso permite crear rápidamente un servidor API Rest y ejecutarlo con middlewares.
- **Mongodb driver (ver. 5.2):** Es un Driver que permite conectar fácilmente con la base de datos en MongoDB además de brindar funcionalidades para interactuar con ésta (crear, modificar o eliminar registros).
- **Nodemon (ver. 2.0.7):** Este módulo se utiliza únicamente en el entorno de desarrollo, permitiendo recargar la aplicación cada vez que éste detecta un cambio en el código.
- **GraphQL (ver. 14.6.0):** Este módulo permite estructurar los datos obtenidos desde MongoDB para posteriormente ser entregados al cliente.
- **Apollo-Server (ver. 2.20.0):** Este módulo permite comunicarse con el cliente cuando éste realiza una petición y servirle con los datos necesarios.
- **Prettier (ver. 2.2.1):** Este módulo se utiliza únicamente en el entorno de desarrollo, marcando errores de sintaxis que se cometen y reparándolos automáticamente.

Adicionalmente se requiere de la extensión **ES7 React/Redux/GraphQL/React-Native snippets (ver. 1.9.3)** instalada en el editor de código a utilizar que aportará ayuda en la sintaxis del código.

6.1.2. Requerimientos Plataforma Web

Para esta plataforma es necesario tener instalado NodeJS desde su página oficial en la versión 17.0 para garantizar la completa compatibilidad con los módulos a instalar posteriormente.

Adicionalmente y después de obtener NodeJS, se deberán utilizar los siguientes módulos en esta plataforma:

- **Create-React-App (latest version):** Este framework incorpora la funcionalidad de ReactJS y permite crear rápidamente una aplicación web con ReactJS.
- **Express (ver. 4.18.1):** Es un Framework minimalista y flexible que provee un amplio set de funcionalidades tanto para portal web como aplicativos móviles. En este caso añade la funcionalidad de middlewares y posteriormente funcionalidades de Handlebars (estructuras HTML con funcionalidades extras).
- **Passport (ver. 0.6.0):** Este Framework permite la gestión de autenticación para usuarios mediante usuario y contraseña u otros medios (Google, Facebook...).
- **Multer (ver. 1.4.2):** Este Framework permite la gestión de subida de archivos al servidor por parte del cliente, además de añadir funciones de seguridad como el filtrado de extensión, de tamaño, y más opciones.
- **Bcryptjs (ver. 2.4.3):** Este Framework permite encriptar datos como contraseñas para posteriormente alojarlas en su correspondiente base de datos. Se utiliza en conjunto con Passport.
- **React-Router (ver. 6.10.0):** Este módulo añade la funcionalidad de navegación entre páginas en base a la dirección URL.

Adicionalmente se requiere de la extensión **ES7 React/Redux/GraphQL/React-Native snippets (ver. 1.9.3)** instalada en el editor de código a utilizar aportará ayuda en la sintaxis del código.

6.1.3. Requerimientos Plataforma APP

Para esta plataforma es necesario tener instalado NodeJS desde su página oficial en la versión 17.0 para garantizar la completa compatibilidad con los módulos a instalar posteriormente.

Adicionalmente y después de obtener NodeJS, se deberán utilizar los siguientes módulos en esta plataforma:

- **Create-React-Native-App (latest version):** Este framework incorpora la funcionalidad de React Native y permite crear rápidamente una aplicación móvil con React Native y JavaScript.
- **Express (ver. 4.18.1):** Es un Framework minimalista y flexible que provee un amplio set de funcionalidades tanto para portal web como aplicativos móviles. En este caso añade la funcionalidad de middlewares y variables como tamaños de pantalla por defecto, gestión de la barra de estado y más opciones.
- **React-Navigation (ver. 6.0):** Este módulo añade la funcionalidad de navegación entre pantallas además de funcionalidades con la barra de opciones inferior y superior.

- **React-Redux (ver. 8.0.5):** Este módulo añade las funcionalidades de Redux para React Native.

Adicionalmente se requiere de la extensión **ES7 React/Redux/GraphQL/React-Native snippets (ver. 1.9.3)** instalada en el editor de código a utilizar aportará ayuda en la sintaxis del código.

6.2. Base de datos

Para el propósito de este proyecto se requiere de una Base de Datos para alojar toda la información requerida de los usuarios, publicaciones, animales y hasta contactos, además de necesitar un rápido acceso a ella para mostrar esta información al usuario sin largas esperas.

Es por ello que se utiliza una Base de Datos en MongoDB, dada su alta disponibilidad, perfecta para cubrir las necesidades de este proyecto, además de permitir una alta escalabilidad y modelación de los datos.

Se utiliza el servicio MongoDB Atlas para el hospedaje de esta base de datos, puesto que permite crear un Clúster compartido sin coste alguno, escogiendo el proveedor y la región a conveniencia (siempre contando con la disponibilidad en ese momento):

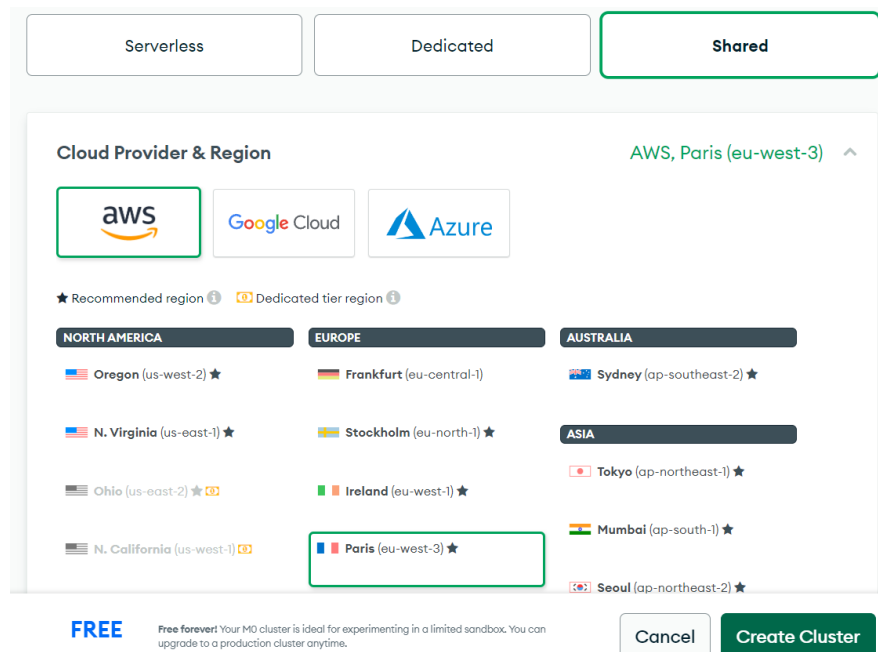


Figura 6.2 Selección de tipo, proveedor y región del Clúster - Fuente: MongoDB Atlas

6.2.1. Estructura de la Base de Datos

MongoDB se caracteriza por almacenar los datos en un sistema de ficheros (NoSQL), en vez de un sistema de tablas como se encuentra en SQL. Partiendo de esta premisa, se nombrará colecciones a todos aquellos conjuntos de documentos que forman un objeto, y documentos a todos aquellos conjuntos de datos que forman un registro.

El diseño de la base de datos consta del siguiente esquema de colecciones y documentos:

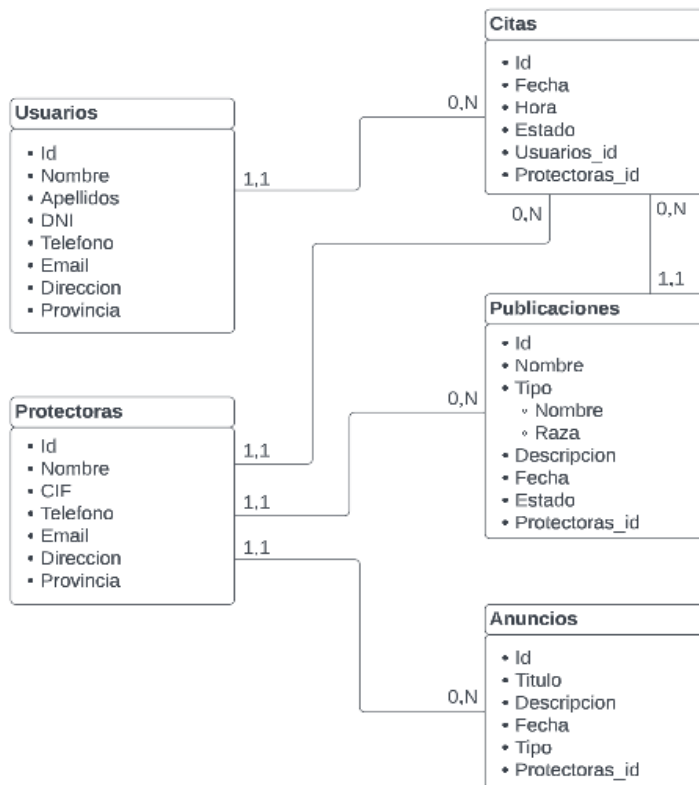


Figura 6.2.1 Diagrama de Base de Datos - Fuente: Propia

A continuación, la definición de cada colección y documento:

- Colección '**Usuarios**' con la siguiente estructura de documento:
 - **Id**: Identificador del usuario en formato UUID alfanumérico.
 - **Nombre**: Nombre del usuario en formato String.
 - **Apellidos**: Apellidos del usuario en formato String.
 - **DNI**: Cadena de 9 caracteres alfanuméricos.
 - **Teléfono**: Cadena de 9 caracteres numéricos.
 - **Email**: Email del usuario en formato String.
 - **Dirección**: Dirección de la vivienda del usuario en formato String.
 - **Provincia**: Provincia de la vivienda del usuario en formato String.

- Colección '**Protectoras** con la siguiente estructura de documento:
 - **Id:** *Identificador de la protectora en formato UUID alfanumérico.*
 - **Nombre:** *Nombre de la protectora en formato String.*
 - **CIF:** *Cadena de 9 caracteres alfanuméricos.*
 - **Teléfono:** *Cadena de 9 caracteres numéricos.*
 - **Email:** *Email de la protectora en formato String.*
 - **Dirección:** *Dirección del local de la protectora en formato String.*
 - **Provincia:** *Provincia del local de la protectora en formato String.*

- Colección '**Citas** con la siguiente estructura de documento:
 - **Id:** *Identificador de la cita en formato UUID alfanumérico.*
 - **Fecha:** *Fecha de la cita en formato Date.*
 - **Hora:** *Hora de la cita en formato Time.*
 - **Estado:** *Marca si ha sido aceptada o no mediante formato Boolean.*
 - **Usuarios_id:** *Referencia al Id del usuario solicitante.*
 - **Protectoras_id:** *Referencia al Id de la protectora pertinente.*

- Colección '**Publicaciones** con la siguiente estructura de documento:
 - **Id:** *Identificador de la publicación en formato UUID alfanumérico.*
 - **Nombre:** *Nombre del animal (si tiene) en formato String.*
 - **Tipo:** *Array con datos del tipo de animal que es.*
 - **Nombre:** *Nombre del tipo de animal (perro, gato, caballo...) en formato String.*
 - **Raza:** *Raza del animal en formato String.*
 - **Descripción:** *Descripción de la publicación en formato Text Area.*
 - **Fecha:** *Fecha de la creación de la publicación en formato Datetime.*
 - **Estado:** *Marcar si está público o no mediante formato Boolean.*
 - **Protectoras_id:** *Referencia al Id de la protectora pertinente.*

- Colección '**Anuncios**' con la siguiente estructura de documento:
 - **Id:** *Identificador del anuncio en formato UUID alfanumérico.*
 - **Título:** *Título del anuncio en formato String.*
 - **Descripción:** *Descripción del anuncio en formato Text Area.*
 - **Fecha:** *Fecha de la creación del anuncio en formato Datetime.*
 - **Tipo:** *Tipo de publicación (Informativa, Advertencia...) en formato String.*
 - **Protectoras_id:** *Referencia al Id de la protectora pertinente.*

6.3. Diseño de interfaces

En el diseño de las interfaces, se opta por predefinir un estilo que se crea como marca diferenciadora y persistente en cada una de las pantallas del proyecto.

Se establece el uso de colores Blanco y Verde como símbolo de marca, predominante en fondos, algunas tipografías y encabezados de secciones o detalles en relieves.

Como tipografía, se ha decantado por utilizar la fuente '**Roboto**' en todas las secciones excepto en el nombre del sitio, donde se utilizará la fuente '**IBM Plex Sans**' para denotar diferencia entre el título y el resto de tipografías.

Los tamaños de fuente estandarizados para las diferentes secciones aún están por definir.

6.3.1. Interfaz de plataforma APP

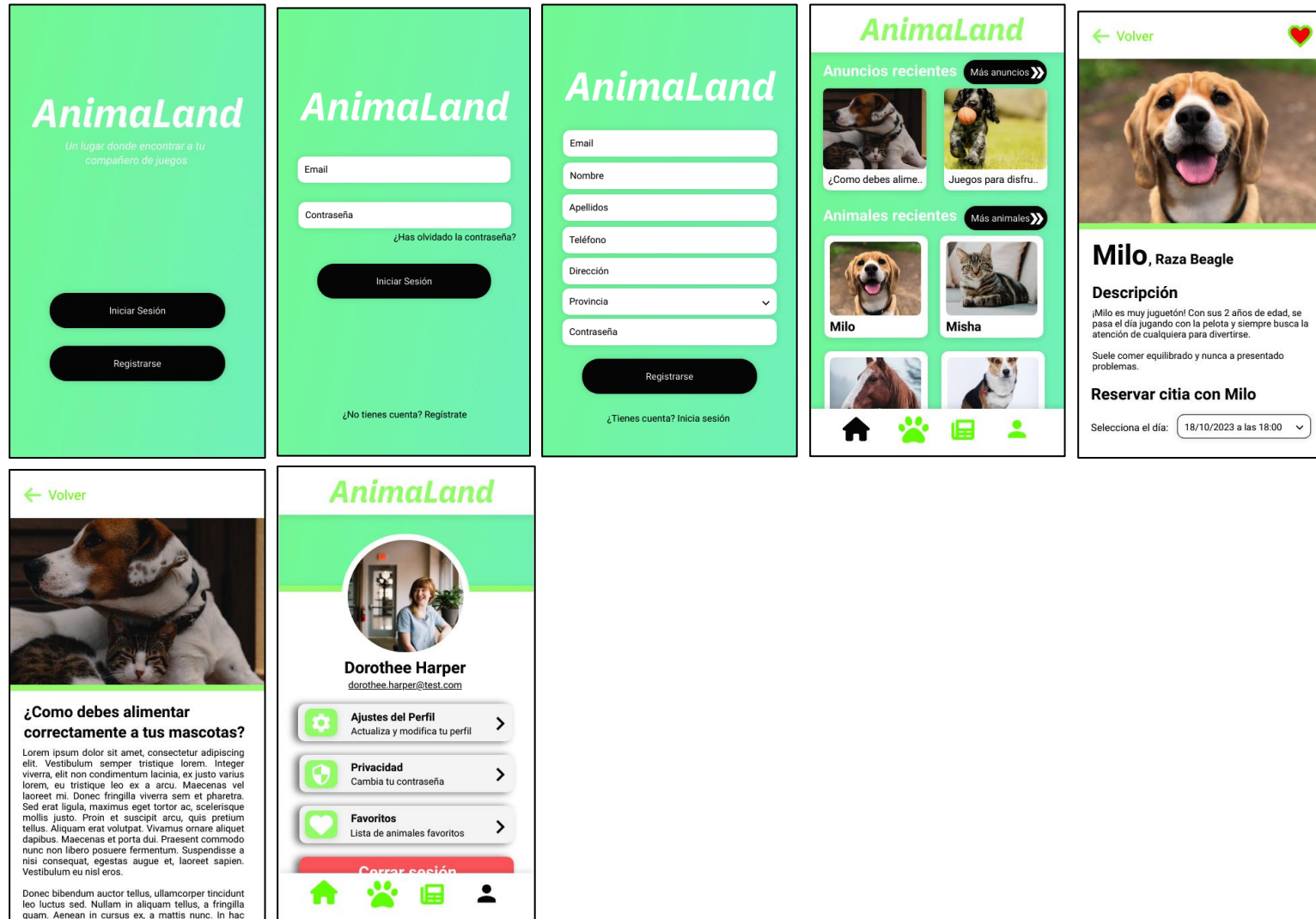


Figura 6.3.1 Galería de secciones Plataforma APP - Fuente: Propia

6.3.2. Interfaz de plataforma Web

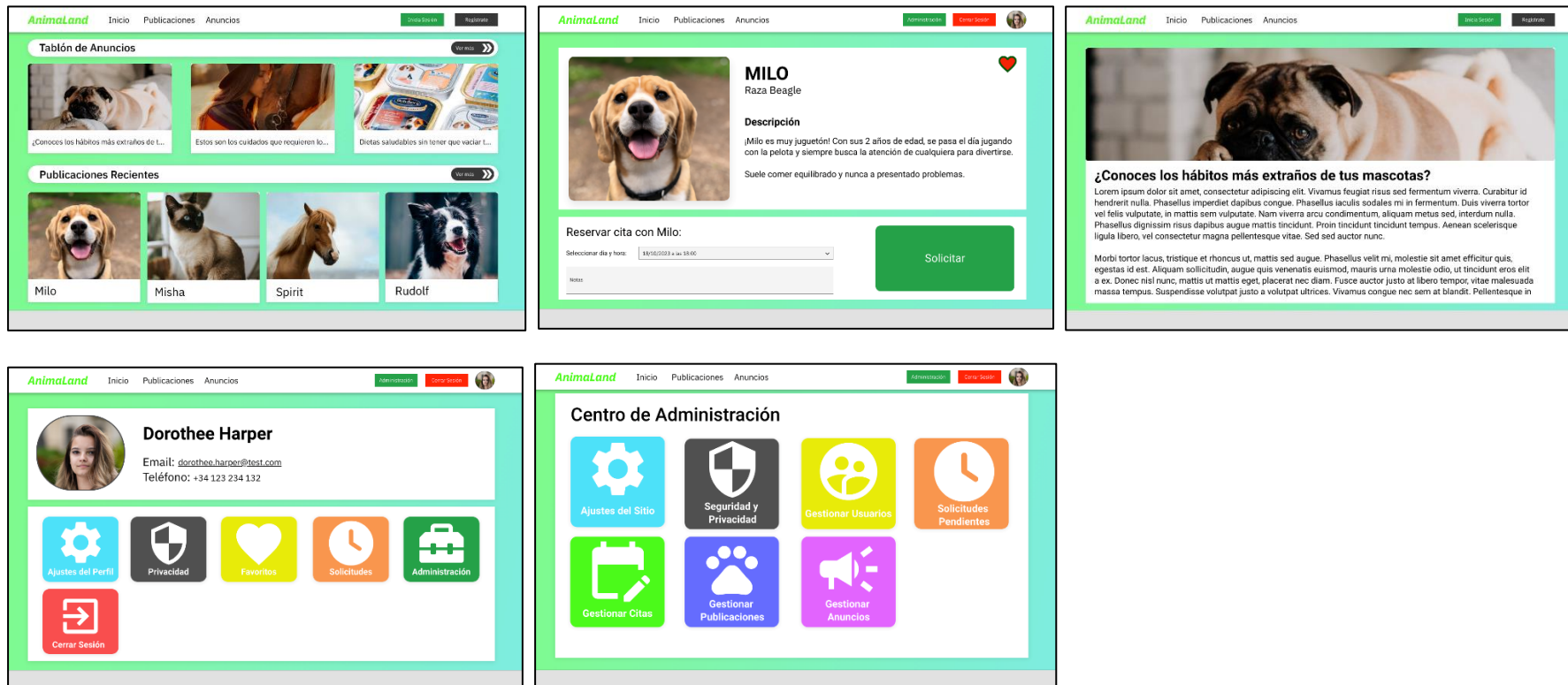


Figura 6.3.2 Galería de secciones Plataforma Web - Fuente: Propia

6.4. Definición de casos de uso

6.4.1. Registrar un usuario

| | |
|-----------------------|--|
| Actor | Usuario |
| Precondición | <ul style="list-style-type: none"> - El usuario debe tener conexión a Internet. - El usuario no debe existir. - El usuario no debe haber iniciado sesión. |
| Descripción | Para registrar un usuario, se le pedirán datos como Nombre Completo, Email, Contraseña e URL de imagen de perfil (opcional). |
| Pasos a seguir | <ol style="list-style-type: none"> 1. El usuario abre el aplicativo o página web. 2. El usuario se encuentra en la página o pantalla principal. 3. El usuario presiona el botón de “Registrarse” o “Crear una cuenta”. 4. El sistema redirige al usuario a la página o pantalla de registro. 5. El usuario introduce los datos y presiona en “Registrarse”. 6. El sistema comprueba los datos y los añade a la base de datos. 7. El sistema autentica la sesión y redirige al usuario a la página o pantalla principal. |
| Postcondición | El usuario se encuentra autenticado y en la pantalla o página principal. |
| Excepción | <ol style="list-style-type: none"> 5.1. El usuario introduce los datos incorrectamente. 5.2. El sistema no puede validar los datos y muestra un mensaje de error. 6.1. El sistema comprueba los datos y el usuario ya existe. 6.2. El sistema muestra un mensaje de error al usuario. |

Tabla 6.4.1 Caso de uso: Registrar un usuario

6.4.2. Iniciar sesión

| | |
|-----------------------|--|
| Actor | Usuario Administrador |
| Precondición | - El usuario o administrador debe tener conexión a Internet. - El usuario o administrador debe existir. - El usuario o administrador no debe haber iniciado sesión. |
| Descripción | Para iniciar sesión se le pedirán datos como Email y Contraseña. |
| Pasos a seguir | <ol style="list-style-type: none"> 1. El usuario o administrador abre el aplicativo o página web. 2. El usuario o administrador se encuentra en la página o pantalla principal. 3. El usuario o administrador presiona el botón de “Iniciar sesión”. 4. El sistema redirige al usuario o administrador a la página o pantalla de inicio de sesión. 5. El usuario o administrador introduce los datos y presiona en “Iniciar sesión”. 6. El sistema comprueba los datos. 7. El sistema autentica la sesión y redirige al usuario o administrador a la página o pantalla principal. |
| Postcondición | El usuario o administrador se encuentra autenticado y en la pantalla o página principal. |
| Excepción | <ol style="list-style-type: none"> 5.1. El usuario o administrador introduce los datos incorrectamente. 5.2. El sistema no puede validar los datos y muestra un mensaje de error. |

Tabla 6.4.2 Caso de uso: Iniciar sesión

6.4.3. Cerrar sesión

| | |
|-----------------------|--|
| Actor | Usuario Administrador |
| Precondición | - El usuario o administrador debe tener conexión a Internet. - El usuario o administrador debe haber iniciado sesión. |
| Descripción | Se permite cerrar sesión para desautenticarse del aplicativo. |
| Pasos a seguir | <ol style="list-style-type: none"> 1. El usuario o administrador abre el aplicativo o página web. 2. El usuario o administrador se encuentra en la página o pantalla principal. 3. El usuario o administrador se dirige a su página o pantalla de perfil. 4. El usuario o administrador presiona el botón de “Cerrar sesión”. 5. El sistema desautentica al usuario o administrador y elimina los registros en el aplicativo o navegador. 6. El sistema redirige al usuario o administrador a la pantalla principal. |
| Postcondición | El usuario o administrador se encuentra desautenticado y en la pantalla o página principal. |
| Excepción | <p>5.1. El usuario o administrador no está autenticado o la sesión ha expirado.</p> <p>5.3. El sistema redirige al usuario o administrador a la pantalla o página principal.</p> |

Tabla 6.4.3 Caso de uso: Cerrar sesión

6.4.4. Visualizar un animal

| | |
|-----------------------|--|
| Actor | Usuario Administrador |
| Precondición | - El usuario o administrador debe tener conexión a Internet. - El usuario o administrador debe haber iniciado sesión. |
| Descripción | Se permite visualizar el perfil de un animal registrado en el sistema. |
| Pasos a seguir | <ol style="list-style-type: none"> 1. El usuario o administrador abre el aplicativo o página web. 2. El usuario o administrador se encuentra en la página o pantalla principal. 3. El usuario o administrador se dirige a la página o pantalla de “Lista de Animales” o “Publicaciones”. 4. El sistema devuelve una lista con los animales disponibles en el sistema. 5. El usuario o administrador selecciona un animal a visualizar. 6. El sistema muestra una pantalla o página con todos los detalles del animal seleccionado. |
| Postcondición | El usuario o administrador puede visualizar todos los datos referentes a un animal seleccionado. |
| Excepción | 6.1. El sistema no encuentra el animal seleccionado y devuelve un mensaje de error. |

Tabla 6.4.4 Caso de uso: Visualizar un animal

6.4.5. Visualizar un anuncio

| | |
|-----------------------|---|
| Actor | Usuario Administrador |
| Precondición | - El usuario o administrador debe tener conexión a Internet. - El usuario o administrador debe haber iniciado sesión. |
| Descripción | Se permite visualizar un anuncio o evento registrado en el sistema. |
| Pasos a seguir | <ol style="list-style-type: none"> 1. El usuario o administrador abre el aplicativo o página web. 2. El usuario o administrador se encuentra en la página o pantalla principal. 3. El usuario o administrador se dirige a la página o pantalla de “Lista de Anuncios” o “Anuncios”. 4. El sistema devuelve una lista con los anuncios disponibles en el sistema. 5. El usuario o administrador selecciona un anuncio a visualizar. 6. El sistema muestra una pantalla o página con todos los detalles del anuncio seleccionado. |
| Postcondición | El usuario o administrador puede visualizar todos los datos referentes a un anuncio seleccionado. |
| Excepción | 6.1. El sistema no encuentra el anuncio seleccionado y devuelve un mensaje de error. |

Tabla 6.4.5 Caso de uso: Visualizar un anuncio

6.5. Problemáticas encontradas durante el desarrollo

Este proyecto en su fase temprana no ha tenido demasiados errores más allá de los que se pueden encontrar en la configuración inicial. Es por ello que los errores encontrados en el momento son los listados a continuación:

6.5.1. Problemas con versiones desfasadas de módulos

Al inicio del desarrollo, se encontraron problemas en el momento de añadir ciertos módulos necesarios en la Plataforma APP los cuales se usaron con una versión demasiado inferior (versiones citadas en varios ejemplos obsoletos en tiempo) y han implicado ciertos cambios en las formas de implementarlos y en sus funciones.

El ejemplo se produce con el módulo ‘**React-Navigation**’, el cual producía problemas al intentar incorporarse con la versión de ‘**Create-React-Native-App**’ y no terminaba de mostrarse. Además de este problema inicial, también hubo problemas al momento de

actualizarlo a la versión que se utiliza actualmente, puesto que la sintaxis de implementación también había variado.

Así pues, la solución a este problema fue actualizar este módulo a la versión en uso actual (ver. 6.0) y buscar nuevamente en la documentación las nuevas sintaxis de implementación.

6.5.2. Problemas de conexión con la Base de Datos en MongoDB Atlas

En el momento de la implementación del sistema de seguridad mediante el módulo ‘**Passport**’ en Plataforma Web, se ha encontrado que al momento de pasar las peticiones hacia el Backend y alojar los inicios de sesión, se obtienen problemas por la forma sintáctica en la que se guarda.

Este problema sucedía en el momento del registro de un usuario, no permitiendo utilizarse posteriormente el inicio de sesión, puesto que el usuario no se llega a crear.

La solución a este problema ha sido utilizar temporalmente el sistema ‘**Authentication**’ proporcionado por Firebase junto con su servicio de base de datos ‘**Realtime Database**’, permitiendo de esta forma la gestión de usuarios con estos servicios.

Cabe destacar que Realtime Database proporciona una base de datos NoSQL, mediante documentos, al igual que ofrece MongoDB, facilitando su posterior fusión con este último.

6.6. Entrevistas con el cliente

6.6.1. Primera entrevista con el cliente

En esta primera toma de contacto, se dedica especial énfasis a los requerimientos del cliente en base a este proyecto para poder responder a la cuestión básica de “¿Qué se busca conseguir?”.

Con esta explicación, se marcarán las bases para desarrollar los aplicativos cumpliendo con las necesidades del cliente, marcadas en puntos anteriores (5. *Definición de requerimientos funcionales y tecnológicos*).

6.6.2. Segunda entrevista con el cliente y usuarios

En esta segunda entrevista, se presenta una versión muy inicial de la Plataforma APP en conjunto con su parte correspondiente de la Plataforma Backend.

Se muestran las pantallas Inicial, Registro, Inicio de sesión, Inicio, Publicación de animal, Anuncio y Perfil (observables en el apartado de Desarrollo del Proyecto – Diseño de interfaces).

El sistema de pantallas se muestra adecuado para el cliente y cumple con los requerimientos mínimos establecidos, aunque no hay total acuerdo sobre el tipo de fuente y colores utilizados, siendo estos demasiado llamativos. Se requerirá un cambio a una fuente más amigable y utilizar colores menos Neón (resultados actualizados en el apartado Diseño de interfaces).

Los usuarios se muestran acordes con la funcionalidad y fluidez entre pantallas. Descartando los comentarios ofrecidos por el cliente, no añaden ningún otro problema.

Los compromisos en este punto del proyecto se establecen en cambiar la tipografía de la fuente y utilizar colores no tan llamativos y se marcan como próximos pasos la muestra funcional del portal web y las actualizaciones pertinentes en la App.

6.6.3. Última entrevista con el cliente y usuarios

En esta última entrevista se muestra una versión final del aplicativo web y app que permite al cliente y los usuarios interactuar con la interfaz y todo su sistema.

Antes de realizar las pruebas, se explica detalladamente cuáles son las secciones y pantallas de las que se componen cada aplicativo y se contrarresta con las necesidades especificadas por el cliente.

Finalizada la explicación y con el visto bueno del cliente, se procede a entregar una muestra tanto al cliente como a los usuarios para poder probarla y detectar posibles problemas en el momento del uso.

A la finalización de las pruebas, se proporciona todos los entregables al cliente junto con la documentación del proyecto para finalizarlo.

7. Análisis y conclusiones

7.1. Análisis del producto

El desarrollo de dos aplicativos orientados a diferentes plataformas ha supuesto un gran punto de inflexión sobre la tecnología a utilizar, debido a que, si se desarrollaran con dos sistemas diferentes, aumentaría significativamente el tiempo de desarrollo y podría no cumplir con los plazos de entrega estipulado.

Es por esto que se opta por realizar todo el ámbito Frontend en base a la tecnología React, permitiendo completar los diferentes requerimientos (divididos dentro del producto en módulos) para seguidamente utilizarse en el posterior aplicativo sin problemas, además de aportar una gran escalabilidad en términos de añadir nuevas funcionalidades cuando se requieran sin modificar la base del proyecto.

Pudiendo ser un proyecto muy ambicioso por la cantidad de funcionalidades que se podrían añadir, el hecho de realizar un estudio previo de los objetivos, los requerimientos (funcionales y tecnológicos), y definición del alcance ayudó a determinar hasta qué punto se puede desarrollar el proyecto y no desviarse de la planificación inicial y estimación del tiempo.

Por problemas técnicos al momento de conexión entre base de datos, autenticación y proyecto, se desvió el uso de la base de datos a “Firebase Realtime Database”, manteniendo la misma estructura de documentos, pero cambiando el proveedor, puesto que la autenticación se realizaría con otro sistema de la misma marca bajo el nombre de “Authentication”.

Al momento de entregar este proyecto se han concluido todos estados de desarrollo, aunque con algunos detalles, puesto que el entorno de React es significativamente grande y algunas funcionalidades se han visto modificadas y reducidas de calidad por cumplir con el plazo de entrega, incluyendo algunos mensajes de error como en el momento en que no hay ningún animal o anuncio añadido, o el sistema de citas, que aun estando en funcionamiento, la introducción de datos por parte del cliente no es automática y se deben introducir uno a uno.

7.2. Conclusiones del proyecto

Como conclusión final de este proyecto, se podría catalogar como completado a grandes rasgos puesto que las funcionalidades especificadas desde un principio se cumplen.

En términos más específicos, se puede observar algunos detalles restantes que se podrían pulir o mejorar, pero dado la curva de aprendizaje de éste Framework no utilizado anteriormente y el tener que seguir con el estudio de nuevos módulos que cumplieran con los requisitos durante su fase de desarrollo han provocado un aumento en el tiempo de ésta fase, lo cual ha llevado a reducir las posteriores tareas y dejar algunas funcionalidades sin mensajes de error como los mencionadas en el punto anterior.

Si bien este Framework es muy amplio, es bastante fácil poder encontrar módulos que cumplan con lo que necesites realizar, y más con grandes marcas como MongoDB o Firebase, que cuentan con sus propios módulos diseñados específicamente para interactuar con sus interfaces, además de la fácil integración entre las plataformas web y apps puesto que comparten gran parte de los módulos y funcionalidades.

Es por todo esto que seguiría utilizando las tecnologías escogidas para continuar fortaleciendo este proyecto, contando ahora con más conocimiento previo.

8. Posibles ampliaciones

Este proyecto no ha abarcado varias partes en términos de funcionalidad y practicidad para los usuarios que podrían ser de gran utilidad si se desarrollaran.

Es por esto que como posibles ampliaciones se podría continuar desde esta base con:

8.1. Filtraje de listas

Como funcionalidad extra, se podría acceder a las listas de Animales y Anuncios, además de poder filtrar los resultados según cada tipo:

- Para los Animales: Por Localización, raza, tipo de animal, centro, etiquetas.
- Para los anuncios: Por centro, autor, categorías.

8.2. Etiquetado y categorización

Siguiendo con la funcionalidad de filtraje, se podrían establecer etiquetas y categorías en cada anuncio o publicación para permitir una fácil búsqueda por parte del usuario y reducir más el rango de resultados que obtiene.

8.3. Generación de citas automáticas

Se podría desarrollar de nuevo el módulo de citas permitiendo un sistema automático de creación de espacios basado en las horas laborales y en las que no se deba depender del administrador para crearlas y establecerlas. Con esto se reduciría el gasto de tiempo por parte del cliente en crear los diferentes espacios libres para cada animal, así como mantener siempre actualizadas las fechas y eliminando las pasadas.

8.4. Autenticación mediante redes sociales

El módulo utilizado para las autenticaciones (Firebase Authentication) permite utilizar varios métodos de registro y autenticación que incluye las populares redes sociales como Facebook, Twitter y Google.

Modificando el módulo de inicio de sesión y registro se podría conseguir integrar esta funcionalidad que facilitaría el acceso de nuevos usuarios por su practicidad a la hora del uso.

7. Bibliografía

[1] La diferencia entre una protectora de animales y una perrera [en línea] [consulta: 09 de febrero de 2023].

Disponible en https://www.eldiario.es/murcia/patrulla-animal/diferencia-protectora-animales-perrera_132_1731242.html

[2] La diferencia entre perreras municipales y protectoras [en línea] [consulta: 09 de febrero de 2023].

Disponible en <https://www.adiestrar-perros.com/blog/la-diferencia-entre-perreras-municipales-y-protectoras/>

[3] AdoptaMe, la aplicación que permite adoptar animales en muy pocos pasos [en línea] [consulta: 09 de febrero de 2023].

Disponible en https://los40.com/los40/2021/05/15/bigbang/1621072918_248148.html

[4] AdoptaMe: adoptar animales ahora es más fácil gracias a esta app española [en línea] [consulta: 09 de febrero de 2023].

Disponible en <https://www.xatakandroid.com/aplicaciones-android/adoptame-adoptar-animales-ahora-facil-gracias-a-esta-app-espanola>

[5] Hablamos con el equipo de KLYGO, la app para adoptar y acoger animales [en línea] [consulta: 09 de febrero de 2023].

Disponible en <https://appmarketingnews.io/hablamos-con-el-equipo-de-klygo-la-app-para-adoptar-y-acoger-animales/>

[6] KLYGO, una app para adoptar gatos y perros [en línea] [consulta: 09 de febrero de 2023].

Disponible en <https://www.whatsnew.com/2022/01/28/klygo-una-app-para-adoptar-gatos-y-perros/>

[7] Qué es MongoDB y razones para usarlo [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://platzi.com/blog/7-razones-mongodb/>

[8] ¿Qué es React.js? Un Vistazo a la Popular Biblioteca de JavaScript [en línea]

[consulta: 19 de abril de 2023].

Disponible en <https://kinsta.com/es/base-de-conocimiento/que-es-react-js/>

[9] 5 aplicaciones famosas que están usando ReactJS [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://medium.com/codika/5-aplicaciones-famosas-que-est%C3%A1n-usando-reactjs-e8b90d36238f>

[10] ¿Qué es React Native? [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html>

[11] 5 Populares Aplicaciones que usan tecnología React (React Native, React JS) [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://blog.nubecolectiva.com/5-populares-aplicaciones-que-usan-tecnologia-react/>

[12] Android SDK vs AngularJS [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://stackshare.io/stackups/android-vs-angularjs#:~:text=Android%20provides%20a%20rich%20application,you%20had%20a%20smarter%20browser.>

[13] Angular vs React: Cuál Elegir Para tu Aplicación [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://www.freecodecamp.org/espanol/news/angular-vs-react-cual-elegir-para-su-aplicacion/>

[14] Qué es Svelte y Otros Detalles [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://blog.nubecolectiva.com/que-es-svelte-y-otros-detalles/>

[15] React vs. Angular vs. Vue vs. Svelte [en línea] [consulta: 19 de abril de 2023].

Disponible en <https://platzi.com/blog/react-angular-vue-svelte/>