



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE COMPUTACIÓN

**DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA PARA LA
GESTIÓN DE GASTOS PERSONALES Y FAMILIARES CON UN ENFOQUE A
TOMA DE DECISIONES MEDIANTE ANALÍTICA DE DATOS**

Trabajo de titulación previo a la obtención del
título de Ingeniero en Ciencias de la Computación

AUTORES: JORGE MARTIN BOJORQUE MACANCELA
IVÁN MARCELO ULLOA BERMEO
TUTORA: ING. BERTHA KATERINE TACURI CAPELO

Cuenca - Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Jorge Martin Bojorque Macancela con documento de identificación N° 0105070569 e Iván Marcelo Ulloa Bermeo con documento de identificación N° 0106993058; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 20 de julio del 2022

Atentamente,

Jorge Martin Bojorque Macancela
0105070569

Iván Marcelo Ulloa Bermeo
0106993058

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Jorge Martin Bojorque Macancela con documento de identificación N° 0105070569 e Iván Marcelo Ulloa Bermeo con documento de identificación N° 0106993058, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Desarrollo de una aplicación móvil multiplataforma para la gestión de gastos personales y familiares con un enfoque a toma de decisiones mediante analítica de datos”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 20 de julio del 2022

Atentamente,

Jorge Martin Bojorque Macancela
0105070569

Iván Marcelo Ulloa Bermeo
0106993058

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Bertha Katerine Tacuri Capelo con documento de identificación N° 0102972627, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA PARA LA GESTIÓN DE GASTOS PERSONALES Y FAMILIARES CON UN ENFOQUE A TOMA DE DECISIONES MEDIANTE ANALÍTICA DE DATOS**, realizado por Jorge Martin Bojorque Macancela con documento de identificación N° 0105070569 y por Iván Marcelo Ulloa Bermeo con documento de identificación N° 0106993058, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico, que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 20 de julio del 2022

Atentamente,

Ing. Bertha Katerine Tacuri Capelo

0102972627

Resumen

Este proyecto aborda la problemática de manejar las finanzas de la familia y ofrece una manera simple y conveniente de hacerlo: el uso de una aplicación móvil. Esta busca controlar los gastos realizados en el lapso de un mes, realiza el seguimiento de presupuestos y presenta reportes con información útil basada en su uso. Nuestro objetivo fue conseguido con el uso de múltiples tecnologías tales como el ambiente de Node.js en conjunto con el framework de desarrollo multiplataforma Ionic, orientado a construir un frontend atractivo. Adicionalmente, se usa Google Firebase y su colección de características como un backend confiable que escala dinámicamente con la aplicación. También se utiliza inteligencia de negocios y análisis de datos para presentar gráficas útiles para el monitoreo de los gastos de la familia. El presente documento detalla las actividades realizadas para cumplir los objetivos impuestos, incluyendo: la problemática estudiada, conceptos generales, la metodología seguida junto con las distintas fases del desarrollo y varias pruebas que comprueban su desempeño.

Palabras clave – gastos, finanzas, aplicación, multiplataforma, analítica, cómputo en la nube

Abstract

This project addresses the problem of managing family finances and offers a simple and convenient way to do it: a mobile application. This application seeks to control the expenses made in a month, tracks different budgets and presents useful reports with information based on their use. Our objective was achieved with the use of multiple technologies such as the Node.js environment in conjunction with the Ionic cross-platform development framework, to building an attractive frontend. Additionally, Google Firebase and its collection of features are used as a reliable backend that dynamically scales with the application. Business intelligence and data analysis are also used to present useful graphs for monitoring family expenses. This document details the activities carried out to meet the set objectives, including studied problem, general concepts, the followed methodology along with the different phases of development and various tests that verify its performance.

Keywords – *spences, finances, application, multiplatform, analytics, cloud computing*

ÍNDICE DE CONTENIDO

Resumen.....	4
Abstract.....	5
I. Introducción.....	8
II. Problema.....	9
III. Objetivos Generales y Específicos	9
Objetivo General.....	9
Objetivos específicos (OE).....	9
IV. Revisión de la literatura o fundamentos teóricos.....	10
A. Trabajos Relacionados	10
B. Apps de organización personal	11
C. Analítica de Datos	12
D. Inteligencia de Negocios.....	12
E. Node.js.....	13
• NPM	13
• Simple Statistics	13
• regresions-js.....	13
• Chart.js	13
• Frameworks.....	13
G. Firebase.....	14
• Cloud Functions para Firebase.....	14
• Cloud Firestore.....	14
• Firebase Authentication	15
H. NGINX	15
I. Metodología de desarrollo Scrum.....	15
• Planificación del sprint.....	15
• Reuniones diarias	15
• Revisión del sprint.....	15
• Retrospectiva del sprint.....	15
• Scrum master	16
• Scrum team.....	16

•	Product owner	16
V.	Marco Metodológico	16
A.	Requerimientos de la Aplicación	16
a)	Requisitos funcionales	17
b)	Requisitos no funcionales	18
B.	Diseño y Codificación	19
C.	Pruebas y Manuales	27
VI.	Resultados	29
A.	Resultados por Funcionalidad	30
B.	Resultados de las Pruebas de Integración	32
a)	Integración de funcionalidades para miembros de la familia	32
b)	Integración de funcionalidades para el jefe de hogar	33
C.	Pruebas de Rendimiento	36
D.	Pruebas de aceptación	38
E.	Manuales	40
VII.	Cronograma	42
VIII.	Presupuesto	46
IX.	Conclusiones	47
X.	Recomendaciones	48
	REFERENCIAS BIBLIOGRÁFICAS	50
	ANEXOS	52
	Anexo 1: PROTOTIPOS DE INTERFACES DE USUARIO	52
	Anexo 2: PRUEBAS UNITARIAS	55
	Anexo 3: PRUEBAS DE INTEGRACIÓN	65
	Anexo 4: PRUEBAS DE RENDIMIENTO	66
	Anexo 5: ENCUESTA PARA PRUEBA DE ACEPTACIÓN	72
	Anexo 6: RESULTADOS DE LAS ENCUESTAS PARA LAS PRUEBAS DE ACEPTACIÓN ..	72
	Anexo 7: MANUAL DE USUARIO	75
	Anexo 8: MANUAL TÉCNICO DEL SISTEMA	81

I. Introducción

Conocimiento previo y experiencia son necesarios para un manejo adecuado de las finanzas de un hogar. Se han presentado varios métodos conforme avanza la tecnología y el conocimiento en estas áreas, mas no todos han tenido acceso a estos avances. Gracias a la introducción de los dispositivos móviles muchas aplicaciones que buscan suplir los requisitos antes mencionados son liberadas al público en general. Muchas de las antes mencionadas hacen uso de infraestructuras con recursos en la nube, como las provistas por compañías como: Google, Amazon o Microsoft; esto crea un ambiente de desarrollo óptimo para la innovación y desarrollo de nuevas propuestas.

Así como se presenta abundancia en las infraestructuras diseñadas para los propósitos ya mencionados se observa igualmente una abundancia en las herramientas y bases para el desarrollo de aplicaciones. Una técnica definida para esto es el desarrollo para múltiples plataformas, con este acercamiento se busca la unificación de las funcionalidades independientemente del sistema operativo; es decir que las características y comportamientos del software no varían por limitaciones arbitrarias que pueden ser impuestas al desarrollar para un hardware específico.

Específicamente para suplir el conocimiento previo y la experiencia se busca un enfoque para la toma de decisiones en la presentación de resultados. De forma independiente del lenguaje en el que la aplicación fue construida, un gráfico puede ser interpretado de manera exitosa. Los elementos de dicho gráfico deben ser la estructura de un mensaje que se desea transmitir al espectador. La mayoría de los gráficos recurren a: colores, tamaño, ejes; para expresar diferentes características del mensaje que buscan transmitir. Un enfoque en las funcionalidades naturalmente nos lleva a la utilización de tecnologías de desarrollo que favorecen este enfoque.

El presente documento detalla el problema, conceptos, metodología y resultados del proyecto de desarrollo de una aplicación multiplataforma para la gestión de gastos personales y familiares denominada “GestionarGastos”. Las secciones incluyen: detalle del problema y su propuesta de solución, objetivos generales y específicos del proyecto, conceptos teóricos para el desarrollo y trabajos relacionados, la metodología seguida para el desarrollo de la propuesta de solución, los resultados que describen los productos de la metodología, cronograma de actividades, presupuestos tentativos, conclusiones y finalmente recomendaciones. A continuación, se dará a conocer en detalle, el problema de estudio.

II. Problema

En un entorno tanto familiar como personal existen diversos aspectos a tener en cuenta que involucran nuestra salud física, alimentación y finanzas. La falta de tiempo juega un papel importante en la mala gestión de aspectos personales; esto puede llevar a malos hábitos como una dieta no balanceada o la falta de ejercicio. La gestión de las finanzas familiares puede resultar un contratiempo de igual manera, inflando el presupuesto familiar o limitando el presupuesto para gastos esenciales.

Una solución propuesta para el manejo de los gastos familiares es la utilización de aplicaciones, estas permiten la gestión de un presupuesto disponible con el objetivo de generar estrategias dirigidas al manejo del dinero, seguir y controlar las transacciones realizadas, generar planes de ahorro y así conseguir satisfacción financiera (Rossellinee, 2019). Estas herramientas permiten visualizar los gastos y crear una base para el control del dinero, todo esto en conjunto con consultoría financiera especializada en situaciones específicas.

El presente proyecto propone el desarrollo de una aplicación para la gestión de gastos personales y familiares utilizando tecnologías para aplicaciones móviles y herramientas en la nube, esto con el objetivo de generar un seguimiento de los gastos en distintas categorías de cada miembro de la familia dado un determinado presupuesto total. Además de mostrar grafos de los gastos en diferentes categorías aportando en la proyección de gastos futuros.

La siguiente sección expondrá los objetivos planteados para este proyecto.

III. Objetivos Generales y Específicos

Se definen los objetivos generales y específicos del proyecto, detallando así la extensión de la solución propuesta.

Objetivo General: Desarrollar una aplicación móvil multiplataforma para la gestión de gastos personales y familiares con un enfoque a toma de decisiones mediante analítica de datos

Objetivos específicos (OE):

- **OE1:** Estudiar y conocer el desarrollo de aplicaciones móviles multiplataforma junto con los fundamentos de analítica de datos e inteligencia de negocios
- **OE2:** Definir los requerimientos funcionales y no funcionales de la aplicación

- **OE3:** Desarrollar los módulos de la aplicación para la gestión de gastos personales y familiares, presentación de grafos dirigidos a toma de decisiones
- **OE4:** Diseñar y ejecutar pruebas funcionales de la aplicación de gestión de gastos
- **OE5:** Elaborar el manual técnico y de usuario de la aplicación de gestión de gastos

Esta información actuará como guía general para detallar los requerimientos de nuestra aplicación propuesta.

La siguiente sección y subsecciones detallarán los conceptos utilizados para el desarrollo y su implementación.

IV. Revisión de la literatura o fundamentos teóricos

En esta sección daremos una introducción a los conceptos técnicos utilizados para la realización de la propuesta de solución. Incluye la revisión de algunos trabajos relacionados como conocimiento general del estado de arte. Además, esta se compone de conocimientos teóricos generales y técnicos que servirán como punto de partida para las actividades de desarrollo propio del proyecto.

A. Trabajos Relacionados

Para comprender las necesidades cubiertas por aplicaciones similares a nuestra propuesta de solución, su lugar en el mercado y determinar la relevancia que tiene el proyecto presente, se dará a conocer algunos ejemplos. Se han investigado aplicaciones dispuestas en Google Play que cumplan ciertas características: ser usadas para el manejo de finanzas personales, poseer manejo de gastos y poseer reportes gráficos.

Como primer ejemplo tenemos la aplicación “Gestor de gastos, ingresos y presupuesto, dinero” desarrollada por Innim Mobile Exp; esta tiene más de un millón de descargas y su reseña en Google Play es de 4.9 sobre 5 estrellas. Sus características incluyen: control de gastos (gráfica de resumen), categorías de gastos y la posibilidad de crearlas, ingreso de transacciones, permitir pagos con distintas monedas y recordatorios. Tiene un aspecto gráfico simple y el manejo de los gastos es exclusivo para una persona.

La aplicación “Wallet” desarrollada por BudgetBakers.com, con más de cinco millones de descargas y con reseñas promedio de 4.8 sobre 5 estrellas, es la una de las aplicaciones líder

entre las mencionadas. Sus características incluyen: control de gastos, categorías, establecer presupuestos, control de créditos, recordatorios y otras varias. La aplicación es muy profunda en términos de manejo de finanzas, pero es posible que muchas de estas no sean usadas por la mayoría de los usuarios.

“Monefy” desarrollada por Reflectly, que tiene más de cinco millones de descargas y una puntuación de 4.7 sobre 5 estrellas, es otra de las más grandes aplicaciones. Sus características incluyen: reportes gráficos, categorías e historiales de gastos. También tenemos “Expense Manager” desarrollada por CASorin que tiene más de un millón de descargas con una media de reseñas de 4.7 sobre 5 estrellas. Sus características incluyen: ingreso de gastos, categorías de gastos, resúmenes gráficos. Estas aplicaciones son mucho más simples que las anteriores y como resultado tienen características limitadas, lo cual no es un impedimento para generar un gran atractivo en la tienda.

Por último, tenemos la aplicación “Registro contable” desarrollada por Realbyte Inc. Con más de diez millones de descargas y con reseñas medias de 4.9 sobre 5 estrellas. Es la aplicación con más funcionalidades dentro de esta categoría, contiene todas las características mencionadas para las otras aplicaciones expuestas en esta sección adicionando otras como: fotografías de las facturas y compras, presupuestos con filtros reforzados, calendarios, gráficas de activos y una versión web con un Dashboard de las finanzas.

B. Apps de organización personal

Es una categoría de aplicaciones móviles que permiten mantener un registro de actividades de toda índole, de esta manera se gestionan actividades: diarias, semanales y anuales; con el objetivo de mantener una vida más organizada, controlar ciertos aspectos de manera rigurosa y/o generar buenos hábitos de comportamiento. Estas tecnologías han sido adaptadas en los últimos años principalmente por adultos jóvenes, estos utilizan todas las herramientas disponibles acorde a sus necesidades con el objetivo de mantener un mejor estilo de vida; resultando así en un mayor número de personas con mayor flexibilidad, recontextualización de necesidades y generación de nuevas tecnologías que busquen acomodarse a esas necesidades (Tully & Alfaraz, 2017). Estas aplicaciones pueden encontrarse en distintas categorías como finanzas, fitness, salud, alimentación, entre otras.

C. Analítica de Datos

La analítica de datos vio su apogeo en la década de los 2000. Como su nombre lo indica busca analizar datos para obtener nuevo conocimiento que afecte su influencia sobre aquellos que los estudian. Cuando hablamos de análisis nos referimos a la aplicación de sistemas informáticos al análisis de grandes volúmenes de datos para toma de decisiones (Runkler, 2020). El análisis de datos es un campo con muchas temáticas y tópicos que a su vez ha adoptado aspectos de muchas otras disciplinas científicas como: estadística, teoría de señales, reconocimiento de patrones, inteligencia computacional, aprendizaje automático e investigación de operaciones.

Los proyectos típicos de análisis de datos se ven divididos en diferentes fases: evaluación y selección, limpieza y filtración; y finalmente, visualización y análisis. El descubrimiento del conocimiento en las bases de datos (KDD) comprende las seis fases de selección, preprocesamiento, transformación, extracción de datos, interpretación y evaluación. También tenemos el proceso estandarizado de la industria cruzada para minería de datos (CRISP-DM) que indica las seis fases de comprensión de datos: preparación de datos como proceso preliminar, modelado, evaluación y por último implementación. Para simplificar, destacamos solo cuatro fases: preparación, preprocesamiento, análisis y pos-procesamiento.

D. Inteligencia de Negocios

La inteligencia empresarial es un término que se creó a mediados de la década de 1990 para describir la acción de tomar datos en su forma original y convertirlos en algo utilizable, lo cual puede ser la base de decisiones empresariales. Es un término que acoge a otras varias disciplinas de datos como: minería de datos, análisis estadístico, pronósticos y apoyo a la toma de decisiones (Pareek, 2006). La inteligencia de negocios es una estrategia de negocios dirigida a comprender y anticipar las necesidades de los requisitos actuales de una empresa, los cuales incluyen: conocimiento de los clientes de la empresa, sus competidores, socios comerciales, ambiente competitivo y sus operaciones internas, esto le da a la gerencia de la empresa la capacidad de hacer efectivas varias decisiones comerciales estratégicas. En resumen, estos sistemas incorporan inteligencia de negocios la cual combina datos actuales con herramientas de análisis para visualizar información compleja e importante para aquellas personas que

toman decisiones y aquellos que las planifican. Tiene como objetivo, además, mejorar la oportunidad y calidad de la entrada al proceso de decisión.

E. Node.js

Concebido como un entorno de código abierto y multiplataforma controlado por eventos en tiempo de ejecución del lado del servidor utilizando JavaScript, Node.js se enfoca en el desarrollo de aplicaciones escalables, permitiendo establecer y gestionar varias conexiones simultaneas de manera asíncrona, la comunicación es simplificada al no usar subprocesos y no depender de multi-threading para soportar la ejecución de la lógica de negocios. Un proceso de Node se ejecuta en forma de un demonio de un solo hilo (single-thread) que a su vez es embebido con el motor de JavaScript para soportar la personalización (Tilkov & Vinoski. 2010).

- **NPM:** Es el gestor de paquetes de la plataforma Node JavaScript, los paquetes son provistos por uno de los registros de software más grandes para desarrolladores independientes. Con la ayuda de *npm* es posible descargar herramientas por separado que pueden ser usadas inmediatamente, correr paquetes sin descargarlos, adaptar e incorporar código para tus aplicaciones, manejar versiones del código, entre otras características.
- **Simple Statistics:** Es una librería que implementa métodos estadísticos utilizando JavaScript. Está diseñada para funcionar en todos los navegadores modernos, la librería esta apropiadamente documentada, escrita en código amigable y rigurosamente probada.
- **regresions-js:** Es un módulo de JavaScript que contiene una colección de métodos lineales basados en mínimos cuadrados para análisis simple de datos. Contiene código para regresiones lineales, exponenciales, logarítmicas y polinomiales.
- **Chart.js:** Librería gratuita y de código abierto de JavaScript enfocada en la construcción de gráficos utilizando la HTML como base. Soporta varios tipos de gráficos como: Barras, área polar, burbuja, líneas, dispersión, etc.
- **Frameworks:** Es un esquema de trabajo que permite el desarrollo de aplicaciones de manera más sencilla utilizando plantillas que proveen funcionalidades genéricas que sirven como punto de partida, también conocido como un entorno reusable de software. Node.js ofrece Frameworks como IONIC, Express o Koa que pueden ser instalados utilizando NPM

u otros métodos aprovechando todas las ventajas del entorno de ejecución antes mencionado.

F. Ionic

De la documentación oficial provista por Ionic se tiene que: *“Ionic es un kit de herramientas de código abierto para el desarrollo de apps móviles y de escritorio de alta calidad y desempeño utilizando tecnologías web HTML, CSS y JavaScript con integraciones de Frameworks como Angular, React y Vue”*. Caracterizada por permitir el desarrollo de aplicaciones como una página web estándar, siendo capaz de correr en varias plataformas incluidas iOS y Android, esto en conjunto con la disponibilidad de varios plugins para el uso de servicios extra y hardware específico dentro del dispositivo, crean un entorno amigable de desarrollo que resulta en aplicaciones estéticamente agradables con alto nivel de personalización (Chaudhary, 2018).

G. Firebase

Firebase es una plataforma propietaria de Google que ofrece herramientas para el desarrollo de aplicaciones web y móviles, su oferta incluye infraestructura de backend completamente administrada, supervisión del rendimiento de aplicaciones, generación de estadísticas de los usuarios y uso de recursos. El uso de esta plataforma reduce los problemas de desempeño, facilita la conexión de la aplicación con una base de datos y un almacén de archivos. Su objetivo de negocios está fundamentado en tres pilares: desarrollo, crecimiento y ganancias; eliminando molestias en el desarrollo y añadiendo herramientas que permiten el escalamiento sencillo de aplicaciones, obteniendo así mejores beneficios (Moroney, 2017).

- **Cloud Functions para Firebase:** De acuerdo con la documentación oficial: “Es un framework sin servidores que permite ejecutar de forma automática el código de backend en respuesta a las solicitudes HTTPS. El código JavaScript o TypeScript se almacena en la nube de Google y se ejecuta en un entorno administrado”
- **Cloud Firestore:** De acuerdo con la documentación oficial: “Usa nuestra base de datos NoSQL flexible, escalable y en la nube a fin de almacenar y sincronizar datos para el desarrollo tanto del lado del cliente como del servidor”. Existe también la posibilidad de la implementación de una base de datos en tiempo real para mantener los datos sincronizados entre apps junto con soporte sin conexión.

- **Firestore Authentication:** De acuerdo con la documentación oficial: “Admite la autenticación mediante: contraseñas, números de teléfono, proveedores populares de identidad federada (Google, Facebook y Twitter), y mucho más”

H. NGINX

Es un servidor WEB/Proxy gratuito, de código abierto y alto rendimiento concebido por Igor Sysoev, su desarrollo inició alrededor de 2002 y fue lanzado al mercado en 2004. Es conocido por su escalabilidad, rendimiento, cantidad de características, configuración simple y bajo consumo de recursos (Soni, 2016). La funcionalidad que será utilizada en este proyecto en específico será como servidor de aplicaciones Web, que funciona de manera sencilla permitiendo desplegar cualquier aplicación compilada, con JavaScript como base, como aplicación Web; esto gracias a la funcionalidad build de Ionic Framework. Esto se basa en una máquina virtual en la nube, la cual esta hospedada en Azure, dando como resultado un entorno de despliegue sencillo y fácil de manejar.

I. Metodología de desarrollo Scrum

Scrum es una de las metodologías ágiles más utilizadas en el desarrollo de proyectos dada su flexibilidad en la selección de nuevas tareas y procesos no especificados; siendo, por lo anterior, ideal si se necesita la obtención de tareas a corto plazo. Esta metodología define al *sprint* como la mínima unidad que, a su vez, se dimensiona como un periodo de tiempo o bloque de entre una y tres semanas en la que al grupo de desarrollo se le asigna una tarea, culminando con la entrega de reportes o incremento del desarrollo. La finalidad de todo lo anterior es la de obtener un producto potencialmente enviable (Srivastada et al., 2017).

- **Planificación del sprint:** El equipo de trabajo se reúne para discutir las tareas, prioridades y definir los objetivos finales en el sprint.
- **Reuniones diarias:** Todos los días el equipo se reúne por 15 minutos o menos para inspeccionar el progreso que se ha realizado para cumplir el objetivo.
- **Revisión del sprint:** Esta revisión se centra en el producto en cuestión siendo desarrollado y el incremento creado durante el sprint.
- **Retrospectiva del sprint:** Se centra en los procesos realizados, se discute que fue bien y que mejoras se puede realizar en el sprint.

En Scrum se definen roles para el proyecto y el equipo de desarrollo, mismos que determinan las actividades a realizar. Estos roles son:

- **Scrum master:** Es el líder encargado de facilitar el desarrollo de los procesos, mediante la planificación de reuniones diarias. En adición se encargará de asegurar que las reglas de Scrum dentro de los Sprints están siendo acatadas y aplicadas de manera correcta. Sus funciones también incluyen motivar al equipo, remover posibles obstáculos y asegurar del bienestar del equipo.
- **Scrum team:** Es un equipo organizado de manera autónoma y conformado por miembros con distintas habilidades, las cuales les permiten solucionar problemas y cumplir con los objetivos propuestos.
- **Product owner:** Representa a los stakeholders o interesados en el desarrollo del proyecto, usualmente clientes, que se encargan de determinar las expectativas de producto y priorizar los objetivos de cada sprint.

En la siguiente sección se enlazarán los conocimientos revisados previamente mediante la implementación de la solución propuesta; la sección se detalla a manera de fases describiendo las actividades realizadas.

V. Marco Metodológico

Por motivos ilustrativos, el proyecto está compuesto de siete fases. La **primera** estuvo dedicada a investigar acerca del desarrollo de aplicaciones multiplataforma con Ionic/Angular, en conjunto con la utilización de Node.js, la implementación de servicios ofrecidos por Google Firebase y los fundamentos teóricos para la visualización de datos mediante la analítica de los mismos e inteligencia de negocios.

A. Requerimientos de la Aplicación

La **segunda** fase se centra en definir los distintos requisitos funcionales y no funcionales, los mismos que conforman una base para el diseño y posterior desarrollo de la aplicación. Adicionalmente, se describen los distintos módulos y características que serán incluidos. Los últimos son esenciales para las etapas finales de la aplicación tales como la realización de pruebas y comprobación del cumplimiento de los requisitos previamente establecidos. Los datos fueron recolectados con el uso de técnicas ya existentes para la gestión de gastos, estas son: hojas de cálculo y aplicaciones similares; en conjunto con información

provista por los dueños de la aplicación (tutor de tesis y el equipo Scrum). Los requerimientos obtenidos fueron los siguientes:

a) Requisitos funcionales

Se definen como las características que se ofrecerán en la aplicación a los usuarios, estas posteriormente se verán reflejadas en los módulos operacionales del software. Se definieron los siguientes requisitos funcionales:

- El usuario dispone de un método de inicio de sesión que consta de los datos: correo electrónico y contraseña.
- La aplicación también permitirá el inicio de sesión con una cuenta de Google.
- La aplicación permitirá crear una familia o unirse a una existente (correo del jefe de hogar).
- La aplicación contará con dos roles con diferentes características: jefe de hogar y miembro de la familia.
- Todos los usuarios podrán crear y modificación un perfil, este constará de un nombre, rol, descripción y fotografía.
- La aplicación permitirá al jefe del hogar ingresar el salario o presupuesto global de la casa, este será único para el periodo de un mes.
- La aplicación permitirá al jefe del hogar especificar y modificar presupuestos para cada una de las categorías, este será único para el periodo de un mes.
- La aplicación permitirá al jefe de hogar configurar el inicio de cada mes (día).
- Todos los usuarios podrán ingresar sus gastos con los datos del monto, fecha, la categoría y una descripción; siendo la descripción la única información opcional.
- El jefe de hogar podrá visualizar los gastos de los miembros de la familia y su persona.
- Los miembros de la familia podrán visualizar los gastos realizados por su persona.
- La aplicación contará con las siguientes categorías de gastos: alimentación, servicios, educación, ocio, transporte, vivienda, salud y otros.
- La aplicación notificará a todos del presupuesto gastado en el día.
- La aplicación notificará a todos si se exceden el presupuesto global.
- La aplicación visualizará una gráfica general que resume los presupuestos y los gastos.

- La aplicación visualizará mediante gráficas para el jefe de hogar estimaciones de presupuesto basadas en los gastos realizados a lo largo del tiempo.
- La aplicación visualizará mediante gráficas para el jefe del hogar reportes personalizados de los gastos para las categorías especificadas (mensual).
- La aplicación visualizará el porcentaje de gastos realizados en cada categoría (mensual).

b) Requisitos no funcionales

Los requisitos que no hacen referencia directa a las funcionalidades o características ofrecidas en la aplicación para los usuarios. Aquí se incluyen propiedades del funcionamiento interno, operabilidad, rendimiento, características visuales, entre otras. Se definieron los siguientes requisitos no funcionales:

- El diseño de la aplicación hará uso adecuado de botones, cajas de texto, funciones, enlaces de navegación, entre otros presentes en la aplicación.
- El tiempo de aprendizaje de la aplicación será corto.
- La aplicación proveerá avisos acerca de errores que se puedan presentar.
- Los campos de cantidad aceptarán valores numéricos enteros y decimales.
- Los campos de texto aceptarán números, letras y símbolos.
- Las categorías de gastos serán mostradas en una lista desplegable.
- Las características de backend serán implementadas con Google Firebase.
- Los inicios de sesión y creaciones de cuentas serán incorporados usando la característica de Authentication.
- Los datos serán almacenados en la base de datos de Firestore.
- Las imágenes serán almacenadas utilizando la característica Firebase Storage.
- Los cálculos para estimación de presupuestos serán realizados en la nube mediante el uso de Cloud Functions.
- Los cálculos para la visualización de reportes serán realizados en la nube mediante el uso de Cloud Functions.
- Los reportes harán uso de distintos tipos de gráficos como área polar, estrellas, líneas y dispersión implementando técnicas de análisis de datos e inteligencia de negocios.
- El sistema deberá de ser capaz de soportar operaciones concurrentes.

- El sistema deberá ser capaz de soportar usuarios concurrentes.
- El sistema deberá personalizar la información que se mostrará para los dos roles de usuario.
- El sistema deberá poseer una documentación entendible para la reutilización en el futuro.
- La aplicación contará con un manual de usuario.
- Solo existirá una persona con el rol de jefe de hogar en cada familia.
- El rol de miembro de familia podrá ser utilizada por varios usuarios.
- El correo electrónico será único para cada usuario.
- Cada usuario solo podrá modificar su correspondiente perfil.

B. Diseño y Codificación

En la **tercera** fase se toman los requerimientos y módulos, expuestos en la sección anterior, como fundamento para el modelado del sistema. Lo primero será representar los módulos, para esto usaremos un diagrama de componentes. En la *Figura 1*, las funcionalidades de la aplicación, en el centro de la figura, se encuentran distribuidos en dos grupos de interfaces de usuario, esto en función de los requerimientos funcionales: jefe de hogar y miembros de familia. La infraestructura, representada a la derecha de la figura, pertenece al conjunto de características de *backend* que provee la plataforma de Firebase. Para las relaciones entre componentes:

- Los círculos con un arco de circunferencia adicional representan interfaces claramente definidas que proporcionan servicios.
- Los círculos representan interfaces que necesitan esos servicios o datos provenientes de los mismos.

También se muestra la dependencia del resto de módulos con la autenticación denotando que ninguna de esas acciones puede realizarse si no se inicia sesión.

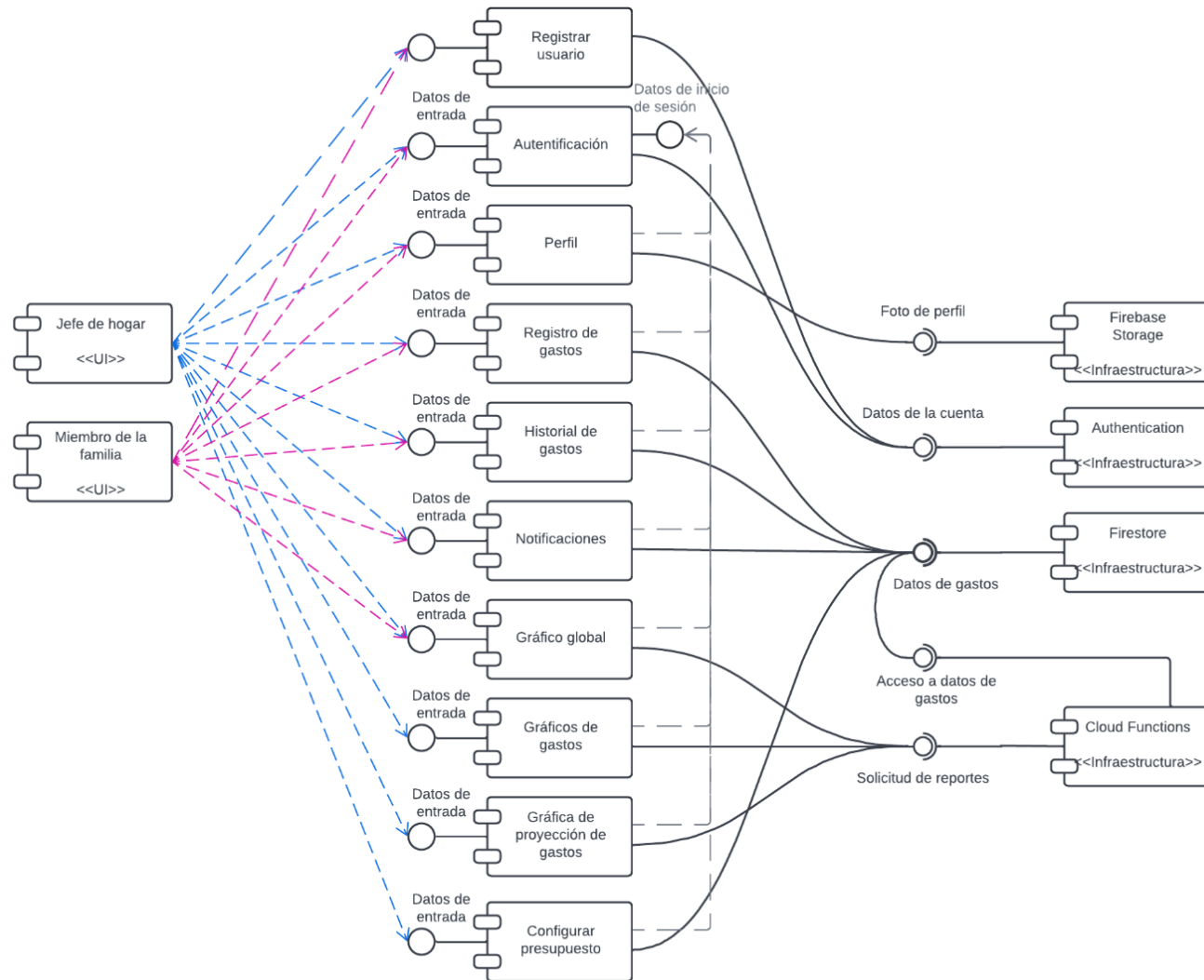


Figura 1. Diagrama de componentes de la aplicación

Para una comprensión más detallada de los datos necesarios para los componentes, utilizamos un diccionario de datos (*Tabla 1*), ya que Firestore guarda registros de manera no estructurada (No-SQL). Los datos se guardan en conjuntos denominados colecciones, cada entrada es llamada documento y contienen un número de atributos que pueden o no ser similares entre sus homólogos.

Tabla 1. Disposición de datos en Firestore Database

Colección	Atributos	Tipo de dato	Descripción
Usuario (users)	uid	Cadena de texto	Hash como ID del documento
	email	Cadena de texto	Correo electrónico de la cuenta
	password	Cadena de texto	Contraseña de la cuenta
	role	Char	“U” para miembro de la familia “A” para jefe de hogar
	displayName	Cadena de texto	Nombre de propietario de la cuenta
	description	Cadena de texto	Mensaje de bienvenida de la cuenta
	photoURL	Cadena de texto	URL de referencia a Firebase Storage
	createdAt	Fecha	Fecha en la que fue creada la cuenta
	lastLogin	Fecha	Fecha de último ingreso a la cuenta
	provider	Cadena de texto	Tipo de inicio de sesión “gestion-gastos” para ingreso con correo y contraseña “google” inicio de sesión con Google
	id_familia	Numérico	Familia asignada a la cuenta
	active	Booleano	Cuenta activa o no
Familia (families)	id	Cadena de texto	Hash como ID del documento
	nombre	Cadena de texto	Nombre de la familia
	presupuesto_global	Numérico	Presupuesto global de la familia
	primer_dia_mes	Numérico	Día de inicio de mes
Presupuesto (presupuestos)	id	Cadena de texto	Hash como ID del documento
	cantidad	Numérico	Monto del presupuesto
	id_categoria	Cadena de texto	Categoría de gastos
	id_familia	Cadena de texto	Familia asignada al presupuesto
	fecha	Fecha	Fecha de modificación del presupuesto
Gasto (gastos)	id	Cadena de texto	Hash como ID del documento
	monto	Numérico	Monto del gasto
	descripcion	Cadena de texto	Razón del gasto
	fecha	Fecha	Fecha de realización del gasto
	id_categoria	Cadena de texto	Categoría de gastos
	id_usuario	Cadena de texto	ID de usuario del gasto
Id_familia	Cadena de texto	Familia asignada al gasto	
Categoría	id	Cadena de texto	Hash como ID del documento

(categories)	nombre	Cadena de texto	Nombre de la categoría
	descripción	Cadena de texto	Descripción de la categoría

Para dar término a esta fase, se realizaron algunos prototipos visuales de las interfaces de usuario (Figura. 2), destinados como base para el desarrollo de las funcionalidades y su presentación al usuario. La disposición de componentes dentro de cada interfaz está sujeta a variaciones, pero ofrecen un contexto general para una visualización mental de la aplicación. La totalidad de estos prototipos se encuentran en el Anexo 1.

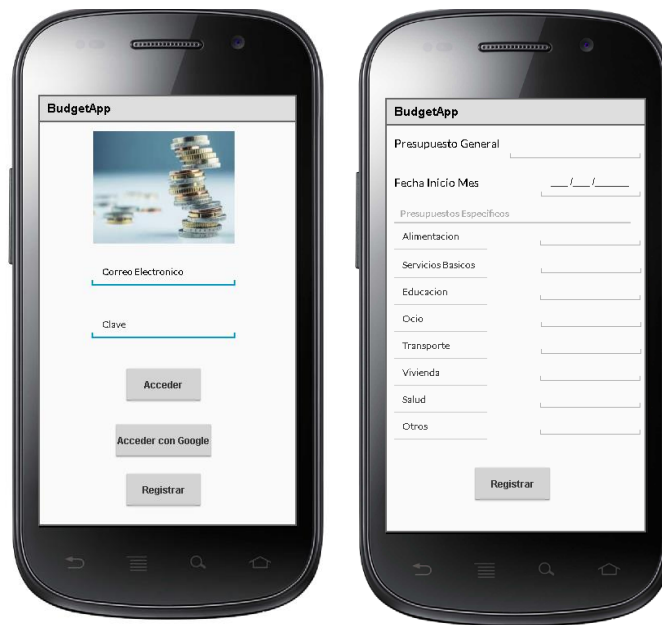


Figura 2. Ejemplo de prototipos de interfaz

La **cuarta** fase se conforma del desarrollo de la aplicación y sus funciones esenciales. El *frontend* o capa de presentación fue desarrollada con Ionic/Angular como framework por medio de Node.js. El *backend* o capa de servicios fue implementado en conjunto con el desarrollo de la aplicación, esto conllevó a no requerir una fase específica para esta parte del proyecto, esto debido al uso de varias herramientas dentro de Firebase. Una de estas herramientas habilita la opción de subir archivos. Los servicios de autenticación y manejo de usuarios (inicio de sesión y creación de cuenta) se realizaron haciendo uso de la característica Authentication, misma que nos permite ingresar usuarios a la aplicación por medios como el correo electrónico y contraseña, o iniciar sesión con una cuenta de Google. Por último, para la inclusión de lógica adicional

necesaria para la analítica de datos se utilizó Cloud Functions. Podemos ver la arquitectura simplificada de la aplicación en la *Figura 3*.

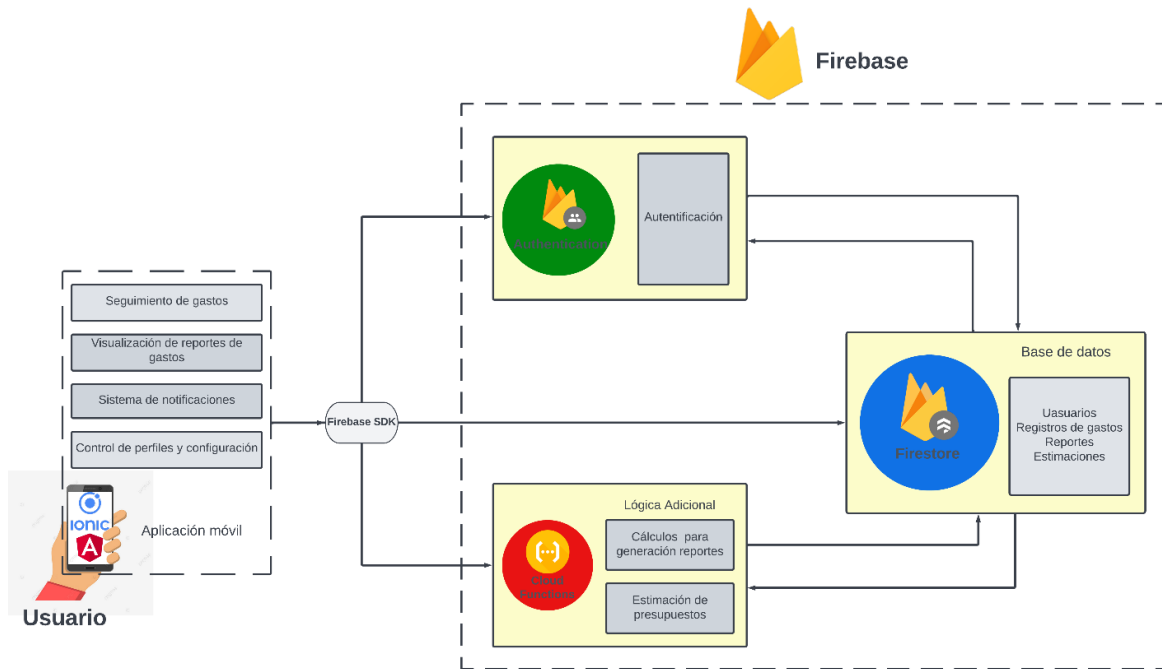


Figura 3. Esquema de la aplicación

El uso de Firebase solo requiere tener una cuenta de Google, el cual incluye gratuitamente el uso de la mayoría de las características usadas en este proyecto, pero cabe recalcar que para el uso de funciones en la nube se requiere un plan prepago que involucra el ingreso de un método de pago.

Otras características incluidas como esenciales fueron: la modificación del perfil de usuario, creación de una familia o unirse a una existente, modificar familia y la navegación básica con un menú lateral.

La **quinta** fase fue el desarrollo del conjunto de módulos destinados al seguimiento y reportes de los gastos personales. En el caso del jefe de familia, podrá definir y modificar el presupuesto tanto global como los específicos, recibirá notificaciones acerca los gastos del día y si se excede del presupuesto global; mientras que en el caso de los miembros de la familia solamente tienen ingreso de gastos, reportes personales y posibilidad de revisar los

presupuestos. Esta fase siguió la misma estructura anterior en cuanto a los recursos y arquitectura que se planteó en la *Figura 3*; las características específicas disponibles para el jefe de hogar y los miembros de la familia se detallan en la *Figura 4*.

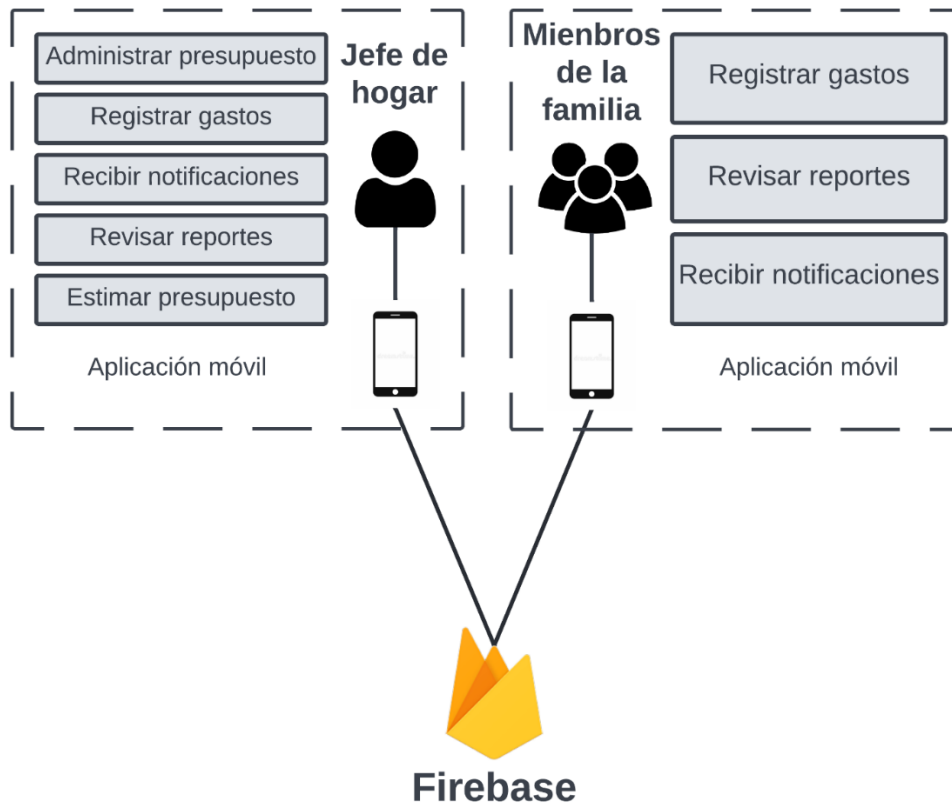


Figura 4. Esquema de funcionalidades

El *backend* se puede administrar directamente en el Dashboard de la plataforma en cuestión, es posible realizar seguimientos de los datos ingresados en la *base de datos*, estos hacen referencia a los usuarios registrados e incluso la inclusión de lógica adicional para la aplicación desplegada en Firebase; esto se explicará en breve.

Las funcionalidades principales fueron integradas directamente en la aplicación utilizando las herramientas de Ionic Framework, Angular y Node.js que en conjunto proveen de una interfaz limpia, con interacciones simples y fáciles de usar.

A continuación, se describirán cada uno de los módulos y submódulos presentes en la aplicación descrita en las *Figura 3* y *Figura 4*:

- **Base de datos:** Es utilizable inmediatamente al configurar Firebase dentro del código de la aplicación, mediante el registro de un conjunto de variables; la base de datos se almacena en el módulo *Firestore Database*.
- **Autenticación:** Permite el ingreso de los usuarios al sistema y la gestión de los roles. La gestión se da al definir al jefe de hogar y los distintos miembros de la familia
- **Seguimiento de gastos:** Permite el *registro de los gastos* realizados en las distintas categorías para todos los usuarios, mientras que la *administración de los presupuestos* es exclusiva del jefe de hogar.
- **Sistema de notificaciones:** Permite a los usuarios *recibir notificaciones* personalizadas de acuerdo con los gastos realizados, presupuestos, estimaciones; estas notificaciones diferirán dependiendo del rol del usuario.
- **Visualización de reportes de gastos y presupuesto:** Permite a los usuarios *revisar reportes*. Estos reportes incluyen: la visualización de un historial de gastos, gráficos de los distintos presupuestos en relación con los gastos y la visualización de una estimación de gastos en relación con el tiempo. Se utilizarán gráficos de área polar para comparar los presupuestos registrados con los reales y los gastos totales con el presupuesto global. Para la estimación se usará un gráfico de líneas/dispersión para mejor visualización de los gastos diarios. Por último, se visualizarán los gastos en contraste al presupuesto específico de su categoría mediante un gráfico de estrella.
- **Control de perfiles y configuración:** La aplicación permite modificar datos del perfil, junto con la revisión y eliminación de miembros. La eliminación está disponible solo para el jefe de hogar.
- **Administrar presupuestos:** Permite ingresar y modificar el presupuesto global y específicos por categoría de gastos.
- **Estimar presupuesto:** Proyección de los gastos utilizando la información a lo largo del tiempo, con el objetivo de dar una estimación de gastos.

Es importante resaltar que el desarrollo de estos módulos dedicará tiempo al aspecto visual de las interfaces y no solo sus funcionalidades. Se utilizaron colores oscuros que contrastan con colores vivos en las gráficas. Se usaron ayudas visuales para los presupuestos, entre otros detalles que hace la experiencia de usuario mucho más satisfactoria.

Para los módulos de **estimación de presupuestos** y **cálculos para la generación de reportes** se hará uso del *framework* sin servidores incluido en la plataforma de Firebase denominado Cloud Functions. El código escrito en TypeScript utilizando el entorno de Node.js es previamente compilado en JavaScript y desplegado en los servidores de Google, de modo que una función incluida en la aplicación móvil de gestión de gastos es ejecutada y desplegada en los servidores con el uso de simples líneas de comando respondiendo a los eventos de nuestra aplicación como si estuviera corriendo nativamente en el dispositivo *Figura 5*.

Nota: Cualquier lógica adicional que se pueda agregar a la aplicación en el futuro podrá ser desplegada y ejecutada remotamente utilizando Cloud Functions.

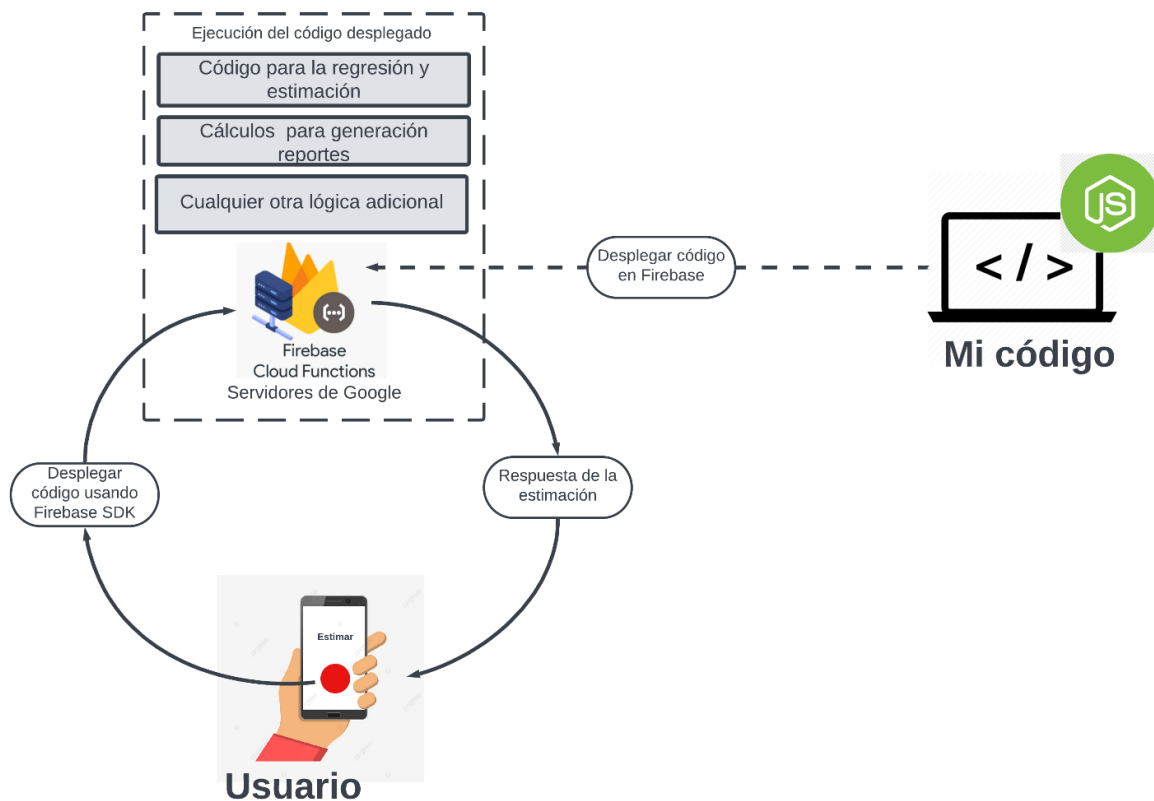


Figura 5. Despliegue de código para estimación de gastos

Se desarrollaron tres funciones que fueron desplegadas en los servicios en la nube de Google y que son llamados a modo de una API desde los módulos de reportes de nuestra aplicación utilizando una URL, estas llamadas toman datos de gastos en la Firestore

dependiendo de la petición, realizan cálculos y envían los resultados en un formato JSON. Las funciones creadas y desplegadas son:

- **Predicción de gastos:** Consta de una regresión lineal utilizando los datos recolectados de los gastos realizados a lo largo del tiempo. Esto da paso a una proyección global a modo de estimación, la cual será visualizada en una gráfica. Estas estimaciones tomarán los datos de los últimos tres meses exclusivamente.
- **Gráfica de inicio:** Sumatoria de los gastos para contrastarlos con el presupuesto global, esto será visualizado en la pantalla de inicio. Los datos usados de esta gráfica son tomados por mes, dependiendo del día de inicio del mes seleccionado por el jefe de hogar.
- **Gráficas por categoría:** Sumatoria de todas las categorías para la visualización de reportes gráficos. Los gráficos describirán la comparación de presupuestos contra gastos. Los datos usados de esta gráfica son tomados por mes dependiendo del día de inicio del mes seleccionado por el jefe de hogar.

Como actividad adicional en esta fase se compiló la aplicación resultante, en archivos de JavaScript, con el objetivo de ser desplegada en un servidor NGINX; este servidor fue desplegado en una máquina virtual en la nube provista por los servicios web de Microsoft Azure. Uno de los objetivos del desarrollo de este proyecto es su uso multiplataforma que puede ser conseguido al usar el navegador de cualquier smartphone sin importar su sistema operativo.

C. Pruebas y Manuales

Para la **sexta** fase el enfoque fue el de diseñar y ejecutar pruebas funcionales (unitarias e integración) de la aplicación de gestión de gastos, esto en base a los requerimientos que se definieron en la segunda fase. Como medio se utilizarán métricas para evaluar la calidad del software. La evaluación está basada en una plantilla.

La plantilla consta de: información del encargado de la prueba, código, tipo de prueba realizada, propósito de la prueba, método de aplicación, datos de entrada y argumentos que justifiquen que la característica se comporta correctamente.

Tabla 2. Diseño de pruebas funcionales

Pruebas Unitarias	Ingreso
	Crear cuenta
	Crear familia
	Inicio
	Perfil
	Miembros
	Registrar presupuesto/ Ver presupuestos
	Registrar gasto
	Ver gastos/ Ver gastos Familia / Ver Gastos Personales
	Reporte de Gastos/Presupuestos
	Predicción de gastos
	Pruebas de Integración
Aplicación Gestión de Gastos (jefe de hogar)	
Prueba de rendimiento	Tiempo de navegación en la aplicación Web
	Tiempo de navegación en la aplicación Android
	Tiempo de carga de datos del formulario
	Tiempo de renderización de gráficas
	Tiempo de respuesta de las funciones des plegadas en la nube
Pruebas de Aceptación	Introducción de la aplicación al público

En la *Tabla 2* se puede apreciar que se optó por la realización de pruebas unitarias de cada uno de los módulos (toda aquella característica que cumpla el mismo objetivo) al ser mucho más sencillo de implementar y nos permite revisar cada una de las características de la aplicación de manera aislada. De igual manera se realizaron pruebas de integración que buscan probar el comportamiento global de la aplicación en conjunto con la navegación, estas nos darán una idea de la experiencia de los dos roles de usuario con la aplicación y sus funcionalidades. También se incluyen pruebas de rendimiento global de la aplicación para determinar futuras optimizaciones, estas engloban: tiempo en la navegación en la aplicación Web y Android, carga de formularios, renderización de gráficas y tiempo de respuesta de las funciones en la nube.

Por último, se tienen las pruebas de aceptación que resultarán en el último lote de resultados, estas muestran la experiencia de los usuarios con proyecto. La ejecución de las pruebas se realizará exponiendo al usuario a la aplicación final con suficiente conocimiento previo (contexto), después podrá interactuar y navegar libremente descubriendo las distintas características. Al final se distribuirá un cuestionario que tendrá una rúbrica de calificación con criterios de usabilidad y experiencia de usuario, al igual que un cuadro de texto para comentarios adicionales.

La **séptima** fase fue empleada para la elaboración de un manual técnico de la aplicación, misma que detalla aspectos teóricos necesarios para el entendimiento de la aplicación como: modelados del sistema, las herramientas usadas para el desarrollo y sus configuraciones, librerías usadas, la estructura del proyecto y los requisitos del sistema; esta documentación será útil para futuro mantenimiento y mejora del proyecto. Del mismo modo se elaborará un manual breve de usuario con la descripción de las características funcionales de la aplicación dependiendo del rol de usuario y las funcionalidades que ofrece la herramienta. Estos se encuentran detallados en los anexos 7 y 8.

Todas las actividades realizadas en las fases descritas anteriormente culminan en la obtención de resultados que serán discutidos en la siguiente sección. Los productos obtenidos en el desarrollo determinan el cumplimiento de los objetivos planteados y si la aplicación propuesta como solución a la problemática tiene valor más allá de los aspectos técnicos que la componen.

VI. Resultados

Para los resultados presentamos un resumen de las pruebas realizadas sobre la aplicación, detallaremos lo que se ha logrado en cada etapa del desarrollo del proyecto, los módulos resultantes (funcionalidades), la experiencia de los módulos en conjunto y las pruebas de rendimiento respectivas.

Se detallará si los objetivos específicos planteados fueron cumplidos, mismos que finalmente darán paso al posible cumplimiento del objetivo general. También se incluyen pruebas de aceptación que brindarán información importante acerca del estado actual de la aplicación, la necesidad de un proyecto así en el mercado y el posible futuro de la aplicación. El proyecto fue compilado en una APK y también fue desplegado en un servidor web en la nube previo a las pruebas.

El estudio de los fundamentos teóricos presentes en el documento, cuyo objetivo fue ser base para la propuesta de solución, demuestra el cumplimiento del objetivo planteado *OE1*.

Una etapa crucial para el desarrollo de la segunda fase del marco metodológico fue la identificación de los requerimientos funcionales y no funcionales, demostrando así el cumplimiento del objetivo específico planteado *OE2*.

A. Resultados por Funcionalidad

En esta subsección se detallarán los productos obtenidos en la cuarta y quinta fase detalladas en la sección anterior, específicamente mediante las pruebas unitarias realizadas (sexta fase). En la *Tabla 3* podemos ver un resumen rápido de las pruebas unitarias (Anexo 2) por funcionalidad destacando información importante y su grado de cumplimiento.

Tabla 3. Resumen de las pruebas unitarias

Funcionalidad	Información a Destacar	Grado de Cumplimiento
Ingreso	Algunos errores con la conexión a Firebase pueden pasar en casos específicos.	PLANEADO
Crear cuenta	Ningún método de confirmación de correo ha sido implementado.	ACEPTABLE
Crear familia	Solo es posible unirse a una familia existente con el email del jefe de hogar.	ACEPTABLE
Inicio	Las notificaciones se muestran muy seguido.	ACEPTABLE
Perfil	Las acciones requieren la interacción con botones y no se sincronizan automáticamente.	ACEPTABLE
Miembros	No es posible asignar un nuevo jefe de hogar.	PLANEADO
Registrar presupuesto/ Ver presupuestos	El funcionamiento es planeado. El registro se hace de manera organizada y bajo una determinada estructura	PLANEADO
Registrar gasto	Tener en cuenta el ingreso de toda la información requerida	PLANEADO
Ver gastos/ Ver gastos Familia / Ver Gastos Personales	Posibles registros incongruentes debido a la constante consulta a Firebase y la naturaleza de la respuesta recibida	ACEPTABLE
Reporte de Gastos/Presupuestos	Se presentan diferentes grafos dirigidos a cada una de las categorías de gasto, una gráfica comparativa y una de aspecto general.	PLANEADO
Predicción de gastos	La escala se da en meses, pero si se quieren datos específicos se puede filtrar.	PLANEADO

El comportamiento independiente y aislado de los módulos funciona de manera esperada sin mayores contratiempos en la mayoría de los casos. Algunos problemas encontrados envuelven casos en los cuales no se tuvo una conexión consistente al internet

y no poderse comunicar con la API de Firebase a tiempo; esto resulta en un comportamiento que se esperaba en un contexto con conexión deficiente a internet. Por otro lado, Firebase tiende a responder con resultados replicados o incompletos de diferentes consultas realizadas a la misma colección, esto tiene repercusión en los historiales de gastos en ocasiones. Existe la posibilidad de bugs que fueron detectados al momento de probar los módulos; la totalidad de los mismos que se encontraron fueron resueltos de manera exitosa.

En cuanto a los aspectos positivos de las funcionalidades, es importante destacar como estos trabajan con la finalidad de cumplir parcialmente nuestro objetivo planteado *OE3*, sin dejar de lado que es necesario un correcto trabajo conjunto. Detalles adicionales al comportamiento esperado de las funcionalidades añaden un grado más de calidad al producto final, algunos de estos detalles que quisiéramos destacar son:

- Existen sugerencias en algunos módulos como *predicción de gastos* que dan a conocer interacciones al usuario que pudo no haberlas notado.
- En el módulo de registrar presupuesto se puede visualizar el presupuesto asignado total entre todas las categorías; se pondrá en rojo si se sobrepasa.
- Optando por el tema oscuro de Ionic Framework el contraste de los colores utilizados en las gráficas del módulo de *reporte de gastos/presupuestos* es mayor, haciendo que llamen la atención del usuario.
- Los historiales en los módulos *ver gastos* contiene filtros que facilitan la visualización.
- Al subir una imagen en el módulo *perfil* se muestra el progreso de la subida del archivo a los servidores.

Una parte importante del sistema de reportes son las funciones en la nube de Cloud Functions, la cuales están detalladas en la quinta fase del marco metodológico. Todas las respuestas a las peticiones realizadas dentro de la aplicación fueron exitosas, pero el tiempo que requirieron es medianamente largo; esto a su vez es un comportamiento esperado y estas tendrán una respuesta más rápida dependiendo de la cantidad de peticiones realizadas.

Se dan otros casos con estas funciones como, por ejemplo; no tener suficientes datos para realizar la *predicción de gastos*, esta dará un error que está controlado dentro de la aplicación.

En el caso de las otras dos funciones utilizadas en las visualizaciones de *gráfica de inicio* y *gráficas por categoría*, se pueden observar que, si no se presentan registros de gastos dentro de la familia uno de los valores es cero. Lo que resulta en un gráfico donde se comparan dos cantidades iguales.

B. Resultados de las Pruebas de Integración

Esta subsección está destinada al segundo grupo de pruebas denominadas de integración (Anexo 3), mismas que nos da información de la experiencia combinada de las funcionalidades, sus interacciones y navegación. Los resultados están repartidos en la experiencia del jefe de hogar y la de los miembros. En la *Figura 6* se puede apreciar las opciones de navegación disponibles para las funcionalidades accesibles para cada rol. Esta sección concluye y confirma el cumplimiento del objetivo específico *OE3*.

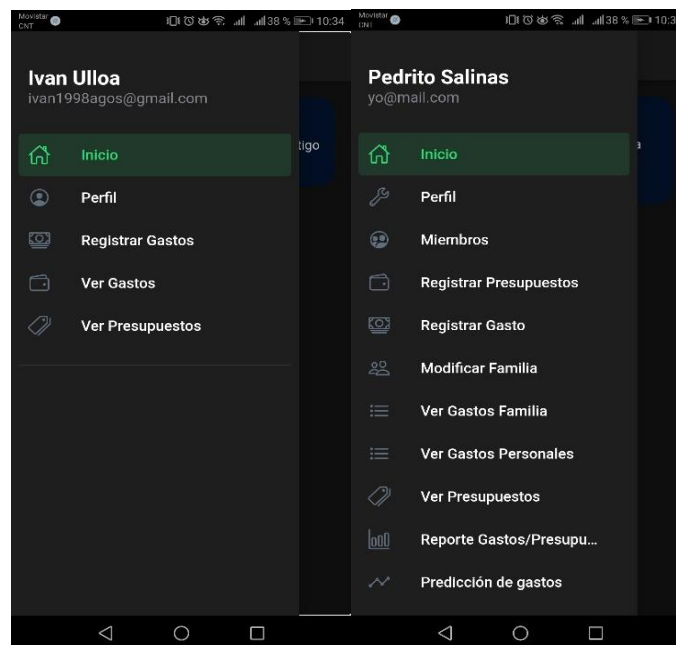


Figura 6. De izquierda a derecha, opciones de navegación para los miembros de la familia y para el jefe de hogar

a) Integración de funcionalidades para miembros de la familia

En el caso del rol de miembro de familia, las funcionalidades son limitadas y solo permiten al usuario: visualizar el resumen global en la pantalla de inicio, revisar y modificar su perfil, registrar sus gastos, ver el historial de sus gastos y por último revisar los presupuestos asignados. El funcionamiento y la interacción entre las distintas características es el esperado. En términos de usabilidad, tenemos que la información brindada en las pantallas es fácil de leer, los colores generan un contraste agradable y dan a la aplicación una impresión de calidad. En la *Figura 7* podemos observar un ejemplo del uso de colores y fuentes.



Figura 7. Pantalla de inicio de la aplicación GestionarGastos

Es importante tener en cuenta que la aplicación utilizada para el proceso de *debug* en dispositivos móviles no mantiene la sesión abierta si se vuelve después de cerrarla, esto puede ser ofrecido en la aplicación como trabajo futuro.

b) Integración de funcionalidades para el jefe de hogar

Para el jefe de hogar se abre una variedad más grande de características que funcionan en conjunto, abriendo paso a más errores potenciales y no detectados.

Muchos de estos errores fueron detectados a tiempo mediante *debugging* en el desarrollo.

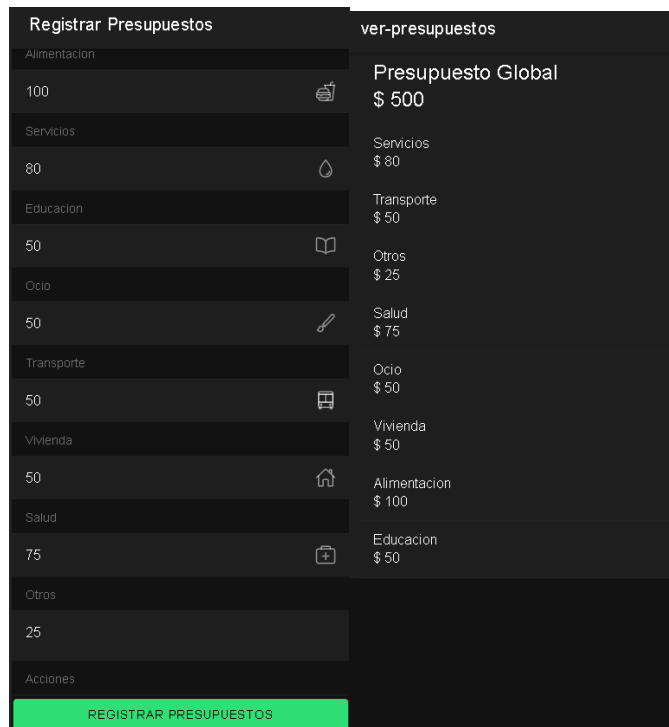


Figura 8. De izquierda a derecha, pantalla de registro de presupuestos y pantalla de listado de presupuestos

La *Figura 8* muestra la función de *registro de presupuesto* y expone los datos que son necesarios para esta, en cuanto al presupuesto general y presupuestos específicos de cada categoría de gastos. Los resultados se obtienen mediante consulta directa a Firebase. La pantalla a la derecha de la figura simplemente visualiza los presupuestos, estando disponible tanto para el jefe de familia como para los demás miembros.



Figura 9. Gráfica comparativa entre categorías

En la *Figura 9* observamos la sección de reportes gráficos, en este caso en particular, una gráfica que compara los gastos generados por los miembros de la familia para cada una de las categorías. Presenta un eje que refleja el porcentaje del presupuesto ya gastado para cada categoría. Esto como referente para saber si podría ser posible un recorte o aumento en uno de los presupuestos.

La funcionalidad permite seleccionar reportes, por cada categoría, por separado mediante una lista desplegable; estos datos se visualizan como gráficas de área polar (similar a la pantalla de inicio). Todo resulta en gráficas de fácil de lectura, donde el presupuesto restante siempre es rápidamente identificable.



Figura 10. Pantalla de Proyección de gastos

Por último, en la *Figura 10* se muestra la proyección de gastos que se realiza mediante uso de los registros de una familia. Los gastos se representan mediante puntos. Se puede interactuar con los gastos si se encuentra uno de interés en el diagrama. Notablemente es la función que más tarda en responder, esto es debido a que el proceso tras la función en la nube que realiza los cálculos requiere este mayor tiempo para su cálculo.

C. Pruebas de Rendimiento

Para las pruebas de rendimiento se tuvo en consideración situaciones que requieren un tiempo de espera para la renderización, obtención o carga de datos. Tener tiempos de espera relativamente cortos ayudan a generar una mejor experiencia para el usuario y calmar la naturaleza impaciente del ser humano. La obtención de estos datos se hizo mediante el registro del tiempo actual del sistema, tanto al inicio como al final del proceso; esto para determinar finalmente el tiempo total que toma el procesamiento. Se tomó en

cuenta esto para cada módulo probado. Los tiempos observados pueden verse a más detalle en el *Anexo 4* de pruebas adjuntos.

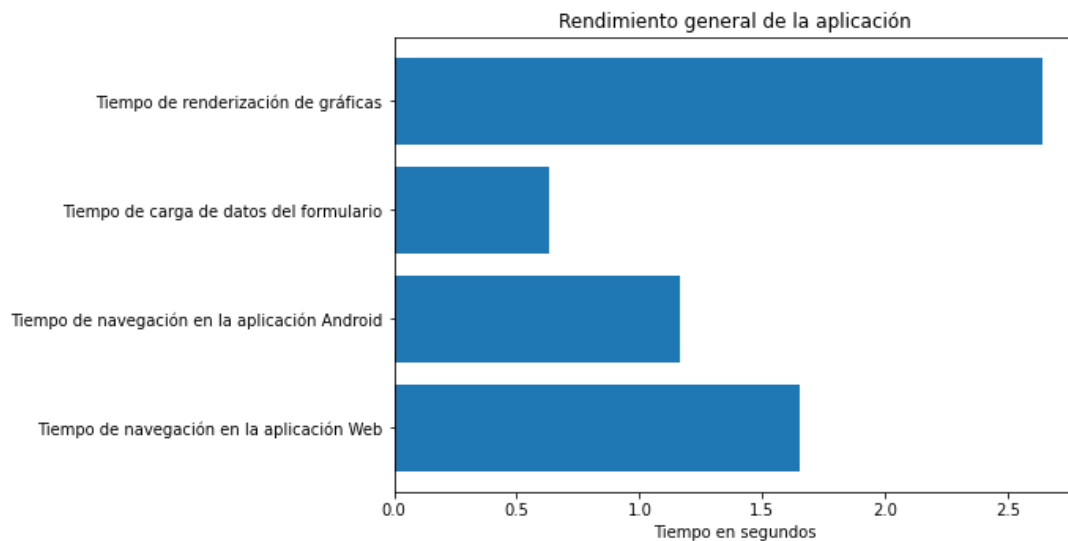


Figura 11. Gráfica de rendimiento promedio general de la aplicación

En la *Figura 11* se observa los tiempos promedio de respuesta para cada una de las interacciones que conforman la experiencia general de la aplicación. Para las muestras solo se tomaron los módulos que contienen los procesos antes expuestos; también dependiendo del número de módulos varía la cantidad de datos tomados.

Se tiene que el proceso más demandante con un tiempo promedio de más de 2.5 segundos es la renderización de gráficas. Este proceso tiene que realizar una petición a la función en la nube, esperar la respuesta y renderizar el gráfico; tiene un paso extra en comparación con la obtención de datos directamente de la Firestore.

También se tiene que el tiempo promedio más corto es el registro de la información del formulario en la base de datos, esto se debe a que las respuestas de Firestore son sumamente rápidas siempre y cuando se tenga una conexión estable. En tanto a la navegación, la aplicación Android tiende a ser más rápido que la aplicación Web por el uso directo del hardware para la renderización de los componentes.

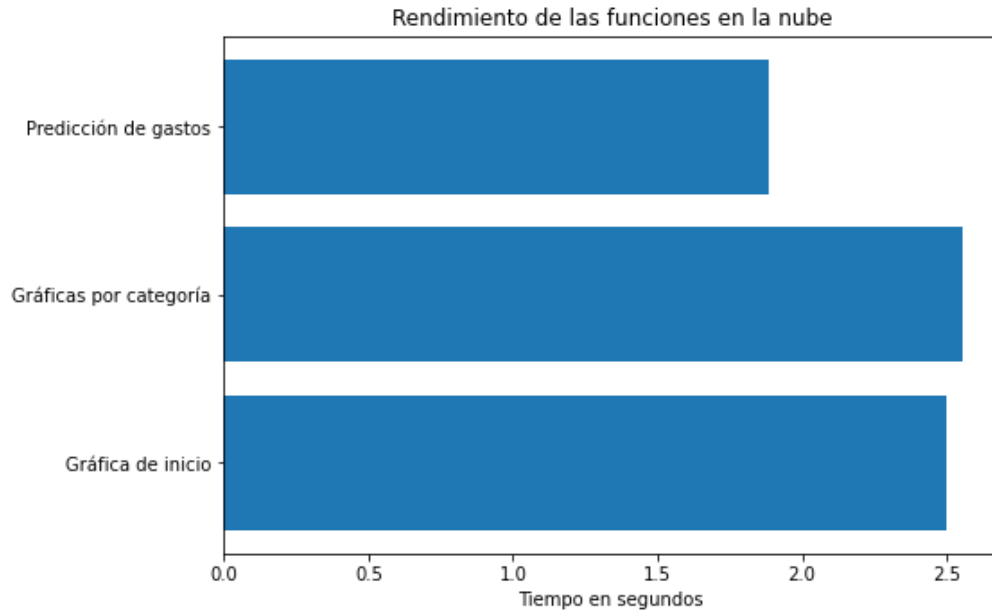


Figura 12. Gráfica de rendimiento promedio de las funciones en la nube

Las funciones desplegadas en los servidores de Google son vitales para los reportes gráficos de la aplicación y sus tiempos de respuesta brindarán una mejor experiencia en los módulos que realizan estas peticiones. En la *Figura 12* podemos observar los tiempos de respuesta promedio de cada una de las tres funciones desarrolladas. La diferencia de estos tiempos no está dada por ninguna situación en particular; los tiempos registrados están directamente relacionados con la conexión a internet. La poca discrepancia de tiempo también prueba que el comportamiento de las funciones es el ideal sin importar las constantes peticiones. Otro aspecto importante en las respuestas es la cantidad de peticiones realizadas a estas funciones, Cloud Functions dinámicamente asigna recursos para que estas puedan responder a la demanda de manera más rápida; es decir, mientras más peticiones existan, las funciones responden más rápido, esto bajo el contexto de una conexión estable al internet.

D. Pruebas de aceptación

Las pruebas de aceptación fueron realizadas tal como se detalló en la sexta fase del marco metodológico sin mayores cambios. Se tuvo una muestra limitada de personas para las pruebas (12 en total), pero se buscó aprovechar al máximo el mecanismo de obtención de resultados, la encuesta realizada. Para ello, se recalcó la importancia de sus respuestas

veraces y se alentó a dejar comentarios adicionales siendo lo más detallado posible. La aplicación estuvo disponible para todos los participantes en el lapso de un día por medio de un hipervínculo que direccionaba a la aplicación desplegada en una máquina virtual en la nube, optando por la conveniencia del uso de la aplicación directamente en el navegador en un ambiente de pruebas. En adición la aplicación para Android fue distribuida para los participantes que podría hacer uso de esta plataforma. El cuestionario aplicado (Anexo 5) fue realizado en Google Forms y distribuido directamente por medios digitales.

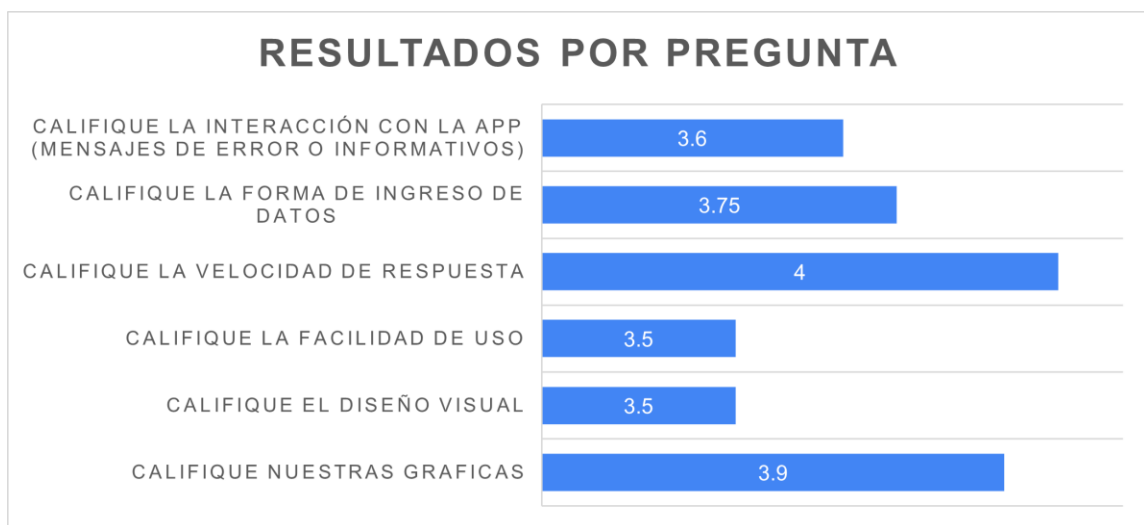


Figura 11. Calificación promedio por pregunta

Existieron seis criterios de calificación dentro de nuestro cuestionario: mensajes de error e informativos, ingreso de datos (cajas de textos, listas y otros métodos), velocidad de respuesta de la navegación entre módulos, facilidad de uso (intuitiva y con una curva de aprendizaje no pronunciada), aspecto visual (colores, íconos, disposición de ítems) y por último las gráficas de reportes usadas. En las calificaciones obtenidas (*Figura 11*) destacan los puntos más débiles de la aplicación siendo uno de estos el aspecto visual, donde un comentario específico resalta el uso del tema oscuro como un defecto del aspecto visual. Con una calificación idéntica tenemos la facilidad de uso, relacionado a la situación que un par de personas comentaron describiendo la inexistencia de navegación al no poder encontrar el menú de lateral por la falta de una señal visual. Los mensajes de error e informativos no fueron suficientes para algunos de los voluntarios encontrando algunas de

las leyendas nada específicas. Otros aspectos de la aplicación parecen haber llenado las expectativas de los usuarios.

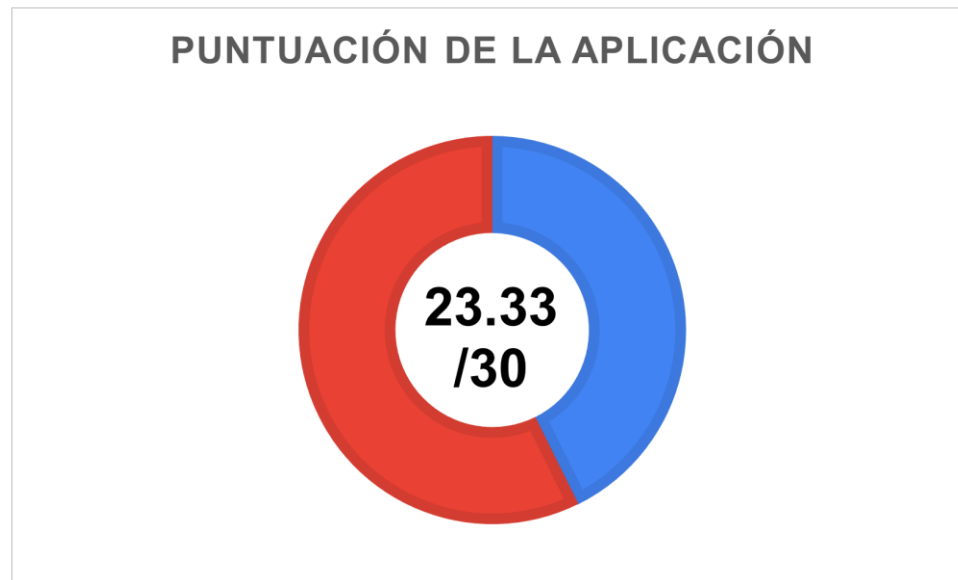


Figura 12. Calificación global de la aplicación

Al final de la adquisición de estos resultados obtuvimos una calificación promedio del sistema sobre 30 puntos como se puede apreciar en la *Figura 12*. Los resultados finales nos presentaron información un tanto desconcertante en relación al estado de la experiencia de usuario de la aplicación final. Muchos de los desafíos al desarrollar los módulos fueron el crear una interfaz y experiencia óptima para nuestros usuarios hipotéticos, pero los resultados demuestran que nuestras intenciones no fueron suficientes en determinadas situaciones. En todo caso, la información obtenida tiene un gran valor dentro del contexto de la aplicación y como una experiencia de interacción con los usuarios finales. Estos datos pueden ser usados para pulir aspectos sugeridos y solucionar errores que no fueron detectados en el desarrollo. Con esto concluimos la presentación de resultados pertenecientes a nuestras pruebas detallando el cumplimiento del objetivo específico planteado *OE4*.

E. Manuales

Como parte de las actividades se elaboró un manual de usuario (Anexo 7) que consta con dos secciones detallando las funcionalidades disponibles para un miembro de

familia y para el jefe de hogar. Se hace uso de texto simple con oraciones cortas evitando información irrelevante para su operación. De igual manera se elaboró un manual técnico (Anexo 8) de la aplicación que se divide en cuatro secciones: aspectos teóricos usados en el desarrollo, diagramas de modelado, aspectos técnicos del desarrollo y requerimientos del sistema. Este último manual incluye información acerca de la instalación y configuración de las herramientas usadas para el desarrollo, estructura de la aplicación y su compilación. Con la realización de estos documentos damos por cumplido el objetivo específico planteado *OE5*.

VII. Cronograma

El presente proyecto se rigió bajo la metodología SCRUM, teniendo cuatro *sprints* sobre los que se dividieron las diferentes actividades. Las tareas se asignaron en concordancia a los requerimientos especificados como resultado de la segunda fase. Al término de las actividades de un sprint se realizó una revisión, esto para determinar si se procedía a desarrollar las tareas del siguiente sprint. Los roles en el proceso fueron definidos de la siguiente manera:

Product Owner: Iván Ulloa (IU), Martin Bojorque (MB)

SCRUM Master: Iván Marcelo Ulloa Bermeo (IU)

SCRUM Team:

- **Desarrollador N°1:** Ivan Marcelo Ulloa Bermeo (IU)
- **Desarrollador N°2:** Jorge Martin Bojorque Macancela (MB)

Nombre de la tarea		Responsable	Duración	Comienzo	Fin
	Proyecto		630	12-04-22	08-07-22
	Sprint 1		6		
OE1			6		
	ACT.1 Recolectar información del dueño del proyecto	MB-IU	1	12-04-22	12-04-22
	ACT.2 Definir los requisitos funcionales de la aplicación para gestión de gastos	MB-IU	2	12-04-22	12-04-22
	ACT.3 Definir los requisitos no funcionales de la aplicación para gestión de gastos	MB-IU	2	13-04-22	13-04-22
	Revisión/Corrección	MB-IU	1	13-04-22	13-04-22

Sprint 2			295		
OE.1			30		
	ACT.1 Estudio de los fundamentos de servicios web	MB-IU	15	14-04-22	18-04-22
	ACT.2 Estudio de servicios y herramientas en la nube de Google Firebase para aplicaciones móviles	MB-IU	15	14-04-22	18-04-22
OE.3			15		
	ACT.1 Creación de la aplicación inicial mediante el uso Ionic framework	MB-IU	5	19-04-22	22-04-22
	ACT.2 Creación y configuración del proyecto en Firebase	MB-IU	5	19-04-22	22-04-22
	ACT.3 Creación de módulos genéricos para las distintas funciones de la aplicación de gestión de gastos	MB-IU	5	19-04-22	22-04-22
OE.4			229		
	ACT.1 Diseñar e implementar el submódulo de autenticación, control de roles (jefe de hogar y miembros de la familia)	IU	55	23-04-22	06-05-22
	ACT.2 Diseñar e implementar el submódulo de gestión de gastos	MB	55	23-04-22	06-05-22
	ACT.3 Diseñar e implementar el submódulo de gestión de notificaciones	MB	55	09-05-22	17-05-22
	ACT.6 Diseñar e implementar el submódulo de configuración y administración de perfiles	IU	55	09-05-22	17-05-22

Revisión/Corrección		MB-IU	54	18-05-22	08-07-22
Sprint 3			286		
OE. 1			18		
	ACT.3 Estudio de fundamentos de analítica de datos e inteligencia de negocio para visualización de reportes	MB-IU	18	30-05-22	01-06-22
OE. 4			180		
	ACT.4 Diseñar e implementar el submódulo de estimación de presupuesto	IU	90	02-06-22	13-06-22
	ACT.5 Diseñar e implementar el submódulo de reportes de gastos y presupuesto	MB	90	02-06-22	13-06-22
Revisión/Corrección		MB-IU	88	14-06-22	08-07-22
Sprint 4			43		
OE. 5			18		
	ACT.1 Diseño de pruebas funcionales	IU	3	21-06-22	21-06-22
	ACT.2 Elaboración de plantilla para validación de las pruebas	MB	3	21-06-22	21-06-22
	ACT.3 Ejecución las pruebas y registro de los resultados obtenidos	MB-IU	12	22-06-22	26-06-22
OE. 3			12		

	ACT.4 Creación y compilación de los paquetes los paquetes de aplicación (APK) de la aplicación de gestión de gastos para la instalación en el dispositivo	MB-IU	12	27-06-22	01-07-22
OE. 6			7		
	ACT.1 Elaboración del manual técnico de la aplicación de gestión de gastos	MB	5	04-07-22	06-07-22
	ACT.2 Elaboración del manual de usuario de la aplicación de gestión de gastos	IU	2	04-07-22	06-07-22
	Revisión/Corrección	MB-IU	6	07-07-22	08-07-22

Total de horas: 630

Total de horas Iván Ulloa: 315

Total de horas Martin Bojorque: 315

Fecha de inicio: 12 de abril del 2022

Fecha de fin: 08 de julio del 202

VIII. Presupuesto

Los gastos en la siguiente tabla fueron registrados durante el proceso de desarrollo de la aplicación.

DENOMINACIÓN	CANT.	COSTO UNITARIO	COSTO TOTAL
	Unidades	Dólares	Dólares
1. Bienes			
Papel Bond A-4	2	10.00	20.00
Copias	100	0.05	5.00
2.Tecnológico			
Computadora portátil	2	1000.00	2000.00
Teléfono inteligente	2	200.00	400.00
Servicios de Firebase	3 meses	12.14 por mes	36.42
2. Servicios			
Servicio de internet	3 meses	30.00 por mes	90.00
4. Personal			
Desarrollador	300 horas (2 desarrolladores)	8.00 por hora	4800.00
3. Otros			
Imprevistos	1	100.00	100.00
Total	412	\$ 1360.19	\$ 7451.42

IX. Conclusiones

La construcción de la aplicación para gestionar gastos fue una importante experiencia que nos ayudó a comprender de mejor manera las formalidades y disciplina necesarias para la elaboración de un proyecto. Los conocimientos aprendidos a lo largo de nuestros estudios universitarios fueron utilizados para las fases preliminares y el desarrollo de software en sí. Nuestro marco teórico muestra el aprendizaje adquirido en varios temas como el desarrollo de aplicaciones multiplataforma utilizando herramientas Web. El uso de estas tecnologías estudiadas agilitó el desarrollo en aspectos que incluyen la instalación fácil de paquetes necesarios para el desarrollo, métodos de compilación sencillos, una colección de librerías con comunidades activamente en busca de mejoras, entre muchas otras que pueden ser utilizadas en contextos ajenos a nuestro proyecto. Se aplicaron fundamentos de analítica de datos e inteligencia de negocios en la obtención de la mejor librería de gráficos que se acople a nuestras necesidades de exponer información útil, brindando visualizaciones limpias, concisas y fáciles de leer. Nuestra instrucción sobre los conceptos aplicados en este trabajo ha sido exitosa y ha generado un nuevo entendimiento del desarrollo de aplicaciones móviles multiplataforma.

La obtención de información acerca de las necesidades de los usuarios en busca de un proyecto con nuestras características nos llevó a aplicar diversas técnicas con la finalidad de obtener la mayor cantidad de datos en un corto tiempo y expresarlos en forma de requerimientos que describan nuestro proyecto de una manera más precisa. Tener una lista confiable de las necesidades de nuestra aplicación agilitó mucho el trabajo posterior dándonos objetivos claros en cada uno de los futuros módulos, que resultaron de agrupar requerimientos que se relacionen.

El tiempo de desarrollo de la aplicación fue limitado por la naturaleza del proyecto, pero al final la versión obtenida como producto final es la mejor que puede ser tomando en cuenta esas limitaciones. Las actividades previstas en el cronograma fueron mayormente respetadas, sin embargo, los tiempos que detallamos no resultaron adecuados. Las actividades de construcción de algunos módulos tomaron más tiempo del previsto, pero los desfases de tiempo fueron en su mayoría causados por procesos de depuración de errores al momento de agruparlos. Todas las características que formaron parte de la visión inicial al

momento de escribir el objetivo *OE3* fueron implementadas con distintos grados de satisfacción, pero muchos aspectos tuvieron una reducción de alcance al no poder ser concluidos exitosamente como fueron planteados por falta de tiempo. Algunos de estos aspectos incluyen: más gráficos con resúmenes de interés, más opciones en la pantalla de inicio, más opciones de estimación de gastos; todas estas pueden ser mejoradas o incluso cambiadas si se cree pertinente por un futuro trabajo que quiera retomar este tema.

Cuando definimos los alcances de nuestro proyecto tuvimos en cuenta que la realización de pruebas funcionales es fundamental cuando se desarrolla un software que está enfocado al público en general, al final optamos por la realización de más pruebas de lo planificado. Las pruebas unitarias nos dieron información sobre el comportamiento de las funcionalidades en aislamiento, pero no nos dice lo suficiente sobre la experiencia cuando están acoplados. Creímos pertinente realizar pruebas de integración de los módulos que finalmente nos dio información sobre las experiencias de los dos roles. Las pruebas de rendimiento nos dieron información de posibles optimizaciones y también un poco de la experiencia promedio de navegar en la aplicación. Por último, también decidimos aplicar pruebas aceptación para saber las reacciones de un público seleccionado a lo que habíamos desarrollado.

Los manuales realizados no fueron un mayor contratiempo, pero el desarrollo de la aplicación tuvo prioridad en todo el proceso. Como resultado el manual técnico solo contiene información crucial para el entendimiento del proyecto que podrá ser usada en actividades de mantenimiento. Creemos que lo incluido en el contenido también será suficiente para alguien familiarizado con las tecnologías usadas, también le dará una idea del enfoque de la aplicación y lo que puede ser incluido o excluido en futuras versiones de la aplicación.

X. Recomendaciones

Firebase es una plataforma que debería estar al principio de la lista de desarrolladores de aplicaciones móviles si hablamos de *backend*. Teniendo características como fácil escalamiento que funciona en conjunto con la aplicación, facilidad de administración y otros servicios incorporados. Si se opta por desarrollar la aplicación sin la necesidad de administrar recursos individuales, crear APIs para el manejo de peticiones de datos u otras funcionalidades, esta plataforma lo tiene todo cubierto y sus planes son económicos

acoplándose a los intereses de cada grupo y el tamaño de la aplicación a desarrollar. El entorno de Node.js tiene muchas ventajas para el desarrollo de aplicaciones móviles utilizando Firebase, por lo que se recomienda para aprovechar características como el desarrollo de funciones en la nube (Cloud Functions). También si se optan por otros medios de desarrollo como el uso de Android Studio, el soporte es muy bueno de igual manera.

Si el trabajo es retomado en el futuro la inclusión de personalización de las plantillas por defecto de Ionic Framework darían a la aplicación una personalidad que puede ser atractiva para el público. El diseño audio/visual forma parte de la experiencia de software, efectos de sonidos para notificaciones, ingresos de datos, clicks en botones en conjunto con colores y formas que estimulan sus sentidos pueden hacer que su respuesta sea incluso más positiva que solo con las funcionalidades ofrecidas por defecto. Diseñar una excelente experiencia de usuario podría ser un buen motivo para revisar y mejorar funcionalidades existentes dando forma a una nueva aplicación.

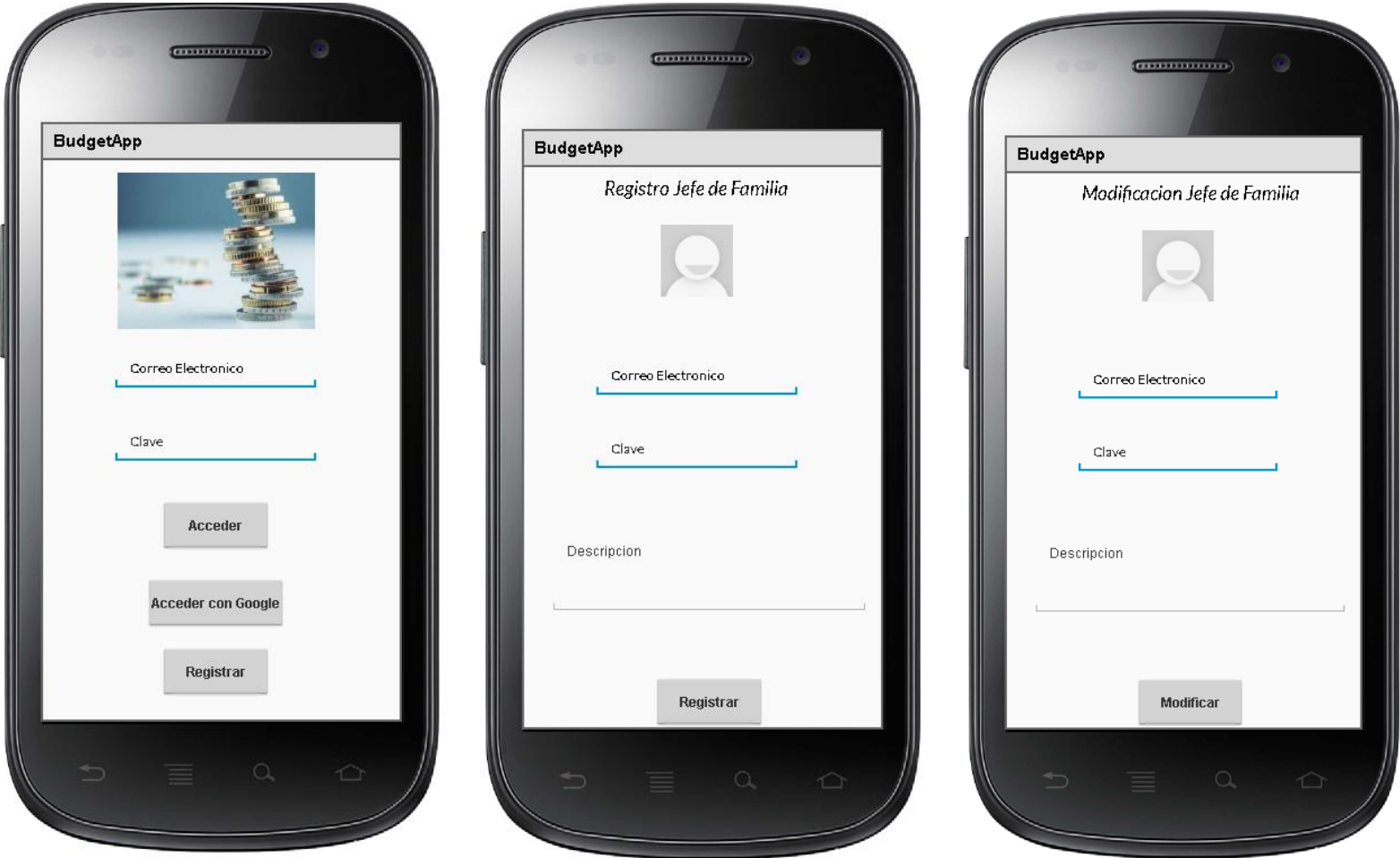
REFERENCIAS BIBLIOGRÁFICAS

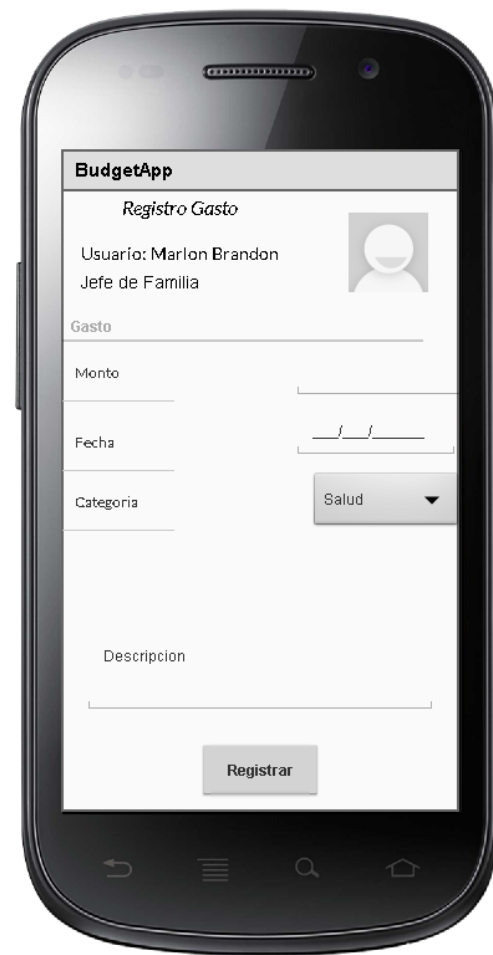
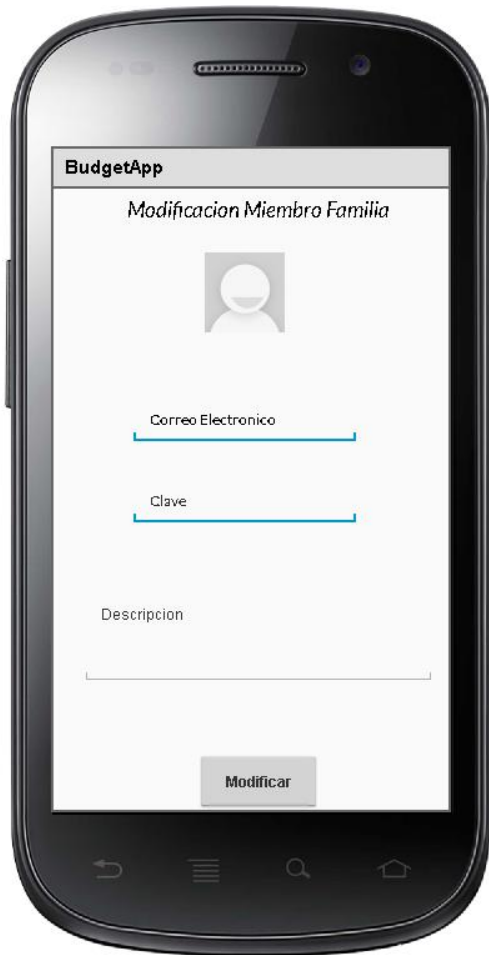
- BudgetBakers.com. 2011. Wallet: rastreador de finanzas (Versión 8.4.101) [Aplicación Móvil]. Google Play. <https://play.google.com/store/apps/details?id=com.droid4you.application.wallet&hl=es>
- CASorin. 2016. Expense Manager (Versión 1.3.2) [Aplicación Móvil]. Google Play. <https://play.google.com/store/apps/details?id=com.smartexpenditure&hl=es>
- Chaudhary, P. (2018). Ionic Framework. *International Research Journal of Engineering and Technology*, 5(05), 3181-3185.
- Gómez, R. (2019). Finances management App. Universitat Autònoma de Barcelona. <https://ddd.uab.cat/record/211504/>
- Innim Mobile Exp. 2020. Gestor de gastos, ingresos y presupuesto, dinero (Versión 0.9.12) [Aplicación Móvil]. Google Play. https://play.google.com/store/apps/details?id=ru.innim.my_finance&hl=es
- Moroney, L., Moroney, A., & Anglin. (2017). *Definitive Guide to Firebase* (pp. 51-71). California: Apress.
- Morrow, C., & Dees, B. (2020). Money Management for Young Adults with Autism. Autism at-a-Glance. *Center on Secondary Education for Students with Autism Spectrum Disorder*.
- Mueller, R. (2020). Exploring family nurse practitioners' practices in recommending mhealth apps to patients. *CIN: Computers, Informatics, Nursing*, 38(2), 71-79.
- Pareek, D. (2006). *Business Intelligence for telecommunications*. Auerbach Publications.
- Patil, S., Patil, N., Bhatambrekar, S., & Kulkarni, S. (2008). Android App: Personal Finance (Pf). *Link: https://bit.ly/3hjsKTr*.
- Preston, S. (2016). Microsoft Azure Terminology and Concepts. In *Using Chef with Microsoft Azure* (pp. 29-53). Apress, Berkeley, CA.

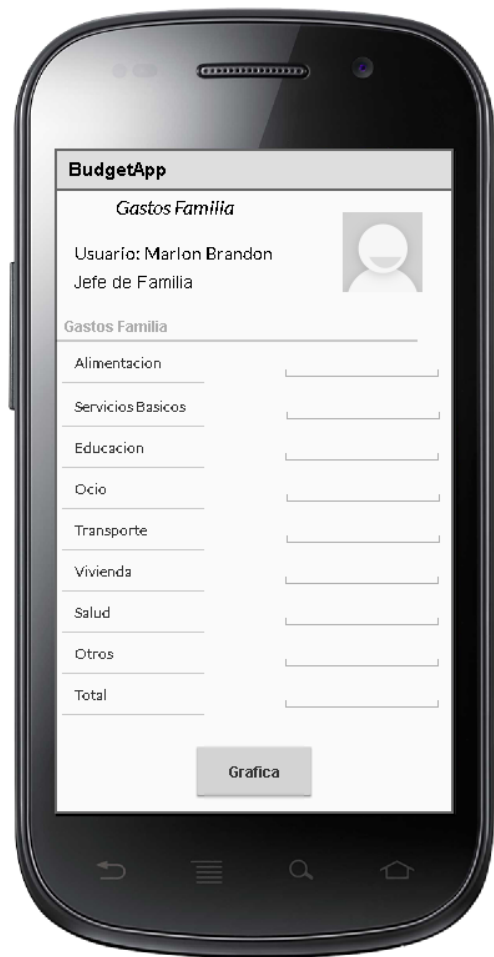
- Rossellinee, I. (2019). Useful Digital Tools And App Services Will Simplify Money Management.
- Realbyte Inc. 2013. Registro contable (Versión 4.6.20) [Aplicación Móvil]. Google Play. <https://play.google.com/store/apps/details?id=com.realbyteapps.moneymanagerfree&hl=es>
- Reflectly. 2014. Menefy – App de control de gastos e ingresos (Versión 1.16.0) [Aplicación Móvil]. Google Play. <https://play.google.com/store/apps/details?id=com.monefy.app.lite&hl=es>
- Runkler, T. A. (2020). *Data analytics*. Springer Fachmedien Wiesbaden.
- Soni, R. (2016). Introduction to nginx web server. In *Nginx* (pp. 1-15). Apress, Berkeley, CA.
- Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017, May). SCRUM model for agile methodology. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 864-869). IEEE.
- Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, *14*(6), 80-83.
- Tully, C., & Alfaraz, C. (2017). Youth and mobility: The lifestyle of the new generation as an indicator of a multi-local everyday life. *Applied Mobilities*, *2*(2), 182-198.

ANEXOS

Anexo 1: PROTOTIPOS DE INTERFACES DE USUARIO







Anexo 2: PRUEBAS UNITARIAS

Verificación de Pruebas					
Encargados	Iván Ulloa				
Código de Requerimiento:	LoginPrueba				
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Probar el ingreso a la aplicación en la página de <i>login</i> , su funcionamiento, disposición de elementos y éxito en la función asignada.					
Método de aplicación					
No se usan aplicaciones adicionales. Se aplican un conjunto de datos y se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
4 inicios de sesión con cada cuenta que se encuentra registrada		Correo: outlook.com Contraseña: 123456			
Correo: nada (Google) Contraseña: nada (Google)		Correo: prueba1@outlook.com Contraseña: 123			
Correo: yo@mail.com Contraseña: 123456		Correo: prueba1@outlook.com Contraseña: prueba1#contra			
Correo: usuario1@gmail.com Contraseña: 123456					
Argumentación					
Las configuraciones del inicio de sesión están por defecto y por lo tanto solo verifica que la contraseña tenga 6 caracteres. La sintaxis del correo electrónico se verifica al enviar. Existe una rueda de carga para mostrar que se está realizando alguna acción y que debemos esperar. Algunos errores con la conexión a Firebase pueden pasar en casos específicos. Las redirecciones a otras páginas funcionan bien de acuerdo al contexto. SE verifica la combinación correcta de correo electrónico y contraseña.					
Evaluación:	No cumple		Cumple	X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
				X	

Verificación de Pruebas					
Encargados		Iván Ulloa			
Código de Requerimiento:			CrearCuentaPrueba		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Probar la creación de cuentas en la página <i>signup</i> , su funcionamiento, disposición de elementos y éxito en la función asignada.					
Método de aplicación					
No se usan aplicaciones adicionales. Se aplican un conjunto de datos y se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
<p>Correo: yo@mail.com Contraseña: 123456 Nombre: Pedrito Salinas</p> <p>Correo: usuario1@gmail.com Contraseña: 123456 Nombre: Mary Salinas</p> <p>Correo: outlook.com Contraseña: 1234 Nombre: Lucas Prado</p> <p>Correo: prueba1@outlook.com Contraseña: prueba1#contra Nombre: Lucas Prado</p>					
Argumentación					
El funcionamiento es el esperado. Los elementos describen que clase de información es necesaria y están claros. Existe una rueda de carga para mostrar que se está realizando alguna acción y que debemos esperar. Las pruebas no demostraron ningún problema de conexión con Firebase. Las redirecciones después de crear la cuenta funcionan correctamente. Ningún método de confirmación de correo ha sido implementado.					
Evaluación:	No cumple		Cumple	X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
	X				

Verificación de Pruebas					
Encargados		Iván Ulloa			
Código de Requerimiento:			CrearFamiliaPrueba		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Probar la creación de una familia en la página createfamily, su funcionamiento, disposición de elementos y éxito en la función asignada.					
Método de aplicación					
No se usan aplicaciones adicionales. Se aplican un conjunto de datos y se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Nombre: Salinas Presupuesto global: 500 Primer día del mes: 6 Nombre: Prado Presupuesto global: 1000 Primer día del mes: 9					
Argumentación					
El funcionamiento es el esperado. Los elementos describen que clase de información es necesaria y están claros. El funcionamiento incluye unirse a una familia existente, pero no existe un sistema de incitaciones. Existe una rueda de carga para mostrar que se está realizando alguna acción y que debemos esperar. Las pruebas no demostraron ningún problema de conexión con Firebase. Las redirecciones después de crear la cuenta funcionan correctamente. Ser redireccionado a esta pantalla parece siempre funcionar de acuerdo a lo anticipado. No es posible unirse a otra familia de otra manera que no sea por medio del correo electrónico del jefe de hogar.					
Evaluación:		No cumple		Cumple	X
Grado de Cumplimiento		ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO
		X			

Verificación de Pruebas					
Encargados		Iván Ulloa			
Código de Requerimiento:			PerfilPrueba		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Probar la visualización y modificación del perfil de usuario en la página <i>profile</i> , su funcionamiento, disposición de elementos y éxito en la función asignada.					
Método de aplicación					
No se usan aplicaciones adicionales. Se aplican un conjunto de datos y se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
<p>Email: algomas@mail.com Descripción: Hola, voy a comenzar con mis ahorros.</p> <p>Email: @mail.com Descripción: [dejarlo sin modificar]</p> <p>Email: [dejarlo sin modificar] Descripción: Hola, voy a comenzar con mis ahorros.</p> <p>Email: algomas@mail Descripción: [dejarlo sin modificar]</p> <p>Imagen de perfil: [múltiples imágenes subidas]</p>					
Argumentación					
El funcionamiento es el esperado. Los elementos describen que clase de información es necesaria y están claros. Existe una rueda de carga para mostrar que se está realizando alguna acción y que debemos esperar. Las pruebas no demostraron ningún problema de conexión con Firebase, aunque pueden darse al subir una foto de perfil en casos muy específicos. Se muestra el progreso de subida de la imagen al sistema. Permite cerrar sesión y su funcionamiento es lo esperado. Las acciones requieren la interacción con botones y no se sincronizan automáticamente.					
Evaluación:	No cumple		Cumple	X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
	X				

Verificación de Pruebas					
Encargados		Iván Ulloa			
Código de Requerimiento:			MiembrosPrueba		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Probar la visualización y modificación los miembros de la familia junto con sus gastos en la página <i>list-familymembers</i> , su funcionamiento, disposición de elementos y éxito en la función asignada.					
Método de aplicación					
No se usan aplicaciones adicionales. Se aplican un conjunto de datos y se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Eliminación de todos los miembros. Eliminación de un miembro de la familia. Revisar los gastos de cada miembro en la lista con gastos ya ingresados. Revisar los gastos de cada miembro en la lista sin gastos ingresados.					
Argumentación					
El funcionamiento es el esperado. Los elementos describen que clase de información es necesaria y están claros. Existe una rueda de carga para mostrar que se está realizando alguna acción y que debemos esperar. Las pruebas no demostraron ningún problema de conexión con Firebase. No es posible la eliminación del jefe de hogar, pero se lista como miembro. Los gastos se visualizan de manera de una lista en una ventana emergente. No es posible asignar un nuevo jefe de hogar.					
Evaluación:		No cumple		Cumple	X
Grado de Cumplimiento		ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO
					X

Verificación de Pruebas					
Encargados		Iván Ulloa			
Código de Requerimiento:			PredicciónGastosPrueba		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Probar la visualización de una regresión lineal usando los datos de gastos de la familia de en modo de predicción en la página <i>spence-prediction</i> , junto con su funcionamiento, disposición de elementos y éxito en la función asignada.					
Método de aplicación					
No se usan aplicaciones adicionales. Se aplican un conjunto de datos y se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Monto: 29.69 Fecha: 2022-05-16T19:52:00-05:00			Monto: 12 Fecha: 2022-06-21T15:30:00-05:00		
Monto: 4 Fecha: 2022-06-07T15:44:00-05:00			Monto: 17.1 Fecha: 2022-06-14T17:17:00-05:00		
Monto: 230.49 Fecha: 2022-06-21T11:07:00-05:00			Monto: 10 Fecha: 2022-06-21T18:23:00-05:00		
Monto: 23.5 Fecha: 2022-05-14T15:20:00-05:00			Monto: 45 Fecha: 2022-06-10T18:06:00-05:00		
Argumentación					
El funcionamiento es el esperado. Los elementos describen que clase de información es necesaria y están claros. Existe una rueda de carga para mostrar que se está realizando alguna acción y que debemos esperar. La escala se da en meses, pero si se quieren datos específicos se puede dar click en cada punto. También se puede hacer zoom para poder seleccionar un punto fácilmente. Las pruebas no demostraron ningún problema de conexión con Firebase.					
Evaluación:	No cumple		Cumple	X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
				X	

Verificación de Pruebas					
Encargados		Martin Bojorque			
Código de Requerimiento:		Reporte de Gastos/Presupuestos			
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
					X
Propósito					
Probar la generación de los reportes dedicados a los gastos y presupuestos					
Método de aplicación					
No se usan aplicaciones adicionales. Se almacenan un conjunto de datos, se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Registros de gastos almacenados en firebase			Registros de presupuestos almacenados en firebase		
Argumentación					
El funcionamiento es el esperado. Los diferentes grafos están acorde a los registros almacenados en firebase. Existe una rueda de carga para mostrar que se está realizando la consulta de la información y que se debe esperar. Las pruebas no demostraron ningún problema de conexión con Febeas.					
Evaluación:		No cumple		Cumple	X
Grado de Cumplimiento		ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO
					X

Verificación de Pruebas					
Encargados		Martin Bojorque			
Código de Requerimiento:			Ver gastos/ Ver gastos Familia / Ver Gastos Personales		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
					X
Propósito					
Probar la visualización de los diferentes listados de gastos dependiendo del rol del usuario					
Método de aplicación					
No se usan aplicaciones adicionales. Se almacenan un conjunto de datos, se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Registros de gastos del usuario almacenados en firebase			Registros de gastos de la familia almacenados en firebase		
Argumentación					
El funcionamiento es aceptable. Posibles registros incongruentes debido a la constante consulta a Firebase y la naturaleza de la respuesta recibida					
Evaluación:		No cumple		Cumple	X
Grado de Cumplimiento		ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO
		X			

Verificación de Pruebas					
Encargados		Martin Bojorque			
Código de Requerimiento:			Registrar presupuesto/ Ver presupuestos		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
					X
Propósito					
Probar el registro de los presupuestos y su respectivo reporte para la cabeza de familia					
Método de aplicación					
No se usan aplicaciones adicionales. Se almacenan un conjunto de datos, se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Presupuesto general Presupuesto Salud Presupuesto Alimentación Presupuesto Salud Presupuesto Educación Presupuesto Ocio Presupuesto Transporte Presupuesto Vivienda Presupuesto Otros					
Argumentación					
El funcionamiento es planeado. El registro se hace de manera organizada y bajo una determinada estructura.					
Evaluación:	No cumple		Cumple	X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
				X	

Verificación de Pruebas					
Encargados		Martin Bojorque			
Código de Requerimiento:			Registrar presupuesto/ Ver presupuestos		
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
					X
Propósito					
Probar el registro de los presupuestos y su respectivo reporte para la cabeza de familia					
Método de aplicación					
No se usan aplicaciones adicionales. Se almacenan un conjunto de datos, se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
<p>Presupuesto general</p> <p>Presupuesto Salud</p> <p>Presupuesto Alimentación</p> <p>Presupuesto Salud</p> <p>Presupuesto Educación</p> <p>Presupuesto Ocio</p> <p>Presupuesto Transporte</p> <p>Presupuesto Vivienda</p> <p>Presupuesto Otros</p>					
Argumentación					
El funcionamiento es planeado. El registro se hace de manera organizada y bajo una determinada estructura					
Evaluación:		No cumple		Cumple	X
Grado de Cumplimiento		ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO
					X

Anexo 3: PRUEBAS DE INTEGRACIÓN

Verificación de Pruebas					
Encargados	Iván Ulloa				
Código de Requerimiento:	MiembroFamiliaPrueba				
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Prueba de integración de los módulos enfocados a miembros de la familia, junto con su funcionamiento, disposición de elementos y éxito en la función asignada.					
Método de aplicación					
No se usan aplicaciones adicionales. Se aplican un conjunto de datos y se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Utilizando cuenta de Google	Correo: usuario1@gmail.com				
Monto: 11.0	Contraseña: 123456				
Fecha: 2022-05-16T19:52:00-05:00	Nombre: Mary Salinas				
	Monto: 45				
	Fecha: 2022-06-10T18:06:00-05:00				
Monto: 12	Monto: 34.5				
Fecha: 2022-06-21T15:30:00-05:00	Fecha: 2022-06-22T11:27:00-05:00				
Monto: 230.49					
Fecha: 2022-06-21T11:07:00-05:00					
Monto: 23.5					
Fecha: 2022-05-14T15:20:00-05:00					
Argumentación					
El funcionamiento es el esperado. Los elementos describen que clase de información es necesaria y están claros. Existe una rueda de carga para mostrar que se está realizando alguna acción y que debemos esperar. Las pruebas no demostraron ningún problema de conexión con Firebase. Las opciones ofrecidas permiten a los miembros usar funcionalidades básicas de ingreso de gastos y visualización de historiales. El esquema de color de la aplicación en general es el adecuado y el uso de un tema oscura ayuda a resaltar ciertos aspectos de la funcionalidad como el gráfico de inicio de mejor manera. La información presentada es pertinente y correcta de acuerdo al usuario que se encuentra en la sesión.					
Evaluación:	No cumple		Cumple	X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
				X	
Verificación de Pruebas					

Encargados		Martin Bojorque			
Código de Requerimiento:		Integración de funcionalidades para miembros de la familia			
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional
	X				X
Propósito					
Probar el uso de las diferentes funcionalidades destinadas para el jefe de familia					
Método de aplicación					
No se usan aplicaciones adicionales. Se almacenan un conjunto de datos, se observa el comportamiento y de acuerdo al resultado se determinará si su funcionamiento es ideal.					
Entradas					
Presupuesto general			Gastos personales		
Presupuesto Salud			Gastos familiares		
Presupuesto Alimentación					
Presupuesto Salud					
Presupuesto Educación					
Presupuesto Ocio					
Presupuesto Transporte					
Presupuesto Vivienda					
Presupuesto Otros					
Argumentación					
El uso de las funcionalidades en conjunto para la observación de los diferentes resultados presenta resultados planeados en su totalidad. Los valores registrados se observan de manera óptima dentro de cada una de las gráficas en conjunto con los diferentes gastos registrados en la base de datos.					
Evaluación:	No cumple		Cumple	X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
				X	

Anexo 4: PRUEBAS DE RENDIMIENTO

Verificación de Pruebas						
Encargados		Iván Ulloa				
Código de Requerimiento:			RendimientoNavegacionWebPrueba			
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional	
		X				
Propósito						
Prueba de rendimiento de la navegación entre las distintas funcionalidades de la aplicación en la Web. Se toma en cuenta que todos los componentes hayan sido renderizados (excepto gráficas).						
Método de aplicación						
No se usan aplicaciones adicionales. Se toma el tiempo del sistema y se distribuyen en tres pruebas por módulo, solo para los módulos que se pueden navegar dentro del menú lateral.						
Entradas						
U1		U2		U3		
Utilizando cuenta de Google Role: Miembro de familia		Correo: usuario1@gmail.com Contraseña: 123456 Nombre: Mary Salinas Role: Miembro de familia		Correo: yo@mail.com Contraseña: 123456 Nombre: Pedrito Salinas Role: Jefe de familia		
Resultados (tiempo en segundos)						
Módulo				P1	P2	P3
Inicio				1.84	2.32	1.38
Perfil				0.91	2.29	2.43
Miembros				0.75	2.01	1.69
Registrar presupuesto/ Ver presupuestos				1.95	1.09	1.04
Registrar gasto				1.18	2.37	1.99
Ver gastos/ Ver gastos Familia / Ver Gastos Personales				1.50	1.77	2.19
Reporte de Gastos/Presupuestos				0.92	1.91	1.68
Predicción de gastos				1.64	1.76	1.1
Argumentación						
La aplicación web es lo suficientemente rápida en sus respuestas, pero se dan casos de estrés que pueden ser ocasionados por el dispositivo o por la conexión al internet. Algunos de estos casos son aislados, pero otros parecen repetirse con más frecuencia. Las predicciones de gastos parecen tener un tiempo de procesamiento alto, pero sería peor si esta se realizara directamente en el dispositivo.						
Evaluación:		No cumple		Cumple	X	
Grado de Cumplimiento		ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
		X				

Verificación de Pruebas						
Encargados		Iván Ulloa				
Código de Requerimiento:			RendimientoNavegacionAndroidPrueba			
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional	
		X				
Propósito						
Prueba de rendimiento de la navegación entre las distintas funcionalidades de la aplicación Android. Se toma en cuenta que todos los componentes hayan sido renderizados (excepto gráficas).						
Método de aplicación						
No se usan aplicaciones adicionales. Se toma el tiempo del sistema y se distribuyen en tres pruebas por módulo, solo para los módulos que se pueden navegar dentro del menú lateral.						
Entradas						
U1		U2		U3		
Utilizando cuenta de Google Role: Miembro de familia		Correo: usuario1@gmail.com Contraseña: 123456 Nombre: Mary Salinas Role: Miembro de familia		Correo: yo@mail.com Contraseña: 123456 Nombre: Pedrito Salinas Role: Jefe de familia		
Resultados (tiempo en segundos)						
Módulo				P1	P2	P3
Inicio				1.55	1.90	1.65
Perfil				0.49	0.45	1.51
Miembros				1.29	0.57	0.69
Registrar presupuesto/ Ver presupuestos				0.37	1.85	0.80
Registrar gasto				1.31	1.95	1.57
Ver gastos/ Ver gastos Familia / Ver Gastos Personales				1.79	1.78	0.58
Reporte de Gastos/Presupuestos				0.78	1.92	0.44
Predicción de gastos				0.83	0.33	1.48
Argumentación						
La aplicación web es lo suficientemente rápida en sus respuestas, pero se dan casos de estrés que pueden ser ocasionados por el dispositivo o por la conexión al internet. Algunos de estos casos son aislados, pero otros parecen repetirse con más frecuencia. Las predicciones de gastos parecen tener un tiempo de procesamiento alto, pero sería peor si esta se realizara directamente en el dispositivo.						
Evaluación:		No cumple		Cumple	X	
Grado de Cumplimiento		ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO	
		X				

Verificación de Pruebas							
Encargados	Iván Ulloa						
Código de Requerimiento:				TiempoGraficas			
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional		
		X					
Propósito							
Tiempo de espera para la renderización de gráficas de reportes en la aplicación.							
Método de aplicación							
Se toma el tiempo con la ayuda del tiempo del sistema realizando 7 pruebas por cada módulo que posee renderización de gráficas.							
Entradas							
<i>Irrelevante para esta prueba</i>							
Resultados (tiempo en segundos)							
Módulo	P1	P2	P3	P4	P5	P6	P7
Inicio	2.4	2.41	2.57	3.55	3.56	3.24	3.46
Reporte de Gastos/Presupuestos	3.05	2.24	2.94	2.53	2.96	1.10	1.25
Predicción de gastos	1.04	2.20	2.90	2.38	3.55	2.44	3.70
Argumentación							
La aplicación web es lo suficientemente rápida en sus respuestas, pero se dan casos de estrés que pueden ser ocasionados por el dispositivo o por la conexión al internet. Algunos de estos casos son aislados, pero otros parecen repetirse con más frecuencia. Las predicciones de gastos parecen tener un tiempo de procesamiento alto, pero sería peor si esta se realizara directamente en el dispositivo.							
Evaluación:	No cumple			Cumple		X	
Grado de Cumplimiento	ACEPTABLE		NO ACEPTABLE		EXCESO		PLANEADO
	X						

Verificación de Pruebas						
Encargados	Iván Ulloa					
Código de Requerimiento:			TiempoFormularios			
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional	
		X				
Propósito						
Rendimiento al momento de envío de formularios, tiempo de espera para que se guarde la información.						
Método de aplicación						
Se toma el tiempo con la ayuda del tiempo del sistema realizando 5 pruebas por cada módulo que posee un formulario.						
Entradas <i>Irrelevante para esta prueba</i>						
Resultados (tiempo en segundos)						
	Módulo	P1	P2	P3	P4	P5
	Crear cuenta	0.52	0.97	0.59	0.98	1.06
	Crear familia	0.82	0.11	0.71	0.97	0.70
	Perfil	0.06	1.34	1.35	0.02	0.53
	Registrar presupuesto/ Ver presupuestos	0.28	0.76	0.58	0.03	0.82
	Registrar gasto	0.17	0.59	0.54	0.66	0.66
Argumentación						
La aplicación web es lo suficientemente rápida en sus respuestas, pero se dan casos de estrés que pueden ser ocasionados por el dispositivo o por la conexión al internet. Algunos de estos casos son aislados, pero otros parecen repetirse con más frecuencia. Las predicciones de gastos parecen tener un tiempo de procesamiento alto, pero sería peor si esta se realizara directamente en el dispositivo.						
Evaluación:	No cumple		Cumple		X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO		PLANEADO	
	X					

Verificación de Pruebas							
Encargados		Iván Ulloa					
Código de Requerimiento:				TiempoFunciones			
Tipo Prueba:	Usabilidad	Eficiencia de desempeño	Mantenimiento	Seguridad	Adecuación Funcional		
		X					
Propósito							
Tiempo de respuesta de las funciones desplegadas utilizando Cloud Functions.							
Método de aplicación							
Se toma el tiempo con la ayuda del tiempo del sistema realizando 7 pruebas por cada función desplegada.							
Entradas <i>Irrelevante para esta prueba</i>							
Resultados (tiempo en segundos)							
Módulo	P1	P2	P3	P4	P5	P6	P7
Gráfica de inicio	3.55	1.39	1.98	3.40	1.64	3.4	2.11
Gráficas por categoría	3.28	3.38	2.77	2.83	2.44	1.84	1.33
Predicción de gastos	3.54	1.45	1.23	1.61	3.04	1.27	1.05
Argumentación							
La aplicación web es lo suficientemente rápida en sus respuestas, pero se dan casos de estrés que pueden ser ocasionados por el dispositivo o por la conexión al internet. Algunos de estos casos son aislados, pero otros parecen repetirse con más frecuencia. Las predicciones de gastos parecen tener un tiempo de procesamiento alto, pero sería peor si esta se realizara directamente en el dispositivo.							
Evaluación:	No cumple			Cumple		X	
Grado de Cumplimiento	ACEPTABLE	NO ACEPTABLE	EXCESO	PLANEADO			
	X						

Anexo 5: ENCUESTA PARA PRUEBA DE ACEPTACIÓN

- **Pregunta 1:** Califique nuestras gráficas
Calificación del 1-5. La calificación más bajo en “Incomprensible” y la más alta es “Excelente”.
- **Pregunta 2:** Califique el diseño visual
Calificación del 1-5. La calificación más bajo en “Malo” y la más alta es “Excelente”.
- **Pregunta 3:** Califique la facilidad de uso
Calificación del 1-5. La calificación más bajo en “Difícil de manejar” y la más alta es “Fácil de manejar”.
- **Pregunta 4:** Califique la velocidad de respuesta
Calificación del 1-5. La calificación más bajo en “Demorado” y la más alta es “Rápido”.
- **Pregunta 5:** Califique la forma de ingreso de datos
Calificación del 1-5. La calificación más bajo en “Difícil de usar” y la más alta es “Muy fácil de usar”.
- **Pregunta 6:** Califique la interacción con la app (Mensajes de error o informativos)
Calificación del 1-5. La calificación más bajo en “Inútiles” y la más alta es “Muy útiles”.
- **Pregunta 6:** Sugerencias de mejora
Comentarios escritos detallando aspectos relevantes para el encuestado.

Anexo 6: RESULTADOS DE LAS ENCUESTAS PARA LAS PRUEBAS DE ACEPTACIÓN

Marca temporal	Puntuación /30	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6
7/11/2022 14:59:24	22	5	3	2	4	4	4
7/11/2022 15:08:56	16	2	2	3	4	2	3
7/11/2022 15:19:17	20	3	3	3	4	4	3
7/12/2022 12:49:29	23	4	3	5	3	5	3
7/12/2022 12:58:44	30	5	5	5	5	5	5

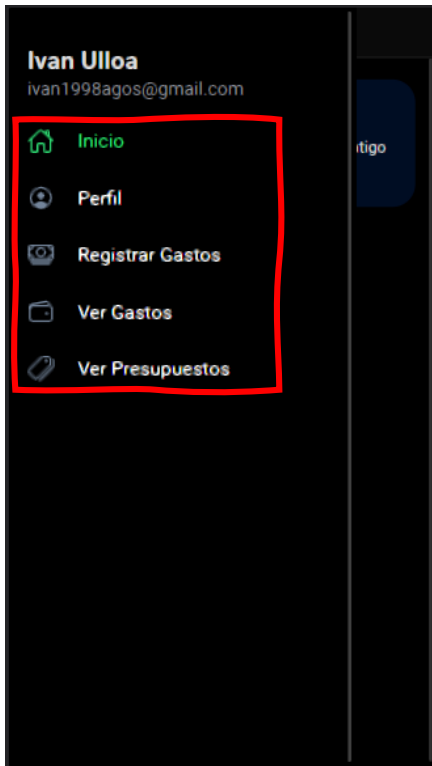
7/12/2022 13:29:50	23	5	4	3	5	3	3
7/12/2022 14:29:40	14	2	1	2	4	2	3
7/12/2022 14:30:51	24	4	5	4	3	4	4
7/12/2022 15:07:55	23	4	3	4	4	4	4
7/12/2022 16:16:09	23	4	4	3	4	4	4
7/12/2022 16:28:24	30	5	5	5	5	5	5
7/12/2022 17:32:21	20	4	4	3	3	3	3
TOTAL	22.33	3.91	3.5	3.5	4	3.75	3.667

Marca temporal	Sugerencias de mejora
7/11/2022 14:59:24	No existe navegabilidad
7/11/2022 15:08:56	Incluir botones para redireccionamiento, incluir detalles de las opciones, es una app poco intuitiva
7/11/2022 15:19:17	Mejoramiento de la interfaz de usuario, hacerlo más acorde al contexto de gastos. Utilizar el logo de la página de login con transparencia. Implementar accesibilidad en la aplicación.
7/12/2022 12:49:29	Mensaje que indique el usuario ya se encuentra registrado con ese correo. Ya que intente crear otra cuenta con el mismo correo y no indica que este ya fue registrado y posterior o esto se queda esperando solicitud de respuesta.
7/12/2022 12:58:44	Para mí todo fue una buena experiencia y no creo que se necesite mejoras
7/12/2022 13:29:50	Manejo de la aplicación
7/12/2022 14:29:40	Mejorar la parte visual, el color negro en toda la aplicación no se hace agradable, algunos textos y selección de entradas son de difícil de entender, también fuera bueno unas gráficas más entendibles ya que las que se muestran son muy técnicas y no me dice realmente algo en lo que pueda mejorar, como un dashboard para una persona que sabe de estas cosas está entendible pero como cliente realmente no me dice nada, es más tengo que ponerme a entender las gráficas,

	<p>haciendo que me tome tiempo en vez de ayudarme, tambien dentro de la app, no hay control de entradas de valores, puedo colocar valores absurdos creo que esto deberia de ser modificado, cuando doy refrescar me sale error de pagina web, tengo que regresar al login, y eso que ya tenia ingresado.</p>
7/12/2022 14:30:51	<p>Mejorar velocidad de respuesta del sistema Ser más específicos en los mensajes de ser posible. Al pedir iniciar sesión nuevamente debería borrarse la contraseña de la cuadrícula, salvo agregaran una opción de 'Recordarme'</p>
7/12/2022 15:07:55	<p>Falta una introducción en cuál se indique a qué va dirigida la app o la función que se va a cumplir antes de que el usuario cree una cuenta.</p>

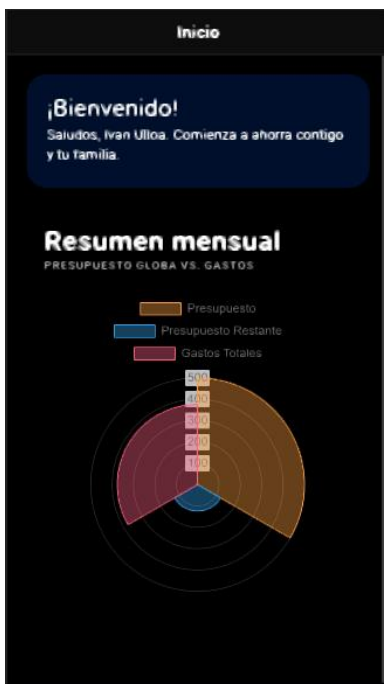
Anexo 7: MANUAL DE USUARIO

Menú lateral



En este menú se verán todo lo que puedes hacer con la aplicación Gestionar Gastos, diferentes opciones se mostrarán si eres un miembro de familia o un jefe de familia.

Pantalla de inicio

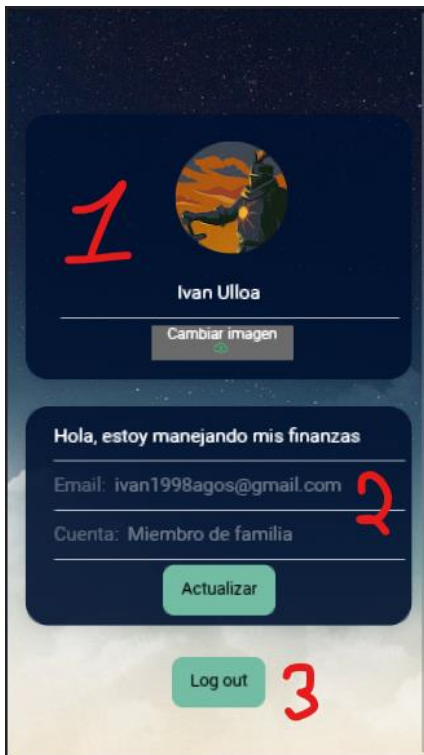


En esta pantalla se visualizará un gráfico comparativo de los gastos versus el presupuesto junto con un mensaje de bienvenida.

Perfil

Aquí se visualizará tu información básica del perfil.

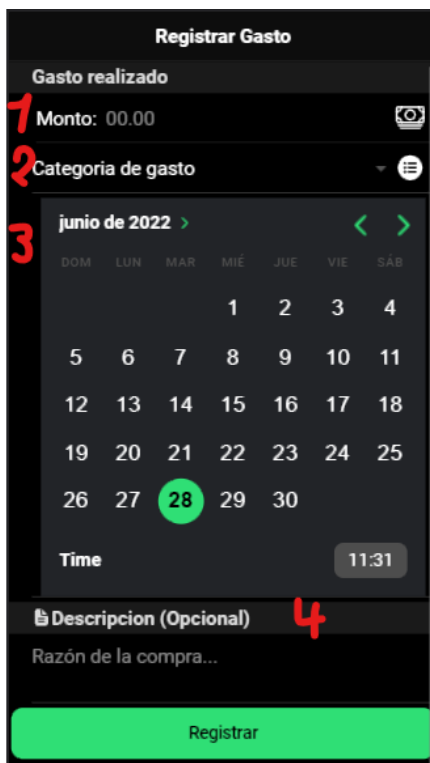
1. Puedes subir una imagen de perfil nueva
2. Puedes visualizar y cambiar tu información como nombre de perfil y descripción.
3. Puedes cerrar sesión.



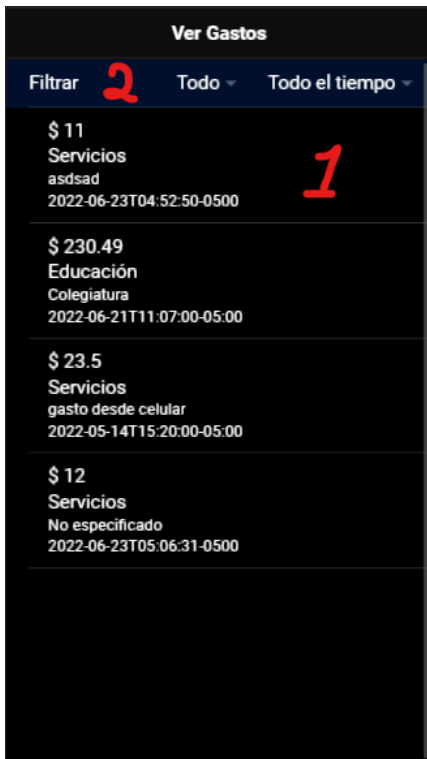
Registrar gastos

Aquí podrás registrar un gasto que realizaste.

1. Monto del gasto.
2. Tipo de gasto que estás realizando: Alimentación, Salud, Educación, etc.
3. Fecha del gasto (fecha actual por defecto).
4. Una descripción para explicar mejor la razón del gasto.



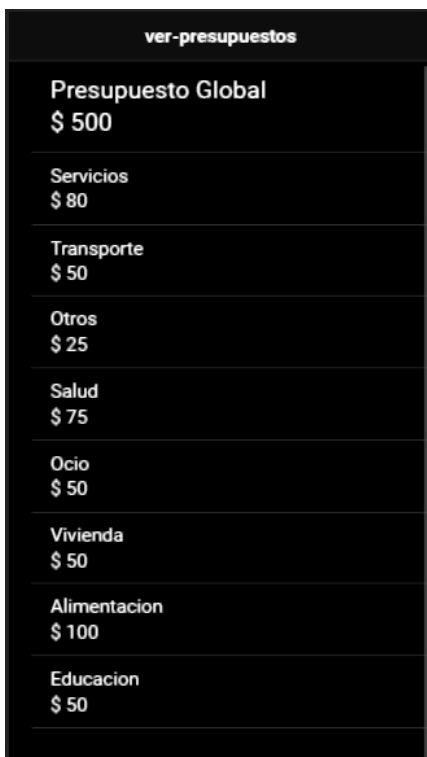
Ver gastos



Podrás ver los gastos que has realizado.

1. Lista de gastos que has realizado.
2. Filtrar lista. Podrás filtrar por tipo de gastos (alimentación, salud, etc.) y por tiempo (todo el tiempo o el último mes).

Ver presupuestos



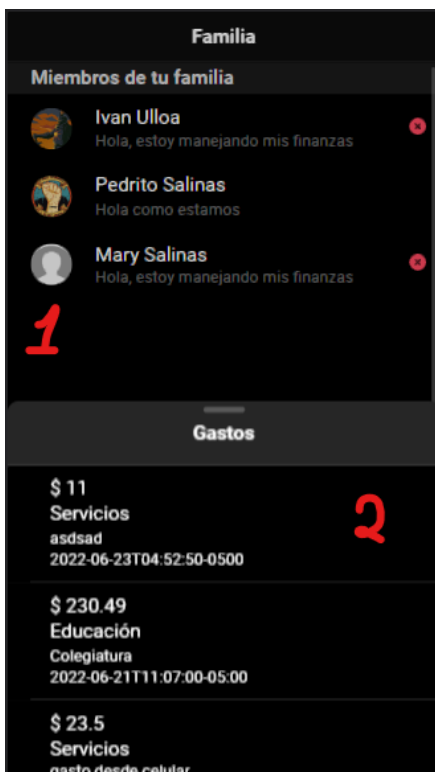
Revisa los presupuestos para cada una de las categorías ofrecidas por la app incluido el presupuesto global.

Menú lateral



Como jefe de familia tendrás acceso a otras características de la aplicación Gestionar Gastos.

Miembros



Aquí podrás revisar a los miembros que tiene tu familia. Podrás eliminar la cuenta de tu familia utilizando el ícono (X).

1. Lista de miembros de la familia.
2. Gastos realizados por el miembro seleccionado.

Registrar presupuestos

Registrar Presupuestos	
Presupuesto General	
1500	Asignado: 480
Alimentacion	
100	
Servicios	
80	
Educacion	
50	
Ocio	
50	
Transporte	
50	
Vivienda	
50	
Salud	
75	
Otros	
00	

Aquí podrás ingresar todos tus presupuestos tentativos por cada categoría, al igual que modificar el presupuesto global.

1. Presupuesto global
2. Cantidad asignada del presupuesto global
3. Asignar presupuesto a las categorías

Ver gastos de familia

Ver gastos familia	
Filtrar	Todo ▾
	Todo el tiempo ▾
\$ 11	Servicios asdsad 2022-06-23T04:52:50-05:00
\$ 29.69	Transporte gasto de prueba 2022-05-16T19:52:00-05:00
\$ 4	Salud medicina 2022-06-07T15:44:00-05:00
\$ 230.49	Educación Colegiatura 2022-06-21T11:07:00-05:00
\$ 23.5	Servicios gasto desde celular 2022-05-14T15:20:00-05:00
\$ 12	Educación 2022-06-21T15:30:00-05:00

Similar a ver los gastos personales, pero enfocado en los gastos globales de todos los miembros. Incluye las mismas opciones de filtrado.

Reportes gráficos



Reportes gráficos comparativos de todas las categorías de gastos.

1. Gráfica actual.
2. Seleccionar categoría a graficar.

Predicción de gastos



Visualizar la predicción de sus gastos en el futuro e inspeccionar los gastos de interés.

1. Gráfica de predicción.
2. Información del gasto seleccionado en la gráfica.

Anexo 8: MANUAL TÉCNICO DEL SISTEMA



MANUAL TÉCNICO DEL SISTEMA

Elaborado por:
Martín Bojorque
Iván Ulloa

RESUMEN

Se detallan aspectos técnicos del desarrollo de la aplicación Gestionar Gastos esclareciendo aspectos como la estructura del sistemas, comportamiento, funcionalidades, plataformas y herramientas usadas. El manual contiene información de teórica, modelado, aspectos técnicos específicos y requerimientos del sistema. Aspectos técnicos y teóricos abordados: Node.js y algunas librerías usadas, Ionic Framework, Google Firebase y sus funciones.

OBJETIVO

Dar a conocer los conocimientos generales empleados en el desarrollo de la aplicación Gestionar Gastos en conjunto con descripciones de la estructura y funcionalidad de la aplicación dando un entendimiento global del sistema al lector.

INTRODUCCIÓN

El objetivo de este documento es brindar una guía técnica de la Aplicación Gestionar Gastos a futuros desarrolladores que busquen administrar, editar, configurar o dar mantenimiento al sistema apropiadamente. A continuación, se detallan las secciones a tratar:

- **ASPECTOS TEÓRICOS:** Dar a conocer definiciones, conceptos y conocimiento básico teórico acerca de los componentes que conforman el desarrollo del aplicativo. Estos conocimientos incluyen plataformas, editores, librerías y herramientas utilizadas en la construcción del sistema.
- **DIAGRAMAS DE MODELADO:** Algunos diagramas que describen el comportamiento y funcionamiento de la aplicación.
- **ASPECTOS TÉCNICOS DEL DESARROLLO:** Guías que busca instruir al lector sobre los componentes del sistema. Los temas incluyen almacenamiento de datos, configuración de plataformas de desarrollo, estructura de la aplicación y recomendaciones de uso.
- **REQUERIMIENTOS DEL SISTEMA:** Detalla los requerimientos de software y hardware de las herramientas de desarrollo y el aplicativo.

1. ASPECTOS TEÓRICOS DEL DESARROLLO

Repositorio: https://github.com/IvanUlloa098/Trabajo_de_Titulacion-App_de_gastos_familiares

La aplicación Gestionar Gastos busca dar al usuario una herramienta para llevar registro de sus gastos personales y de otros miembros de la familia. Esto combinado con una gestión de presupuestos, historiales de gastos, visualizaciones gráficas de los gastos y una predicción global de los gastos.

1.1. CONCEPTOS GENERALES

1.1.1. Apps de organización personal

Es una categoría de aplicaciones móviles que permiten mantener un registro de actividades de toda índole de esta manera gestionando actividades: diarias, semanales y anuales; con el objetivo de mantener una vida más organizada, controlar ciertos aspectos de manera rigurosa y/o generar buenos hábitos de comportamiento.

1.1.2. Analítica de datos

Busca analizar datos para obtener nuevo conocimiento que afecte la influencia que tiene sobre estos a aquel que los estudia. En el contexto de informática, análisis se refiere a la aplicación de sistemas informáticos al análisis de grandes volúmenes de datos para toma de decisiones (Runkler, 2020).

1.1.3. Inteligencia de negocios

La inteligencia empresarial es un término que se creó a mediados de la década de 1990 para describir la acción de tomar datos en su forma original y convertirlos en algo utilizable, lo cual puede ser la base de decisiones empresariales. Es un término que acoge a otras varias disciplinas de datos como: minería de datos, análisis estadístico, pronósticos y apoyo a la toma de decisiones (Pareek, 2006).

1.2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO

1.2.1. Node.js

Concebido como un entorno de código abierto y multiplataforma controlado por eventos en tiempo de ejecución del lado del servidor utilizando JavaScript, Node.js se enfoca en el desarrollo de aplicaciones escalables, permitiendo establecer y gestionar varias conexiones simultáneas de manera asíncrona, la comunicación es simplificada al no usar subprocesos y no depender de multi-threading para soportar la ejecución de la lógica de negocios.

1.2.2. Librerías adicionales utilizadas

Todas las librerías fueron obtenidas usando el gestor de paquetes de Node.js o también conocido como NPM.

- **regresions-js:** Es un módulo de JavaScript que contiene una colección de métodos lineales basados en mínimos cuadrados para análisis simple de datos. Contiene código para regresiones lineales, exponenciales, logarítmicas y polinomiales.
- **Chart.js:** Librería gratuita y de código abierto de JavaScript enfocada en la construcción de gráficos utilizando la HTML como base. Soporta varios tipos de gráficos como: Barras, área polar, burbuja, líneas, dispersión, etc.
- **firebase:** Librería que provee herramientas para incorporar todos los servicios e infraestructura de Google Firebase a tu proyecto.

1.2.3. Ionic Framework

De la documentación oficial proveída por Ionic se tiene que: *“Ionic es un kit de herramientas de código abierto para el desarrollo de apps móviles y de escritorio de alta calidad y desempeño utilizando tecnologías web HTML, CSS y JavaScript con integraciones de Frameworks como Angular, React y Vue”*. Caracterizada por permitir el desarrollo de aplicaciones como una página web estándar, siendo capaz de correr en varias plataformas incluidas Android y iOS, esto en conjunto con la disponibilidad de varios plugins para el uso de servicios extra y hardware específico dentro del dispositivo crean un entorno amigable de desarrollo que resulta en aplicaciones estéticamente agradables con alto nivel de personalización (Chaudhary, 2018). Requiere conocimientos previos de Angular (también permite el desarrollo en Vue o React).

1.2.4. Firebase

Firebase es una plataforma propietaria de Google que ofrece herramientas para el desarrollo de aplicaciones web y móviles, su oferta incluye infraestructura de backend completamente administrada, supervisión del rendimiento de aplicaciones, generación de estadísticas de los usuarios y uso de recursos. Las características que son ofrecidas pueden ser activadas o desactivadas de acuerdo con las necesidades de los desarrolladores y los precios varían dependiendo del uso de recursos y almacenamiento.

- **Cloud Functions para Firebase:** De acuerdo con la documentación oficial: “Es un framework sin servidores que te permite ejecutar de forma automática el código de backend

en respuesta a las solicitudes HTTPS. Tu código JavaScript o TypeScript se almacena en la nube de Google y se ejecuta en un entorno administrado”

- **Cloud Firestore:** De acuerdo con la documentación oficial: “Usa nuestra base de datos NoSQL flexible, escalable y en la nube a fin de almacenar y sincronizar datos para el desarrollo tanto del lado del cliente como del servidor”. Existe también la posibilidad de la implementación de una base de datos en tiempo real para mantener los datos sincronizados entre apps junto con soporte sin conexión.
- **Firebase Authentication:** De acuerdo con la documentación oficial: “Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y mucho más”

2. DIAGRAMAS DE MODELADO

2.1. ESQUEMA GENERAL DE LA APLICACIÓN

Visualización general de la estructura del sistema en relación con las funciones ofrecidas por Firebase.

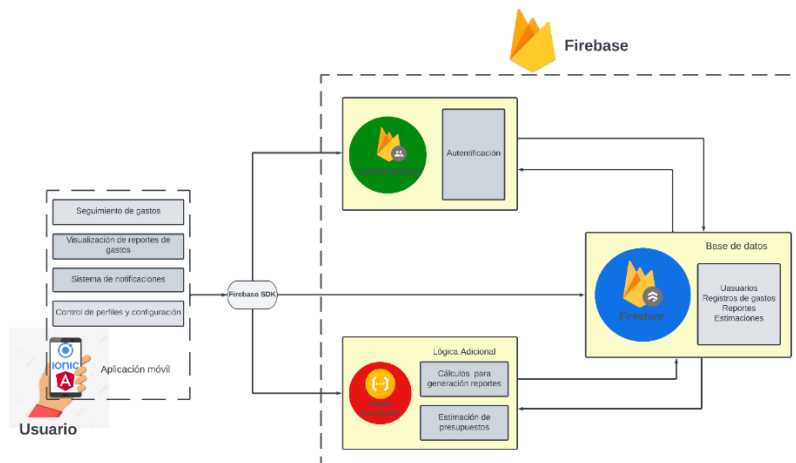


Figura 1. Esquema de la aplicación

2.2. FUNCIONALIDADES Y ROLES

Esquema de las funcionalidades ofrecidas dependiendo de acuerdo con los roles asignados. Roles existentes: Miembros de la familia y jefe del hogar.

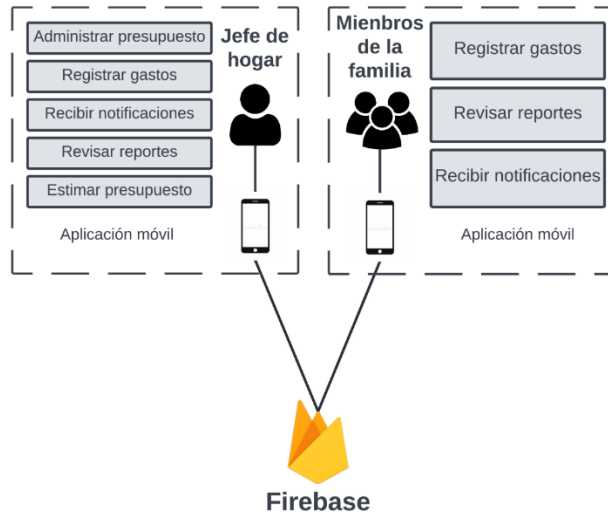


Figura 2. Esquema de funcionalidades

2.3. DICCIONARIO DE DATOS

Disposición de los datos guardados en la base de datos no relacional de Firestore. La información incluye las entidades (y los nombres de las colecciones en Firestore), los nombres de los atributos, el tipo de datos y una descripción.

Tabla 1. Diccionario de datos

Colección	Atributos	Tipo de dato	Descripción
<i>Usuario (users)</i>	uid	Cadena de texto	Hash como ID del documento
	email	Cadena de texto	Correo electrónico de la cuenta
	password	Cadena de texto	Contraseña de la cuenta
	role	Char	“U” para miembro de la familia “A” para jefe de hogar
	displayName	Cadena de texto	Nombre de propietario de la cuenta
	description	Cadena de texto	Mensaje de bienvenida de la cuenta
	photoURL	Cadena de texto	URL de referencia a Firebase Storage
	createdAt	Fecha	Fecha en la que fue creada la cuenta
	lastLogin	Fecha	Fecha de último ingreso a la cuenta
	provider	Cadena de texto	Tipo de inicio de sesión “gestion-gastos” para ingreso con correo y contraseña “google” inicio de sesión con Google
	id_familia	Numérico	Familia asignada a la cuenta
	active	Booleano	Cuenta activa o no
<i>Familia (families)</i>	id	Cadena de texto	Hash como ID del documento
	nombre	Cadena de texto	Nombre de la familia
	presupuesto_global	Numérico	Presupuesto global de la familia
	primer_dia_mes	Numérico	Día de inicio de mes

<i>Presupuesto</i> (presupuestos)	id	Cadena de texto	Hash como ID del documento
	cantidad	Númérico	Monto del presupuesto
	id_categoria	Cadena de texto	Categoría de gastos
	id_familia	Cadena de texto	Familia asignada al presupuesto
	fecha	Fecha	Fecha de modificación del presupuesto
<i>Gasto</i> (gastos)	id	Cadena de texto	Hash como ID del documento
	monto	Númérico	Monto del gasto
	descripcion	Cadena de texto	Razón del gasto
	fecha	Fecha	Fecha de realización del gasto
	id_categoria	Cadena de texto	Categoría de gastos
	id_usuario	Cadena de texto	ID de usuario del gasto
	Id_familia	Cadena de texto	Familia asidnada al gasto
<i>Categoría</i> (categories)	id	Cadena de texto	Hash como ID del documento
	nombre	Cadena de texto	Nombre de la categoría
	descripción	Cadena de texto	Descripción de la categoría

2.4. DIAGRAMA DE COMPONENTES

Esquema de los componentes (módulos) de la aplicación junto con sus relaciones de dependencia con las características del back-end (Firebase).

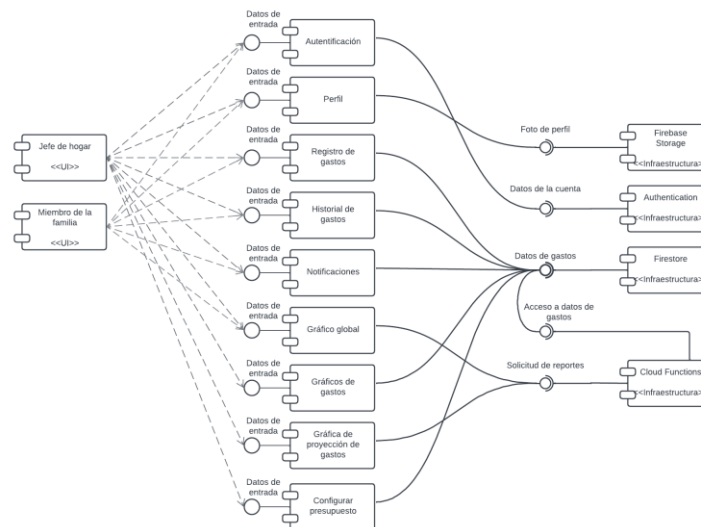


Figura 3. Diagrama de componentes

2.5. EESQUEMA DE FUNCIONALIDAD DE CLOUD FUNCTIONS

Describe brevemente el uso de Cloud Functions dentro de la aplicación de manera visual en donde el código desarrollado en el entorno de Node.js es desplegado en los servidores de Google y estos servicios son solicitados por la aplicación.

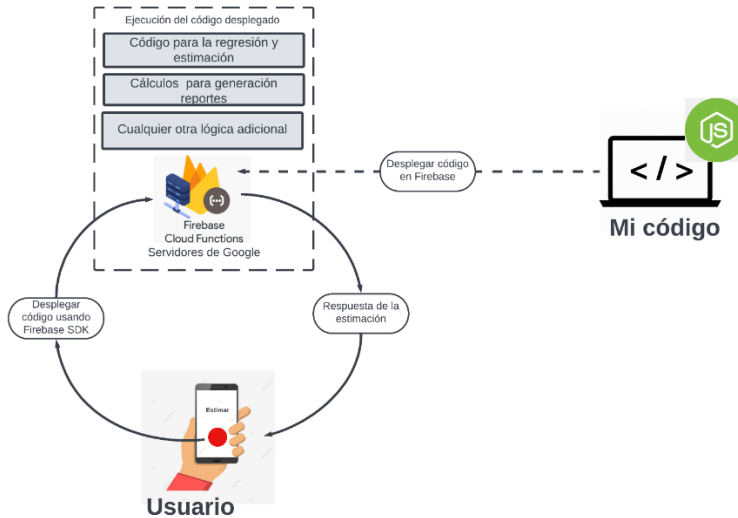


Figura 4. Despliegue de código para estimación de presupuesto

3. ASPECTOS TÉCNICOS DEL DESARROLLO

En esta siguiente sección se realizará una descripción detallada sobre los aspectos técnicos del aplicativo, estos están relacionados con la instalación de las herramientas necesarias para realizar modificaciones requeridas.

3.1. MODIFICACIÓN LOCAL

3.1.1. NodeJs

Se requiere de la instalación de Nodejs 16.13.1, se recomienda la instalación de esta versión en específico. (<https://nodejs.org/download/release/v16.13.1/>)

Archivo	Fecha	Tamaño
docs/	30-Nov-2021 19:33	-
win-x64/	30-Nov-2021 19:42	-
win-x86/	30-Nov-2021 19:41	-
SHASUMS256.txt	01-Dec-2021 16:13	3153
SHASUMS256.txt.asc	01-Dec-2021 16:13	3690
SHASUMS256.txt.sig	01-Dec-2021 16:13	310
node-v16.13.1-aix-ppc64.tar.gz	01-Dec-2021 11:51	43966837
node-v16.13.1-darwin-arm64.tar.gz	01-Dec-2021 11:33	28506601
node-v16.13.1-darwin-arm64.tar.xz	01-Dec-2021 11:33	18427632
node-v16.13.1-darwin-x64.tar.gz	01-Dec-2021 12:14	29938975
node-v16.13.1-darwin-x64.tar.xz	01-Dec-2021 12:15	20056120
node-v16.13.1-headers.tar.gz	01-Dec-2021 11:27	562423
node-v16.13.1-headers.tar.xz	01-Dec-2021 11:27	381248
node-v16.13.1-linux-arm64.tar.gz	01-Dec-2021 11:15	32385016
node-v16.13.1-linux-arm64.tar.xz	01-Dec-2021 11:16	20985584
node-v16.13.1-linux-armv7l.tar.gz	01-Dec-2021 12:31	29679419
node-v16.13.1-linux-armv7l.tar.xz	01-Dec-2021 12:32	17985684
node-v16.13.1-linux-ppc64le.tar.gz	01-Dec-2021 11:16	34444979
node-v16.13.1-linux-ppc64le.tar.xz	01-Dec-2021 11:18	22100280
node-v16.13.1-linux-s390x.tar.gz	01-Dec-2021 11:15	32655582
node-v16.13.1-linux-s390x.tar.xz	01-Dec-2021 11:16	20575440
node-v16.13.1-linux-x64.tar.gz	01-Dec-2021 11:29	32395116
node-v16.13.1-linux-x64.tar.xz	01-Dec-2021 11:30	21686428
node-v16.13.1-win-x64.7z	01-Dec-2021 11:40	16721509
node-v16.13.1-win-x64.zip	01-Dec-2021 11:40	25689871
node-v16.13.1-win-x86.7z	01-Dec-2021 11:38	15557278
node-v16.13.1-win-x86.zip	01-Dec-2021 11:38	24039967
node-v16.13.1-x64.msi	01-Dec-2021 11:40	28020736
node-v16.13.1-x86.msi	01-Dec-2021 11:38	26267648
node-v16.13.1.pkg	01-Dec-2021 11:48	55634645
node-v16.13.1.tar.gz	01-Dec-2021 11:19	64385859
node-v16.13.1.tar.xz	01-Dec-2021 11:24	33907544

Mediante esta página web se obtiene el archivo de instalación correspondiente para cada sistema operativo que puede hacer uso de nodejs y la arquitectura necesaria (X86 o X64).

3.1.2. Angular

Adicionalmente para la modificación es necesario la instalación de Angular 13.1.2.

```
Angular CLI

angular CLI: 13.1.2
node: 16.13.1
Package Manager: npm 8.1.2
OS: win32 x64

angular:
..

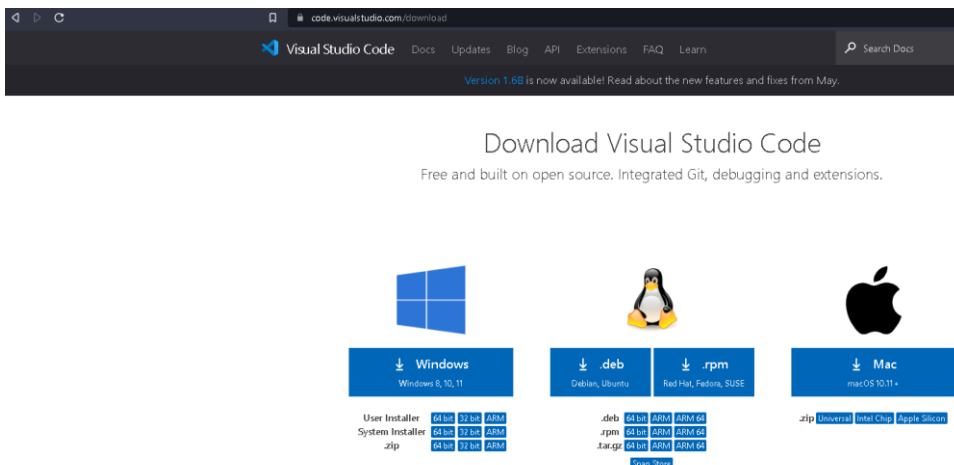
Package                                Version
-----
angular-devkit/architect                0.1301.2 (cli-only)
angular-devkit/core                      13.1.2 (cli-only)
angular-devkit/schematics                13.1.2 (cli-only)
schematics/angular                      13.1.2 (cli-only)
```

Esto se realiza desde la consola de comando, mediante el comando `npm install @angular/cli` y especificando la versión requerida.

3.1.3. Visual Studio Code

El editor de texto utilizado para el desarrollo del proyecto es Visual Studio Code. Para ello se pueden encontrar las diferentes versiones de este en la página web. <https://code.visualstudio.com/download>

Esta herramienta se puede utilizar en varios sistemas operativos.



3.1.4. Paquetes Adicionales y plugins

Para este paso es necesario haber descargado el proyecto, desde Github. Dentro de este se encuentra el archivo *package.json*, este archivo contiene todos los paquetes y plugin que se utilizaron para el proyecto. Ubicados en la carpeta “gestionar-gastos” desde la consola se utiliza el código `npm install`, este se encargará de instalar los paquetes descritos en este.

```

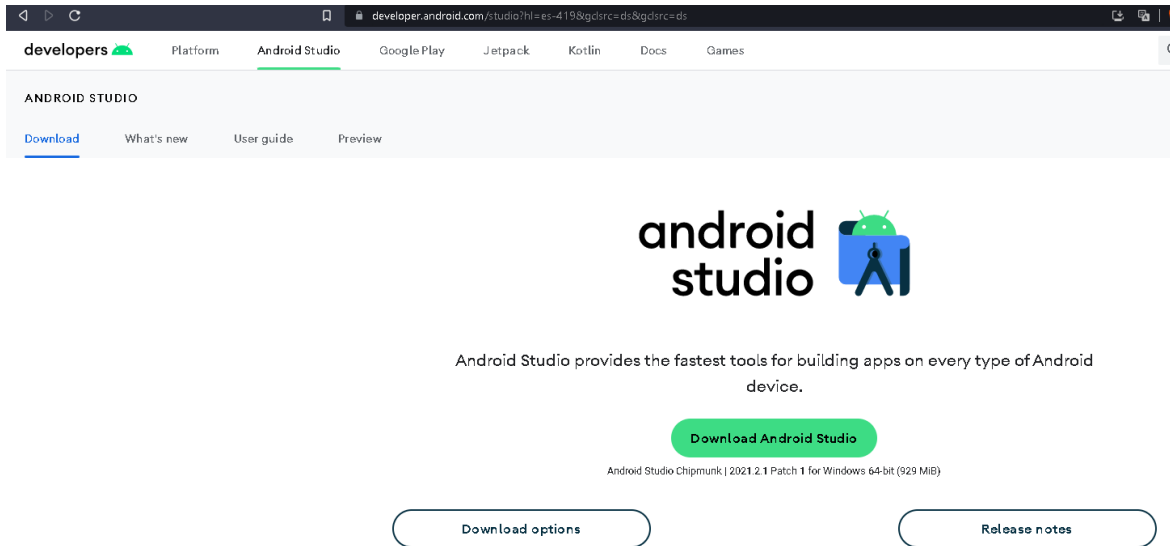
40 "cordova-plugin-local-notification": "^0.9.0-beta.2",
41 "date-fns": "^2.28.0",
42 "firebase": "^9.6.11",
43 "ng2-charts": "^3.0.11",
44 "rxjs": "~6.6.0",
45 "tslib": "^2.2.0",
46 "zone.js": "~0.11.4"
47 },
48 "devDependencies": {
49   "@angular-devkit/build-angular": "~13.2.3",
50   "@angular-eslint/builder": "^12.2.1",
51   "@angular-eslint/eslint-plugin": "~13.0.1",
52   "@angular-eslint/eslint-plugin-template": "~13.0.1",
53   "@angular-eslint/template-parser": "~13.0.1",
54   "@angular/cli": "~13.2.3",
55   "@angular/compiler": "~13.2.2",
56   "@angular/compiler-cli": "~13.2.2",
57   "@angular/language-service": "~13.2.2",
58   "@capacitor/cli": "^3.5.1",
59   "@ionic/angular-toolkit": "^6.0.0",
60   "@ionic/lab": "3.2.12",
61   "@types/jasmine": "~3.6.0",
62   "@types/jasminewd2": "~2.0.3",
63   "@types/node": "^12.11.1",
64   "@typescript-eslint/eslint-plugin": "5.3.0",
65   "@typescript-eslint/parser": "5.3.0",
66   "eslint": "^7.6.0",
67   "eslint-plugin-import": "2.22.1",
68   "eslint-plugin-jsdoc": "30.7.6",
69   "eslint-plugin-prefer-arrow": "1.2.2",
70   "jasmine-core": "~3.8.0",
71   "jasmine-spec-reporter": "~5.0.0",
72   "karma": "~6.3.2",
73   "karma-chrome-launcher": "~3.1.0",
74   "karma-coverage": "~2.0.3",
75   "karma-coverage-istanbul-reporter": "~3.0.2",
76   "karma-jasmine": "~4.0.0",
77   "karma-jasmine-html-reporter": "^1.5.0",
78   "protractor": "~7.0.0",
79   "ts-node": "~8.3.0",
80   "typescript": "~4.4.4"
81 },
82 "description": "An Ionic project"
83 }
84

```

3.1.5. Android Studio

Para la depuración y ejecución del proyecto en la plataforma Android es necesario la instalación de Android Studio y varias funcionalidades dentro de este.

Para comenzar la herramienta está disponible en <https://developer.android.com/studio?hl=es-419&gclid=ds&gclid=ds>

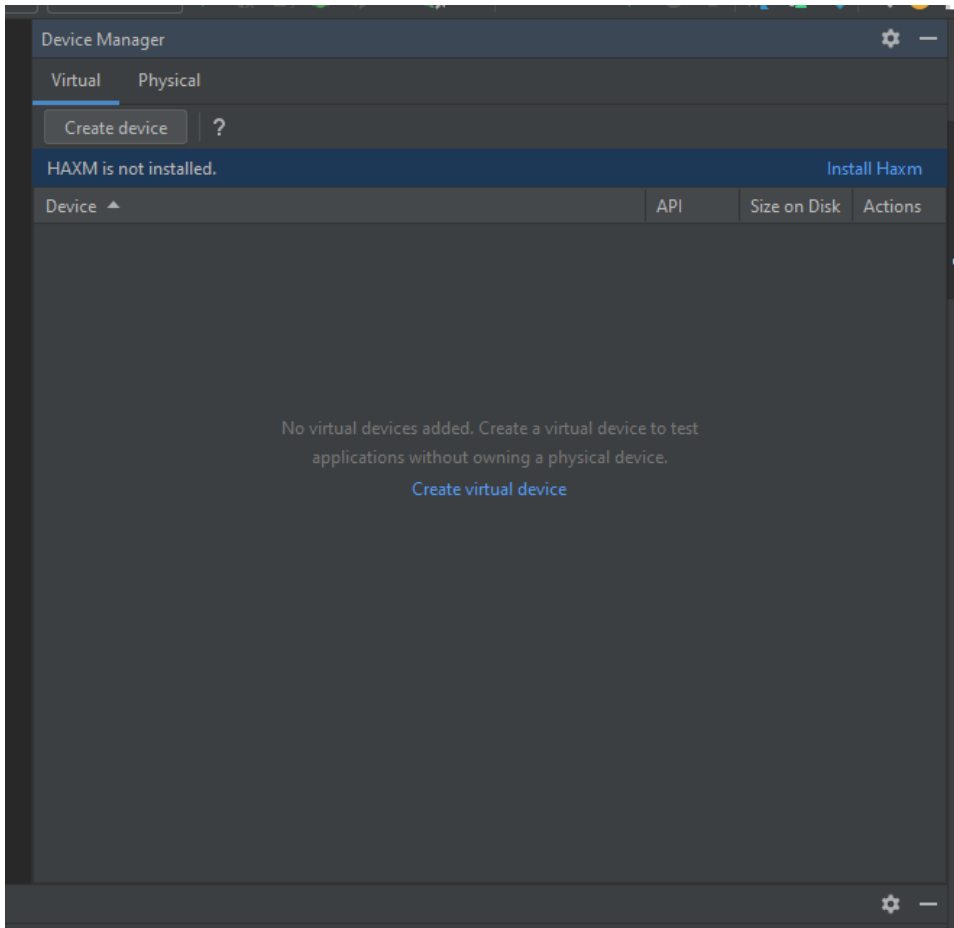


3.1.5.1. SDK

<input type="checkbox"/>	Android TiramisuPrivacySandbox Preview	TiramisuPrivacySandbox	3	Not installed
<input type="checkbox"/>	Android API 33	33	1	Not installed
<input type="checkbox"/>	Android API 32	32	1	Partially installed
<input type="checkbox"/>	Android 12.0 (S)	31	1	Partially installed
<input type="checkbox"/>	Android 11.0 (R)	30	3	Partially installed
<input type="checkbox"/>	Android 10.0 (Q)	29	5	Not installed
<input checked="" type="checkbox"/>	Android 9.0 (Pie)	28	6	Installed
<input type="checkbox"/>	Android 8.1 (Oreo)	27	3	Not installed
<input checked="" type="checkbox"/>	Android 8.0 (Oreo)	26	2	Installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Not installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Not installed
<input type="checkbox"/>	Android 4.4 (KitKat)	19	4	Not installed
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3	Not installed
<input type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3	Not installed

Para el proyecto se utilizaron los SDK, correspondientes a la imagen. Estos pueden ser modificados a discreción y consideración de la persona. Esto dependiendo de la versión de Android que se desee usar.

3.1.5.2. Device manager



El proyecto puede ser desplegado en dispositivos virtuales o físicos. En caso de tener un dispositivo virtual se deberá crear dicho dispositivo según las reglas del software.

En caso de tener un dispositivo físico sobre el cual desplegar el proyecto se deberá tener en cuenta la herramienta ADB y activar las opciones de desarrollador dentro del dispositivo.

3.1.6. IONIC

Este complemento se instalar en el punto 3.1.4 este framework es el utilizado para la realización del proyecto para diferentes plataformas. En caso de requerir arrancar el proyecto para navegadores se deberá utilizar el comando: `ionic serve`; en caso de que se busque desplegar en la plataforma Android se utilizara el comando: `ionic capacitor build android`.

3.1.7. Capacitor

Este proyecto utiliza capacitor para la integración de las plataformas móviles. Sobre esta herramienta se despliegan los diferentes plugins que permitirán las interacciones con los dispositivos móviles. Esto está integrado en el proyecto y se instalar mediante el punto 3.1.4

4. REQUERIMIENTOS DEL SISTEMA

Requisitos del sistema para Android Studio

A continuación, se describen los requisitos del sistema para Android Studio en Windows.

- Microsoft® Windows® 8/10/11 de 64 bits
- Arquitectura de CPU x86_64; procesador Intel Core de segunda generación o posterior, o CPU AMD compatible con un [hipervisor de Windows](#)
- 8 GB de RAM o más
- 8 GB de espacio disponible en el disco como mínimo (IDE + SDK de Android + Android Emulator)
- Resolución de pantalla mínima de 1280 × 800

Dado que el software con más requisitos es Android Studio es el que tiene requisitos más definidos se tomará como base para las recomendaciones sobre el sistema sobre el cual se utilizará. La imagen anterior desvela los requisitos de sistema pedidos por este programa para su funcionamiento. Estos y más requerimientos o consejos se encuentran en: <https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419#1>

5. BIBLIOGRAFÍA

Tully, C., & Alfaraz, C. (2017). Youth and mobility: The lifestyle of the new generation as an indicator of a multi-local everyday life. *Applied Mobilities*, 2(2), 182-198.

Runkler, T. A. (2020). *Data analytics*. Springer Fachmedien Wiesbaden.

Pareek, D. (2006). *Business Intelligence for telecommunications*. Auerbach Publications.

Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80-83.

Chaudhary, P. (2018). Ionic Framework. *International Research Journal of Engineering and Technology*, 5(05), 3181-3185.