



I.E.S. INFANTA ELENA



FAMILIA PROFESIONAL DE INFORMÁTICA Y COMUNICACIONES

**CICLO FORMATIVO DE GRADO SUPERIOR DESARROLLO DE
APLICACIONES WEB**

PROYECTO FIN DE CICLO

MI-MASCOTA PERRUNA

Diego González Martínez

CURSO 2023-2024

TÍTULO: MI-MASCOTA PERRUNA

AUTOR: Diego González Martínez

TUTOR DEL PROYECTO: María Irina Romero Molina

FECHA DE LECTURA:

CALIFICACIÓN:

Fdo: Diego González Martínez

Tutor/a del Proyecto

RESUMEN:

"Mi-Mascota Perruna" es un proyecto fin de ciclo que nace con el objetivo de desarrollar un espacio donde conectar a dueños de perros con cuidadores de perros, mostrar sus servicios y contenidos y dar visibilidad y respuesta a una necesidad actual en nuestra sociedad.

El proyecto se divide en tres partes: una aplicación móvil para Android que ayuda al usuario a contactar y reservar al cuidador ideal para su mascota y dos páginas web funcionales e intuitivas. En la primera página web se puede consultar información sobre los servicios ofrecidos y cuenta con una interfaz que permite al administrador crear, borrar y editar usuarios que se registran a través de la aplicación móvil. La segunda página incluye información sobre los servicios ofrecidos, opiniones de usuarios y una calculadora de ubicación.

La página web del administrador está desarrollada con Angular, HTML y CSS y la página web informativa en HTML, CSS y JavaScript. Para la aplicación se ha utilizado Java y XML.

Tanto el sitio web informativo como la aplicación van a contar con integración de API's, extensiones de Google para depurar la página web, herramientas de Firebase y aplicaciones como Google Analytics y Meta que utilizarán píxeles de seguimiento para proporcionar estadísticas y métricas detalladas.

ABSTRACT:

"Mi-Mascota Perruna" is an end-of-course project created with the aim of developing a platform that connects dog owners with dog sitters, showcases their services and content, and addresses a current need in our society.

The project is divided into three parts: a mobile application for Android that helps users contact and book the ideal sitter for their pet, and two functional and intuitive websites. The first website provides information about the services offered and features an interface that allows the administrator to create, delete, and edit users who register through the mobile app. The second website includes information about the services offered, user reviews, and a location calculator.

The administrator website is developed using Angular, HTML, and CSS, while the informational website is built with HTML, CSS, and JavaScript. The application is developed using Java and XML.

Both the informational website and the application will feature API integration, Google extensions for website debugging, Firebase tools, and applications like Google Analytics and Meta that use tracking pixels to provide detailed statistics and metrics.

AGRADECIMIENTOS

Al concluir esta etapa de mi formación académica, deseo expresar mi más sincero agradecimiento a todas las personas que de una u otra forma, han contribuido a la realización de este Trabajo de Fin de Ciclo.

En primer lugar, agradezco a los profesores y profesoras del Infanta Elena que han sido esenciales para mi crecimiento académico.

Agradezco a mi tutor de prácticas en la empresa Kyocera por los conocimientos impartidos y el tiempo que me ha dedicado.

A mi compañera Joana Mateo por su ayuda y su constante ánimo.

A mi familia, por su incondicional apoyo, comprensión y amor. Gracias por estar siempre a mi lado, motivándome a superar cada obstáculo y celebrando cada logro conmigo.

Finalmente, agradezco a todas las personas que de alguna manera contribuyeron con su tiempo y recursos para que este trabajo fuera posible.

A todos, mi más profundo agradecimiento.



Esta obra se distribuye bajo una licencia Creative Commons.

Se permite la copia, distribución, uso y comunicación de la obra si se respetan las siguientes condiciones:

- Se debe reconocer explícitamente la autoría de la obra incluyendo esta nota y su enlace.
- La copia será literal y completa
- No se podrá hacer uso de los derechos permitidos con fines comerciales, salvo permiso expreso de los autores.

El texto precedente no es la licencia completa sino una nota orientativa de la licencia original completa (jurídicamente válida) que puede encontrarse en: <http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es>



ÍNDICE

INTRODUCCIÓN.....	2
1.1. OBJETIVOS Y MOTIVACIÓN	3
DESARROLLO DE LA PRÁCTICA	5
1.2. ANÁLISIS	5
1.2.1. <i>Requerimientos.....</i>	5
1.2.2. <i>Arquitectura del sistema.....</i>	6
1.3. DISEÑO	11
1.4. COMPONENTES	13
1.4.1. <i>Componentes de la interfaz de usuario</i>	13
1.4.2. <i>Componentes de la lógica de negocio.....</i>	17
1.5. TECNOLOGÍAS Y LENGUAJES.....	18
1.6. PLANIFICACIÓN.....	19
1.7. HERRAMIENTAS	21
1.8. INTEGRACIÓN DE FIREBASE EN EL PROYECTO.....	23
CONCLUSIONES	26
1.9. PRUEBAS, RESULTADOS Y VALIDACIONES.....	26
1.10. EXPLOTACIÓN.....	29
1.11. PRESUPUESTO	30
1.12. TRABAJO FUTURO	32
BIBLIOGRAFÍA Y WEBGRAFÍA	34
ANEXOS	I
1.13. PROTOTIPOS.....	I
1.13.1. <i>Prototipo App Móvil.....</i>	<i>I</i>
1.13.2. <i>Prototipo Página Web Informativa</i>	<i>V</i>
1.14. RESULTADOS APLICACIÓN MÓVIL	VIII
1.15. RESULTADOS PÁGINA WEB INFORMATIVA	XI
1.16. RESULTADOS PÁGINA WEB ADMINISTRADOR CRUD	XIV
1.17. GLOSARIO.....	XVI
1.18. CONFIGURACIÓN	XVII
1.18.1. <i>Configuración del hosting en Firebase.....</i>	<i>XVII</i>
1.18.2. <i>Configuración de las herramientas de firebase</i>	<i>XXV</i>
1.19. CARTAS DE COMPROBACIÓN.....	XXVIII
1.20. ANUNCIOS PUBLICITARIOS.....	XXX
1.21. MANUAL ANGULAR.....	XXXI



INTRODUCCIÓN

El 35% de la población mundial tiene un perro como mascota. Según el último estudio de censos (2021), España cuenta con más de 9 millones de perros domésticos. Los perros son una de las mascotas que más tiempo de cuidados requiere. Se recomienda sacarlos a pasear 3 veces al día durante 30 minutos, llevarlos a la peluquería cada 2-3 meses, acudir al veterinario, además de tener que adiestrarlos y darles de comer.

El teletrabajo ha hecho que podamos pasar más tiempo con nuestras mascotas, pero a pesar de que nuestra sociedad es cada vez más “pet friendly”, sigue habiendo bastantes obstáculos a la hora de compaginar nuestra vida familiar y laboral con el cuidado de nuestro perro.

Es por ello, por lo que surge la idea de Mi-Mascota Perruna, una aplicación dirigida a aquellos que quieran mimar a su perro cuando no se dispone de suficiente tiempo para ello. El principal objetivo de esta aplicación es facilitar a los dueños de perros que necesiten atención para sus mascotas, conectar fácilmente con cuidadores que ofrezcan este servicio. De esta manera, los dueños de perros pueden estar seguros de que sus mascotas están siendo bien cuidadas, mientras que ellos pueden centrarse en otras responsabilidades.

La aplicación Mi-Mascota Perruna permitirá a los usuarios buscar y reservar cuidadores de perros, y también ofrecerá una plataforma de pago segura para la transacción. Además, los cuidadores de perros podrán registrarse y crear un perfil, indicando sus habilidades y experiencia en el cuidado de perros, así como su disponibilidad. De esta forma, los usuarios pueden buscar cuidadores que se ajusten a sus necesidades específicas y preferencias.

Por otro lado, se encuentran las dos páginas web. La primera será diseñada utilizando Angular con TypeScript, HTML y CSS, permitiendo realizar operaciones CRUD (Create, Read, Update y Delete) en los usuarios registrados en la aplicación. Esta página web estará exclusivamente disponible para el administrador, quien accederá a la plataforma mediante un inicio de sesión. La segunda página web contendrá información sobre los servicios ofrecidos por Mi-Mascota Perruna, opiniones de usuarios y una calculadora de ubicación.

En resumen, Mi-Mascota Perruna será una solución práctica y eficaz para aquellos que necesitan ayuda para cuidar a su perro y no tienen el tiempo o los recursos para hacerlo ellos mismos.



1.1. Objetivos y motivación

El objetivo de este proyecto es desarrollar mediante herramientas de programación una aplicación móvil en Android Studio usando lenguaje Java y conectarla a la página web elaborada con HTML, CSS y JavaScript mediante una base de datos de FireBase NoSQL.

Tanto la página web como la aplicación deberán ser seguras, accesibles, útiles y con un diseño atractivo y fácil de usar. Asimismo, se utilizarán tecnologías innovadoras como la inteligencia artificial para elaborar un mejor diseño de la aplicación, como por ejemplo Kaiber AI.

El proyecto se divide en 3 partes:

1. Una aplicación Android donde los usuarios podrán registrarse e iniciar sesión como dueño o cuidador de perros.

En el caso de registrarse como dueño, se mostrarán los siguientes botones:

- Mi perfil, donde aparecerán datos del dueño y de los perros (número de perros, raza, edad...).
- Cuidadores, donde aparecerá un listado con las personas que se han registrado como cuidadores de perros. Se podrá ver el perfil de cada uno de ellos y la opción para contratarlos.
- Atención al cliente.
- Cerrar sesión.

En el caso de registrarse como cuidador:

- Mi perfil, donde aparecerá una foto del cuidador y sus datos de contacto. Además, podrán establecer los servicios que ofrecen, el precio y una breve descripción.
2. Una página web CRUD que tendrá una pantalla de inicio de sesión. La interfaz de la web será de tipo CRUD, cuatro operaciones que se pueden realizar en las bases de datos y sistemas de gestión de información, donde el administrador se encargará de configurar los usuarios que se hayan registrado en la app.
 - 3.- Una página web informativa con las siguientes funcionalidades:
 - Una interfaz donde se informe a los usuarios de los servicios que se ofrecen.
 - Opiniones de los clientes.
 - Calculadora donde el dueño podrá ver a qué distancia se encuentra el cuidador de su casa para contratarle.



Mi-Mascota Perruna quiere proporcionar a los dueños de mascotas un servicio de cuidado de alta calidad, personalizado y accesible en todo momento. Actualmente, muchos dueños de mascotas tienen horarios ocupados que les impiden dedicar el tiempo y la atención adecuados a sus perros. Este proyecto busca solucionar estos problemas al proporcionar una herramienta que conecta a los dueños de mascotas con cuidadores de mascotas confiables en su área. Asimismo, se espera que se fomente la creación de una comunidad de dueños y cuidadores de mascotas, lo que facilitará la colaboración y el intercambio de información entre ellos. Esto mejorará la experiencia del usuario y también puede llevar a un mayor conocimiento y conciencia sobre el cuidado de perros y la importancia de la atención adecuada.



DESARROLLO DE LA PRÁCTICA

1.2. Análisis

1.2.1. Requerimientos

Requerimientos funcionales:

- Registro y autenticación en la aplicación para permitir crear una cuenta e iniciar sesión.
- Creación de perfiles de usuarios:
 - Permitir a los usuarios crear perfiles de dueño y/o cuidador de perros.
 - Proporcionar campos para añadir información sobre la mascota (nombre, raza, edad, tamaño, necesidades especiales) o el cuidador.
- Listado de cuidadores:
 - Facilitar un listado que permita a los dueños buscar cuidadores según sus necesidades y las de su mascota.
- Reservas:
 - Permitir a los dueños solicitar servicios de cuidado de su perro a los cuidadores disponibles.
 - Opción para seleccionar la fecha de reserva.
 - Ofrecer una plataforma de pago online para que los dueños paguen por los servicios de cuidado a sus mascotas.
- Mensajería y notificaciones:
 - Posibilitar que tanto los dueños como los cuidadores reciban notificaciones y mensajes sobre las reservas a través de email.
- Recordatorio:
 - En la pantalla de menú del dueño aparecerá un recordatorio de la reserva que ha realizado.
- Calculadora de distancia:
 - Redirigir desde la aplicación a un mapa de la página web donde se podrá calcular la distancia que hay entre el dueño y el cuidador.
 - Atención al cliente y ayuda:
 - Ofrecer un servicio de atención al cliente para responder las preguntas de los usuarios y resolver problemas desde la aplicación y desde la página web.
- Login página administrador:



- Permitir iniciar sesión al administrador usando un id y una contraseña única.
- Interfaz CRUD:
 - Tener un listado con los dueños y los cuidadores permitiendo eliminar, editar, crear y leer su información.

Requerimientos no funcionales:

La aplicación y las páginas web deben cumplir con los siguientes requisitos:

- Rendimiento rápido y eficiente, sin tiempos de espera largos ni retrasos en la actualización de la información.
- Proporcionar seguridad, protegiendo la privacidad y la información personal de los usuarios, así como los datos de pago.
- Usabilidad, deben ser fáciles de usar y accesible para los usuarios, con una interfaz intuitiva y atractiva.
- Mantenimiento sencillo, con una arquitectura clara, debiendo funcionar correctamente y sin errores.

1.2.2. Arquitectura del sistema

La aplicación está basada en la jerarquía de clases modelo-vista-controlador (MVC) para separar los datos, la lógica de negocio y la interfaz gráfica de usuario. Esta separación de responsabilidades facilita el diseño modular y la independencia de cada componente, es decir, que cada parte pueda ser reemplazada por otra que realice la misma función de manera diferente, pero que siga utilizando la misma interfaz sin afectar a los demás componentes que dependen de él. Además, esta modularidad permite la expansión de los servicios ofrecidos por la aplicación.

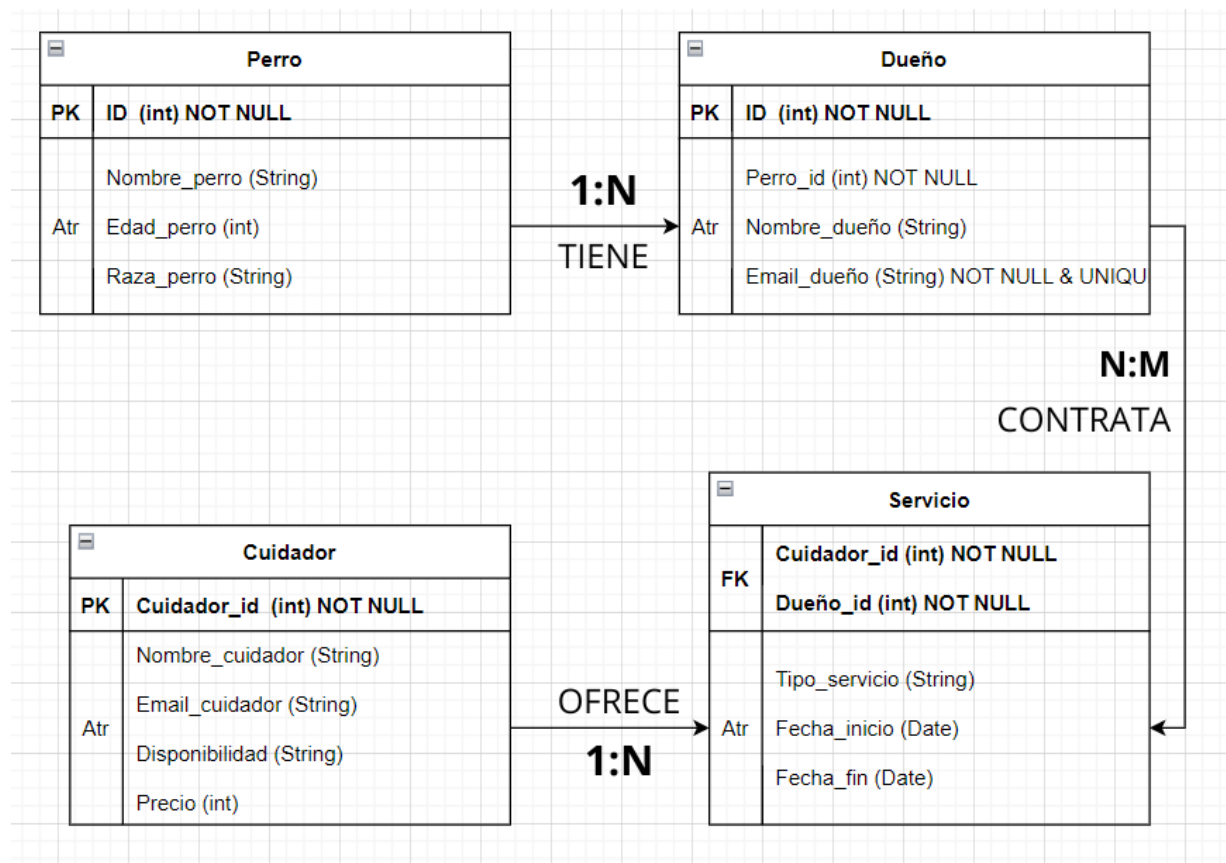
El modelo es el responsable de manejar los datos y la lógica de negocio de la aplicación. Incluye las entidades del sistema, como Dueño, Perro, Cuidador, así como las operaciones CRUD para cada una de ellas. También incluye el cálculo de tarifas para los dueños y la búsqueda de cuidadores disponibles para pasear perros. En este caso, el modelo es Firebase.

La vista se encarga de la interfaz de usuario de la aplicación. Incluye todas las páginas y pantallas de la aplicación, así como los elementos de la interfaz de usuario (botones, campos de entrada de datos y listas). La vista se comunica con el controlador para enviar y recibir información y actualizaciones. En la aplicación móvil la vista es XML, mientras que en las páginas web es HTML, CSS, JavaScript y Angular.



El controlador actúa como intermediario entre el modelo y la vista. Recibe las solicitudes de la vista y las envía al modelo para realizar la operación. Una vez realizada, el controlador envía la respuesta a la vista para actualizar la interfaz de usuario. También es el responsable de manejar el proceso de registro y el flujo de reserva de servicios. El controlador de la aplicación es Java y en la página web CRUD TypeScript.

Figura 1. Diagrama de Entidad-Relación y Diagrama de Clases:



Fuente: Elaboración propia.

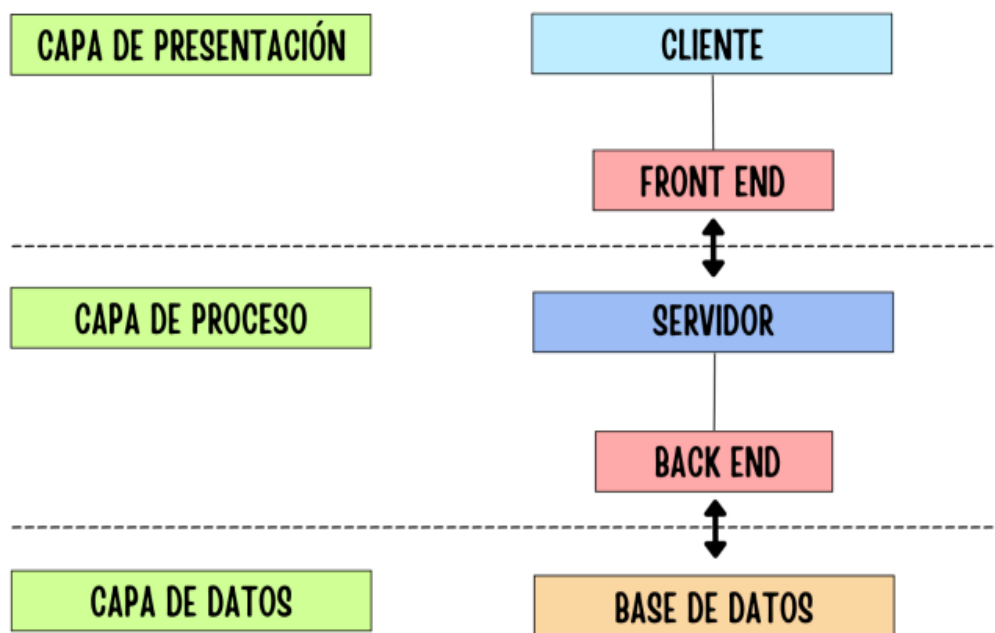
Por otra parte, la arquitectura de la página web informativa está basada en el modelo cliente/servidor. El cliente es quien inicia las solicitudes y espera la respuesta del servidor. El servidor procesa las solicitudes enviadas por el cliente para obtener información y recursos, y le envía una respuesta de vuelta. En este caso, el servidor también se encarga de manejar la autenticación y el registro de los usuarios con la base de datos.

La aplicación web se configura a través de un modelo de tres capas: capa de presentación (front-end), capa de proceso (back-end) y capa de datos (base de datos). La capa de presentación es responsable de la interfaz de usuario y la presentación de la información al usuario. La capa de



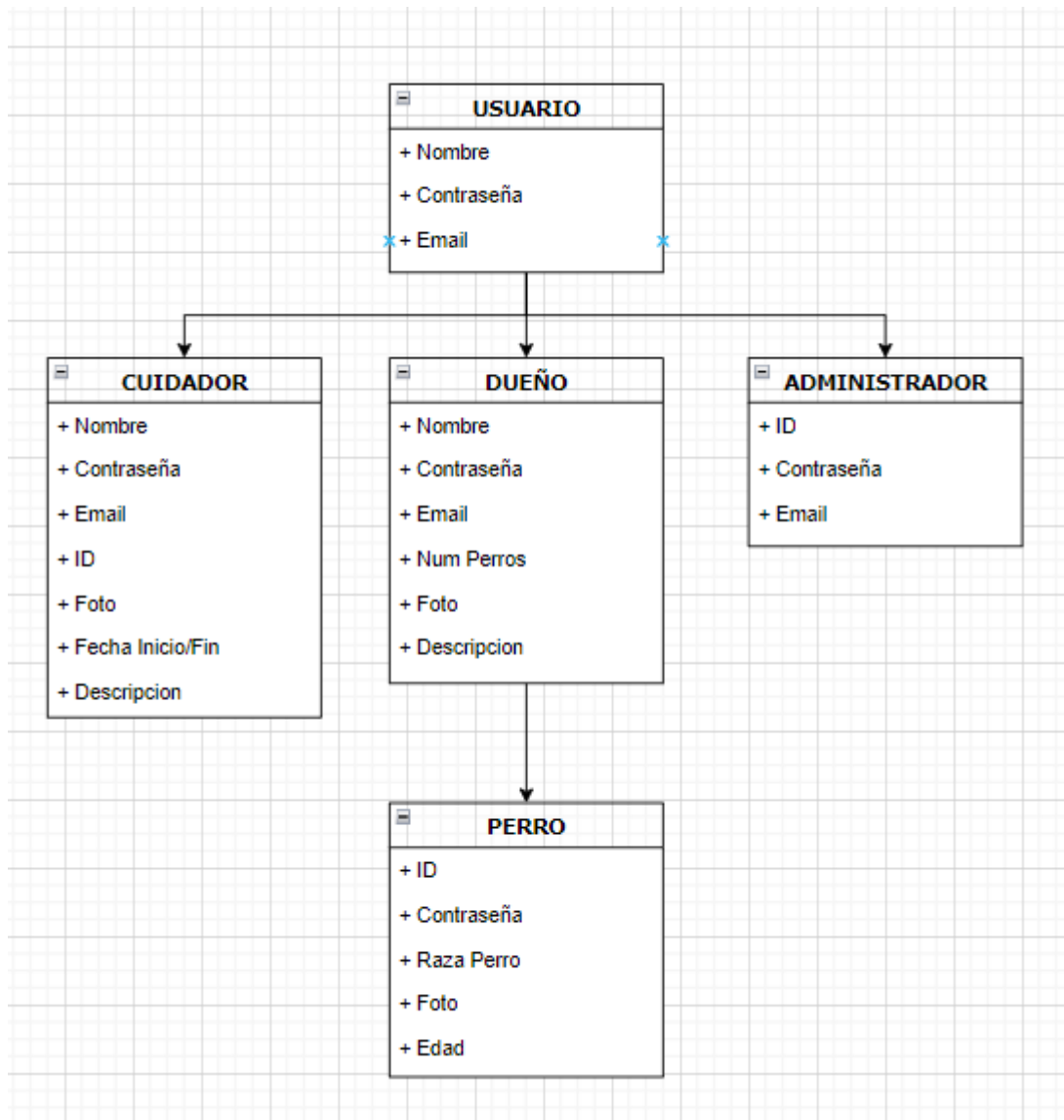
proceso es donde se realiza el procesamiento de la lógica empresarial y se toman las decisiones en función de la información recibida de la capa de presentación. Y la capa de datos es responsable del almacenamiento y recuperación de datos.

Figura 2. Diagrama de arquitectura en capas de la página web informativa:



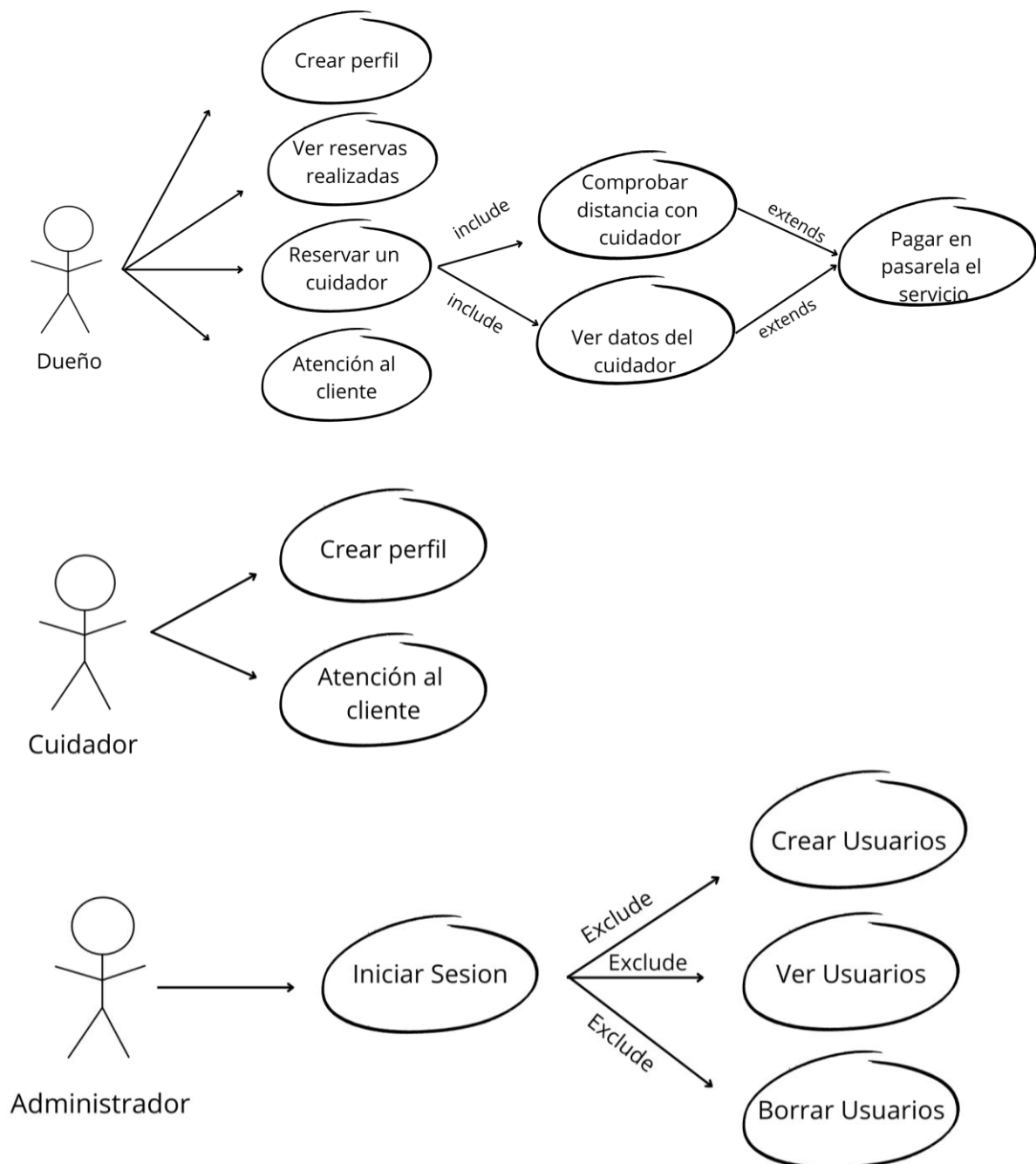
Fuente: Elaboración propia.

Figura 3. Modelo Relacional de la página web informativa:



Fuente: Elaboración propia.

Figura 4, Diagrama de casos de uso:



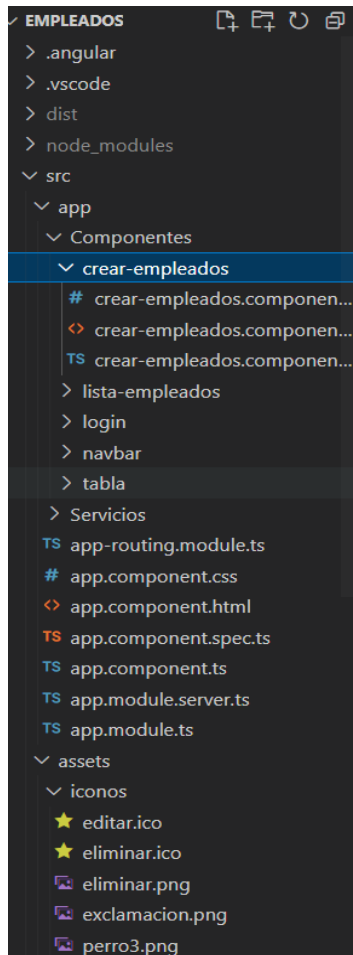
Fuente: Elaboración propia.



1.3. Diseño

Este proyecto se diseña con una arquitectura Modelo-Vista-ViewModel (MVVM): Este es un patrón popular en el desarrollo de aplicaciones de interfaz de usuario, especialmente en el contexto de frameworks como Angular, Vue.js y Android Studio. En MVVM, el ViewModel actúa como un intermediario entre la vista y el modelo, proporcionando una representación del estado de la vista que es independiente de la interfaz de usuario.

En el caso de Angular tenemos:



Dentro de la carpeta principal Empleados tenemos;

src/app: esta carpeta es donde se ubica la mayor parte del código de la aplicación Angular. Aquí se encuentran los componentes, servicios, directivas y otros elementos de Angular que forman la aplicación:

- Login.



-
- Crear empleados.
 - Lista empleados.
 - Navbar.
 - Tabla.

src/assets: donde se almacenan los recursos estáticos de la aplicación, como imágenes o archivos CSS.

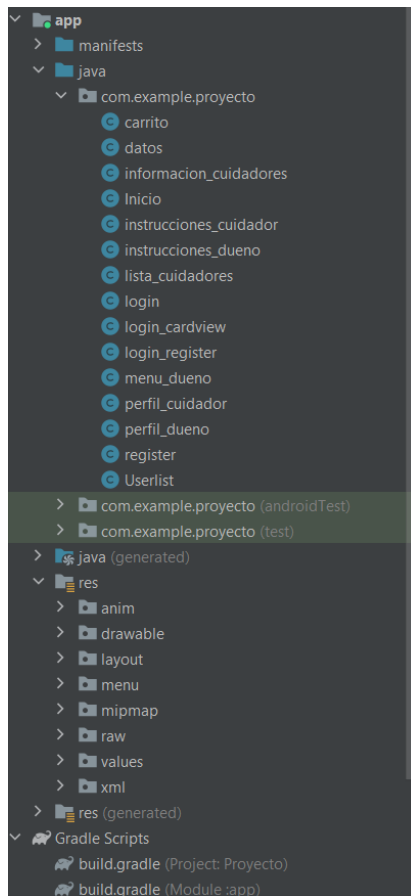
src/environments: contiene archivos de configuración para diferentes entornos como producción y desarrollo.

src/styles: contiene estilos CSS globales.

src/index.html: es el archivo HTML principal de la aplicación, donde se carga el componente raíz de Angular.

src/service: proporciona métodos para agregar, obtener y eliminar empleados de una colección en Firestore, y utiliza AngularFire para interactuar con la base de datos de Firebase.

En el caso de Android Studio tenemos:



La carpeta Java que contiene los modelos de cada layout. Dentro de la carpeta “res” se encuentra:

- Drawable con las fotos y videos que necesita el proyecto.
- Layout con los archivos de las vistas.

Build Gradle: donde se especifican las configuraciones de compilación y las dependencias del proyecto. El archivo build.gradle principal se encuentra en el directorio raíz del proyecto

1.4. Componentes

Los componentes principales de la aplicación y de las páginas web se van a dividir en dos categorías:

1.4.1. Componentes de la interfaz de usuario

Componentes de la aplicación Android:

Pantalla de inicio



En la pantalla de inicio se mostrará el logo de la aplicación y un botón que redirige a la pantalla de registro/inicio de sesión. Si no se interactúa con la pantalla, a los 5 segundos pasará automáticamente.

Ver **Ilustración 1**: prototipo pantalla de inicio (**Anexo 1**).

Registro/Inicio de sesión

Permite realizar un registro en la aplicación introduciendo nombre, email y contraseña, y seleccionar el perfil “dueño o cuidador”. Una vez registrado, el acceso se realizará a través de la pantalla inicio de sesión. En ella, se introducirán las credenciales o se iniciará sesión con Google. En caso de olvidar la contraseña, tiene la opción de recuperarla.

Ver **Ilustración 2 y 3**: prototipo pantalla de inicio de sesión y de la pantalla de registro (**Anexo 1**).

Pantalla de instrucciones

Con el perfil de dueño se mostrarán las instrucciones básicas para el correcto uso de la aplicación. Con el perfil de cuidador aparecerá un mensaje de bienvenida con recomendaciones para el usuario.

Ver **Ilustración 4 y 5**: prototipo pantalla de instrucciones cuidador y dueño (**Anexo 1**).

Perfil dueño/cuidador

Una vez leídas las instrucciones, tanto el dueño como el cuidador, tendrán acceso a una pantalla para configurar su perfil con datos de contacto, foto y otro tipo de información.

Ver **Ilustración 6 y 7**: prototipo pantalla de perfil de dueño y cuidador (**Anexo 1**).

Listado de cuidadores

Se visualizarán todos los posibles cuidadores con su foto, nombre, tarifa y dos botones, uno en el cardview para desplegar una ventana con más información y otro para reservar el servicio.

Ver **Ilustración 8**: prototipo pantalla listado de cuidadores (**Anexo 1**).

Información del cuidador

Se mostrarán de una forma más detallada los datos del cuidador y los servicios que ofrece.

Ver **Ilustración 9**: prototipo pantalla de información del cuidador (**Anexo 1**).



Pop-up de reserva

Una vez seleccionado el cuidador que se quiera contratar, se deberá clicar en el botón de reservar. Automáticamente, aparecerá un pop up donde se podrá concretar el servicio, el día y la hora e información relevante de la mascota.

Ver **Ilustración 10**: prototipo pantalla de pop-up para reservar (**Anexo 1**).

Pop-up de pre-compra

Una vez reservado, se abrirá un pop-up informando de que se ha añadido el servicio al carrito.

Ver **Ilustración 11**: prototipo de pop-up de compra del servicio (**Anexo 1**).

Mi carrito

En el carrito se podrá ver la reserva y finalizar la compra.

Ver **Ilustración 12**: prototipo pantalla de compra (**Anexo 1**).

Pop-up de pago

Una vez realizada la compra, se mostrará un mensaje de pago realizado con éxito.

Ver **Ilustración 13**: prototipo pantalla de pop-up pago realizado (**Anexo 1**).

Componentes de la página web informativa

Pantalla de inicio

En la pantalla de inicio se mostrará el logo de la aplicación y dos menús, uno desplegable y otro fijo, con un botón que te llevará a iniciar sesión.

Ver **Ilustración 14 y 15**: prototipo pantalla de inicio y menú (**Anexo 2**).

Inicio de sesión

Se podrá iniciar sesión para ver la información específica del usuario, en caso de haberse creado una cuenta con anterioridad.

Ver **Ilustración 16**: prototipo pantalla de inicio de sesión (**Anexo 2**).

Registro

Si el usuario no tiene cuenta podrá registrarse como dueño o cuidador rellenando un formulario muy sencillo con sus datos.

Ver **Ilustración 17**: prototipo pantalla de registro (**Anexo 2**).



Calculadora de ubicación

La web se conectará con la página <https://nominatim.openstreetmap.org/ui/search.html> para visualizar un mapa que calculará la distancia entre el cuidador y el dueño. Para ello, se deberá insertar la dirección que proporciona el cuidador y automáticamente la web detectará la ubicación del dueño en ese mismo momento.

Desde la aplicación móvil habrá un botón que redireccionará a la página de la calculadora de ubicación.

Ver **Ilustración 18**: prototipo calculadora de ubicación (**Anexo 2**).

Servicios de los cuidadores

Habrán 4 tipos de servicios que podrán ofrecer los cuidadores para el cuidado de los perros: (cita con el veterinario, paseos, 24 horas y cita en la peluquería). Seleccionando la opción que desee podrá ver su contenido.

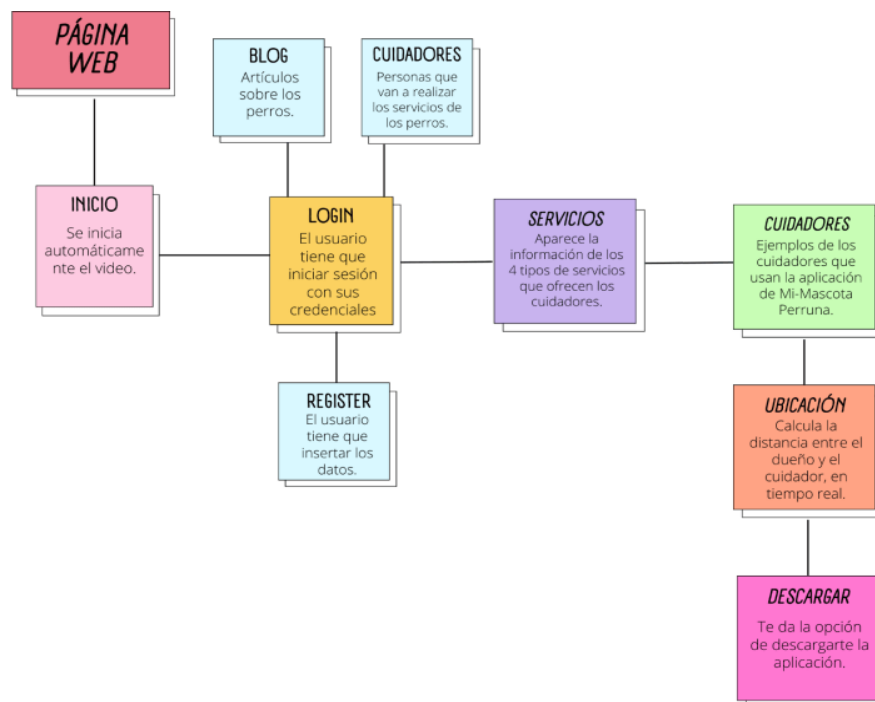
Ver **Ilustración 19**: prototipo pantalla de servicios (**Anexo 2**).

Listado de cuidadores

Se visualizará un ejemplo de cuidadores, con su foto, nombre y más información.

Ver **Ilustración 20**: prototipo de listado de cuidadores (**Anexo 2**).

Figura 5. Esquema de los componentes de la página web informativa.



Fuente: Elaboración propia.

Componentes de la página web CRUD

Inicio de Sesión Administrador

El Administrador podrá iniciar sesión con su usuario y contraseña únicos.

Listado CRUD

Los cuidadores y dueños que se registren aparecerán en un listado y al pinchar encima de cada uno se abrirá una página con los datos de cada persona. Además, habrá dos botones: uno para eliminar a cuidadores o dueños del listado y otro para crearlos.

1.4.2. Componentes de la lógica de negocio

Autenticación y registro de usuarios

A través de Firebase se verifican las credenciales de inicio de sesión y se crean cuentas de usuario tanto en la página web informativa como en la aplicación. De esta manera, si un usuario está previamente registrado en la página web, los datos quedarán almacenados y no será necesario volver a registrarse en la aplicación, y viceversa.



Uso de interfaz de programación de aplicaciones (API's)

Permiten iniciar sesión con la cuenta de Google, unir la página web informativa con la aplicación y mejorar la seguridad de las mismas.

Píxeles

Uso de píxeles en aplicación Meta para obtener datos y métricas más detalladas. Es un gráfico de 1×1 píxel que se utiliza para realizar un seguimiento del comportamiento del usuario, las conversiones del sitio, el tráfico web y otras métricas a nivel del servidor de un sitio.

Visualización de información de cuidadores y reserva de servicios

Posibilita el almacenamiento de la información de los usuarios y de las reservas en Firebase.

Mensajería

Notifica vía email las credenciales de inicio de sesión y de la reserva realizada con los datos del cuidador y muestra un pop-up, en caso de olvidar la contraseña y se enviará un link para recuperarla.

Pasarela de pago

Facilita al usuario pagar a través de la app utilizando sistemas de pago confiables.

Publicidad:

Uso de anuncios y panfletos para dar a conocer la aplicación.

Pluggins:

Extensiones para detectar errores, dar mayor fluidez a la página informativa y poder obtener métricas más detalladas.

1.5. Tecnologías y lenguajes

La elaboración de las páginas web se van a usar varios lenguajes de programación diferentes. HTML se va a utilizar para estructurar el contenido, definir encabezados, párrafos, imágenes, CSS se va a emplear para definir el estilo visual, como por ejemplo la fuente y el tamaño del texto, colores y disposición de los elementos en pantalla, JavaScript y Angular junto con TypeScript para añadir interactividad y dinamismo a las páginas, creando efectos visuales y permitiendo interactuar con el contenido en tiempo real.

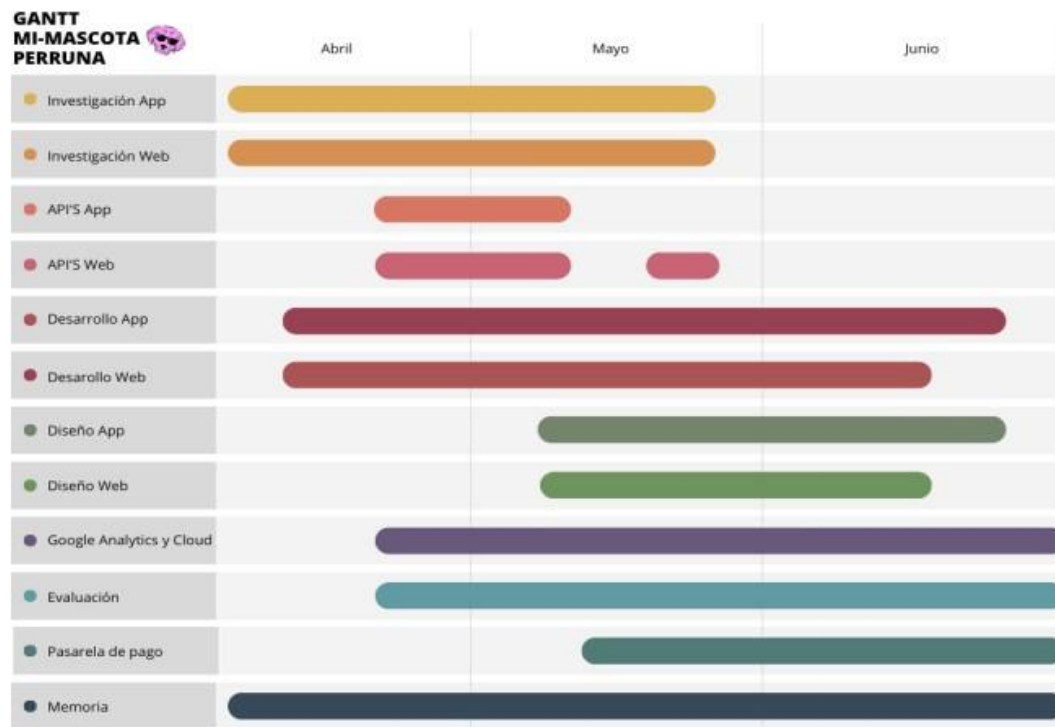
Por otra parte, para la aplicación móvil se va a utilizar el lenguaje XML para la creación de diseños de la interfaz del usuario, definir los elementos visuales y su disposición en la pantalla y recursos como colores, estilos, dimensiones y cadenas de texto; y Java para escribir el código de



la interfaz de usuario de la aplicación, donde se describen las clases, métodos, variables y estructuras de control necesarias para el funcionamiento de la aplicación.

1.6. Planificación

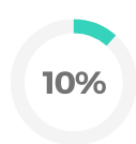
Figura 6. Tiempo que voy a emplear en cada parte del proyecto.



Fuente: Elaboración Propia



PÁGINA WEB INFORMATIVA MI-MASCOTA PERRUNA



INICIO

17 HORAS
DEDICADAS



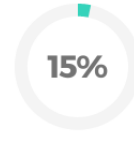
UBICACIÓN

20 HORAS
DEDICADAS



DISEÑO

20 HORAS
DEDICADAS



SERVICIOS

12 HORAS
DEDICADAS

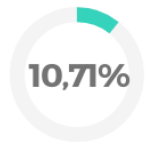


PÁGINAS

17 HORAS
DEDICADAS

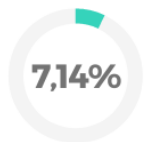
Fuente: Elaboración Propia

APLICACIÓN MI-MASCOTA PERRUNA



MENU DUEÑO

25 HORAS
DEDICADAS



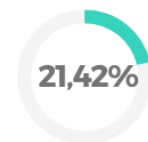
MENU CUIDADORES

20 HORAS
DEDICADAS



CARRITO

35 HORAS
DEDICADAS



LISTADO CUIDADORES

30 HORAS
DEDICADAS



LOGIN / REGISTER

25 HORAS
DEDICADAS



POPUPS

10 HORAS
DEDICADAS



HERRAMIENTAS FIREBASE

30 HORAS
DEDICADAS



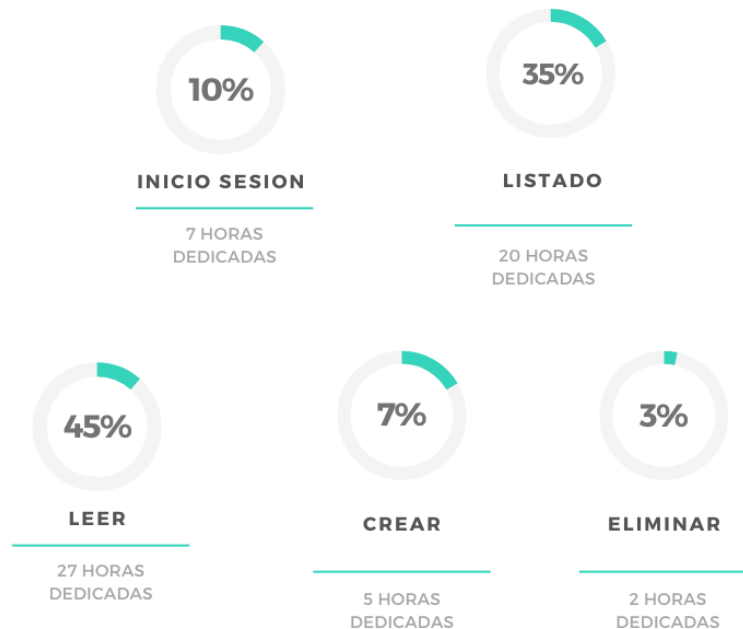
EXTENSIÓN

15 HORAS
DEDICADAS

Fuente: Elaboración Propia



PÁGINA WEB CRUD MI-MASCOTA PERRUNA



Fuente: Elaboración Propia

1.7. Herramientas

Este proyecto se va a desarrollar mediante la utilización de las siguientes herramientas:

Android Studio

IDE (Integrated Development Enviroment) que proporciona una amplia gama de herramientas y funcionalidades como el emulador de dispositivo móvil, herramientas de depuración y desarrollo de pruebas.

Visual Studio Code

Una herramienta que permite editar el código HTML, CSS , Angular y TypeScript de la página web.

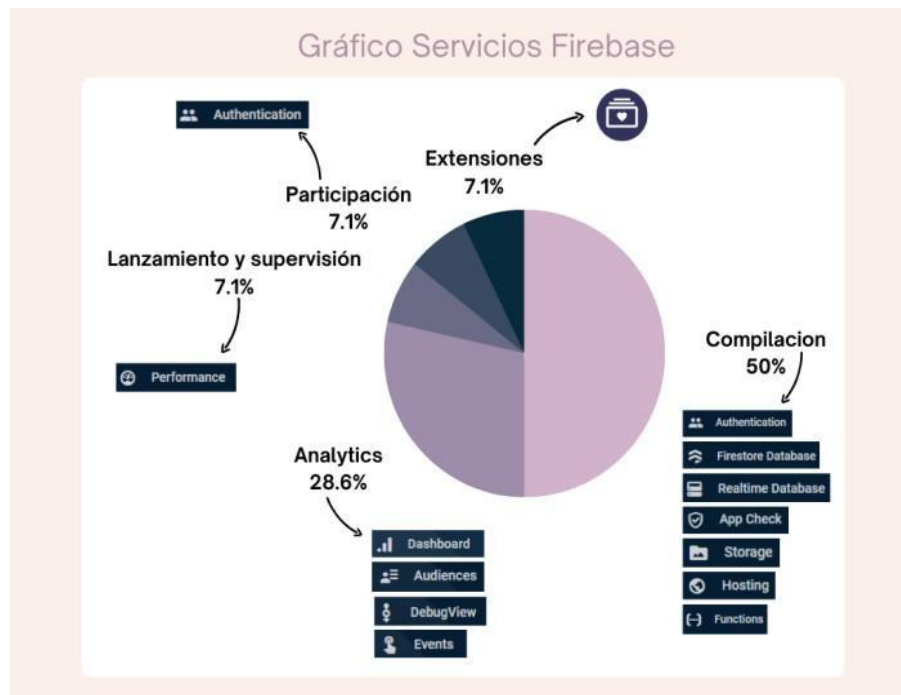
Firebase

Android Studio se conecta con Firebase, una plataforma de desarrollo móvil de Google, que facilita la gestión de datos, almacenamiento en la nube, autenticación de usuarios, base de datos en tiempo real y análisis. Además, permite el uso de APIs, plugins, herramientas de seguridad,



entre otros. La página web también se conecta a Firebase para obtener un dominio y hosting gratuito que ofrece estadísticas y datos.

Figura 7. Servicios Firebase.



Fuente: Elaboración Propia

Herramientas hosting y app check

Herramientas para la utilización de un hosting y dominio gratuito que aporte seguridad tanto a la página web informativa como a la aplicación. Del mismo modo, la API de Google Service, reCAPTCHA Enterprise, comprueba y asegura la identidad de los usuarios, evitando bots.

Node JS

Un entorno de ejecución para JavaScript con el que se puede leer, editar, escribir y ejecutar un proyecto. En este caso, se ha usado NodeJS en la terminal para subir la página web informativa a la herramienta de hosting en Firebase y así ver los cambios que se han realizado.

Plugins de Google

Tag Assistant Companion o Google Analytics Debugger son plugins que sirven para depurar las distintas ventanas que tenga la página web o app. Se conectan al Google Analytics mediante la API y el ID del proyecto de Google Cloud. Una vez esté en funcionamiento el proyecto con las



extensiones, se puede ver una gran variedad de métricas y datos que se podrán comparar, exportar, etc.

Figma

Herramienta para crear prototipos web interactivos empleada para la elaboración de la propuesta del proyecto.

Canva

Herramienta online de diseño gráfico que permite añadir y crear diseños con mayor calidad y facilidad de uso.

Envato Elements

Tienda online que ofrece productos digitales relacionados con diseños e ideas para optimizar la app y la página web informativa.

Kaiber AI

Herramienta que permite elaborar vídeos a través de inteligencia artificial. Se ha empleado para la elaboración de un video en la página de inicio de la web informativa.

1.8. Integración de Firebase en el proyecto

1. Conexión entre Firebase y la aplicación de Android Studio:

- La aplicación de Android Studio se conecta a Firebase utilizando el SDK de Firebase para Android.
- La configuración de Firebase se realiza en el proyecto de Android Studio, donde se agrega el archivo de configuración de Firebase (google-services.json). Esto permite que la aplicación se comuniquen con los servicios de Firebase, como la base de datos en tiempo real, el almacenamiento en la nube, la autenticación, etc.

2. Integración de Stripe con Firebase:

- Firebase ofrece una extensión para integrar Stripe como pasarela de pago en la aplicación. La extensión de Stripe en Firebase simplifica la configuración y el procesamiento de pagos mediante la generación de un backend seguro y escalable. Los usuarios pueden realizar pagos a través de la aplicación utilizando las API de Stripe y Firebase se encarga de la comunicación con la pasarela de pago.

3. Vinculación con Google Cloud utilizando el ID del proyecto:



- Firebase y Google Cloud están estrechamente relacionados, ya que Firebase es parte de la plataforma de Google Cloud.
 - Al crear un proyecto en Firebase, se genera automáticamente un proyecto correspondiente en Google Cloud.
 - Firebase utiliza los servicios de infraestructura y almacenamiento de Google Cloud para respaldar sus características como las bases de datos en tiempo real y el almacenamiento en la nube.
4. Vinculación de Firebase con Google Analytics:
- Firebase está integrado con Google Analytics, lo que permite recopilar y analizar datos sobre el uso de la aplicación.
 - A través de Google Analytics, se pueden obtener métricas, estadísticas y datos relacionados con el rendimiento de la aplicación, la adquisición de usuarios, el comportamiento de los usuarios, etc.
 - Esta integración proporciona información valiosa para tomar decisiones informadas sobre mejoras en la aplicación y estrategias de marketing.

A continuación, se muestran dos gráficos separando las herramientas que se han usado en la aplicación y en la página web informativa:

Figura 8. Herramientas utilizadas para la elaboración de la página web.



Fuente: Elaboración propia.

Figura 9. Herramientas utilizadas para la elaboración de la aplicación.



Fuente: Elaboración propia.



CONCLUSIONES

Durante el desarrollo del proyecto, se encontraron numerosos desafíos que se resolvieron mediante investigación, consulta de documentación oficial y prueba y error. Uno de los mayores obstáculos del proyecto fue la integración de las API's, la conexión de Firebase con la web y la aplicación, la configuración del hosting y dominio gratuitos a través de Firebase y la pasarela de pago.

En el trabajo de desarrollo de la aplicación, se tuvieron que abordar varios retos, como el manejo de una misma base de datos para cuidadores y dueños, el diseño de un sistema de registro único por correo electrónico para cada usuario y la creación de un listado de cuidadores compatible con la aplicación. En cuanto a la página web informativa, el mayor desafío fue lograr que las pantallas se adaptaran a diferentes dispositivos y conectar la API de Google Maps para mostrar un mapa que calculara la distancia entre el propietario y el cuidador. En lo que respecta a la página web CRUD del administrador, se tuvo que enfrentar el tema de la seguridad en el inicio de sesión y la gestión de las bases de datos con las listas de cuidadores y propietarios.

Aunque el proyecto presentó limitaciones de tiempo y requirió esfuerzos adicionales, ha sido gratificante poder lograr los objetivos propuestos. Desde el principio, el proyecto era muy ambicioso y requeriría de mucha dedicación y habilidad en resolución de problemas. El resultado final, ha sido poder elaborar con éxito una aplicación y dos páginas web funcionales, integrando más de 6 API's, utilizando 9 herramientas de Firebase y explorando el seguimiento de 3 aplicaciones.

A nivel académico, el proyecto me ha ofrecido la oportunidad de mejorar mis habilidades de programación, superar mis expectativas y no poner límites a mis objetivos.

1.9. Pruebas, resultados y validaciones

Comprobaciones de la aplicación.

Para comprobar el correcto funcionamiento de la aplicación se han realizado una serie de pruebas. Para el registro se utilizó el correo prueba@gmail.com, el usuario del cuidador es: cuidadorprueba@gmail.com y el usuario del dueño: dueñoprueba@gmail.com ambos con contraseña 123456.

- Pantalla de inicio:

Al compilar la aplicación se debe observar que no se produce ningún error y verificar el funcionamiento del cronómetro para que se inicie automáticamente la pantalla de inicio de sesión.

- Pantalla Registro/Inicio de sesión:



Asegurarse que el estilo de la aplicación se mueve y que podemos pasar del inicio de sesión al registro moviendo la pantalla con el dedo. Además, hay que revisar la parte interna viendo cómo se conecta Firebase con la aplicación para la autenticación de usuarios. Se tienen que registrar “cuidador/dueño” e identificarse.

Al clicar en resetear contraseña debe saltar un Pop-Up para introducir el correo electrónico y cuando se dé a aceptar, se reciba un mensaje personalizado que permita cambiar la contraseña.

Al pulsar en el botón de Google se tiene que abrir una pantalla para poder insertar la cuenta de Google e iniciar automáticamente la aplicación.

- Pantalla del perfil del cuidador:

Iniciada la actividad se comprueba que se pueden insertar todos los datos correctamente y subir una foto al firestore.

A través de la herramienta realtime database se suben los datos al firebase para que se incorporen automáticamente en la opción listado cuidadores al identificarse como dueño.

- Instrucciones dueño y cuidador:

Una vez inicie sesión, se activará la clase de instrucciones.

- Pantalla del dueño:

Se comprobará que se despliega un menú donde el usuario puede ver varias opciones, entre ellas crear un perfil de dueño donde se subirán los datos con realtime database.

En caso de tener una reserva ya hecha, deberá saltar un recordatorio en la pantalla con los datos del cuidador y servicio.

- Pantalla del listado de cuidadores:

Con la herramienta realtime database de firebase se puede ver en tiempo real las creaciones o modificaciones que se hacen. Por lo tanto, al insertar los datos en la clase perfil cuidador, automáticamente dichos datos se muestran en la actividad listado cuidadores donde se muestran todas las personas que quieren cuidar perros.

Al pinchar encima de la tarjeta de un cuidador se abre otra actividad que ofrece más información. Al seleccionar un cuidador y darle a aceptar se abre un pop-up con las opciones de cancelar o reservar. Si se selecciona “reservar” se enviarán los datos del servicio y del cuidador a la clase de carrito para realizar la compra.

- Información del cuidador:

En esta clase hay que ver que se envían correctamente todos los datos del cuidador cuando pinchamos encima de la tarjeta.

- Carrito de compra:



Se tienen que pasar los datos del servicio y cuidador seleccionado y realizar correctamente la compra recibiendo automáticamente un email al correo.

Finalmente, se visualizarán las estadísticas de Firebase asegurándose a la vez de que está bien conectada con la aplicación.

Comprobaciones de la página web informativa.

- **Pantalla de Inicio:**

Al iniciar la página web debe reproducirse el video de la pantalla de inicio y, una vez que finalice, pasar a la pantalla de inicio de sesión.

Al clicar en el submenú de la web, se confirmará que se abren las 5 opciones que el usuario puede elegir.

- **Inicio de sesión:**

Se comprobará que funciona el botón de registrar redirigiendo a una nueva página llamada registro. Al clicar el botón de contraseña olvidada, debe saltar un pop-up para introducir el correo electrónico y cuando se da a aceptar, se recibe un mensaje personalizado que permite cambiar la contraseña.

La identificación tiene que estar correctamente conectado con Firebase y verificar que el usuario no puede volverse a registrar con el mismo usuario con el que ha iniciado en la aplicación. Al ser un usuario existente debe iniciar sesión con sus credenciales.

Observar en la parte inferior de la página web el blog y la rotación de los cuidadores.

- **Registro:**

Los datos que se vayan insertando se tienen que ver reflejados en el Firebase.

- **Cuidadores:**

Se mostrarán los listados con todos los perfiles de los cuidadores con los datos e información de sus perros.

- **Servicios:**

Se mostrará la información de los 4 servicios que van a realizar los cuidadores a los dueños de los perros.

- **Ubicación:**

Una vez que se acepta la ubicación actual del usuario, se inserta la dirección del dueño y al pulsar en el botón de mostrar el mapa se calcula la distancia que hay entre las dos coordenadas.



En el Anexo 4 se muestran las tarjetas de comprobación de la aplicación (ver ilustraciones 42-48).

Para realizar estas pruebas y obtener estadísticas sobre el funcionamiento de la web, se utilizan dos plugins: Tag Assistant Companion y Google Analytics Debugger. Estos plugins están conectados a la herramienta Performance de Firebase, que proporciona datos detallados y estadísticas sobre el rendimiento de la web.

Para comprobar el correcto funcionamiento de la aplicación se usa la herramienta Performance de Firebase sin extensiones. Además del SDK con Firebase y a su vez con Google Cloud, podremos ver si hay alguna variación en las métricas, datos y estadísticas del proyecto.

1.10. Explotación

Estrategia de SEO:

- Utiliza palabras clave relevantes en el contenido de la página web y aplicación móvil. Por ejemplo, incluye términos como "cuidadores de perros", "servicio de cuidado de mascotas", "paseadores de perros" y otros similares, usando la extensión de Google: Keyword Planner.
- Crea contenido útil y relevante en el blog de la web. Elaboración de guías de cuidado de perros, consejos para entrenamiento o recomendaciones de productos. Esto ayuda a posicionar el sitio web como una de las mejores en el cuidado de perros y atrae tráfico orgánico.
- Optimiza la velocidad de carga del sitio web y aplicación móvil para mejorar la experiencia del usuario y el posicionamiento en los motores de búsqueda usando la extensión Google PageSpeed Insights.

Publicidad en línea:

- Utilización de Google Adds para crear anuncios de búsqueda y anuncios de display dirigidos a personas interesadas en servicios de cuidado de perros. Se puede segmentar los anuncios por ubicación geográfica y palabras clave relevantes.
- Uso de anuncios en diferentes canales para mostrar publicidad en sitios web y aplicaciones relacionadas con mascotas y cuidado de animales. Esto ayuda a llegar a una audiencia específica y aumentar la visibilidad del negocio. Algunos canales o herramientas utilizadas son Display programático (Placebo, Outbrain), RRSS (Business Manage Meta) y Reporting (Looker Studio).



Marketing de contenidos:

- Crear contenido valioso y atractivo en canales de redes sociales, como consejos de cuidado de perros, historias de éxito de cuidadores o testimonios de clientes satisfechos.
- Colaboración con influencers y bloggers especializados en mascotas para promocionar el servicio.
- Los usuarios pueden realizar reseñas de la aplicación o compartir su experiencia utilizando cuidadores.

Panfletos y publicidad tradicional:

- Diseño de panfletos atractivos para distribuirlos en lugares estratégicos, como parques para perros, clínicas veterinarias, tiendas de mascotas y cafeterías amigables con las mascotas.
- Anuncio del servicio en periódicos locales o revistas especializadas en mascotas.

En el **Anexo 5** se muestran los anuncios publicitarios de la aplicación (**ver ilustraciones 49-53**).

1.11. Presupuesto

Para la elaboración del presupuesto se tendrán en cuenta una serie de factores. Por un lado, las funcionalidades que incluye la aplicación y la tecnología que se utilizará para desarrollarla y, por otro, los costes adicionales relativos a la Inversión en marketing como, por ejemplo, el coste de publicar la app en Play Store, uso de las RRSS, publicación de anuncios en meta y su posicionamiento en tiendas.

Funcionalidades:

- Creación de una app para Android tipo tienda online, intuitiva con un diseño claro y atractivo.



Principales características de la app	Componentes de la aplicación Android
Geolocalización en un mapa	Pantalla de inicio
Accesibilidad	Login/Register
Seguridad verificación en dos pasos	Pantalla de instrucciones
Subida de archivos	Perfil dueño/cuidador
Estrategia de monetización mediante los servicios que compran los dueños y las descargas de la app	Listado de cuidadores
Desarrollo de una opción multi idioma para poder utilizarla internacionalmente.	Información del cuidador
Seguimiento de uso de la aplicación para su mejora (mantenimiento, actualizaciones y corrección de errores)	Pop-up de reserva
	Pop-up de pre-compra
	Mi carrito
	Pop-up de pago

Fuente: Elaboración propia.

- Diseño a medida de una página web informativa para aumentar las descargas de la app.

Principales características de la Web	Componentes de la página web
Diseño a medida	Pantalla de inicio
Con más de 10 págs.	Inicio de sesión
Con registro de usuarios	Registro
Multiidioma	Calculadora de ubicación
Cumplimiento RGPD	Servicios de los cuidadores
Con directorio o buscador interno	Ranking de cuidadores
Con plataforma de pago	Perfil dueño-cuidador

Fuente: Elaboración propia.



- Diseño a medida de una página web del administrador para que pueda editar, crear, ver y borrar usuarios.

Principales características de una página web administrador
Login administrador
Leer listado cuidadores/dueños
Editar listado cuidadores/dueños
Borrar listado cuidadores/dueños
Crear listado cuidadores/dueños

Fuente: Elaboración propia.

Costes fijos	Importe	Costes variables	Importe
Nombre de dominio web informativo.	10€/año	Alojamiento web en Hosting	70€/año
Diseño app, web informativa y administrador.	2.500 €	Herramientas de Firebase	mensual
Desarrollo app, web informativa y administrador.	27€/hora	Posicionamiento SEO y promoción en redes sociales	a partir de 500€ al mes
Publicación app en Play Store	20€ + 30% de comisión por venta	Mantenimiento	del 10% al 25% del pto.
		Generación de pasarela de pago	del 1,5% al 2% del pto.

Fuente: Elaboración propia.

El presupuesto total para la elaboración de las páginas web y la aplicación incluyendo los costes asociados a marketing y publicidad será aproximadamente de unos 5000-7000€.

1.12. Trabajo futuro

El proyecto tiene un gran potencial de crecimiento y mejoras de funcionalidades tanto en la página web informativa como en la aplicación. La limitación de tiempo para su elaboración ha impedido implementar mejoras adicionales. A continuación, se mencionan algunas de las funcionalidades que se podrían considerar a futuro:

- Intercambiar la base de datos de Firebase por una más grande como phpmyadmin con características avanzadas y soporte completo de SQL.
- Posibilidad de reservar más de un servicio a la vez.
- Funcionalidad de chat online que permita poner en contacto al dueño con el cuidador.



-
- Instalación de plugins y extensiones de pago que faciliten el funcionamiento, SEO y rendimiento de la aplicación y/o web.
 - Habilitar un botón de idioma que permita traducir la aplicación y la página web.
 - Desarrollar la App en una aplicación IOS como Xcode para ampliar el Target de usuarios y obtener más beneficios económicos.
 - Añadir en la aplicación una opción para filtrar la búsqueda de cuidadores según sus características.



BIBLIOGRAFÍA Y WEBGRAFÍA

Introducción:

Food, A. P. (2015, 5 febrero). Estadísticas del Mundo del Pet Food ¿Qué debes Conocer? ¿Cómo Adaptarte y Aprovecharlas a tu Favor? All Pet Food.

<https://allextruded.com/entrada/estadisticas-del-mundo-del-pet-food-que-debes-conocercomo-adaptarte-y-aprovecharlas-a-tu-favor-22808#:~:text=Las%20estad%C3%ADsticas%20mundiales%20afirman%20que,un%20animal%20a%20su%20cargo.>

Figma:

Figma. (s.f.-b). Figma.

[https://www.figma.com/proto/Lic3jaBZphDE0x4XXZxBtT/Página-Web\(Community\)?node-id=3-2&scaling=scale-down&page-id=0:1&starting-pointnode-id=3:2](https://www.figma.com/proto/Lic3jaBZphDE0x4XXZxBtT/Página-Web(Community)?node-id=3-2&scaling=scale-down&page-id=0:1&starting-pointnode-id=3:2)

Página web (community). (s/f). Figma. de

[https://www.figma.com/proto/Lic3jaBZphDE0x4XXZxBtT/P%C3%A1gina-Web\(Community\)?node-id=3-2&scaling=scale-down&page-id=0%3A1&starting-pointnodeid=3%3A2](https://www.figma.com/proto/Lic3jaBZphDE0x4XXZxBtT/P%C3%A1gina-Web(Community)?node-id=3-2&scaling=scale-down&page-id=0%3A1&starting-pointnodeid=3%3A2)

Login y Register:

Creando nuestro sistema de registro y login: Android (I) - Nosinmiubuntu | Ubuntu en concreto, GNU/Linux en general. (s.f.). Nosinmiubuntu | Ubuntu en concreto, GNU/Linux en general.

<https://www.nosinmiubuntu.com/creando-nuestro-sistema-de-registro-y/>

Codes Easy. (2022, 7 de diciembre). Login and Registration using Firebase in Android [Video]. YouTube. <https://www.youtube.com/watch?v=QAKq8UBv4GI> **Generar APK y Firmar**

APP:

Firma tu app | Android Studio | Android Developers. (s.f.). Android Developers.

<https://developer.android.com/studio/publish/app-signing?hl=es-419>

Arif, A. (2021, 29 de agosto). Cómo generar archivos APK y APK firmados en Android Studio. Code Envato Tuts+.

<https://code.tutsplus.com/es/tutorials/how-to-generate-apk-and-signedapk-files-in-android-studio--cms-37927>

Plugins:

Tag Assistant Companion. (s.f.). Google Chrome - Download the Fast, Secure Browser from Google.

<https://chrome.google.com/webstore/detail/tag-assistantcompanion/jmekfmbnaedfebfnmakmokmlfpblbfdm>



Google Analytics Debugger. (s.f.). Google Chrome - Download the Fast, Secure Browser from Google.

<https://chrome.google.com/webstore/detail/google-analyticsdebugger/jnkmfdileelhofcijamephohjechna>

Analytics:

Envato Elements: vídeos, música, fotos y gráficos de archivo ilimitados. (s/f). Envato Elements, de <https://elements.envato.com/es/>

Kaiber. (s/f). Kaiber.Ai. de <https://www.kaiber.ai/>

(S/f). Canva.com, de <https://www.canva.com/>

Performance monitoring:

Primeros pasos con Performance Monitoring para Android. (s/f). Firebase, de

<https://firebase.google.com/docs/perf-mon/get-started-android?authuser=0&hl=es>

<https://www.youtube.com/watch?v=k3zoVAMuW5w>

Debug View:

[GA4] Monitorizar eventos en DebugView. (s/f). Google.com. Recuperado el 22 de mayo de 2023, de https://support.google.com/analytics/answer/7201382?hl=es&utm_id=ad

Diseño:

Envato Elements: vídeos, música, fotos y gráficos de archivo ilimitados. (s/f). Envato Elements, de <https://elements.envato.com/es/>

Kaiber. (s/f). Kaiber.Ai. de <https://www.kaiber.ai/>

(S/f). Canva.com, de <https://www.canva.com/>

Arquitectura del sistema:

(S/f-b). Uam.es. de

https://repositorio.uam.es/bitstream/handle/10486/662281/gomez_matesanz_alfonso_tfg.pdf?sequence=1&isAllowed=y

Arquitectura de las aplicaciones Web. (2013, noviembre 14). Programacion Web.

<https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>

Diagrama de clases:

Diagrama UML: Qué es, cómo hacerlo y ejemplos. (s/f). <https://miro.com/>, de <https://miro.com/es/diagrama/que-es-diagrama-uml/>

Documents — SmartDraw. (s/f). Smartdraw.com, de <https://cloud.smartdraw.com/?nsu=1>

Lista/RecyclerView:



WorldClub, A. [@AndroidWorldClub]. (2018, diciembre 14). Custom listView in android with item click. Youtube. <https://www.youtube.com/watch?v=5Tm--PHhbJo&list=PLhRVVZ-e6n9XSjSCv7LdzUhjUPqJ8bChM&index=2>

AristiDevs, P. A. by [@AristiDevs]. (2022, enero 13). Tutorial RECYCLERVIEW DEFINITIVO en ANDROID 📱 con KOTLIN en español + GRID [Android en Kotlin 2022]. Youtube. <https://www.youtube.com/watch?v=k3zoVAMuW5w>

RealTime Database:

Firebase realtime database. (s/f). Firebase, de <https://firebase.google.com/products/database/> [23] Documentación Firebase (2017). Structure Your Database. Consultada el 15 de junio de 2017, en <https://firebase.google.com/docs/database/web/structuredata?>

Hosting y Node.js:

Referencia de Firebase CLI. (s/f). Firebase, de <https://firebase.google.com/docs/cli?hl=es-419>

Firebase Hosting. (s/f). Firebase, de <https://firebase.google.com/docs/hosting?hl=es-419>

reCAPTCHA:

Verificación de aplicaciones de. (s/f). Firebase, de <https://firebase.google.com/docs/app-check?hl=es-419>

Comienza a usar la Verificación de aplicaciones con reCAPTCHA v3 en apps web. (s/f). Firebase, de <https://firebase.google.com/docs/app-check/web/recaptcha-provider?hl=es-419>

reCAPTCHA v3. (s/f). Google for Developers, de <https://developers.google.com/recaptcha/docs/v3?hl=es-419>

Implementa reCAPTCHA Enterprise para la integración de WAF y Google Cloud Armor. (s/f). Google Cloud, de <https://cloud.google.com/recaptcha-enterprise/docs/implement-tokens?hl=es-419>

Androide, S. [@SociedadAndroide]. (2017, noviembre 29). Firebase - Como subir tu propia pagina al hosting de firebase. Youtube. <https://www.youtube.com/watch?v=drvXMz75k5s>

Instala claves de sitios basadas en puntuaciones (sin desafíos) en sitios web. (s/f). Google Cloud, de <https://cloud.google.com/recaptcha-enterprise/docs/instrument-web-pages?hl=es-419>

Crea una evaluación. (s/f). Google Cloud, de https://cloud.google.com/recaptcha-enterprise/docs/create-assessment?hl=es-419&_ga=2.186762728.-1365570470.1678553625

Iniciar sesión con Google:

Skillz, T. [@technicalskillz]. (2020, abril 14). Google sign in | Login with Google using firebase in android app without error | exception. Youtube. <https://www.youtube.com/watch?v=AUOZXy28CJ0&list=PLYx38U7gxBf3lFg7YVv-mEOY27J9T-cIU&index=3>



Icono:

¿Por qué el ic_launcher de mi aplicación queda con fondo? (s. f.). Stack Overflow en español. <https://es.stackoverflow.com/questions/182306/por-qu%C3%A9-el-ic-launcher-de-mi-aplicaci%C3%B3n-queda-con-fondo>

La cueva del programador. (2022, 2 enero). Android Studio.- Cambiar el ícono y nombre de tu aplicación [Vídeo]. YouTube. <https://www.youtube.com/watch?v=aUPgYLfl6f0>

Errores:

Mensaje conexión reCAPTCHA. (s/f). Foro WordPress en Español, de <https://www.webempresa.com/foro/preguntas-sobre-prestashop/mensaje-conexion-recaptcha>
Firebase deploy error and firebase.json empty file. (s/f). Stack Overflow, de <https://stackoverflow.com/questions/44323081/firebase-deploy-error-and-firebase-json-empty-file>

Media Player (música):

Descripción general de MediaPlayer. (s/f). Android Developers, de <https://developer.android.com/guide/topics/media/mediaplayer?hl=es-419>

De Ernesto, L. G. [@LaGeekipediaDeErnesto]. (2018, mayo 11). Curso Android desde cero #39 | Reproducción de audio – Las clases MediaPlayer y SoundPool. Youtube. <https://www.youtube.com/watch?v=shzbPeL21cA>

Menú Action Bar:

¿Como hacer para definir un título diferente para cada pantalla en Android Studio 2.3.3? (s. f.). Stack Overflow en español. <https://es.stackoverflow.com/questions/129273/como-hacer-para-definir-un-titulo-diferente-para-cada-pantalla-en-android-studi>

hcode_. (2020, 26 junio). Agregar botones a la ActionBar (Forma simple) [Ejemplo] - Android con Kotlin [Vídeo]. YouTube. https://www.youtube.com/watch?v=F_JqJrom-VA

Bibliografía apartado presupuesto:

¿Cuánto cuesta crear una app? Configura tu presupuesto. (2020, enero 21). Presupuesto APP. <https://presupuestoapp.com/cuanto-cuesta-app/>

YeePLY. (2020, enero 24). ¿Cuánto cuesta crear una app? YeePLY. <https://www.yeeply.com/blog/cuanto-cuesta-crear-una-app/>

¿Cuál es el precio de una página web? (s/f), de <https://joseramonbernabeu.com/precio-pagina-web/>

Firebase. Pricing plans., de <https://firebase.google.com/pricing/>



ANEXOS

1.13. Prototipos

1.13.1. Prototipo App Móvil



Ilustración 1. Prototipo de la pantalla de inicio.



Ilustración 2. Prototipo de la pantalla de registro.



Ilustración 3. Prototipo de la pantalla de inicio de sesión.



Ilustración 4. Prototipo pantalla de instrucciones del cuidador.



Ilustración 5. Prototipo pantalla de instrucciones del dueño.



Ilustración 6. Prototipo pantalla de perfil del dueño.



Ilustración 7. Prototipo pantalla de perfil del cuidador.



Ilustración 8. Prototipo pantalla de listado de cuidadores.



Ilustración 9. Prototipo pantalla de información del cuidador.



Ilustración 10. Prototipo pop-up para reservar un servicio.



Ilustración 11. Prototipo del pop-up de compra de un servicio.



Ilustración 12. Prototipo de la pantalla de compra.

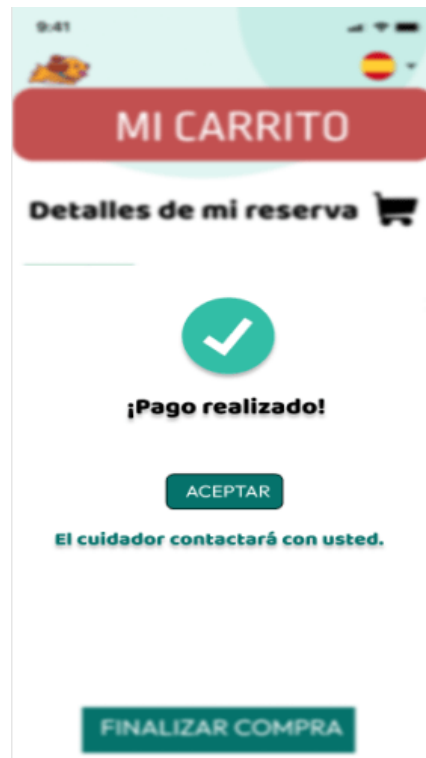


Ilustración 13. Prototipo del pop-up pago realizado.



1.13.2. Prototipo Página Web Informativa

Ilustración 14. Prototipo de la pantalla de inicio.



Ilustración 15. Prototipo de la pantalla menú de inicio.

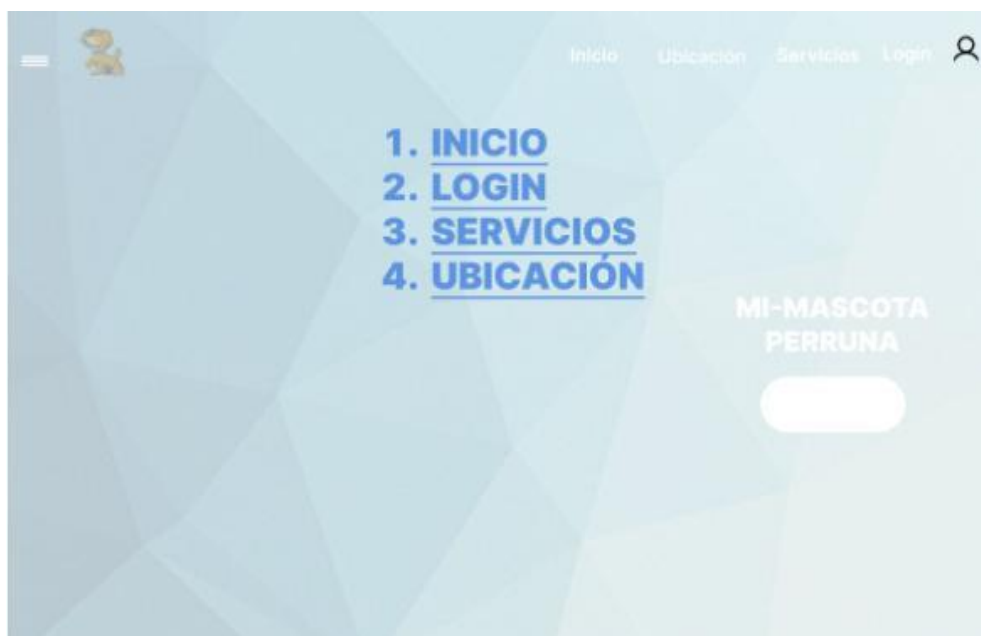




Ilustración 16. Prototipo de la pantalla de inicio de sesión.

Ilustración 17. Prototipo de la pantalla de registro.



Ilustración 18. Prototipo de la pantalla de calculadora de mapa.



Ilustración 19. Prototipo de la pantalla servicios de los cuidadores.



Ilustración 20. Prototipo de la pantalla del listado de los cuidadores.



1.14. Resultados Aplicación móvil



Ilustración 21. Pantalla de inicio app móvil.

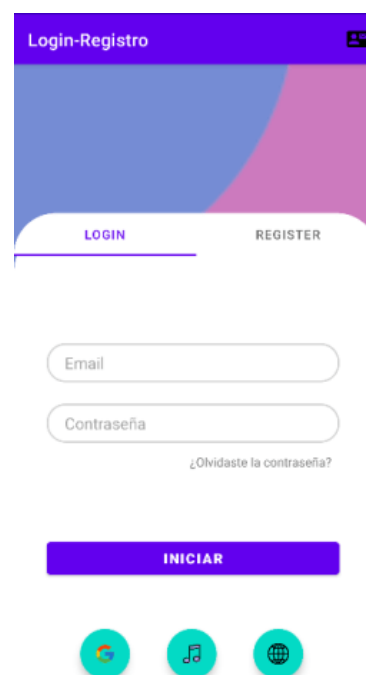


Ilustración 22. Inicio de sesión app móvil.

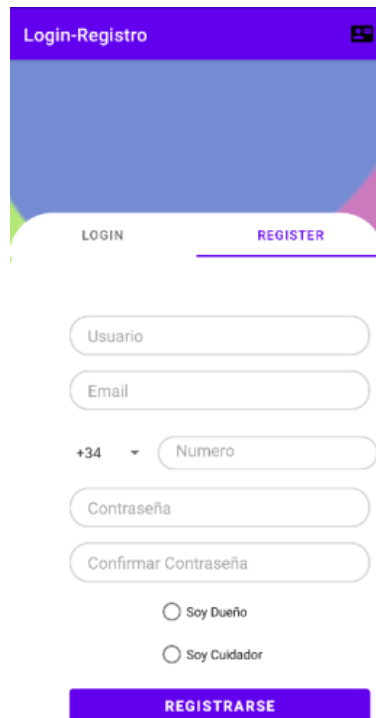


Ilustración 23. Pantalla de registro app móvil.

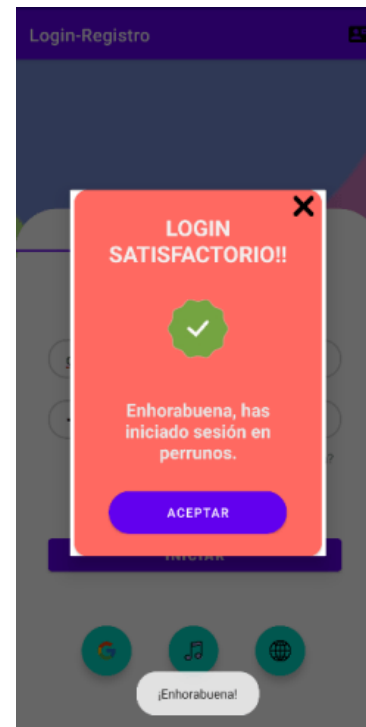


Ilustración 24. Pop-up identificación satisfactoria app móvil.

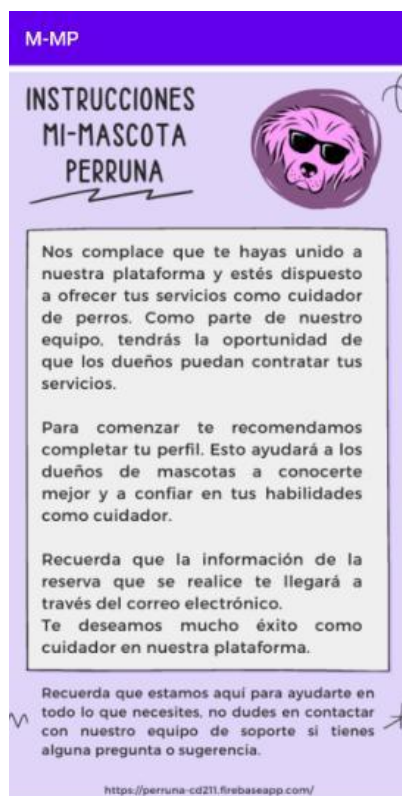


Ilustración 25. Instrucciones cuidador app móvil.



Ilustración 26. Instrucciones dueño app móvil.



Perfil Cuidador

< Inserta detalles Usuario

pablo

Email

Direccion

Telefono

FECHA DE INICIO

FECHA DE FIN

☐ Pasear

☐ 24horas

☐ Veterinario

☐ Peluqueria

INSERTAR DATOS

Ilustración 27. Formulario cuidador app móvil.

Menu Dueño

<

RECORDATORIO

PELUQUERIA CANINA

pablo
peluqueria
54789123

Ilustración 29. Menú dueño app móvil.

Perfil Dueño

< Inserta detalles Dueño

Nombre

juan

Direccion

Numero de perros:

Datos de la mascota

Nombre perro

Raza perro

Edad perro

GUARDAR DATOS

Ilustración 28. Formulario dueño app móvil.

Lista Cuidadores

pablo

54789123

Calle princess 34

7/5/2023

11/5/2023

ELEGIR

Peluqueria

prueba

calle dominguez 23

29/5/2023

31/5/2023

ELEGIR

Pasear

Ilustración 30. Listado de cuidadores app móvil

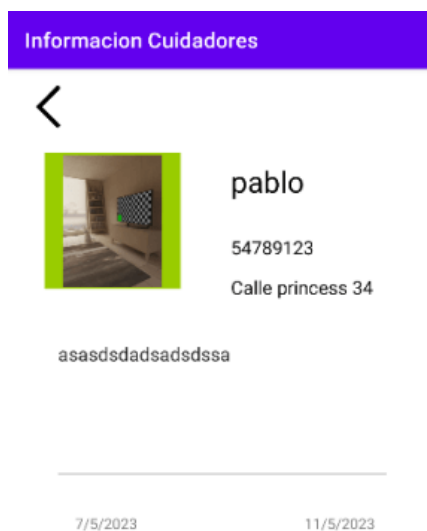


Ilustración 31. Información cuidadores app móvil.



Ilustración 32. Pantalla de compra app móvil.

1.15. Resultados página web informativa



Ilustración 33. Pantalla de inicio página web.

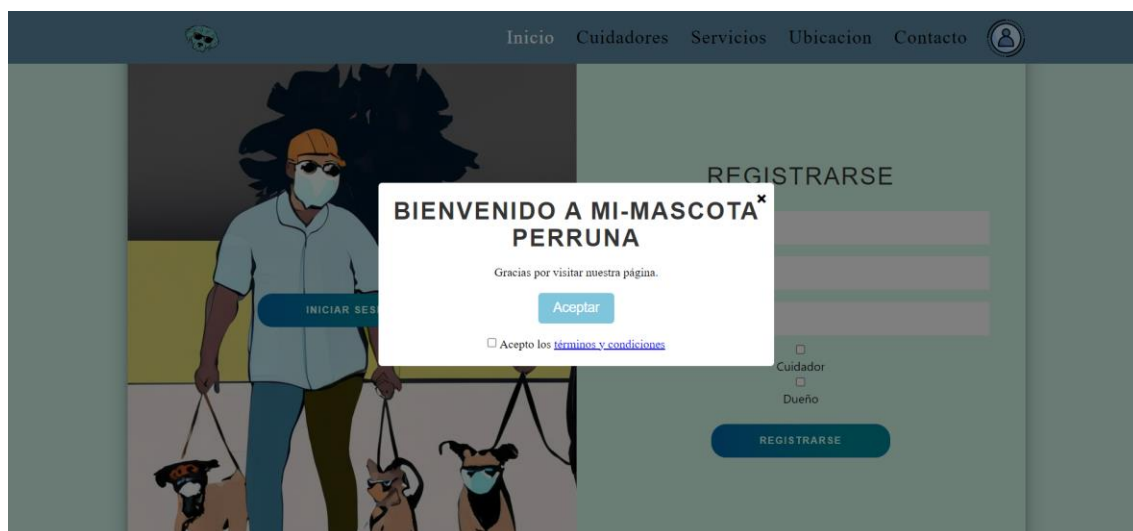


Ilustración 34. Pop-up términos de privacidad página web.

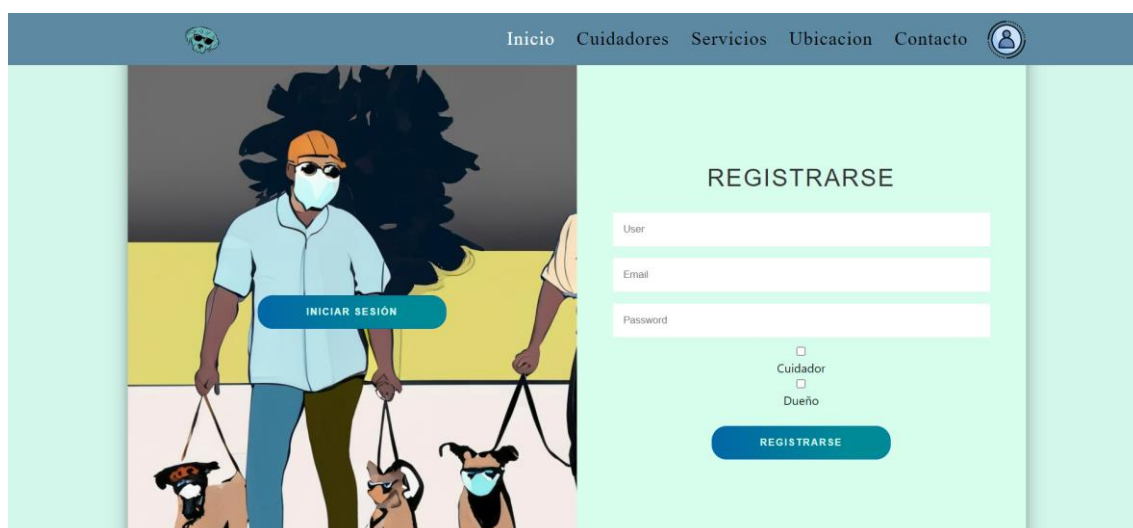


Ilustración 35. Registro/Inicio de sesión página web.



Ilustración 37. Listado de cuidadores página web.

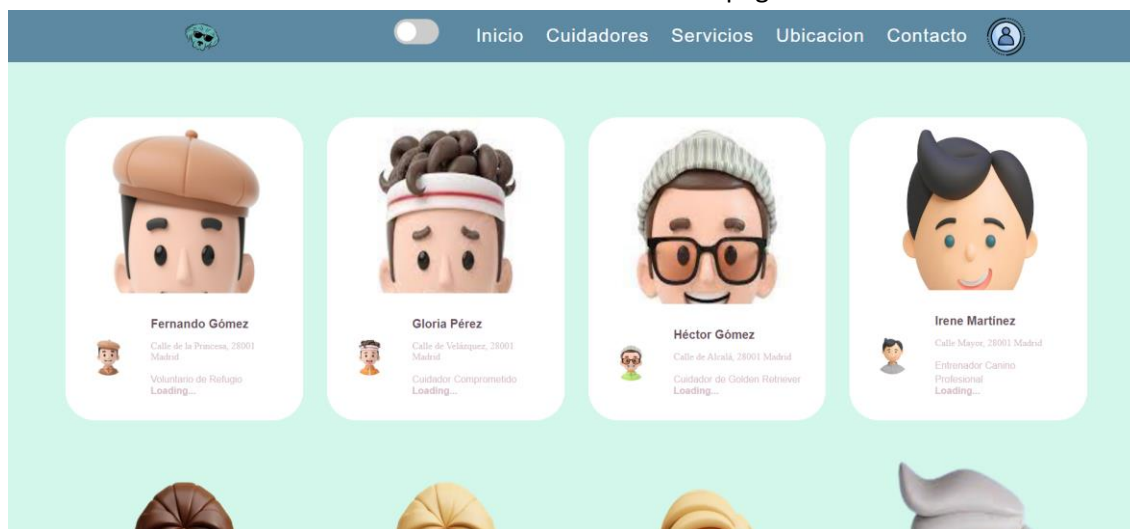


Ilustración 38. Listado de cuidadores página web.



Ilustración 39. Servicios página web.

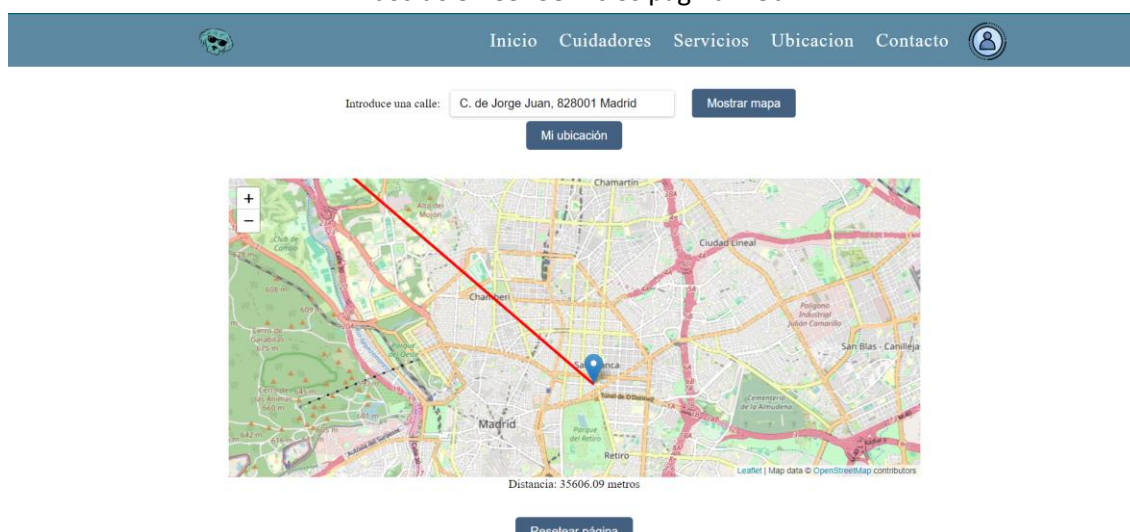


Ilustración 40. Calculadora ubicación página web.



Ilustración 41. Datos de contacto y descarga de app desde la página web.

1.16. Resultados página web administrador CRUD

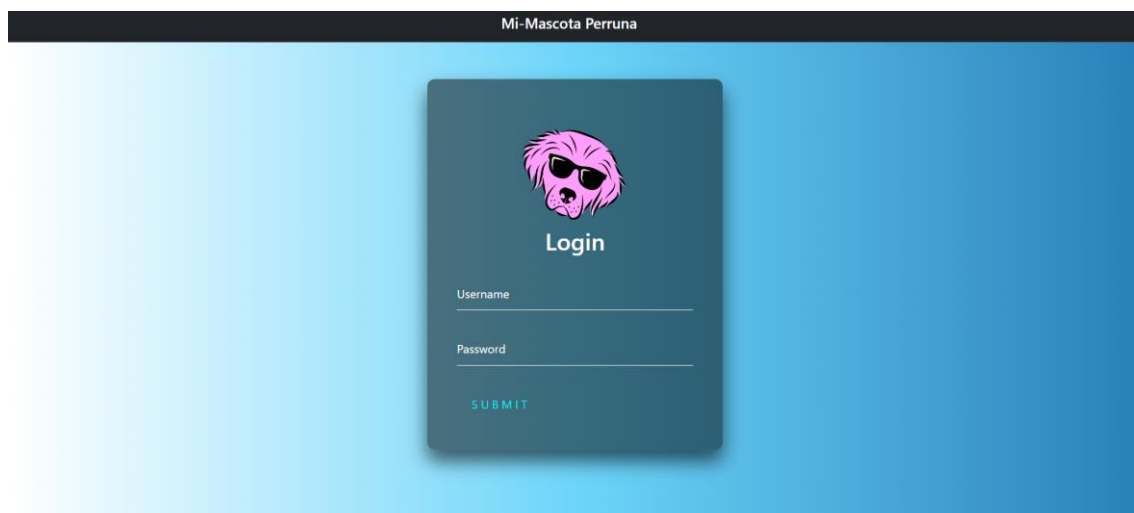


Ilustración 42 Pantalla inicio de sesión administrador.



LISTA CUIDADORES			
Buscar por nombre...			Agregar
Nombre	Email	Telefono	Acciones
ana2	ana2@gmail.com	123456	✎ 👤 👁
bryan1	dueñobryan1@gmail.com	342342342	✎ 👤 👁
carla1	carla1@gmail.com	654321	✎ 👤 👁
carlos1	carlos1@gmail.com	987654	✎ 👤 👁
david1	david@gmail.com	876589	✎ 👤 👁
diego1	diego1@gmail.com	45874587	✎ 👤 👁
diego2	deigo2@gmail.com	456456	✎ 👤 👁
elena1	elena1@gmail.com	963258	✎ 👤 👁
ivan1	ivan1@gmail.com	124578	✎ 👤 👁
juan	juan@gmail.com	453345345	✎ 👤 👁

Ilustración 43. Listado cuidadores y dueños.

Agregar Empleado

VOLVER

AGREGAR

Ilustración 44. Panel para agregar cuidadores y dueños.

Detalles del Empleado	
Nombre	Ana
Dirección	Calle de las Azucenas 78
Email	ana2@gmail.com
Nombre Perro	Cooper
Raza Perro	Great Dane
Edad Perro	7
Ruta de Imagen	ana2.jpeg

Ilustración 45. Ver información del cuidador o dueño.



1.17. Glosario

- **MVC:** es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, Modelo-Vista-Controlador.
- **CRUD:** es el acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar).
- **API:** significa “interfaz de programación de aplicaciones”. Son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos.
- **Firebase:** Plataforma de desarrollo de aplicaciones web y móviles que proporciona herramientas para la autenticación, bases de datos en tiempo real, almacenamiento y más.
- **Angular:** Framework de desarrollo web basado en TypeScript utilizado para construir aplicaciones de una sola página.
- **HTML:** Lenguaje de marcado para la creación de páginas web.
- **CSS:** Lenguaje de estilo utilizado para describir la presentación de un documento escrito en HTML o XML.
- **Inteligencia Artificial (IA):** Uso de tecnologías innovadoras como Kaiber AI para mejorar el diseño de la aplicación.
- **TypeScript:** Lenguaje de programación desarrollado y mantenido por Microsoft, que es un superconjunto de JavaScript, usado en el desarrollo con Angular.
- **Firebase Hosting:** Servicio de alojamiento web para contenido estático que puede conectarse a aplicaciones web y móviles.
- **Google Analytics:** Herramienta que utiliza píxeles de seguimiento para proporcionar estadísticas y métricas detalladas sobre el uso de la web y la aplicación.
- **Android Studio:** Entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android, basado en IntelliJ IDEA.
- **Interfaz:** Conjunto de métodos y propiedades que una clase debe implementar, permitiendo la comunicación y la interacción entre diferentes componentes de software.
- **Framework:** Conjunto de herramientas y bibliotecas que proporciona una estructura de soporte reutilizable para el desarrollo de aplicaciones, facilitando la creación y el mantenimiento de proyectos.
- **Firestore:** Base de datos NoSQL en tiempo real y en la nube ofrecida por Firebase, utilizada para almacenar y sincronizar datos entre aplicaciones cliente y servidores.
- **Layout:** Diseño o disposición de elementos visuales en una interfaz de usuario, definido generalmente mediante XML en el desarrollo de aplicaciones Android.
- **Pop-up:** Ventana o mensaje emergente que aparece en la pantalla para proporcionar información adicional o interactuar con el usuario sin cambiar la vista actual de la aplicación.
- **Plugins:** Complementos o extensiones que añaden funcionalidades específicas a un programa principal, como un IDE, permitiendo la personalización y ampliación de sus capacidades.
- **Meta (Facebook):** Plataforma que utiliza píxeles de seguimiento para proporcionar estadísticas detalladas sobre la interacción de los usuarios con la aplicación.



- **Kaiber AI:** Herramienta de inteligencia artificial utilizada para mejorar el diseño de la aplicación.
- **XML:** Lenguaje de marcado utilizado para definir documentos con un formato que puede ser leído tanto por humanos como por máquinas. Utilizado en la aplicación móvil desarrollada en Java.
- **NoSQL:** Sistema de gestión de bases de datos que no utiliza el modelo relacional tradicional, adecuado para grandes volúmenes de datos y esquemas flexibles.

1.18. Configuración

1.18.1. Configuración del hosting en Firebase

Para conectar el acceso de la aplicación con el acceso de la página web, se han tenido que obtener los siguientes datos de la configuración:

```
var firebaseConfig = {  
  apiKey: "AlzaSyA_SM6d1nCztP2JON5DHytWOOWs7JI_Ff4",  
  authDomain: "perruna-cd211.firebaseio.com",  
  projectId: "perruna-cd211",  
  storageBucket: "perruna-cd211.appspot.com",  
  messagingSenderId: "796763331783",  
  appId: "1:796763331783:web:2fd75de7e305206f66b2e2",  
  measurementId: "G-RHB6WS2L9H"  
};
```

A continuación, en el registro se guardan los datos en Firebase comprobando que no existen previamente.

Para conectar la API de Recatchap con la página web desde Firebase se realizan los siguientes pasos:

1. En la consola de Google Cloud, vamos a la página reCAPTCHA Enterprise.

<https://console.cloud.google.com/security/>

2. Creamos un proyecto, en este caso “Perruna”

Google Cloud

Buscar (/) recursos, docu...

Proyecto nuevo

Tienes 8 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)

Nombre del proyecto *
perruna

ID de proyecto: perruna. No se podrá cambiar más tarde. [EDITAR](#)

Ubicación *
Sin organización [EXPLORAR](#)

Organización o carpeta superior

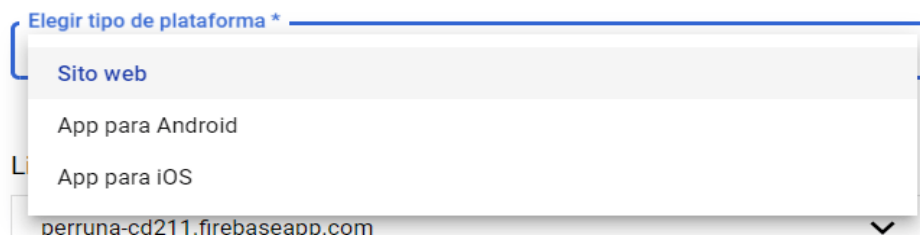
[CREAR](#) [CANCELAR](#)



3. Hacemos clic en **Crear clave**.



4. En el campo **Nombre visible**, ingresamos un nombre visible. En el menú desplegable **Elegir tipo de plataforma**, seleccionamos **Sitio web**.



5. Añadimos el dominio correspondiente a la aplicación.


Lista de dominios

perruna-cd211.firebaseio.com	▼
perruna-cd211.web.app	▼
AGREGAR UN DOMINIO	

6. Añadimos el nivel de seguridad que queremos tener en la página web.



Se debe agregar al menos un dominio a la clave de la plataforma del sitio web

☒ Firewall de aplicación web (WAF) 
Para las implementaciones en la capa perimetral.


[Google Cloud Armor](#) es la integración predeterminada del servicio de WAF. Para habilitar otros servicios de WAF, comunícate con el [equipo de asistencia de reCAPTCHA](#).


Atributo

Token de acción

No se pueden usar la página de desafío ni el token de sesión con el desafío de la casilla de verificación. Determina qué función de WAF se ajusta mejor a tu caso de uso. [Más información](#)

☐ Inhabilitar la verificación del dominio

☒ Usar el desafío de la casilla de verificación 
Verifica a los usuarios con solo pedirles que marquen la casilla de verificación "No soy un robot".
No se puede cambiar después de crear la clave.

 El uso de una casilla de verificación puede aumentar la fricción del usuario y no mejora significativamente la exactitud.

[MÁS INFORMACIÓN](#)

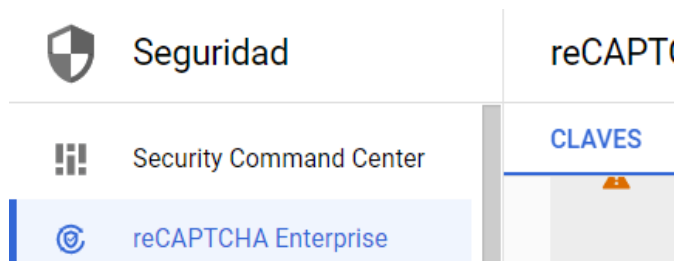
Seguridad del desafío

La seguridad del desafío de la integración de una casilla de verificación establece el nivel de dificultad del desafío mediante el aumento del escrutinio y la frecuencia con la que se presenta a los usuarios.


- ☐ Dificultad mínima del desafío
- ☐ Dificultad media (equilibrio entre la seguridad y el nivel de dificultad)
- ☒ Dificultad mayor (más seguro contra los bots)

☐ Esta es una clave de prueba
Esta clave mostrará la misma puntuación para cada evaluación. No se puede cambiar una clave de prueba por una normal.

7. La clave creada aparece en la página de reCAPTCHA en el apartado de claves.




8. En el estado de la clave, vemos la API incompleta. Para completarla, vamos a la opción de detalles.

**Perruna**


perruna-cd211.firebaseio.com; perruna-cd211.w...


ID de clave


6LcVTm4IAAAAIa1xbmSDFpE_N1BRvg...




Estado de la clave

 Incompleta

 MÉTRICAS

 → DETALLES

 EDITAR O BORRAR



9. En los detalles de la clave encontramos la clave de seguridad, la secreta heredada y los dominios.

[← Detalles clave](#) [✎ EDITAR CLAVE](#) [🗑 BORRAR ESTA CLAVE](#) [📊 VER MÉTRICAS](#) [🎓 APRENDIZAJE](#)

Perruna ID: 6LcVTm4IAAAAAIDa1xbmSDFpE_N1BRvg6jhGugqi 📋

Integración web

[Obtén más información](#) sobre la configuración de la clave WAF para la integración de Google Cloud Armor.

Clave secreta de reCAPTCHA heredada

⚠ Solo se debe usar cuando se realice una integración en un tercero que solicite tu clave secreta/privada (o cualquier integración heredada mediante el extremo `https://www.google.com/recaptcha/api/siteverify` en su backend).

Tu clave secreta heredada es
6LcVTm4IAAAAAO_UaCCeiWimR7Ymv5jk4GeeJf3x

If you want to integrate reCAPTCHA with a 3rd-party application (typically a website builder/host), it is possible that the 3rd-party is using legacy reCAPTCHA and did not upgrade to Enterprise yet. That is easy to differentiate:

- a 3rd-party integrated with legacy reCAPTCHA will ask for your secret/private key
- a 3rd-party integrated with reCAPTCHA Enterprise will ask you for your API key instead

Clave del sitio web

Configuración

Creada el 8 de abril de 2023

Firewall de aplicación web (WAF)	Habilitado
Función WAF	ACTION_TOKEN
Páginas de AMP	Bloqueado
Desafío de casilla de verificación	Habilitado
Verificación del dominio	Habilitado
Clave de prueba	Inhabilitado

Dominios

perruna-cd211.firebaseio.com
perruna-cd211.web.app

10. Accedemos a la aplicación VisualStudio Code donde tenemos el código HTML y JS de la página web. La clave pública se pone en el siguiente código del HTML.

```
<script
```

```
src="https://www.google.com/recaptcha/api.js?render=6LcVTm4IAAAAAIDa1xbmSDFpE_N1BRvg6jhGugqi"></script>
```

Añadimos un DIV donde queramos que esté ubicado el Captcha. En este caso, lo agregamos antes del botón de registrar.

```
<div class="g-recaptcha" data-sitekey="6LcVTm4IAAAAAIDa1xbmSDFpE_N1BRvg6jhGugqi"></div>
```

Al código JS le introducimos la clave heredada.

```
function onSubmit(token) {  
  // Acción que desee realizar después de verificar la respuesta  
  console.log('El token ha sido verificado y se ha enviado a tu servidor');  
}  
  
grecaptcha.execute('6LcVTm4IAAAAAO_UaCCeiWimR7Ymv5jk4GeeJf3x', {action: 'submit'}).then(function(token) {  
  // Enviar el token a su servidor para verificar la respuesta  
  onSubmit(token);  
});
```



11. Descargamos la aplicación Node.js en <https://nodejs.org/es> es un entorno de ejecución para JavaScript.

- Abrimos el cmd en la carpeta donde tengamos la página web (HTML, CSS, JS).
- Instalamos las herramientas.

```
C:\Users\Administrador\OneDrive\Escritorio\Perrruna>npm install -g firebase-tools
npm WARN deprecated har-validator@5.1.3: this library is no longer supported
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environm
or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is i
/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
changed 703 packages in 27s
```

- Iniciamos sesión con Firebase.

```
C:\Users\Administrador\OneDrive\Escritorio\Perrruna>firebase login
Already logged in as diegogmarti@gmail.com
```

- Iniciamos Firebase.

```
C:\Users\Administrador\OneDrive\Escritorio\Perrruna>firebase init

#####  #####  #####  #####  #####  ###  #####  #####
##      ##  ##  ##  ##      ##      ##  ##  ##  ##      ##
#####      ##  #####  #####  #####  #####  #####  #####
##      ##  ##  ##  ##      ##      ##  ##  ##      ##
##      #####  ##  ##  #####  #####  ##      ##  #####  #####

You're about to initialize a Firebase project in this directory:

C:\Users\Administrador\OneDrive\Escritorio\Perrruna
```

- Subimos los archivos a la carpeta que se nos ha creado en la ubicación donde tenemos nuestra página web. Por defecto, se crea la carpeta public, donde pasamos todos los archivos y hacemos una actualización en el firebase.

```
C:\Users\Administrador\OneDrive\Escritorio\Perrruna>firebase deploy

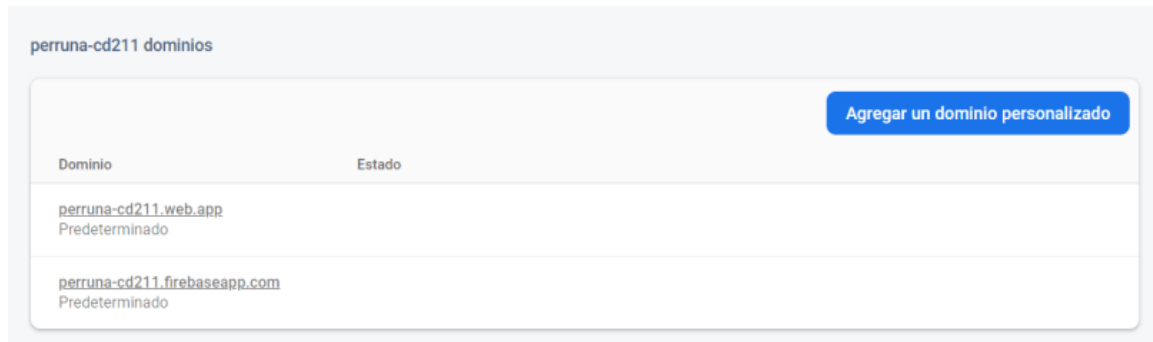
=== Deploying to 'perruna-cd211'...

i  deploying hosting
i  hosting[perruna-cd211]: beginning deploy...
i  hosting[perruna-cd211]: found 6 files in public
+  hosting[perruna-cd211]: file upload complete
i  hosting[perruna-cd211]: finalizing version...
+  hosting[perruna-cd211]: version finalized
i  hosting[perruna-cd211]: releasing new version...
+  hosting[perruna-cd211]: release complete

+  Deploy complete!
```



12. Nos dirigimos a la página de firebase al apartado hosting y entramos en el dominio de firebaseapp.com.



13. Si queremos hacer alguna actualización de la página en el código y luego que se vean en el dominio del Firebase actualizado, tenemos que abrir CMD de nuevo, iniciar sesión con firebase desde consola, hacer el init, sobrescribir el archivo de la página web y actualizar la carpeta.

```
C:\Users\Administrador\OneDrive\Escritorio\Perruna>firebase deploy --only hosting

== Deploying to 'perruna-cd211'...

i  deploying hosting
i  hosting[perruna-cd211]: beginning deploy...
i  hosting[perruna-cd211]: found 38 files in public
+  hosting[perruna-cd211]: file upload complete
i  hosting[perruna-cd211]: finalizing version...
+  hosting[perruna-cd211]: version finalized
i  hosting[perruna-cd211]: releasing new version...
+  hosting[perruna-cd211]: release complete

+  Deploy complete!

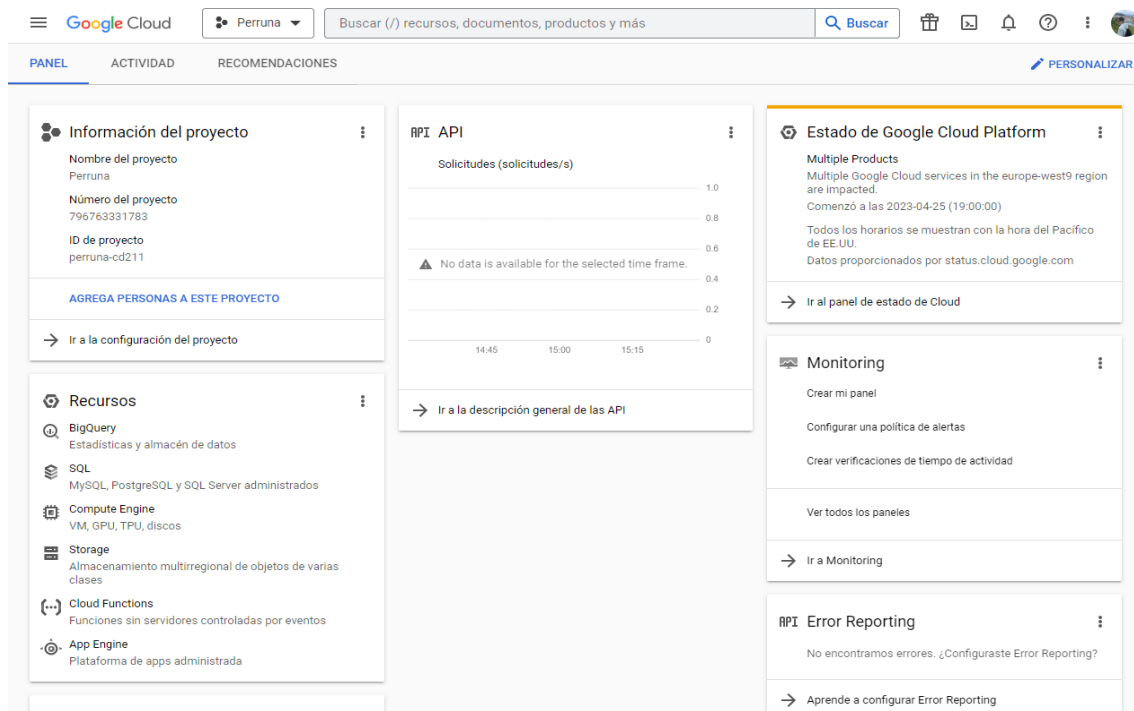
Project Console: https://console.firebase.google.com/project/perruna-cd211/overview
Hosting URL: https://perruna-cd211.web.app

C:\Users\Administrador\OneDrive\Escritorio\Perruna>
```

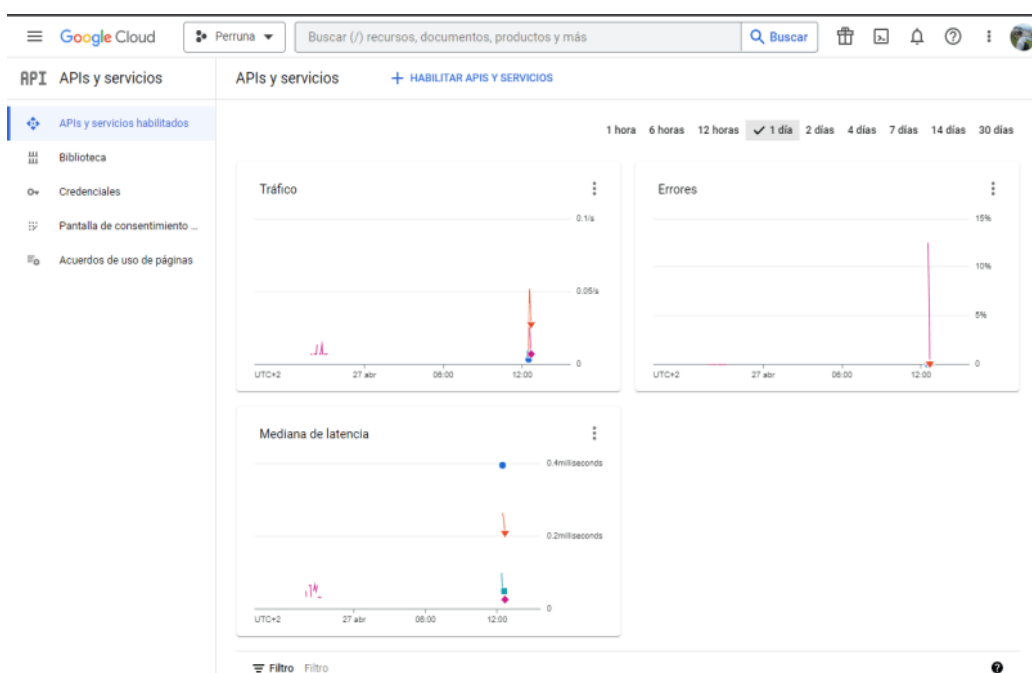
Desde la consola de Google Cloud se puede acceder a la información de las API's que se han utilizado:

<https://console.cloud.google.com/welcome?hl=es-419&ga=2.117040518.1241376742.1682601807-1726461330.1682601807&project=perruna-cd211>

En primer lugar, hay que poner el nombre del proyecto, en este caso, perruna-cd211, y una vez dentro, en panel podemos ver una gran variedad de opciones.



Si entramos en la descripción general de las API podemos ver los gráficos de tráfico, errores y mediana de latencia, información adicional sobre el uso de la API, número total de solicitudes, número de solicitudes exitosas y fallidas, promedio de tiempo de respuesta, cantidad de datos transferidos, entre otros datos relevantes. También se puede acceder a información sobre las cuotas de uso de la API y los límites de la tasa de solicitud, así como a recursos de ayuda y documentación para la integración de la API en aplicaciones.





El gráfico de tráfico muestra la cantidad de solicitudes que se han realizado a la API de reCAPTCHA en un periodo de tiempo determinado.

En el gráfico de errores podemos ver la cantidad de errores que se han producido al utilizar la API de reCAPTCHA en un periodo de tiempo determinado. Si la cantidad de errores es alta, puede indicar que hay algún problema en la implementación de la API.

Por último, el gráfico de mediana de latencia indica el tiempo medio que tarda la API en responder a una solicitud. Si la latencia es alta, puede manifestar que hay un problema en la red o en el servidor que está afectando al rendimiento de la API. También puede significar que se está haciendo un uso ineficiente de la API y se deben optimizar las solicitudes.

También podemos ver las APIs que se van activando según vamos usando la aplicación, es decir, la API reCAPTCHA está desactivada hasta que entras en el registro y pulsas que se verifique, es ahí cuando se activa y vemos los datos en las estadísticas.

Nombre	↓ Solicitudes	Errores (%)	Latencia mediana (ms)	95% de latencia (ms)
Identity Toolkit API	42	2	49	130
reCAPTCHA Enterprise API	25	0	227	1,029
Cloud Firestore API	7	0	53	216
Cloud Storage for Firebase API	1	0	393	511
App Engine				
App Engine Admin API				
BigQuery API				
BigQuery Migration API				



1.18.2. Configuración de las herramientas de firebase

A continuación, se muestran los pasos a seguir para realizar una implementación y configuración de Firebase detallando cada uno de los puntos.

MI-MASCOTA
PERRUNA

FIREBASE

CREAR PROYECTO

Desde la consola de Firebase en la web, crear nuevo proyecto.



INSTALAR SDK

Integrar el SDK en el Gradle de Android Studio.

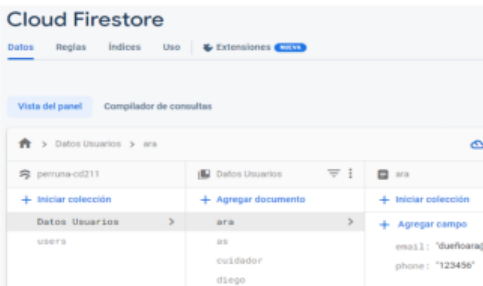


MI-MASCOTA
PERRUNA

FIREBASE

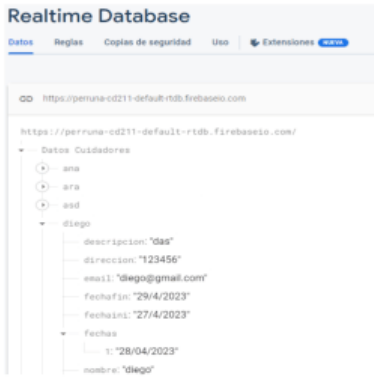
3. AGREGAR USUARIOS

Crear documentos y colecciones en la base de datos.



4. AGREGAR DATOS

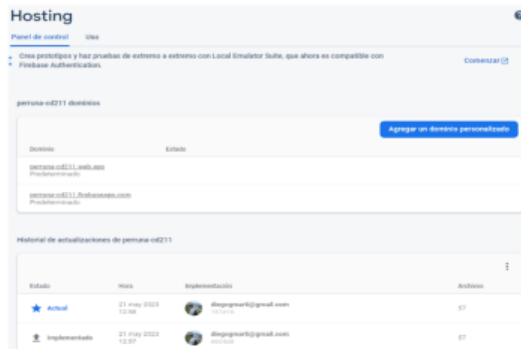
Crear consultas para tener una base de datos en tiempo real





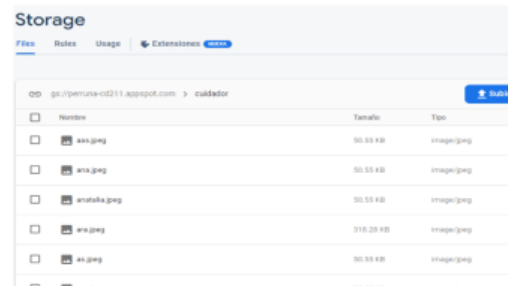
5. USO DE HOSTING Y DOMINIO

Añadir el dominio de la aplicación y subirla a la herramienta hosting de Firebase para que aparezca en Google, teniendo un hosting gratuito.



6. SUBIR FOTOS

Uso de la herramienta Firebase Storage para guardar en carpetas las fotos de los usuarios.



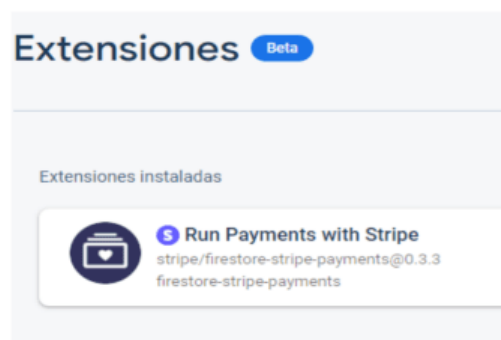
7. RENDIMIENTO DE LA APP

Ver estadísticas y métricas del rendimiento de la aplicación, desglosado en las clases que tenga en Android Studio.



8. EXTENSIONES

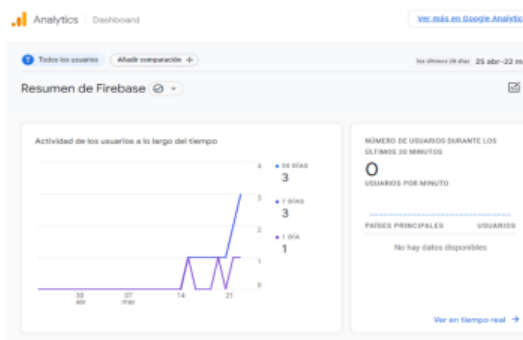
Extensión de pago en Firebase para la pasarela de pago.





7. GOOGLE ANALYTICS

Ver estadísticas y métricas del rendimiento de la web, desglosado en las páginas.



8. FUNCTIONS

Cada vez que se use una api se verá la activación en esta herramienta de Firebase.

The screenshot shows the Firebase Functions console. It displays a list of functions with columns: Función, Activador, Versión, Solicitudes (24 h), Cantidad mínima o máxima de instancias, and Tiempo de espera. The functions listed are related to Stripe API calls.

Función	Activador	Versión	Solicitudes (24 h)	Cantidad mínima o máxima de instancias	Tiempo de espera
ext-firebase-stripe->stripe-test	document.create (firebase.functions.document)	v1	0	0 / 10000	1 min
ext-firebase-stripe->stripe-test	user.create	v1	0	0 / 10000	1 min
ext-firebase-stripe->stripe-test	select	v1	0	0 / 10000	1 min
ext-firebase-stripe->stripe-test	select	v1	0	0 / 10000	1 min
ext-firebase-stripe->stripe-test	document.delete (firebase.functions.document)	v1	0	0 / 10000	1 min
ext-firebase-stripe->stripe-test	user.delete	v1	0	0 / 10000	1 min



9. ANALYTICS EVENTS

Se puede ver más detalladamente donde hacen clicks los usuarios.

The screenshot shows the Google Analytics 'Eventos' page. It displays a table of existing events with columns: Nombre del evento, Recuento, % de cambio, Usuarios, % de cambio, and Marcar como conversión. The events listed are first_visit, page_view, scroll, and session_start.


Nombre del evento	Recuento	% de cambio	Usuarios	% de cambio	Marcar como conversión
first_visit	0	-	0	-	<input type="checkbox"/>
page_view	26	-	0	-	<input type="checkbox"/>
scroll	0	-	0	-	<input type="checkbox"/>
session_start	0	-	0	-	<input type="checkbox"/>

10. ANALYTICS AUDIENCES

Se reciben los datos de la cantidad de usuarios que entran en la app.

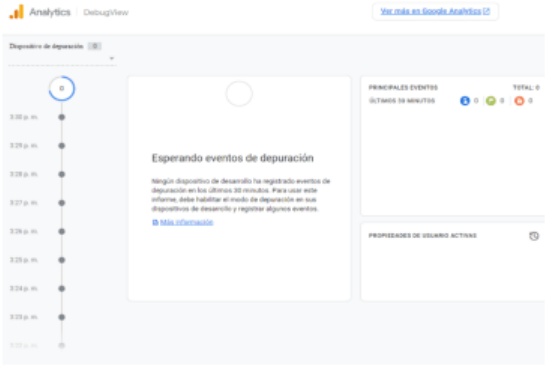
The screenshot shows the Google Analytics 'Audiences' page. It displays a table of audiences with columns: Nombre de la audiencia, Descripción, and Usuarios. The audiences listed are All Users and Purchasers.

Nombre de la audiencia	Descripción	Usuarios
All Users	All users	0
Purchasers	Users who have made a purchase	Menos de 10 usuarios

MI-MASCOTA PERRUNA 
FIREBASE

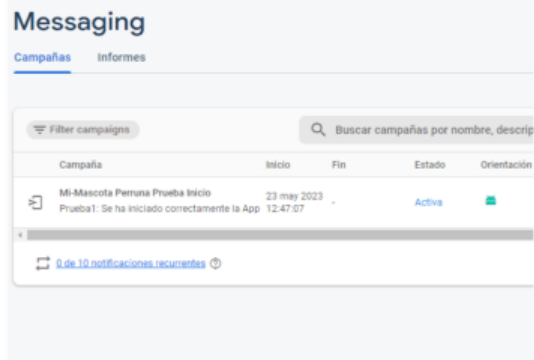
11. DEBUG VIEW

Depurar la web para comprobar su correcto funcionamiento.



12. MESSAGING

Envío de notificaciones cada vez que el usuario entre en la app o haga una compra.



1.19. Cartas de comprobación

TARJETA DE COMPROBACIÓN GENERAL

- ✓ Los usuarios se pueden registrar e identificarse.
- ✓ Crear perfil de cuidador o dueño.
- ✓ Se muestran los cuidadores en el listado.
- ✓ Puedes contratar un servicio.
- ✓ Seleccionar una fecha del rango.
- ✓ Realizar el pago en la pasarela.
- ✓ Comprobar que se guardan los datos en Firestore.
- ✓ Confirmar que la Api se activa en Google Cloud.

Ilustración 42. Tarjeta comprobación general.



TARJETA DE COMPROBACIÓN
REGISTRO/INICIO DE SESIÓN

- ✓ Asegurarnos de que el estilo se mueve.
- ✓ Al deslizar el dedo se pasa al registro.
- ✓ Comprobar que se conecta a Firestore.
- ✓ Al clicar en “resetear contraseña” salta un pop-up y tiene que recibir un email.
- ✓ Al pulsar el botón de Google se tiene que iniciar sesión automáticamente.

Ilustración 43. Tarjeta Registro/Inicio de sesión.

TARJETA DE COMPROBACIÓN
PERFIL CUIDADOR

- ✓ Comprobar que se insertan los datos correctamente en Firestore.
- ✓ Asegurarnos de que Recibe la foto en Firestore.

Ilustración 44. Tarjeta perfil cuidador.

TARJETA DE COMPROBACIÓN
PERFIL DUEÑO

- ✓ Comprobar que se insertan los datos correctamente en Firestore.
- ✓ Asegurarnos de que Recibe la foto en Firestore.

Ilustración 45. Tarjeta perfil dueño.

TARJETA DE COMPROBACIÓN
LISTADO CUIDADORES

- ✓ Al crear un cuidador comprobar que aparece en el listado.
- ✓ Revisar que los datos están bien insertados en la tarjeta del listado.
- ✓ Al clicar una tarjeta se tiene que abrir una actividad con información más detallada.
- ✓ Podemos deslizar con el dedo hacia arriba y abajo.
- ✓ Al seleccionar un cuidador y darle al botón de aceptar se abre un pop-up con opciones de reserva.

Ilustración 46. Tarjeta Listado cuidadores.

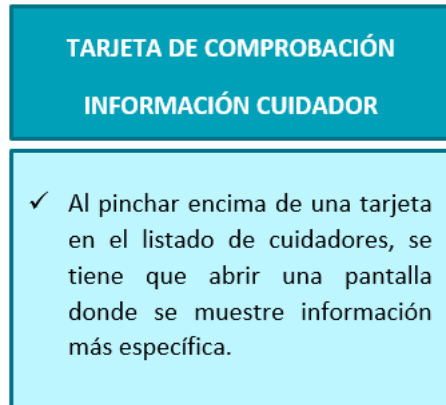


Ilustración 47. Tarjeta Información cuidador.

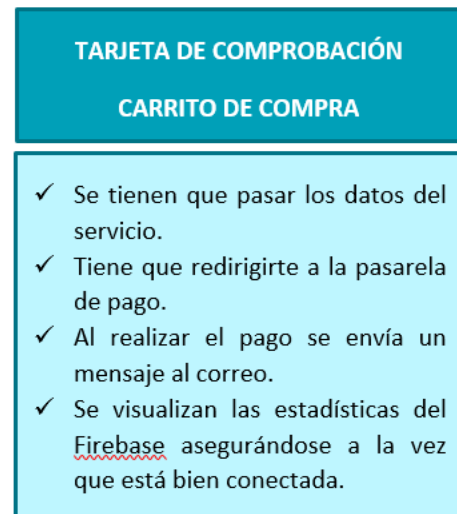


Ilustración 48. Tarjeta perfil dueño.

1.20. Anuncios publicitarios



Ilustración 49. Anuncio en red social Facebook.



Ilustración 50. Anuncio en red social Instagram.

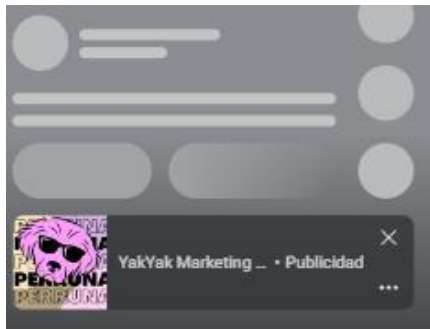


Ilustración 51. Anuncio en aplicación de música.

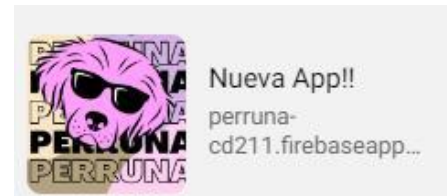


Ilustración 52. Anuncio en aplicación de juego.



Ilustración 53 Tarjeta perfil dueño.

1.21. Manual Angular

Angular es un framework de diseño de aplicaciones y plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas.

Para usar el framework Angular, debes conocer los siguientes lenguajes:

- JavaScript
- HTML
- CSS

Para instalar Angular en el sistema local se necesita node.js.

Para **descargar e instalar paquetes npm**, necesitas un administrador de paquetes npm. Hay que utilizar la interfaz de línea de comandos del cliente npm, que se instala con Node.js por defecto.



Para comprobar que tienes instalado el cliente npm, ejecuta `npm -v` en una terminal.

```
C:\Users\diego.gonzalez>npm -v
9.5.0
```

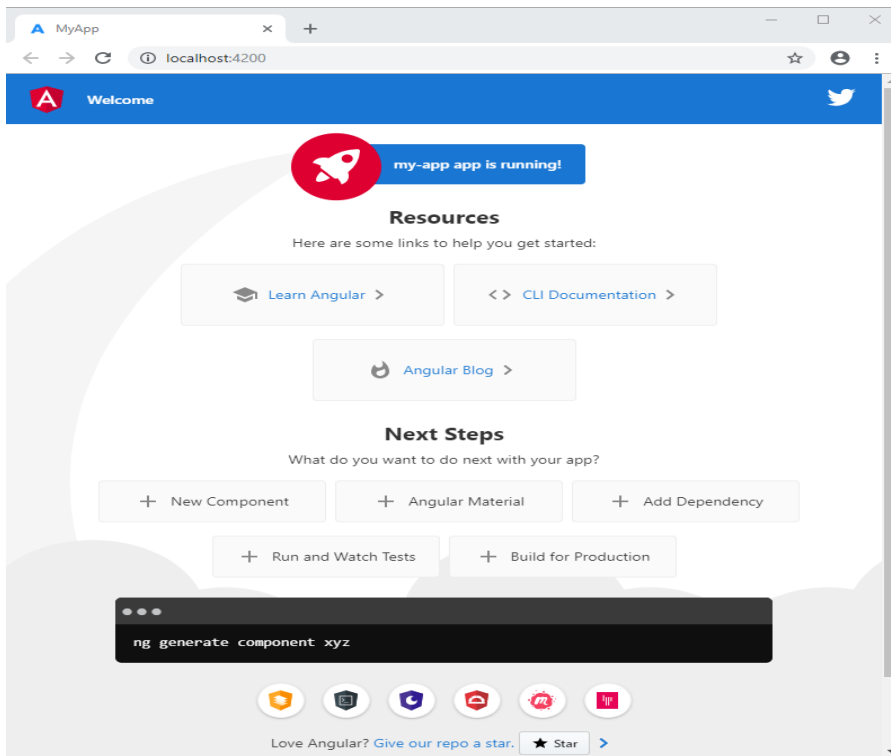
Instala la CLI de Angular para crear proyectos, generar código de aplicaciones y bibliotecas, realizar tareas...

Para instalar CLI de Angular, abre una terminal y ejecuta el siguiente comando: `content_copy npm install -g @angular/cli`.

Crea un espacio de trabajo y una aplicación inicial. Ejecuta el comando CLI `ng new` y proporciona el nombre `my-app`, como se muestra aquí: `content_copy ng new my-app`

Ejecutar la aplicación La CLI de Angular incluye un servidor, de modo que puede crear y servir su aplicación localmente. Navega a la carpeta del espacio de trabajo, como `my-app`. Ejecuta el siguiente comando: `content_copy cd my-app ng serve -o`.

El comando `ng serve` inicia el servidor, observa tus archivos, y reconstruye la aplicación a medida que realizas cambios en esos archivos. La opción `-o` abre automáticamente tu navegador en `http://localhost:4200/`. Si tu instalación y configuración fue exitosa, deberías ver una página similar a la siguiente.



The screenshot displays the Visual Studio Code interface with the following components:

- Explorer View:** Shows the file structure of the 'AppBecarios' project. The 'package-lock.json' file is selected.
- Source Control View:** Shows the 'package-lock.json' file is staged for commit.
- Run and Debug View:** Shows the output of the 'npm run build' command. The output indicates that the application bundle generation is complete and the application is running on <http://localhost:4200/>.

XXXIII

