

TRABAJO FIN DE GRADO

Desarrollo de Aplicaciones Multiplataforma:

Mi-Mascota Perruna

**Alumno:** Diego González Martínez

**Tutor:** Guillermo Fraire de Arada

IES María de Zayas y Sotomayor

**Curso:** 2022-2023

ÍNDICE

[RESUMEN 0](#_Toc133756991)

[1. INTRODUCCIÓN 1](#_Toc133756992)

[2. OBJETIVO Y JUSTIFICACIÓN 2](#_Toc133756993)

[3. ANÁLISIS 4](#_Toc133756994)

[4. DISEÑO 6](#_Toc133756995)

[4.1. ARQUITECTURA DEL SISTEMA 6](#_Toc133756996)

[4.2. COMPONENTES 8](#_Toc133756997)

[4.2.1. COMPONENTES DE LA INTERFAZ DE USUARIO 8](#_Toc133756998)

[4.2.2. COMPONENTES LÓGICA DE NEGOCIO 18](#_Toc133756999)

[4.2.3. TECONOLOGÍAS Y LENGUAJES 19](#_Toc133757000)

[4.3. HERRAMIENTAS UTILIZADAS 19](#_Toc133757001)

[5. IMPLEMENTACIÓN 21](#_Toc133757002)

[6. PRUEBAS 28](#_Toc133757003)

[7. EXPLOTACIÓN 31](#_Toc133757004)

[8. PRESUPUESTO 33](#_Toc133757005)

[9. CONCLUSIONES 35](#_Toc133757006)

[10. LÍNEAS DE TRABAJO FUTURAS 36](#_Toc133757007)

[11. BIBLIOGRAFÍA 37](#_Toc133757008)

# RESUMEN

# INTRODUCCIÓN

El 35% de la población mundial tiene un perro como mascota. Según un estudio de censos de 2021, España cuenta con más de 9 millones de perros domésticos. (1) 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, llevarles 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 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 en su zona, y también ofrecerá una plataforma de pago segura para la transacción.

Además, la aplicación también permitirá a los cuidadores de perros 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 de perros que se ajusten a sus necesidades específicas y preferencias.

En resumen, la aplicación 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.

# OBJETIVO Y JUSTIFICACIÓN

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

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 dividirá en 2 partes:

Por un lado, la elaboración de una página web donde se dará a conocer la aplicación y se podrá consultar información sobre los servicios que ofrece, opiniones de usuarios, atención al cliente y una opción para registrarse.

Por otro lado, una aplicación Android donde los usuarios podrán registrarse o 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, cuidados especiales…)
* 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
* Pasarela de pago.
* Atención al cliente.

En el caso de registrarse como cuidador:

* Mi perfil, donde aparecerá una foto del cuidador, datos de contacto, podrán establecer los servicios que ofrecen, el precio, el número de perros y una breve descripción.

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. Además, puede ser difícil encontrar un cuidador de mascotas confiable cuando se necesita. Esta aplicación busca solucionar estos problemas al proporcionar una aplicación que conecta a los dueños de mascotas con cuidadores de mascotas confiables en su área.

Asimismo, se espera que la aplicación 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.

Por todo ello, la creación de la aplicación Mi-Mascota Perruna responde a una necesidad real y actual de los dueños de mascotas y busca mejorar su experiencia al proporcionar un servicio de cuidado personalizado que se adapte a sus necesidades individuales y esté disponible en todo momento.

# ANÁLISIS

Requerimientos funcionales:

* Registro y autenticación tanto en la aplicación como en la página web 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 y hora 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.
* Atención al cliente y ayuda:
  + Ofrecer servicio de atención al cliente para cualquier problema o pregunta desde la aplicación y desde la página web.

Requerimientos no funcionales:

La aplicación y la página 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.

# DISEÑO

## ARQUITECTURA DEL SISTEMA

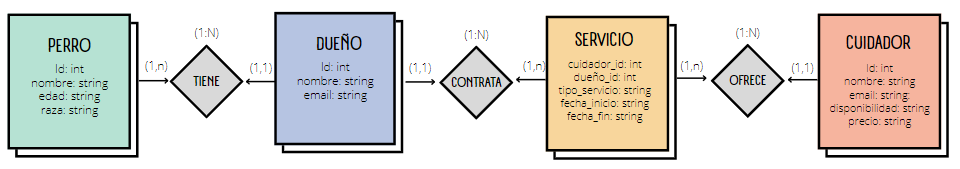
La aplicación está basada en la jerarquía de clases modelo-vista-controlador (MVC) para separar los datos, la lógica de negocio y la interfaz gráfica de usuario. Esta separación de responsabilidades facilita el diseño modular y la independencia de cada componente, es decir, que cada parte pueda ser reemplazada 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 (crear, leer, actualizar y borrar) 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 la página web es HTML y CSS.

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 JavaScript.

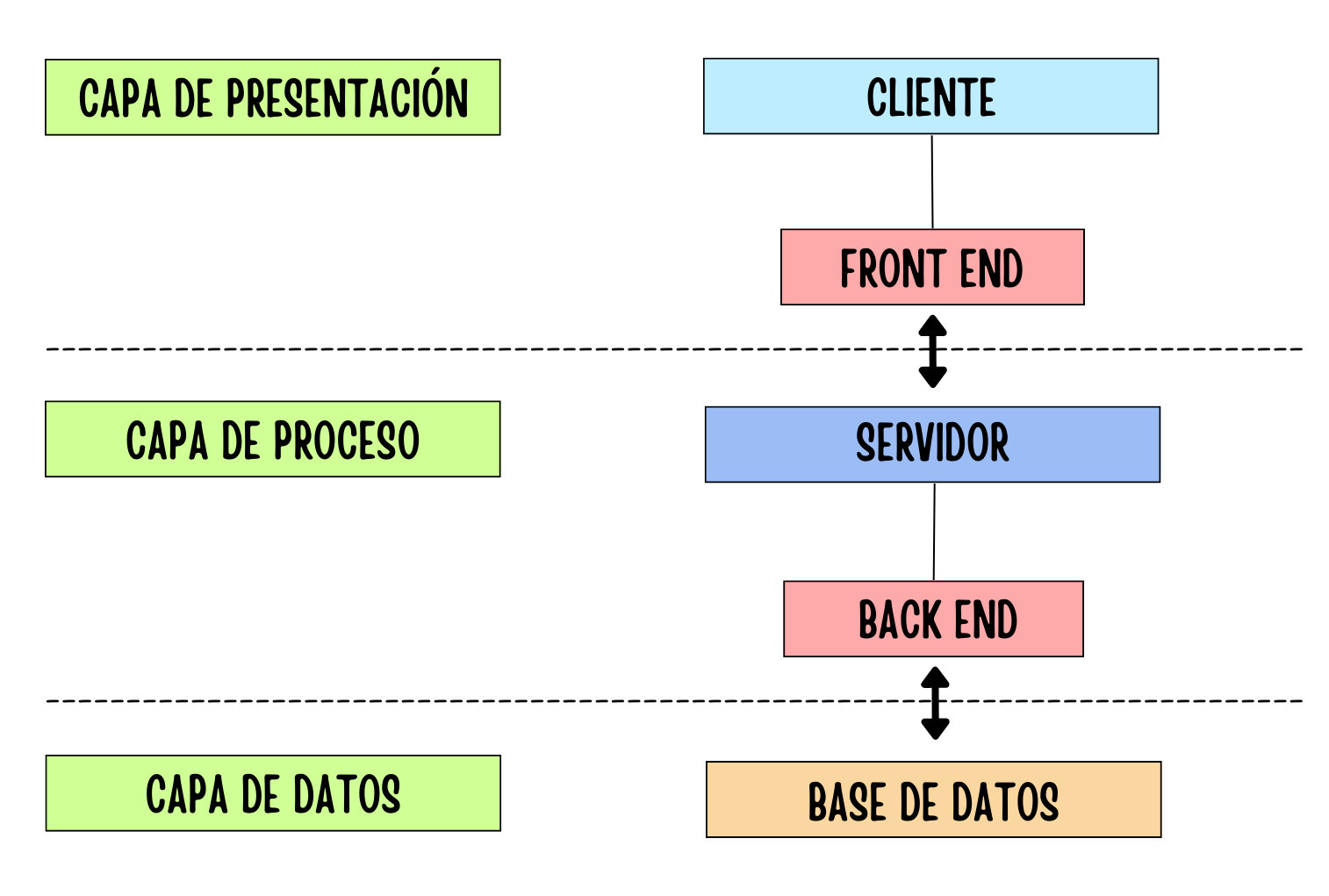
A continuación, se muestra el di0agrama de Entidad-Relación:



Por otra parte, la arquitectura de la página web 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.

A continuación, se muestra un diagrama de la página web:

****

## COMPONENTES

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

### 4.2.1. COMPONENTES DE LA INTERFAZ DE USUARIO

**Componentes de la aplicación Android:**

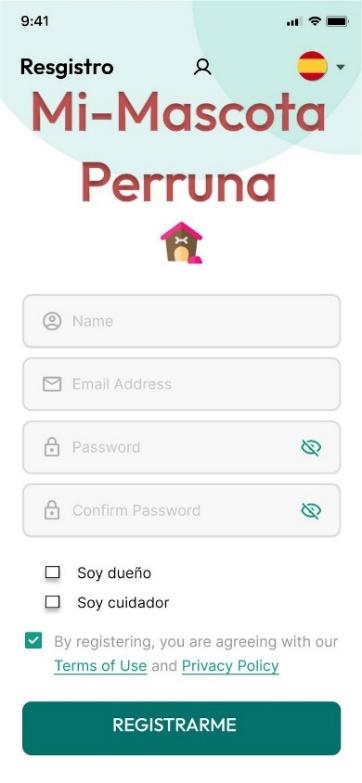
Pantalla de inicio

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

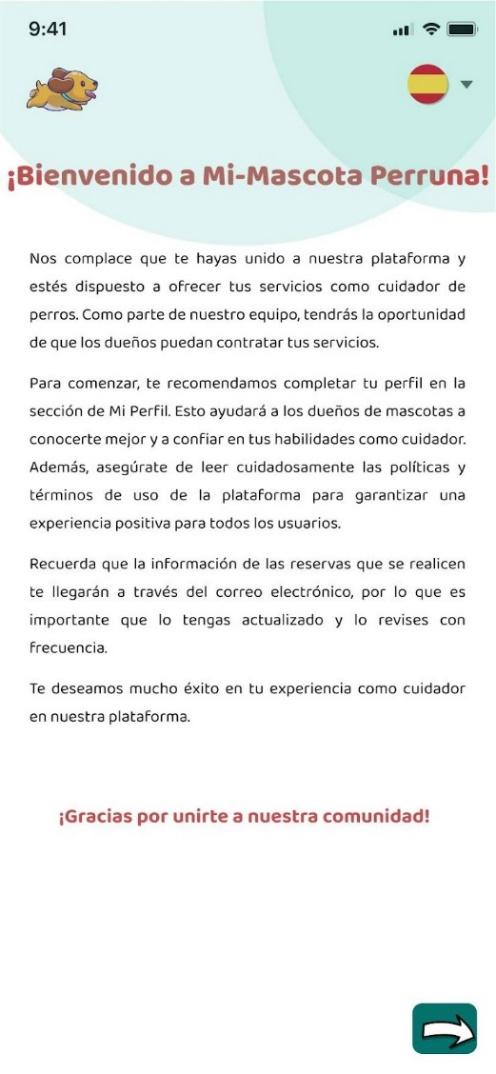


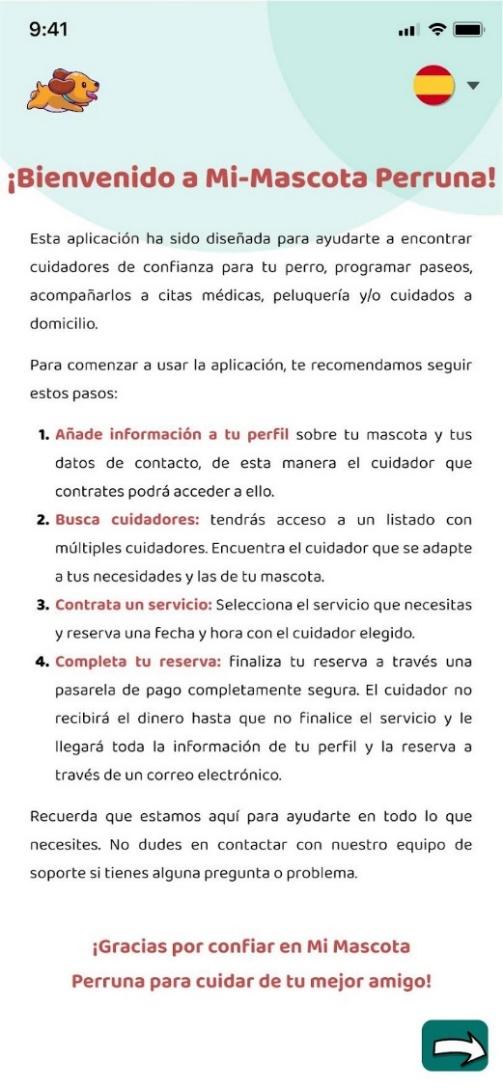
Login/Register

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 Login. En ella, introduciremos las credenciales o iniciaremos sesión con Google. En caso de olvidar la contraseña se tiene la opción de recuperarla.



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.

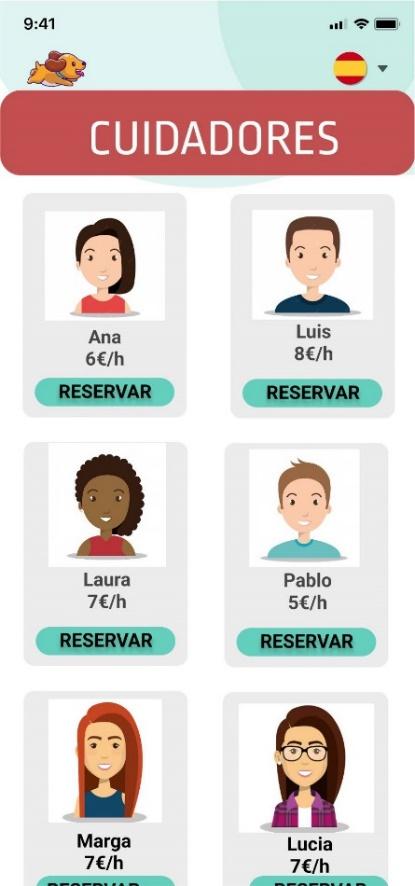


Perfil dueño/cuidador

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.



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.

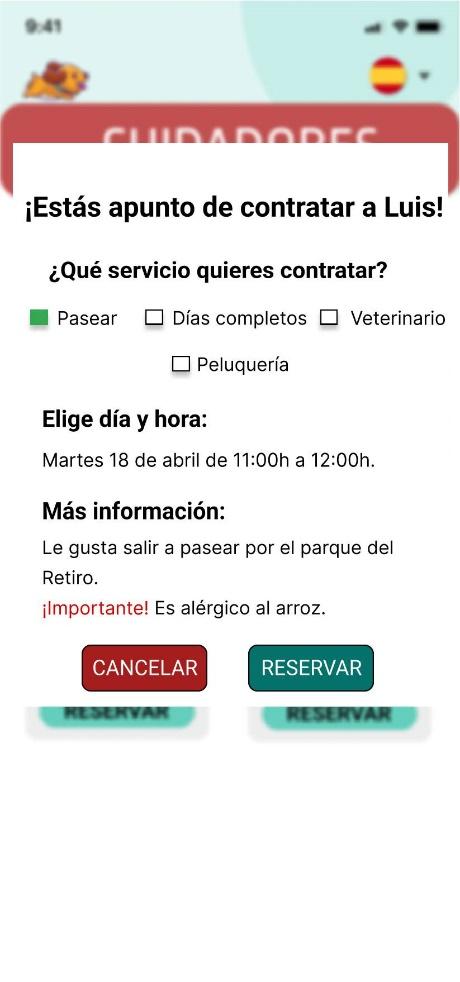
Información del cuidador

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



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.



Pop-up de pre-compra

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



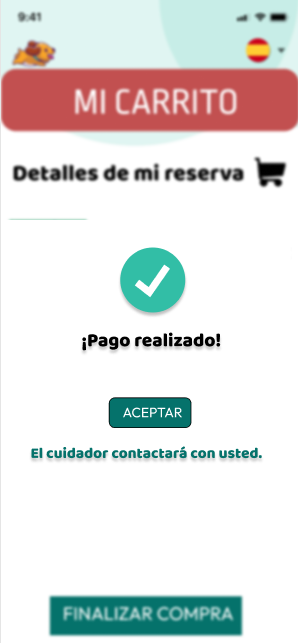
Mi carrito

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

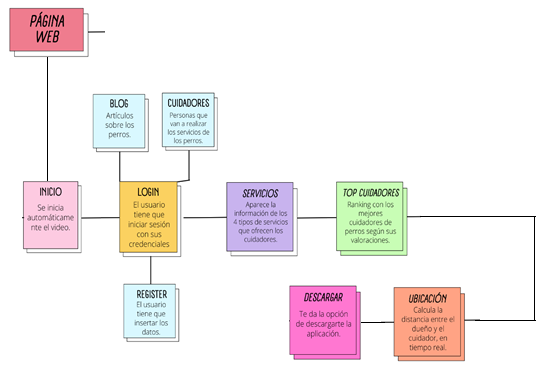


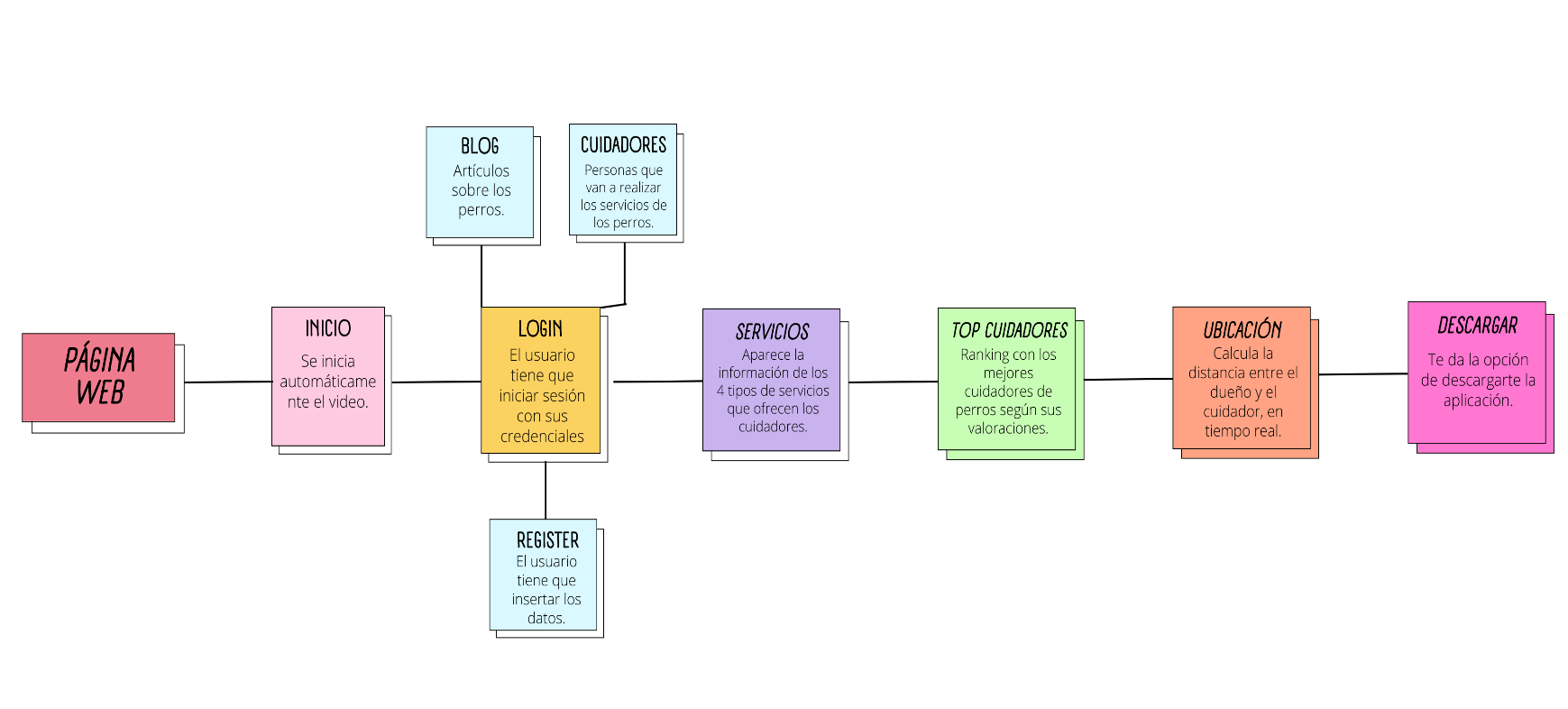
Pop-up de pago

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



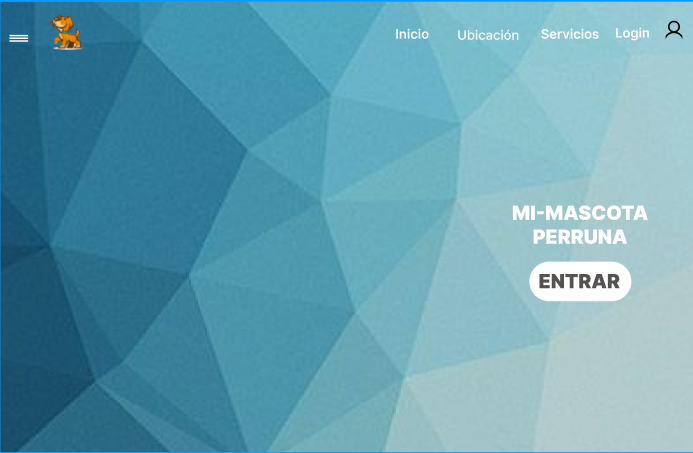
**Componentes de la página web: ( puedes recolocar las cajas, para que la imagen tenga un formato más cuadrado y se vean mejor los textos, al colocarlos todos en línea no te caben bien por el ancho de página )**

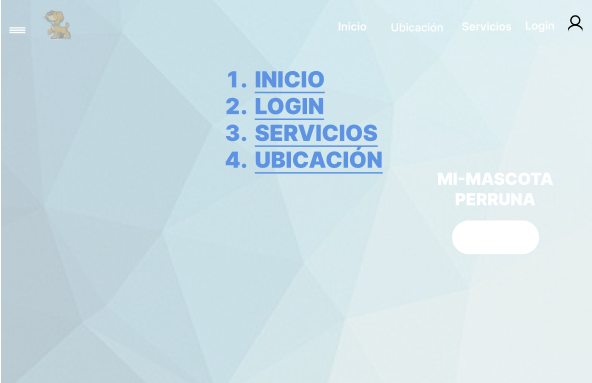




Pantalla de inicio

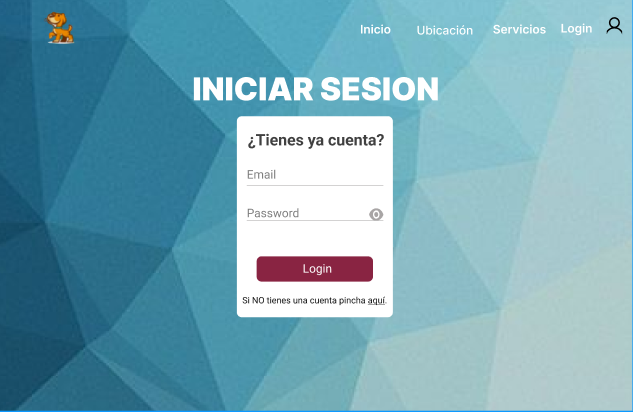
En la pantalla de inicio se mostrará el logo de la aplicación y dos menús, uno desplegable y otro fijo, con un botón que te llevará a iniciar sesión.





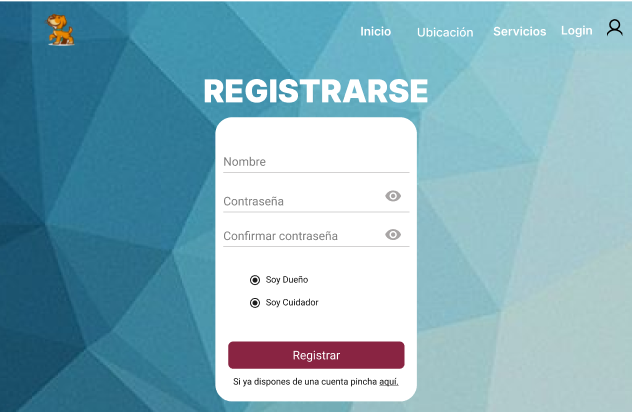
Inicio de sesión

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



Registro

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



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.

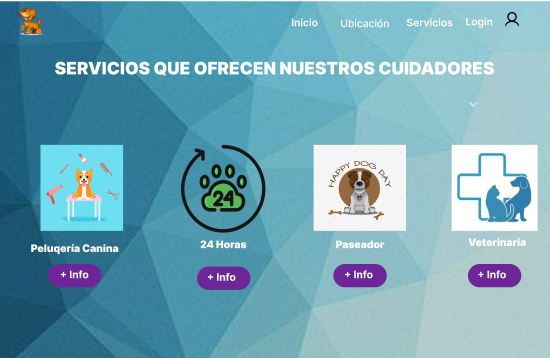
Asimismo, el usuario podrá ver el footer con la política de privacidad y las vías de contacto para poder comunicarse con el soporte de la aplicación.

Aquí he perdido el hilo de la explicación. La calculadora de la ubicación ¿Cuándo sale? Cuando preseleccionas a un cuidador? ¿o cuando lo contratas?



Servicios de los cuidadores

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



Ranking de cuidadores

Se visualizarán los cuidadores que hayan obtenido mejor valoración entre los dueños, con su foto, nombre y un botón que se podrá pulsar para mostrar más información



Perfil Dueño-Cuidador

Pulsando en la ventana de perfil el botón que estará situado en la parte superior a la derecha, mostrará los datos del usuario que inicie sesión.



### 4.2.2. 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 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 (APIs)

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

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. Mostrará 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.

### 4.2.3. TECONOLOGÍAS Y LENGUAJES

Para la elaboración de la página web se han usado tres lenguajes de programación diferentes. HTML ha sido utilizado para estructurar el contenido de la página, definir encabezados, párrafos, imágenes, formularios… CSS se empleó para definir el estilo visual de la página, como por ejemplo la fuente y el tamaño del texto, colores y disposición de los elementos en pantalla, y Java Script para añadir interactividad y dinamismo a la página, creando efectos visuales y permitiendo interactuar con el contenido de la página en tiempo real.

Por otra parte, para la aplicación móvil se ha utilizado el lenguaje XML para la creación de diseños de la interfaz del usuario, donde se definen los elementos visuales y su disposición en la pantalla y para definir 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.

## HERRAMIENTAS UTILIZADAS

Intenta listar/ordenar las herramientas. Te he añadido algunas que citas en la parte de implementación. Visual Studio y NodeJS. ¿NodeJS la utilizas para el back-end? Coméntalo.

Este proyecto ha sido desarrollado mediante la utilización de las siguientes herramientas:

Android Studio

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

Firebase

Android Studio se conecta con Firebase, una plataforma de desarrollo móvil de Google, facilitando la gestión de datos, almacenamiento en la nube, autenticación de usuarios, base de datos en tiempo real y análisis. Además, Firebase permite el uso de APIs, plugins, herramientas de seguridad, entre otros. ¿ realtime database es importante?

Figma

En cuanto al diseño de la aplicación, destaca Figma, una herramienta para crear prototipos interactivos empleada para la elaboración de la propuesta del proyecto.

herramientas hosting y app check

Es importante resaltar las herramientas hosting y app check para la utilización de un hosting y dominio gratuito que aporte seguridad tanto a la página web como a la aplicación. Del mismo modo, la API de Google Service, Recaptcha Enterprise, comprueba y asegura la identidad de los usuarios, evitando bots.

VisualStudio Code

….

Node JS

….

api de Recaptcha en Google Cloud

¿tiene entidad como herramienta o recurso para comentarlo aquí?

Otras herramientas

También se ha utilizado Canvas, para añadir diseños con mayor calidad y facilidad de uso o Envato Elements, una página web de pago que dispone de una gran variedad de diseños e ideas para la app y página web. Asimismo, se ha elaborado un vídeo para el inicio de la página web con la inteligencia artificial proporcionada por la empresa Kaiber AI.

**Si se añade los anuncios en la pag web, hay que añadir las herramientas utilizadas, añadir un nuevo componente y ponerlo en los requerimientos.**

# IMPLEMENTACIÓN

Conectar los login con Firebase

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

 var firebaseConfig = {

            apiKey: "AIzaSyA\_SM6d1nCztP2JON5DHytWOOWs7Jl\_Ff4",

            authDomain: "perruna-cd211.firebaseapp.com",

            projectId: "perruna-cd211",

            storageBucket: "perruna-cd211.appspot.com",

            messagingSenderId: "796763331783",

            appId: "1:796763331783:web:2fd75de7e305206f66b2e2",

            measurementId: "G-RHB6WS2L9H"

        };

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

Conectar la API de Recatchap con la página web desde Firebase:

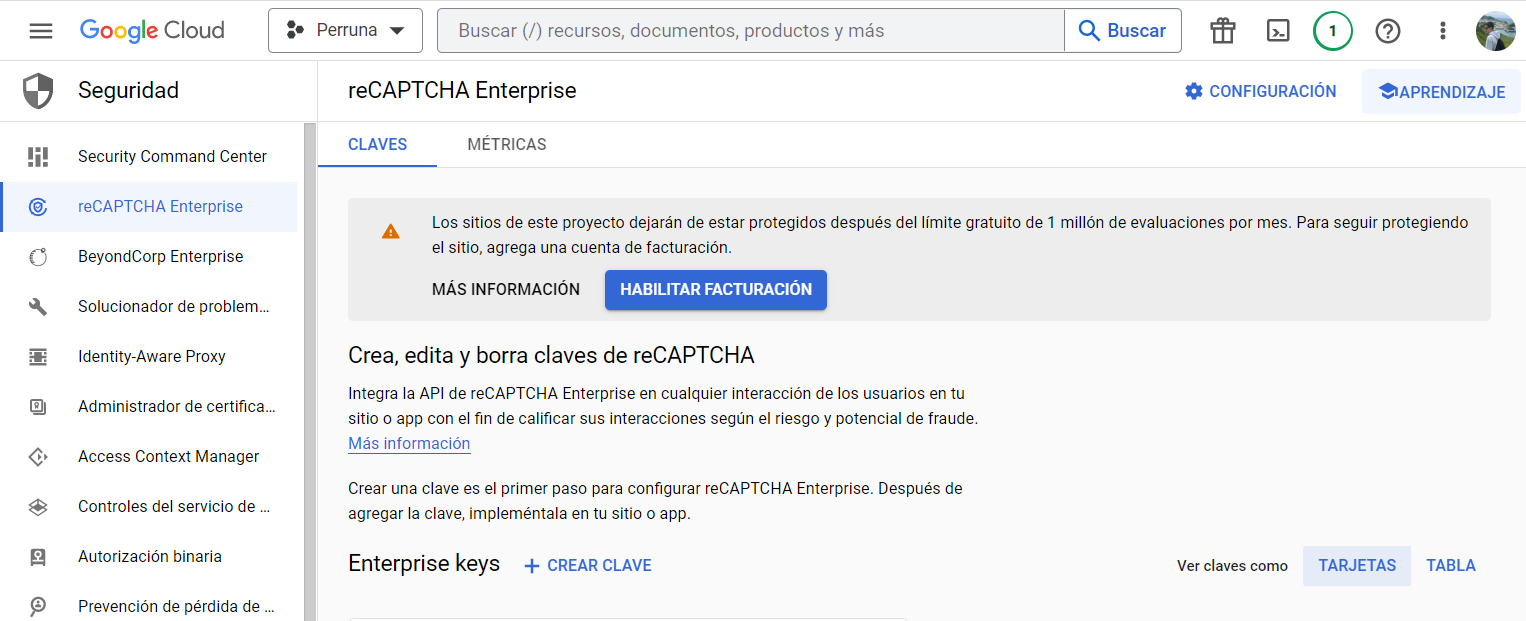
1. En la consola de Google Cloud, vamos a la página reCAPTCHA Enterprise.

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

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

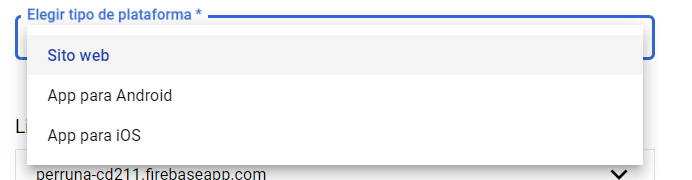


1. Hacemos clic en **Crear clave**.

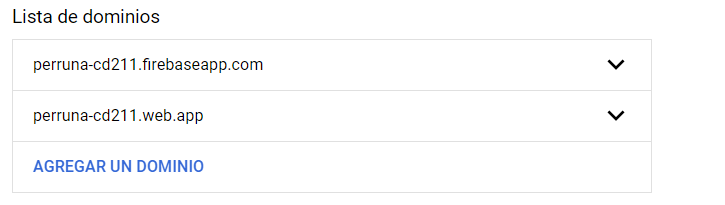


1. En el campo **Nombre visible**, ingresamos un nombre visible.

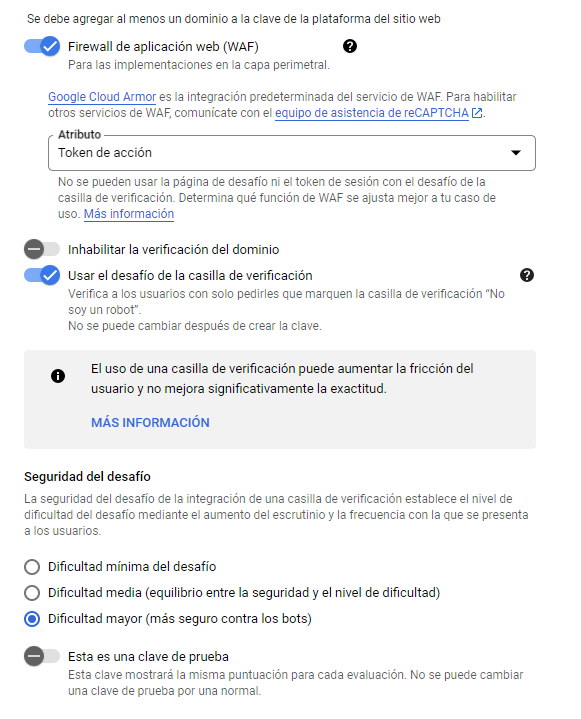
En el menú desplegable **Elegir tipo de plataforma**, seleccionamos **Sitio web**.



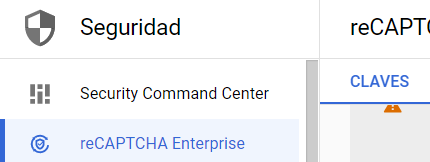
Añadimos el dominio correspondiente a la aplicación.



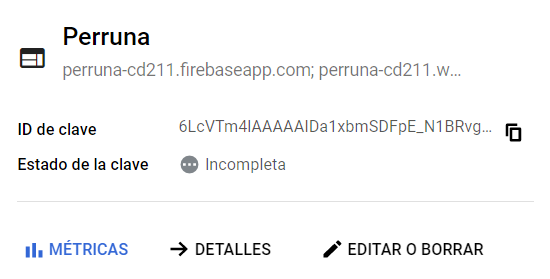
Añadimos el nivel de seguridad que queremos tener en la página web.



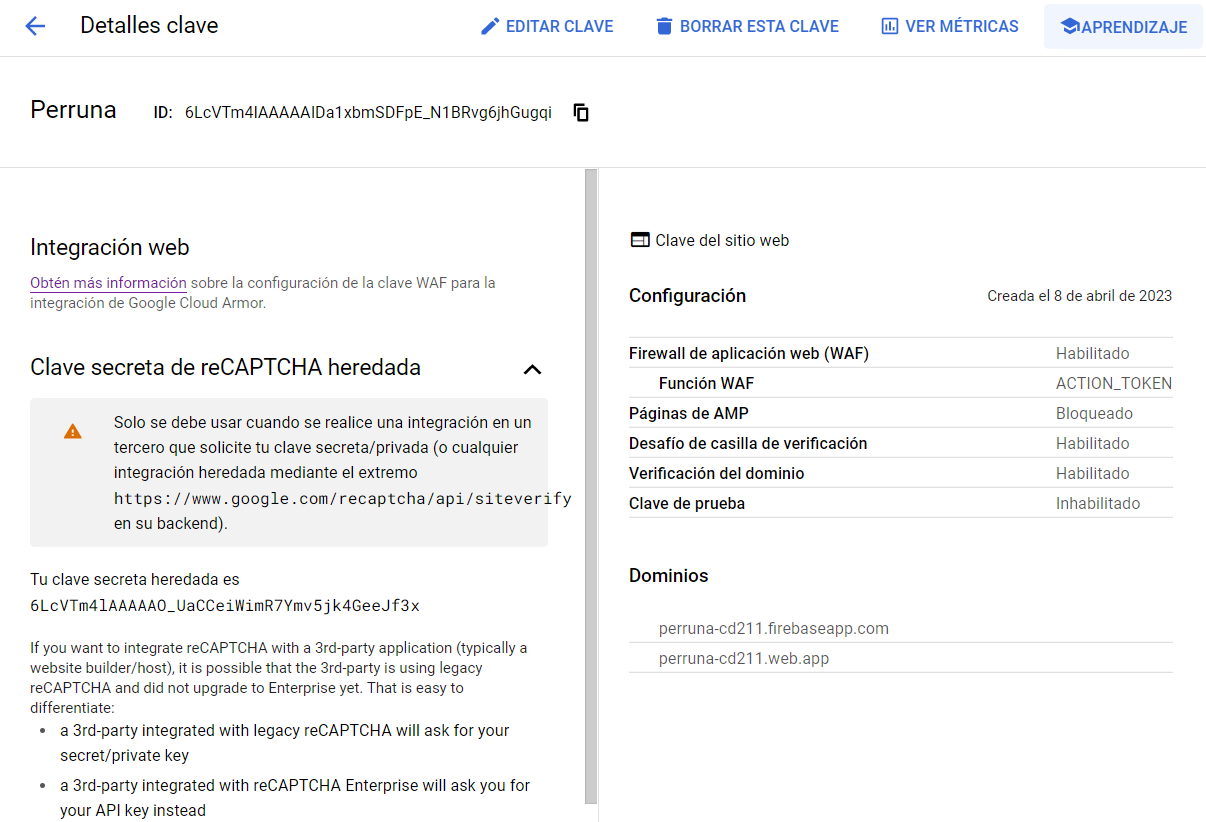
1. La clave creada aparecerá en la página de reCAPTCHA en el apartado de claves.



1. En el estado de la clave, veremos la API incompleta. Para completarla, debemos ir a la opción de detalles.



1. En detalles de la clave encontraremos la clave de seguridad, la secreta heredada y los dominios.



1. Accedemos a la aplicación VisualStudio Code donde tenemos el código HTML y JS de la página web.

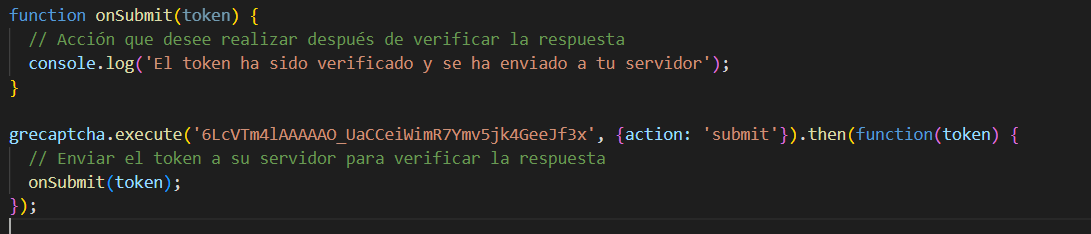
La clave pública la pondremos en el siguiente código del HTML.

<script src="https://www.google.com/recaptcha/api.js?render=6LcVTm4lAAAAAIDa1xbmSDFpE\_N1BRvg6jhGugqi"></script>

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

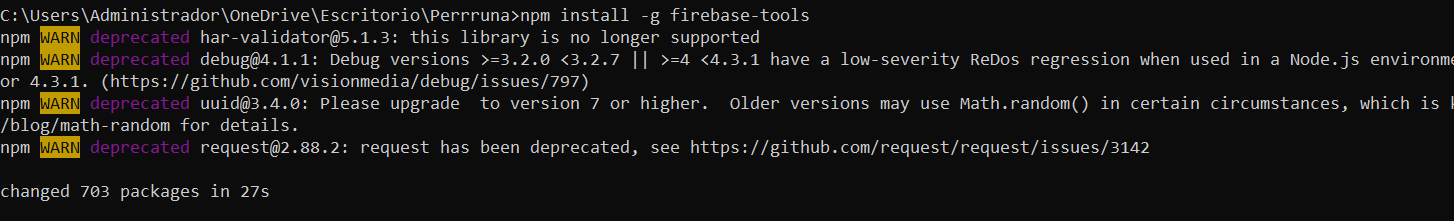
<div class="g-recaptcha" data-sitekey="6LcVTm4lAAAAAIDa1xbmSDFpE\_N1BRvg6jhGugqi"></div>

Al código JS le ponemos la clave heredada.

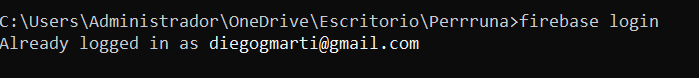


1. Descargamos la aplicación Node.js en <https://nodejs.org/es>  es un entorno de ejecución para JavaScript.

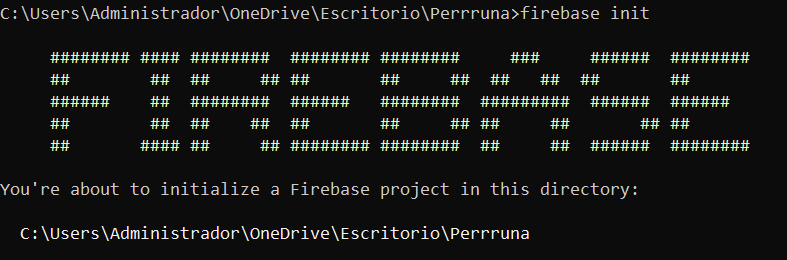
* Abrimos el cmd en la carpeta donde tengamos la página web (HTML, CSS, JS).
* Instalamos las herramientas.



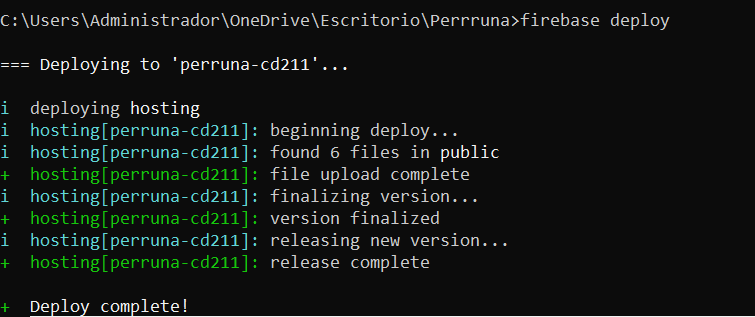
* Iniciamos sesión con Firebase.



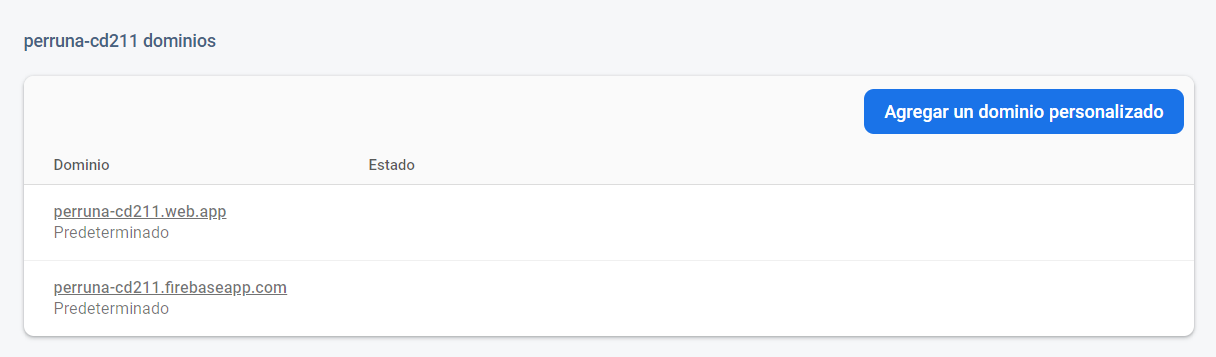
* Iniciamos Firebase.



* 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.



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

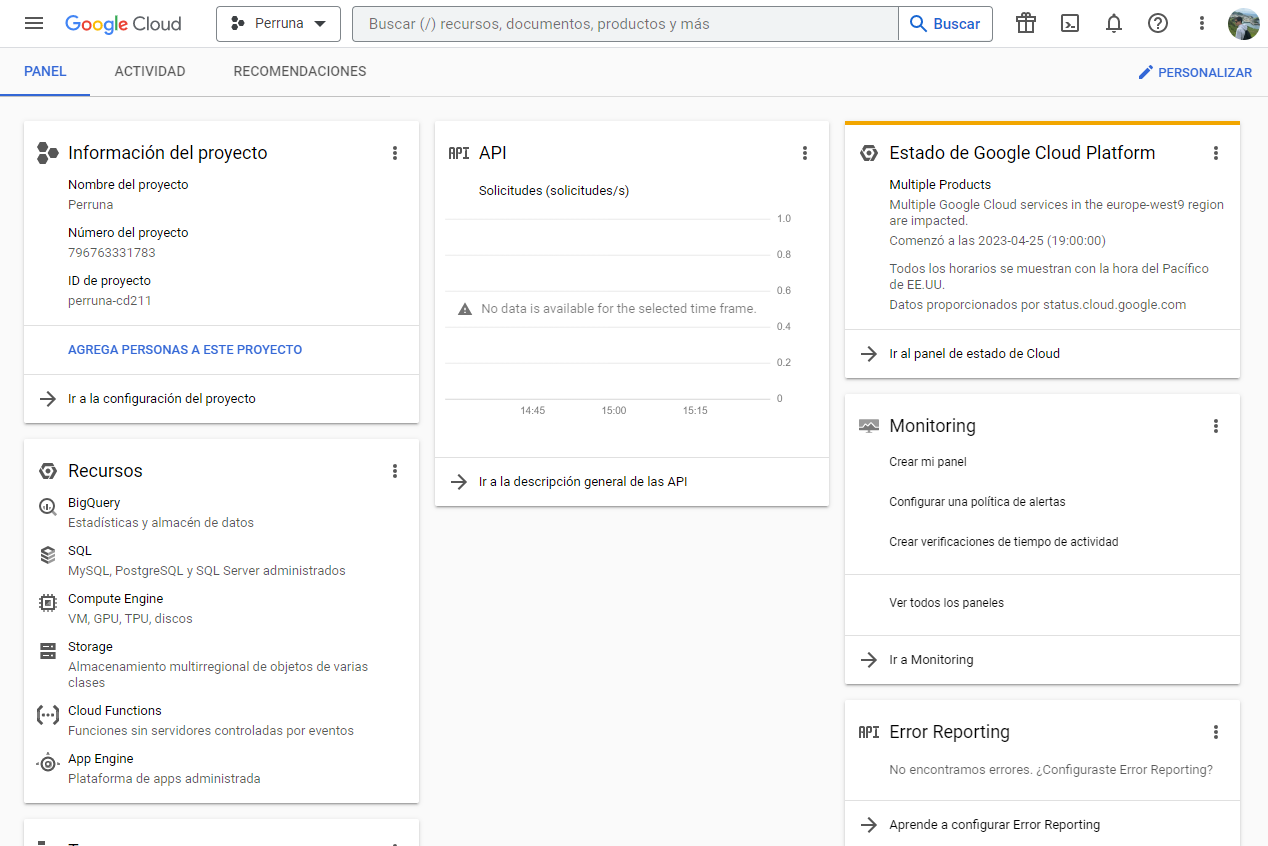


1. Si queremos hacer alguna actualización de la página en el código y luego que se vean en el dominio del Firebase actualizado, tenemos que abrir CMD de nuevo, iniciar sesión con firebase desde consola, hacer el init, sobrescribir el archivo de la página web y actualizar la carpeta.

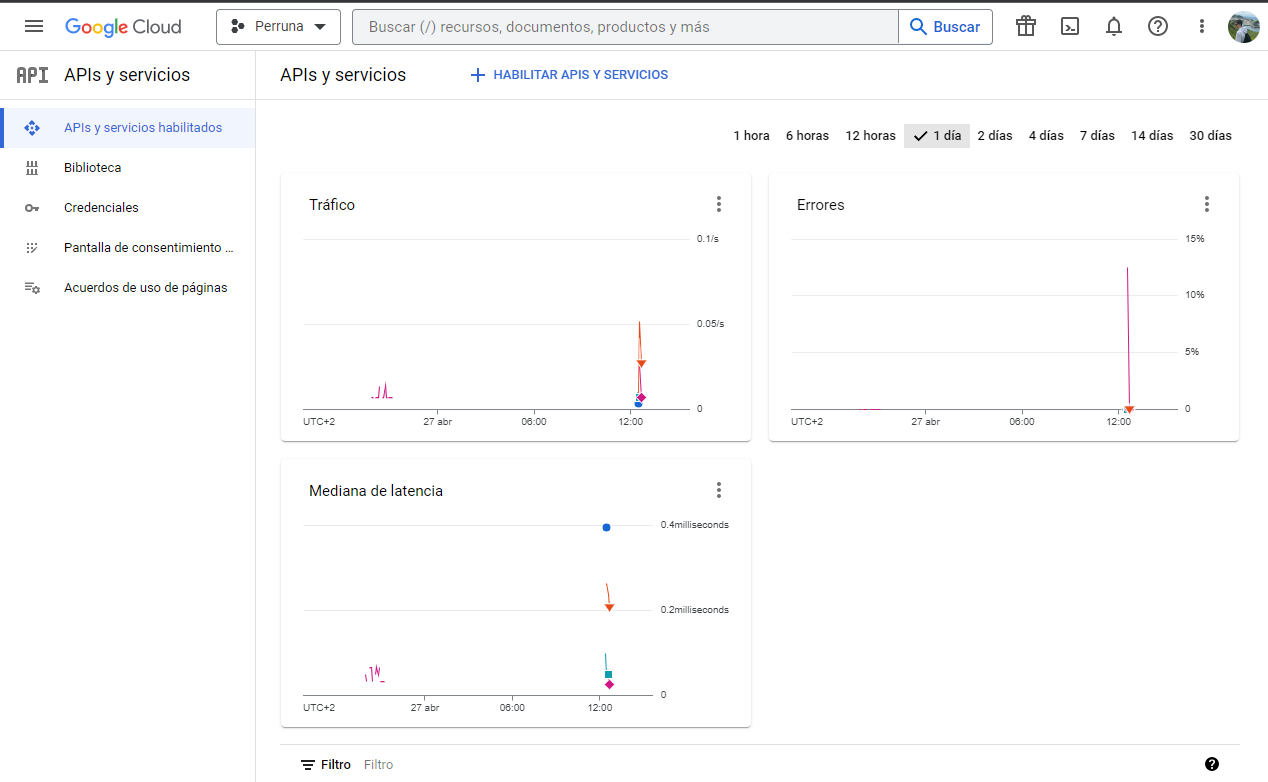
Desde la consola de Google Cloud se puede acceder a la información de las API’s que se han utilizado:

<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.

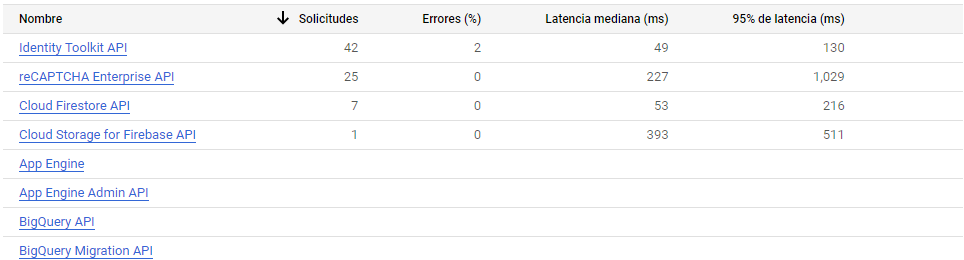


El gráfico de tráfico muestra la cantidad de solicitudes que se han realizado a la API de reCAPTCHA en un periodo de tiempo determinado.

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 API´s 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.



# PRUEBAS

Está muy bien, muy detallado. ¿cómo vas a hacer las pruebas descritas? ¿con alguna herramienta específica de Firebase?

**Comprobaciones de la aplicación**

Para comprobar el correcto funcionamiento de la aplicación se realizarán una serie de pruebas. Para iniciar sesión utilizaremos dos usuarios de prueba, “cuidador” y “dueño”, ambos con contraseña 123456.

* Pantalla 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 Login.

* Pantalla Login-Register

Asegurarse que el estilo de la aplicación se mueve y que podemos pasar de login a register moviendo la pantalla con el dedo. Por otro lado, 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” y loguearse.

Al clicar en resetear contraseña debe saltar un popup 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 Perfil 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 loguearse como dueño.

* Pantalla Dueño

Se comprueba 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.

* Pantalla Listado 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.

Y al seleccionar un cuidador y darle a aceptar se abre un popup 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 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 visualizan las estadísticas de Firebase asegurándose a la vez de que está bien conectada con la aplicación.

**Comprobaciones de la página web**

* Pantalla de Inicio

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

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

* Login

Comprobar que funciona el botón de registrar redirigiendo a una nueva página llamada registro.

Al clicar el botón de contraseña olvidada, debe saltar un popup para introducir el correo electrónico y cuando se da a aceptar, se recibe un mensaje personalizado que permite cambiar la contraseña.

El login 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.

* Register

Los datos que se vayan insertando se tienen que ver reflejados en el firebase y comprobar que se activa la api de Recaptcha en Google Cloud.

* 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.

**TARJETA DE COMPROBACIÓN GENERAL**

* Los usuarios se pueden registrar y logar.
* 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 Firebase.
* Confirmar que la Api se activa en Google Cloud.

Comprobar que se insertan los datos correctamente en Firebase.

Asegurarnos de que Recibe la foto en Firestore.

Comprobar que se insertan los datos correctamente en Firebase.

Asegurarnos de que Recibe la foto en Firestore.

**TARJETA DE COMPROBACIÓN**

**PERFIL CUIDADOR**

**TARJETA DE COMPROBACIÓN**

**PERFIL DUEÑO**

Asegurarnos de que el estilo se mueve.

Al deslizar el dedo se pasa al Register.

Comprobar que se conecta a Firebase.

Al clicar en “resetear contraseña” salta un popup y tiene que recibir un email.

Al pulsar el botón de Google se tiene que iniciar sesión automáticamente.

**TARJETA DE COMPROBACIÓN**

**LOGIN-REGISTER**

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 popup con opciones de reserva.

**TARJETA DE COMPROBACIÓN**

**LISTADO CUIDADORES**

Se tienen que pasar los datos del servicio.

Tiene que redirigirte a la pasarela de pago.

Al realizar el pago se envía un mensaje al correo.

Se visualizan las estadísticas del Firebase asegurándose a la vez que está bien conectada.

**TARJETA DE COMPROBACIÓN**

**CARRITO DE COMPRA**

Al pinchar encima de una tarjeta en el listado de cuidadores, se tiene que abrir una pantalla donde se muestre información más específica.

**TARJETA DE COMPROBACIÓN**

**INFORMACIÓN CUIDADOR**

# EXPLOTACIÓN

Este apartado lo tendría que revisar. Te doy más pistas.

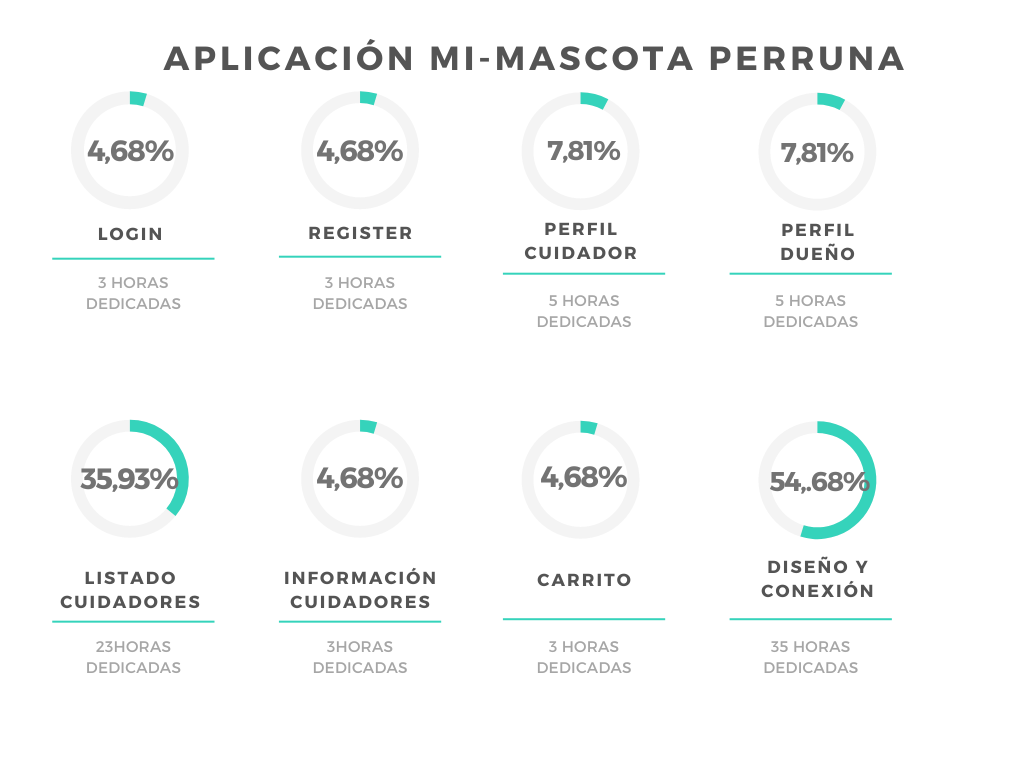
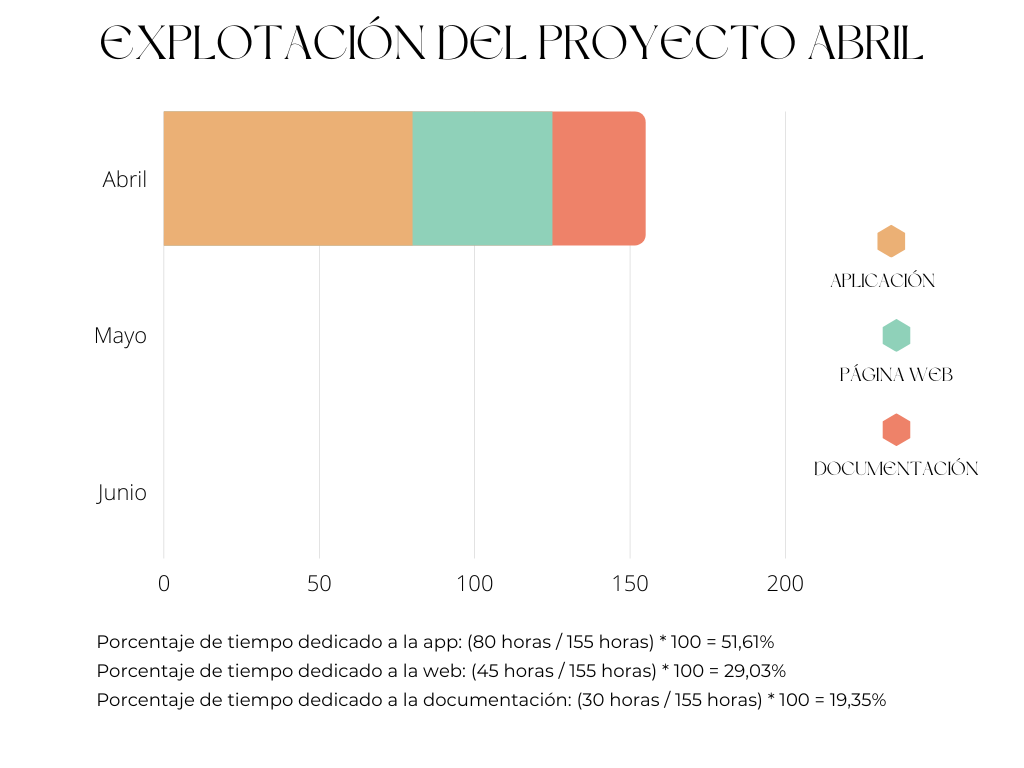
Aquí podrías poner la parte de estadísticas y herramientas de mantenimiento que proporcione Firebase. En implementación comentas estadísticas de uso, errores de la API, eso sería más un tema de administración, monitorización,… explotación del sistema.

Digamos que en implementación lo desarrollas y lo pones en marcha, y un segundo después que esté todo montado lo empiezas a utilizar, lo explotas.

Por ejemplo, el SEO caería dentro de la explotación. Estadísticas de uso, latencias y calidad de servicio de las APIs,… ancho de banda consumido,…

Te paso este enlace, seguro que son competencia de Yak Yak ☺

<https://www.explotacionweb.es/servicios/explotacion-de-producto/>



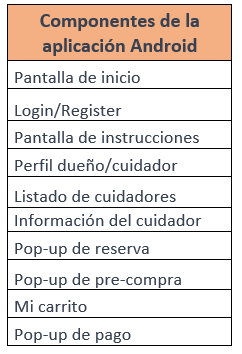


# PRESUPUESTO

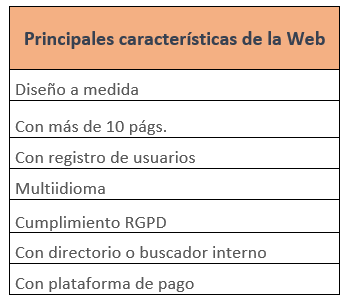
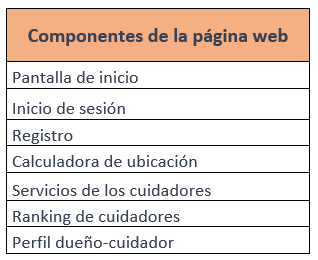
Para la elaboración del presupuesto se tendrán en cuenta una serie de factores. Por un lado las funcionalidades que incluye la aplicación y la tecnología que se utilizará para desarrollarla y, por otro, los costes adicionales relativos a la Inversión en marketing como por ejemplo, el coste de publicar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

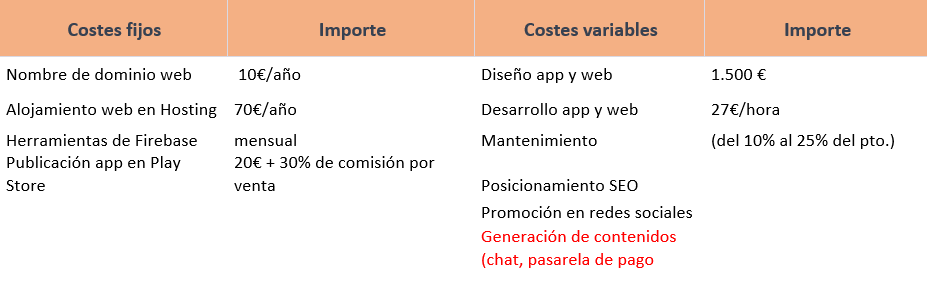
Funcionalidades:

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

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



# Yo creo que el diseño y desarrollo es un coste fijo, puede variar en función del alcance y las horas de desarrollo, pero al final es fijo, lo pagas una vez

# Los costes variables los relaciono más con cuotas mensuales de hosting, firebase, o el mantenimiento, el posicionamiento,…

# Podrías calcular y dar un total sobre cuanto costaría poner en marcha el sistema, y cuanto costaría su mantenimiento y operativa..

# CONCLUSIONES

# LÍNEAS DE TRABAJO FUTURAS

# BIBLIOGRAFÍA

**Figma:**

[**https://www.figma.com/proto/uZ4aIyNXqQKFljzsror54o/Login-(Community)?node-id=4-359&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=4%3A359&show-proto-sidebar=1**](https://www.figma.com/proto/uZ4aIyNXqQKFljzsror54o/Login-(Community)?node-id=4-359&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=4%3A359&show-proto-sidebar=1)

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

**LOGIN Y REGISTER:**

[**https://www.nosinmiubuntu.com/creando-nuestro-sistema-de-registro-y/**](https://www.nosinmiubuntu.com/creando-nuestro-sistema-de-registro-y/)

[**https://www.youtube.com/watch?v=QAKq8UBv4GI**](https://www.youtube.com/watch?v=QAKq8UBv4GI)

**Generar APK y Firmar APP:**

[**https://developer.android.com/studio/publish/app-signing?hl=es-419**](https://developer.android.com/studio/publish/app-signing?hl=es-419)

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

**Diseño:**

[**https://elements.envato.com/es/**](https://elements.envato.com/es/)

[**https://www.kaiber.ai/**](https://www.kaiber.ai/)

[**https://www.canva.com/**](https://www.canva.com/)

**Performance monitoring:**

**https://firebase.google.com/docs/perf-mon/get-started-android?hl=es-419**

**Arquitectura del sistema:**

[**https://repositorio.uam.es/bitstream/handle/10486/662281/gomez\_matesanz\_alfonso\_tfg.pdf?sequence=1&isAllowed=y**](https://repositorio.uam.es/bitstream/handle/10486/662281/gomez_matesanz_alfonso_tfg.pdf?sequence=1&isAllowed=y)

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

**Diagrama de clases:**

[**https://miro.com/es/diagrama/que-es-diagrama-uml/**](https://miro.com/es/diagrama/que-es-diagrama-uml/)

[**https://cloud.smartdraw.com/?nsu=1**](https://cloud.smartdraw.com/?nsu=1)

**Lista/ReciclerView:**

[**https://www.youtube.com/watch?v=5Tm--PHhbJo&list=PLhRVVZ-e6n9XSjSCv7LdzUhjUPqJ8bChM&index=2**](https://www.youtube.com/watch?v=5Tm--PHhbJo&list=PLhRVVZ-e6n9XSjSCv7LdzUhjUPqJ8bChM&index=2)

[**https://www.youtube.com/watch?v=k3zoVAMuW5w**](https://www.youtube.com/watch?v=k3zoVAMuW5w)

**RealTime Database:**

Firebase. Firebase Realtime Database. Store and sync data in real time. Consultada el 14 de junio de 2017, en [**https://firebase.google.com/products/database/**](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**](https://firebase.google.com/docs/database/web/structuredata)**?**

**Hosting y Node.js:**

[**https://firebase.google.com/docs/cli?hl=es-419#install-cli-windows**](https://firebase.google.com/docs/cli?hl=es-419#install-cli-windows)

[**https://firebase.google.com/docs/hosting?hl=es-419**](https://firebase.google.com/docs/hosting?hl=es-419)

**Recaptcha:**

[**https://firebase.google.com/docs/app-check?hl=es-419**](https://firebase.google.com/docs/app-check?hl=es-419)

[**https://firebase.google.com/docs/app-check/web/recaptcha-provider?hl=es-419**](https://firebase.google.com/docs/app-check/web/recaptcha-provider?hl=es-419)

[**https://developers.google.com/recaptcha/docs/v3?hl=es-419**](https://developers.google.com/recaptcha/docs/v3?hl=es-419)

[**https://cloud.google.com/recaptcha-enterprise/docs/implement-tokens?hl=es-419**](https://cloud.google.com/recaptcha-enterprise/docs/implement-tokens?hl=es-419)

[**https://www.youtube.com/watch?v=drvXMz75k5s**](https://www.youtube.com/watch?v=drvXMz75k5s)

[**https://cloud.google.com/recaptcha-enterprise/docs/instrument-web-pages?hl=es-419#frontend\_integration\_score**](https://cloud.google.com/recaptcha-enterprise/docs/instrument-web-pages?hl=es-419#frontend_integration_score)

[**https://cloud.google.com/recaptcha-enterprise/docs/create-assessment?hl=es\_419&\_ga=2.186762728.-1365570470.1678553625#java**](https://cloud.google.com/recaptcha-enterprise/docs/create-assessment?hl=es_419&_ga=2.186762728.-1365570470.1678553625#java)

**Iniciar sesión con Google:**

[**https://www.youtube.com/watch?v=AUOZXy28CJ0&list=PLYx38U7gxBf3lFg7YVv-mEOY27J9T-cIU&index=3**](https://www.youtube.com/watch?v=AUOZXy28CJ0&list=PLYx38U7gxBf3lFg7YVv-mEOY27J9T-cIU&index=3)

**Errores:**

[**https://www.webempresa.com/foro/preguntas-sobre-prestashop/mensaje-conexion-recaptcha**](https://www.webempresa.com/foro/preguntas-sobre-prestashop/mensaje-conexion-recaptcha)

[**https://stackoverflow.com/questions/44323081/firebase-deploy-error-and-fiebase-json-empty-file**](https://stackoverflow.com/questions/44323081/firebase-deploy-error-and-fiebase-json-empty-file)

**Bibliografía apartado presupuesto**

[**https://presupuestoapp.com/cuanto-cuesta-app/**](https://presupuestoapp.com/cuanto-cuesta-app/)

[**https://www.yeeply.com/blog/cuanto-cuesta-crear-una-app/**](https://www.yeeply.com/blog/cuanto-cuesta-crear-una-app/)

[**https://joseramonbernabeu.com/precio-pagina-web/#cual-es-el-precio-de-una-pagina-web**](https://joseramonbernabeu.com/precio-pagina-web/#cual-es-el-precio-de-una-pagina-web)

Firebase. Pricing plans. Consultada 10 de junio de 2017, en https://firebase.google.com/pricing/