

**Aplicativo móvil para el control de activos de pequeños sistemas de producción agrícola
soportados en tecnologías de libre distribución mediante dinámicas de gamificación**

Isabel Cristina Gómez Balvin y Marlon Maurice Mora Carvajal

Trabajo de Grado para optar al título de Ingeniería de Sistemas

Directora

Lola Xiomara Bautista Rozo

PhD. en Ciencias de la Computación

Codirector

Urbano Eliécer Gómez Prada

PhD. en Tecnología Educativa

Universidad Industrial De Santander

Facultad de Ingerirías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2024

Contenido

	Pág.
Introducción	16
1. Planteamiento y justificación del problema.....	17
2. Objetivos.....	18
2.1 Objetivo general.....	18
2.2 Objetivos específicos	18
3. Marco de referencia	19
3.1 Marco conceptual.....	19
3.1.1 Modelo por niveles	19
3.1.2 Tecnología de la información	20
3.1.3 Gamificación.....	20
3.1.4 Pequeña producción agrícola.....	21
3.1.5 Metodologías ágiles	21
3.2 Marco contextual	21
3.2.1 Activo fijo agrícola	22
3.2.2 Control de activos agrícolas en Colombia	22
3.2.3 Aplicativo móvil	22
3.2.4 Tecnologías de libre distribución.....	23
3.3 Marco tecnológico	23
3.3.1 Flutter.....	24
3.3.2 Dart	24

3.3.3 Google Firebase	25
3.3.4 Cloud Firestore.....	26
3.3.5 Firebase Authentication	26
3.3.6 Cloud Storage.....	27
3.3.7 Android Studio.....	27
3.3.8 AVD Manager.....	28
3.3.9 Patrón provider.....	28
3.3.10 Bloc	29
3.3.11 Arquitectura limpia	29
3.3.12 Trello.....	30
3.3.13 SqfLite.....	31
3.4 Antecedentes	31
3.4.1 Aplicativo para el control de activos tecnológicos en una organización	31
3.4.2 Sistema automatizado para el control de activos fijos tangibles.....	32
3.4.3 Aplicación basada en gamificación para apoyar la implementación del sistema de gestión de activos ISO 55000 en Codensa	32
3.5 Marco metodológico	33
3.5.1 Metodología de prototipos	33
3.5.2 Investigación previa y análisis	35
3.5.3 Diseño primordial rápido	35
3.5.4 Construcción del prototipo.....	36
3.5.5 Evaluación.....	36
3.5.6 Mejoras del prototipo.....	37

3.5.7 Implementación.....	37
3.5.8 Metodología Kanban.....	37
4. Desarrollo.....	38
4.1 Requerimientos	38
4.1.1 Requerimientos funcionales.....	39
4.1.2 Requerimientos no Funcionales	40
4.2 Plan de desarrollo.....	41
4.2.1 Tablero de desarrollo	41
4.2.2 Tablero de trabajo de grado	43
4.3 Diseño	46
4.3.1 Diagrama de caso de uso general.....	46
4.3.2 Diagramas de actividades	48
4.3.3 Diagrama de clases	51
4.3.4 Arquitectura de Software	53
4.3.5 Mockups.....	54
4.4 Implementación.....	54
4.5 Prototipo.....	63
4.5.1 Vistas de usuario definitivas para la aplicación.....	63
4.6 Evaluación del funcionamiento del aplicativo.....	64
4.6.1 Pruebas unitarias	64
4.6.2 Pruebas de compatibilidad	64
4.6.3 Prueba de aceptación.....	65
5. Conclusiones.....	65

6. Recomendaciones	67
Referencias Bibliográficas	69
Apéndices.....	73

Lista de Tablas

	Pág.
Tabla 1 Listado de Requerimientos no funcionales.....	39
Tabla 2 Requerimientos no Funcionales.....	40
Tabla 3. Caso de uso registrar usuario #1.....	73
Tabla 4 Caso de uso Registrar Usuario #2.....	74
Tabla 5. Caso de uso Registrar Usuario #3.....	74
Tabla 6. Caso de uso Iniciar Sesión #1.....	75
Tabla 7 Caso de uso Iniciar Sesión #2.....	75
Tabla 8 Caso de uso Iniciar Sesión #3.....	76
Tabla 9 Caso de uso Cerrar sesión.....	76
Tabla 10. Caso de uso Agregar activo #1.....	77
Tabla 11 Caso de uso Agregar activo #2.....	77
Tabla 12 Editar datos del activo.....	78
Tabla 13 Caso de uso Eliminar activo.....	78
Tabla 14. Caso de uso Registrar ingresos por activos.....	79
Tabla 15 Caso de uso Registrar gastos por activo.....	80
Tabla 16 Generar un informe de activos.....	80
Tabla 17. Caso de uso descargar informe de activos.....	81
Tabla 18. Mostrar consejos.....	82
Tabla 19. Visualizar las misiones.....	82
Tabla 20. Visualizar las estadísticas de los activos.....	83

Tabla 21. Comparar activos.	83
Tabla 22. Editar datos del usuario.....	84
Tabla 23 Pruebas de compatibilidad.....	91

Lista de Figuras

	Pág.
Figura 1. Diagrama de la arquitectura limpia	30
Figura 2. Esquema de metodología prototipado incremental del proyecto	34
Figura 3 Espacio de trabajo en Trello	41
Figura 4. Tablero de la metodología Kanban en el inicio del proyecto	42
Figura 5. Tablero de la metodología Kanban en el desarrollo del proyecto	42
Figura 6 Tablero de la metodología Kanban en la finalización del proyecto	43
Figura 7. Tablero del trabajo de grado en el inicio del proyecto	44
Figura 8. Tablero del trabajo de grado durante el desarrollo del proyecto	44
Figura 9. Tablero de trabajo de grado en la finalización del proyecto	45
Figura 10. Pantalla principal de documentación en Notion.....	46
Figura 11. Diagrama de caso de uso general 1	47
Figura 12. Diagrama de caso de uso general 2	48
Figura 13. Diagrama de actividad inicio de sesión.	49
Figura 14. Diagrama de actividad creación de nuevo gasto/ingreso	50
Figura 15. Diagrama de actividad creación de un logro	51
Figura 16. Diagrama de clases	52
Figura 17. Diagrama de arquitectura del proyecto.....	54
Figura 18. Archivo main.dart de la aplicación.....	55
Figura 19. Archivo my app.dart.....	56
Figura 20. Jerarquía de carpetas siguiendo arquitectura limpia.....	57

Figura 21 Modelos implementados en el aplicativo	58
Figura 22 Implementación de la función verificar misión.....	59
Figura 23. Implementación de eventos principales en toda la aplicación 1.....	60
Figura 24. Archivo layoutEvent.dart 2	61
Figura 25. Consola de Firebase authentication de Vivagro	62
Figura 26. Consola de Cloud Firestore usado en Vivagro	63
Figura 27 Diagrama de actividad Agregar Activo.....	85
Figura 28 Diagrama de actividad Cerrar Sesión.....	86
Figura 29 Diagrama de Actividad Editar Información del usuario.....	87
Figura 30 Diagrama de Actividad Descargar Historial.....	88
Figura 31 Diagrama de Actividad Visualizar Estadísticas.	89
Figura 32 Diagrama de Actividad Eliminar Activo.....	90
Figura 33. Pregunta #1, encuesta para los usuarios.	92
Figura 34 Pregunta #2, encuesta para los usuarios.	93
Figura 35 Pregunta #3, encuesta para los usuarios.	93
Figura 36 Pregunta #4, encuesta para los usuarios.	94
Figura 37 Pregunta #5, encuesta para los usuarios.	94
Figura 38 Pregunta #6, encuesta para los usuarios.	95
Figura 39 Pregunta #7, encuesta para los usuarios.	96
Figura 40. Pregunta #8, encuesta para los usuarios.	96
Figura 41 Pregunta #9, encuesta para los usuarios.	97
Figura 42 Pregunta #10, encuesta para los usuarios.	97
Figura 43 Pregunta #11, encuesta para los usuarios.	98

Figura 44 Pregunta #12, encuesta para los usuarios.	98
Figura 45 Pregunta #13, encuesta para los usuarios.	99
Figura 46 Pregunta #14, encuesta para los usuarios.	100
Figura 47 Wireframe Vista Inicio.	101
Figura 48 Wireframe Mostrar consejos.	102
Figura 49 Wireframe Vista activos.	103
Figura 50 Wireframe Vista perfil del activo.	104
Figura 51 Wireframe Vista Estadísticas.	105
Figura 52 Wireframe Vista Perfil de usuario.	106
Figura 53 Wireframe Vista Actualización de datos del usuario.	107
Figura 54 Wireframe Vista Terminos y Condiciones.	108
Figura 55 Wireframe Vista comunidad propuesta.	109
Figura 56 Wireframe Vista Historial de las acciones del usuario.	110
Figura 57 Wireframe Vista listado de misiones.	111
Figura 58 Wireframe Vista Gestión de informes de los activos del usuario.	112
Figura 59 Wireframe Vista Registro de gasto.	113
Figura 60 Wireframe Vista Registro de ingreso.	114
Figura 61 Wireframe Vista Ayuda.	115
Figura 62 Mockup Vista de Inicio.	116
Figura 63 Mockup Vista de Activos.	117
Figura 64 Mockup Vista comparación de activos.	118
Figura 65 Mockup Creación de activo.	119
Figura 66 Mockup Vista de Consejos.	120

Figura 67 Mockup de Bienvenida a la Aplicación.....	121
Figura 68 Mockup Inicio de Sesión.....	122
Figura 69 Mockup Crear Cuenta.....	123
Figura 70 Mockup Estadísticas.....	124
Figura 71 Mockup Actualización Información de Perfil.....	125
Figura 72 Mockup Historial.....	126
Figura 73 Mockup Informe de Activos.....	127
Figura 74 Mockup Nuevo Gasto.....	128
Figura 75 Mockup Nuevo Ingreso.....	129
Figura 76 Mockup Misiones.....	130
Figura 77 Mockup Información de Perfil.....	131
Figura 78 Mockup Términos y Condiciones.....	132
Figura 79 Mockup Información de Activo.....	133
Figura 80 Mockup Activos.....	134
Figura 81 Mockup Ayuda.....	135
Figura 82 Mockup Ayuda Cuenta.....	136
Figura 83 Mockup Configuración.....	137
Figura 84 Mockup NavBar y barras de búsqueda.....	138
Figura 85 Vista de Inicio.....	140
Figura 86 Vista de Registro.....	141
Figura 87 Vista Iniciar Sesión.....	142
Figura 88 Vista Restablecer contraseña.....	142
Figura 89 Vista ver Perfil.....	143

Figura 90 Vista actualizar Perfil y añadir imagen de Perfil.....	144
Figura 91 Vista de Estadísticas.....	145
Figura 92 Vista seleccionar activos a comparar.	146
Figura 93 Vista comparación de activos.....	147
Figura 94 Activo y perfil del Activo.....	148
Figura 95 Vista Menú.	149
Figura 96 Misiones del usuario.....	150
Figura 97 Vista final Gastos o Ingresos.....	151
Figura 98 Historial gastos e ingresos y Editar Gasto/Ingreso.....	152
Figura 99 Vista Historial.....	153
Figura 100 Vista Informe de Activos.....	154
Figura 101 Notificación y visualización del documento descargado.	155
Figura 102 Vista Ajustes.....	156
Figura 103 Vista Ayuda.....	157

Lista de Apéndices

	Pág.
Apéndice A. Casos de uso	73
Apéndice B. Diagramas de actividades	85
Apéndice C. Tablas de pruebas.....	91
Apéndice D. Mockups y Wireframes.....	101
Apéndice E. Vistas de la aplicación.....	139

Resumen

Título: Aplicativo Móvil para el Control de Activos para Pequeños Sistemas de Producción Agrícola Soportados en Tecnologías de Libre Distribución Mediante Dinámicas de Gamificación*

Autores: Marlon Maurice Mora Carvajal, Isabel Cristina Gómez Balvin**

Palabras Clave: Aplicación Móvil, gamificación, campesinos, activos, control de gastos e ingresos.

Descripción:

Objetivo general: Desarrollar un aplicativo móvil para el control de activos en pequeños sistemas de producción agrícola soportados en tecnologías de libre distribución

El sector agrícola en Colombia, se enfrenta a desafíos considerables, marcados por el acceso limitado a la tecnología y la gestión financiera escasa por parte de los campesinos. Esta situación evidencia la importancia de proporcionar apoyo y herramientas adecuadas para mejorar la administración de los activos agrícolas y promover el bienestar económico de los campesinos.

Para abordar estas necesidades, se plantea una aplicación móvil especializada que permita a los campesinos registrar y gestionar sus gastos e ingresos relacionados con sus activos agrícolas. Se empleará la metodología de gamificación para facilitar la adopción y el uso continuo de la aplicación, fomentando así una gestión financiera sobre sus activos de manera más efectiva y proactiva.

Aprovechar el potencial de la tecnología en el sector agrícola es crucial para mejorar la productividad y el desarrollo sostenible. Esta iniciativa busca contribuir a este objetivo al proporcionar a los campesinos las herramientas necesarias para tomar decisiones más acertadas sobre la gestión de sus activos.

* Trabajo de Grado

** Facultad de Ingeniería Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Directora: Lola Xiomara Bautista Rozo. PhD. en Ciencias de la Computación. Codirector: Urbano Eliécer Gómez Prada PhD. en Tecnología Educativa

Abstract

Title: Mobile Application for Asset Control for Small Agricultural Production Systems Supported by Open-Source Technologies Through Gamification Dynamics *

Author: Marlon Maurice Mora Carvajal, Isabel Cristina Gómez Balvin **

Key Words: Mobile Application, gamification, farmers, assets, expense, and income control.

Description:

General Objective: To develop a mobile application for asset control in small agricultural production systems supported by open-source technologies.

The agricultural sector in Colombia faces significant challenges, marked by limited access to technology and poor financial management by farmers. This situation highlights the importance of providing support and appropriate tools to improve the management of agricultural assets and promote the economic well-being of farmers.

To address these needs, a specialized mobile application is proposed that allows farmers to record and manage their expenses and income related to their agricultural assets. Gamification methodology will be employed to facilitate the adoption and continuous use of the application, thus promoting more effective and proactive financial management of their assets.

Harnessing the potential of technology in the agricultural sector is crucial for improving productivity and sustainable development. This initiative seeks to contribute to this goal by providing farmers with the necessary tools to make more informed decisions about the management of their assets.

* Trabajo de Grado

** Facultad de Ingeniería Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Directora: Lola Xiomara Bautista Roza. PhD. en Ciencias de la Computación. Codirector: Urbano Eliécer Gómez Prada PhD. en Tecnología Educativa

Introducción

Colombia posee un gran potencial para el uso de sus tierras, además existe un buen número de personas trabajando en las actividades agrícolas. No obstante, se ha enfrentado a grandes retos en los últimos años en lo que es la competitividad y productividad. Es importante mantener este sector porque representa una gran contribución al PIB y a la generación de empleo, además de las oportunidades y la disponibilidad de crecer sin dañar el medio ambiente (Montalvo, J., Barriga Ospina, et al., 2017).

Para mejorar la economía del sector agrícola, los campesinos deberían implementar el uso de las TIC, una poderosa herramienta que ayuda a tener un mejor acceso y organización a los agricultores (Sociedad de Agricultores de Colombia, 2019). Sin embargo, para fortalecer el uso de estas tecnologías, se debe hacer una labor de concientización en la importancia de su uso, debido a que los campesinos suelen desmotivarse en el uso de estas, por la creencia de que tienen muy poca utilidad.

Hay que tener en cuenta que en el sector agrícola se ha observado que el uso de un celular inteligente y las tabletas 38.4% y 6.4% respectivamente, consideradas TIC de alta complejidad y muy poca capacitación en el manejo de estas tecnologías en zonas rurales (Sociedad de Agricultores de Colombia, 2019).

Se planea desarrollar una herramienta que permita controlar los activos y que este control vaya ampliando su cobertura a medida que se va adquiriendo la capacidad del uso de la herramienta, de tal manera que los campesinos no tengan que utilizar todas las funcionalidades de esta.

Por tanto, para comprenderlo se propone una herramienta que le permita a los campesinos obtener información de cómo ellos están haciendo uso de sus recursos. Este proyecto tendrá la limitación de que debe funcionar en celulares de baja gama y de forma asíncrona.

1. Planteamiento y justificación del problema

En la actualidad, los agricultores tienen un limitado conocimiento respecto a administración, necesaria para sacar un mejor provecho a sus bienes. A su vez, la tecnología disponible no propone una variedad de soluciones tecnológicas adaptadas a la diversidad de las condiciones socioeconómicas y agroecológicas existentes en el país.

Los equipos electrónicos tales como los celulares inteligentes, computadores y tabletas son considerados tecnologías de alta complejidad; en estos equipos se observa un bajo aprovechamiento por parte del sector agrícola (Sociedad de Agricultores de Colombia, 2019).

Para comprender, se propone una herramienta que les permita a los campesinos obtener información de cómo ellos están haciendo uso de sus recursos, con el fin de mejorar la eficiencia económica de los sistemas agrícolas; utilizando la gamificación para afianzar esta tecnología.

Ante el reto del uso de herramientas tecnológicas en este sector, la gamificación se utilizará como método de enseñanza; conforme lo expresa (Aguilera Castillo, Fúquene Lozano & Ríos Pineda, 2014); esta es una estrategia que busca, a través de elementos que se asemejan a un juego, desarrollar las capacidades y habilidades del usuario, con el fin de ofrecer una fuente de aprendizaje motivante para el enseñado en el proceso educativo.

La gamificación será una herramienta ideal para llevar a cabo el proyecto propuesto y tener la oportunidad de mejorar la experiencia de usuario; sin embargo, de acuerdo con (Oliveira & Bittencourt, 2019), no es sencillo manejar ni determinar las técnicas de gamificación que estén acorde al aplicativo. Además, vale aclarar que esta herramienta per se es solo un complemento a las funcionalidades que tendrá el aplicativo.

Por tanto, se desea desarrollar un aplicativo móvil que facilite la adopción mediante una didáctica para su aprendizaje y uso de estas tecnologías, buscando así un mejor manejo de los recursos obtenidos en el sector rural.

2. Objetivos

2.1 Objetivo general

Desarrollar un aplicativo móvil para el control de activos en pequeños sistemas de producción agrícola soportados en tecnologías de libre distribución.

2.2 Objetivos específicos

Definir los requerimientos para un aplicativo móvil, que permita el manejo de los activos que se utilizan en una pequeña producción agrícola.

Diseñar las vistas de usuario y la estructura en UML usando casos de uso, clases, actividades, que permitan la comprensión del alcance del aplicativo móvil.

Implementar el aplicativo siguiendo tecnologías de libre distribución, tales como flutter y Dart.

Evaluar en un ambiente de pruebas el funcionamiento del aplicativo, para la validación del cumplimiento de los requerimientos planteados.

3. Marco de referencia

3.1 Marco conceptual

En este apartado, se establece y se delimita el ámbito conceptual de los principales conceptos que serán objeto de estudio e indagación en la investigación.

3.1.1 Modelo por niveles

El modelo de gamificación por niveles indica el progreso de un usuario dentro del juego, cuyo objetivo es mostrar un progreso al usuario en el sistema, lo que ofrece realimentar su avance, la motivación a progresar y el grado de finalización, lo que queda para llegar a la meta (educativa, s. f.). No obstante, se busca un modelo de gamificación, buscando crear la distinción entre un juego

serio y una aplicación gamificada, que usa elementos de un juego en otro contexto, y no un juego propiamente (Deterding, Dixon, Khaled, & et al., 2011).

3.1.2 Tecnología de la información

La tecnología de la información (TI) se refiere a un conjunto de herramientas, técnicas y recursos diseñados para la manipulación, almacenamiento y transmisión eficiente de información. Este término abarca tanto el hardware como el software, así como las herramientas destinadas a procesar datos. El propósito fundamental de la TI es mejorar la gestión y el flujo de información dentro de organizaciones y sociedades. (Tecnologías de la Información y las Comunicaciones (TIC), s. f.)

3.1.3 Gamificación

El término gamificación es definido como “llevar las distintas mecánicas y técnicas que se encuentran en los juegos a contextos que no tienen nada que ver con juegos para tratar de resolver problemas reales” (Borrás Gené, 2015), con el fin de aumentar el compromiso y motivar cierto comportamiento.

En la vida real hay muchas cosas que pueden tomar tiempo porque son casi aburridas, o en su defecto no generan ese grado de atención para lograrlas con mayor facilidad, por eso se usa la gamificación; esta metodología ayuda a ser consciente de la motivación al hacer una tarea seria. A su vez fortalece la voluntad en la que se hacen estas actividades serias, puesto que no dejan de ser una molestia y se convierten “divertidas” (Borrás Gené, 2015).

3.1.4 Pequeña producción agrícola

Se concibe como pequeño productor a quienes poseen un área de producción pequeña o que la escala de producción es menor, en muchos casos los pequeños productores son familiares. Es necesario reconocerlos como campesinos, que va mucho más allá de la mentalidad actual del mercado. Los campesinos poseen su propio estilo de vida, organización, y llevan consigo un pasado que no debe ser marcado de otra forma porque la sociedad así lo designe. (Herrera, Méndez, Tobón & Sierra, 2016)

3.1.5 Metodologías ágiles

Actualmente se busca que las herramientas y los sistemas de información crezcan a pasos agigantados en un menor tiempo. Esto conlleva la necesidad de utilizar las metodologías de desarrollo ágil. La metodología caracteriza la flexibilidad en su desarrollo y la comunicación constante con el usuario, convirtiéndose en una metodología adaptable a los cambios (Vite Cevallos, Montero, & Cuesta, 2018).

3.2 Marco contextual

Se establece el contexto en el cual se desenvuelve el proyecto de investigación, delimitando el entorno específico en el que se llevará a cabo el trabajo de grado. Se explorarán los temas pertinentes para una comprensión más profunda y detallada del entorno, lo que contribuirá a una mejor comprensión y análisis de los resultados obtenidos.

3.2.1 Activo fijo agrícola

Los activos fijos agrícolas son bienes tangibles usados en la producción y mantenimiento de las actividades agrícolas, tales como herramientas, maquinaria y equipos especializados. Estos activos representan inversiones importantes para los agricultores y son fundamentales para la eficiencia y productividad de las operaciones agrícolas.

3.2.2 Control de activos agrícolas en Colombia

Consiste en la actividad de cultivar o sembrar en la tierra con la finalidad de cosechar vegetales para la alimentación humana, esta actividad está dada por la pre-labranza y la labranza de la tierra. Colombia es uno de los países con mayor potencial productivo, según el DANE se cuenta con 43.1 millón de hectáreas dedicadas a la agricultura, el crecimiento en este sector es bastante amplio, para el 2016 la producción creció en un 14% (DANE, 2014).

3.2.3 Aplicativo móvil

El concepto de aplicación móvil está relacionado a una aplicación diseñada para ser ejecutada en un dispositivo móvil, siendo un teléfono inteligente o una tableta. Requiere en gran medida de un almacenamiento local y algunas veces de un almacenamiento en la nube. Cada funcionalidad del software es limitada y no es un sistema de software integrados, dado que cada aplicación puede tener una sola función, como un juego, una calculadora o un navegador web móvil (Luis, 2021).

3.2.4 Tecnologías de libre distribución

Las tecnologías de libre distribución o software de código abierto son todos aquellos que son modificables según las necesidades del usuario, sin ningún tipo de carga de violación de licencia. Siendo el código fuente del software disponible con licencias de código abierto. Hay que tener en cuenta que las tecnologías de libre distribución difieren en ciertos aspectos del software libre, puesto que las licencias que manejan los tipos de software de código abierto se centran en la disponibilidad del código fuente y la capacidad de modificarlo y compartirlo, a diferencia del software libre, en el que se centran en la libertad del usuario de utilizar el programa, modificarlo e inclusive mejorarlo y compartirlo (Alonso, 2021). **Ej.:** (PHP, Flutter, Python, Perl).

Un claro ejemplo de este concepto es el lenguaje de alto nivel Python que, al ser una tecnología de libre distribución, se han generado cientos de librerías hechas por la comunidad sin restricciones para divulgarlos.

3.3 Marco tecnológico

A continuación, se expondrán las diversas herramientas, plataformas y tecnologías específicas que serán utilizadas en el desarrollo del trabajo de grado, así como su relevancia y aplicación en el contexto del proyecto.

3.3.1 Flutter

Flutter es SDK de código abierto creado por Google que permite la construcción de aplicaciones multiplataforma compiladas nativamente, permitiendo desarrollar para Android, iOS, web y escritorio, con una única base de código. Además de ser uno de los framework para desarrollo de aplicaciones más populares, siendo la base para cerca de 400.000 aplicaciones.

La elección de Flutter es debido a múltiples ventajas que ofrece, como el hot reload una característica muy destacable, debido a que flutter posee una compilación rápida permite a los desarrolladores visualizar instantáneamente los cambios realizados, sin la necesidad de reiniciar la aplicación de forma recurrente. También la gama de widget que ofrece es amplia y personalizable si así se desea, por lo que reduce el tiempo invertido en su diseño. Aunque trabajar con Flutter puede presentar un desafío inicial debido a su novedad, ofrece ventajas significativas.

3.3.2 Dart

Es un lenguaje de programación orientado a objetos de código abierto, diseñado para el desarrollo de aplicaciones tanto para el lado del cliente como para el lado del servidor. Tiene un diseño que lo hace cómodo para los desarrolladores, ofreciendo una sintaxis clara y concisa. (Dart overview, s. f.)

Además, Dart cuenta con un sistema de gestión de paquetes robusto y una biblioteca estándar rica en funcionalidades, lo que simplifica el desarrollo de aplicaciones complejas y fomenta la reutilización de código. Su versatilidad y su sólido conjunto de herramientas lo convierten en una opción atractiva para los desarrolladores que buscan construir aplicaciones

modernas y escalables en diferentes entornos. Es el lenguaje que usa flutter, su rasgo de virtual machine es el que permite el hot reload que tanto destaca flutter.

3.3.3 Google Firebase

Firestore es una plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google. Ofrece una amplia gama de servicios que facilitan el desarrollo, la implementación y la gestión de aplicaciones, lo que permite a los desarrolladores crear aplicaciones de alta calidad de manera rápida y eficiente. (Firestore, s. f.) Una de las principales razones para elegir Firestore es su excelente compatibilidad e integración con Flutter, lo que permite a los desarrolladores aprovechar al máximo las capacidades de ambas plataformas para crear aplicaciones multiplataforma de manera eficiente y efectiva.

Además, Firestore ofrece un plan gratuito que es ideal para proyectos pequeños o en etapas iniciales de desarrollo, lo que permite a los desarrolladores comenzar sin costos iniciales significativos. La plataforma también ofrece una variedad de servicios esenciales, como autenticación de usuarios, almacenamiento en la nube, bases de datos en tiempo real y análisis de rendimiento, lo que simplifica la gestión de varios aspectos del proyecto y permite a los desarrolladores tener todo en un solo lugar, facilitando el desarrollo y la implementación de aplicaciones.

3.3.4 Cloud Firestore

Firestore es una base de datos potente y adaptable, integrada dentro de la plataforma Firebase y Google Cloud. Este servicio ofrece una solución completa para almacenar y gestionar datos en aplicaciones móviles y web, con características adecuadas para proyectos de diversas escalas y complejidades. (Cloud Firestore, s. f.) Firestore utiliza un modelo de datos flexible y jerárquico, permitiendo a los desarrolladores organizar la información en documentos y colecciones de una manera que se adapte a las necesidades específicas de cada proyecto.

Se escogió como base de datos porque una de las principales fortalezas de Firestore radica en su capacidad para mantener los datos sincronizados en tiempo real entre las diferentes instancias de la aplicación, permitiendo una experiencia fluida y coherente para los usuarios finales. Además, ofrece soporte para operaciones sin conexión, lo que garantiza que las aplicaciones puedan funcionar eficazmente incluso con conexión limitada.

3.3.5 Firebase Authentication

Es un servicio de autenticación proporcionado por Firebase, una plataforma de desarrollo de aplicaciones móviles y web de Google. Este servicio permite a los desarrolladores integrar fácilmente la autenticación de usuarios en sus aplicaciones, proporcionando una manera segura y confiable de gestionar la identidad de los usuarios. Ofrece soporte para métodos de autenticación, incluyendo el correo electrónico y la contraseña, redes sociales como Google, Facebook y Twitter, y proveedores de identidad federados como Apple y GitHub. Esto brindando flexibilidad necesaria

para adaptarse a las preferencias de autenticación de los usuarios y proporcionar una experiencia de inicio de sesión conveniente y sin fricciones. (Firebase Authentication, s. f.)

En nuestro proyecto, utilizamos Firebase Authentication para garantizar un acceso seguro y confiable a nuestra aplicación, brindando a nuestros usuarios una experiencia de inicio de sesión fluida y segura.

3.3.6 Cloud Storage

Firebase Storage es un servicio de almacenamiento en la nube que forma parte de la plataforma Firebase de Google. Este servicio permite a los desarrolladores almacenar y compartir archivos de manera segura y confiable, ofreciendo un almacenamiento escalable y rentable para aplicaciones móviles y web. Firebase Storage ofrece características avanzadas de seguridad y control de acceso, como políticas de acceso basadas en roles y firmas de URL, que garantizan que los archivos estén protegidos contra accesos no autorizados. El SDK de Firebase para Cloud Storage, los desarrolladores pueden cargar y descargar archivos directamente desde los clientes, lo que garantiza una experiencia fluida para los usuarios, incluso en condiciones de conectividad deficientes, mejorando la eficiencia de la aplicación. (Cloud Storage, s. f.)

3.3.7 Android Studio

Es el ambiente de desarrollo oficial proporcionado por Google para la creación de aplicaciones móviles Android. Los desarrolladores pueden crear aplicaciones Android de alta calidad de manera eficiente y productiva. El IDE está diseñado para ser fácil de usar, con una

interfaz de usuario intuitiva y una amplia documentación y recursos de ayuda disponibles. (Android Developers, s. f.) Elegimos Android Studio como herramienta de desarrollo principal por su versatilidad y facilidad de uso. La capacidad de realizar pruebas tanto con emuladores como en dispositivos físicos, combinada con su interfaz intuitiva y su integración con otras herramientas de desarrollo de Google, lo convierte en la elección ideal para crear aplicaciones Android de alta calidad

3.3.8 AVD Manager

AVD Manager, abreviatura de Android Virtual Device Manager, es una herramienta esencial integrada en Android Studio que permite a los desarrolladores crear y gestionar emuladores de dispositivos Android para realizar pruebas de sus aplicaciones. Con AVD Manager, los usuarios pueden configurar emuladores con diferentes especificaciones de hardware y versiones de Android para simular una amplia variedad de dispositivos móviles. Para realizar el proyecto se hizo uso del emulador Pixel XL Api 27 para tener una visualización del código en dispositivos de baja resolución. (Android Developers, s. f.)

3.3.9 Patrón provider

Es una técnica de gestión de estado comúnmente utilizada en el desarrollo de aplicaciones Flutter. Proporciona una forma eficaz de administrar y compartir el estado entre los widgets de la aplicación, permitiendo una actualización rápida y eficiente de la interfaz de usuario en respuesta

a cambios en los datos. Este patrón se basa en la idea de proveer un objeto de estado centralizado que pueden acceder varios widgets en diferentes partes de la aplicación, permitiendo escribir código modular y limpio. (Provider, s. f.)

3.3.10 Bloc

El patrón BLoC, o Business Logic Component, es una arquitectura de gestión de estado ampliamente utilizada en el desarrollo de aplicaciones Flutter. Este patrón separa claramente la lógica de negocio de la interfaz de usuario, lo que facilita la reutilización del código y la mantenibilidad de la aplicación. “Bloc (Business Logic Component) es un componente que sirve de intermediario entre la vista y nuestro modelo de datos. Es similar al concepto de ViewModel (de la arquitectura MVVM), ya que nos facilita gestionar de una forma ordenada los cambios de estado que se producen en el modelo de datos de nuestra aplicación para que, de forma reactiva, la interfaz refleje los cambios necesarios. Se basa en el patrón observer, siguiendo la mecánica de escuchar eventos, y transformarlos en nuevos estados.” (Luis Barroso, 2023)

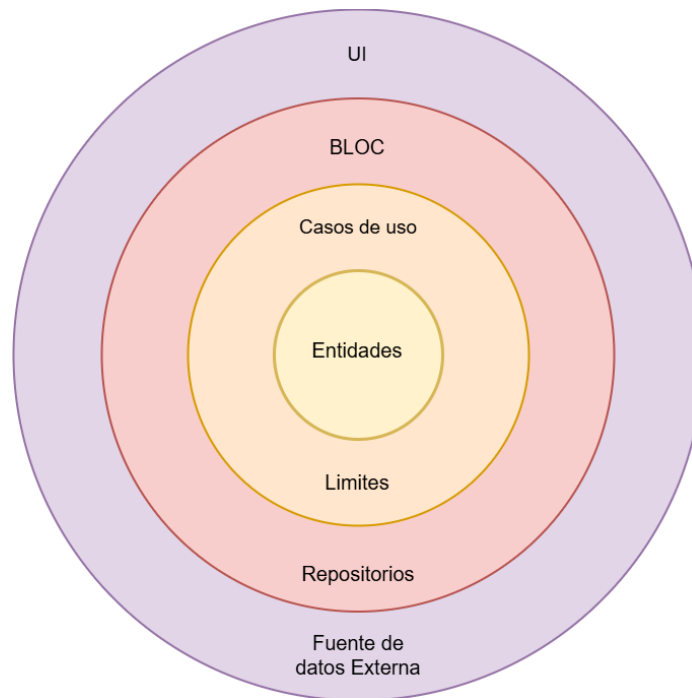
3.3.11 Arquitectura limpia

La arquitectura limpia es un enfoque de diseño de software propuesto por Robert Cecil Martin que promueve la separación clara de responsabilidades y el modularidad del código. Este enfoque se basa en el principio de que el código debe ser fácil de entender, mantener y probar. La arquitectura limpia propone una estructura de capas bien definida, donde cada capa tiene un propósito específico y se comunica con las capas adyacentes a través de interfaces claras y bien

definidas. La idea principal es minimizar las dependencias y el código sea mantenible, para que las aplicaciones sean sólidas y de mayor calidad. (Barroso, L., s. f.)

Figura 1.

Diagrama de la arquitectura limpia



Nota. Adaptado de *One step ahead into methodologies* por M. Arjun, 2020.

3.3.12 Trello

Trello es una plataforma web que facilita la organización y gestión de proyectos o tareas colaborativas. (Trello, s. f.) Está basado en tableros, listas y tarjetas para representar las actividades y su flujo de trabajo en el proyecto. Los usuarios pueden crear tableros personalizados para diferentes proyectos, donde pueden agregar listas de tareas y tarjetas individuales que representan actividades específicas. Además, Trello permite asignar tareas a diferentes miembros del equipo,

establecer fechas límite, adjuntar archivos y comentar en las tarjetas para una comunicación efectiva. Se escogió esta herramienta para llevar a cabo las actividades del proyecto.

3.3.13 SqfLite

Sqflite es un plugin esencial para Flutter que simplifica la gestión de bases de datos SQLite, un motor integrado robusto y confiable. Este motor de bases de datos integrado es altamente fiable y opera de manera autónoma en sistemas operativos como iOS, Android y MacOS. Automáticamente gestiona las versiones de la base de datos al abrirse y ofrece métodos prácticos para ejecutar consultas de inserción, consulta, actualización y eliminación. Su implementación en segundo plano mejora el rendimiento tanto en iOS como en Android (Sqflite. s. f.).

3.4 Antecedentes

En esta sección se presenta una revisión de los proyectos y estudios anteriores relevantes para el desarrollo del presente proyecto.

3.4.1 Aplicativo para el control de activos tecnológicos en una organización

El artículo escrito en el año 2013 fue propuesto como un caso de estudio para el desarrollo y la implementación de una herramienta de gestión de recursos tecnológicos, y surge de la necesidad de implementar una herramienta que permita contribuir al mejoramiento de los procesos

a nivel organizacional. Desarrollando una herramienta bajo estándares y normas de la ingeniería de Sistemas, con aplicaciones a la ingeniería de procesos (Jaramillo& Correa-Henao, 2013).

Como resultado, los autores evidenciaron una mejoría en la toma de decisiones, ya que proporciona un reporte sobre el historial de órdenes de salida y entrada para elementos informáticos de la institución, y estadísticas detalladas de las razones de cada orden.

3.4.2 Sistema automatizado para el control de activos fijos tangibles

El objeto de este proyecto está dedicado a la realización de un sistema para automatizar el proceso de control de activos fijos tangibles en la facultad de ingeniería Eléctrica de la Universidad Central “Marta Abreu” de manera genérica, con el fin de que permitirles a los usuarios intercambiar información con la base de datos y chequear los activos fijos tangibles (Hernández, 2008).

Con este software los autores lograron mejorar la calidad del proceso de control de activos fijos, optimizando y minimizando los errores humanos a través de una estructura fácil e intuitiva que ayuda al usuario a manejar este software sin ningún problema mayor.

3.4.3 Aplicación basada en gamificación para apoyar la implementación del sistema de gestión de activos ISO 55000 en Codensa

En el año 2021 se presenta el desarrollo de una aplicación basada en gamificación para apoyar el sistema de gestión de activos ISO 55000 utilizando la aplicación Energy Assets planteada como una estrategia para introducir los conceptos sobre los sistemas de gestión de activos y

resolución de la comisión de regulación de energía y gas. Usando la metodología de gamificación a través de ocho niveles, los autores buscaron abordar los conceptos básicos de estos sistemas, su ciclo de vida y las disposiciones generales de la resolución (Salazar, Zapata-Madrigal & García Sierra, 2021).

La aplicación implementa acciones de retroalimentación ante una respuesta correcta e incorrecta, un ranking de posiciones según el porcentaje obtenido a lo largo del juego. Siendo esta una estrategia para divulgar de manera más didáctica el camino requerido para la certificación de la ISO 55000.

3.5 Marco metodológico

Se detallarán las metodologías empleadas para llevar a cabo la investigación y el desarrollo del proyecto. En este caso, se aplicaron las metodologías de prototipos y Kanban para gestionar y ejecutar las actividades de manera eficiente. La metodología de prototipos permitió iterar rápidamente sobre el diseño y la funcionalidad de la aplicación. Por otro lado, la metodología Kanban facilitó la organización y visualización del flujo de trabajo, priorizando tareas.

3.5.1 Metodología de prototipos

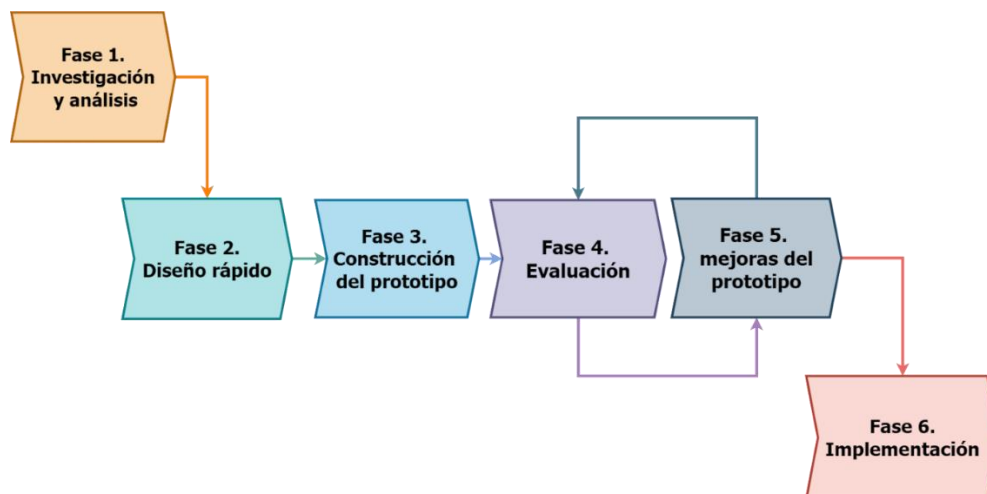
La metodología de prototipos es un enfoque de desarrollo de software que se centra en la creación rápida de versiones preliminares de un producto para obtener retroalimentación temprana y validar conceptos clave. En vez de esperar a que el producto se desarrolle para obtener comentarios de los usuarios. Los prototipos se utilizan para visualizar y probar diferentes ideas,

funcionalidades y diseños, lo que permite a los equipos identificar y corregir problemas potenciales antes de invertir recursos significativos en el desarrollo completo del producto. (Tecno digital, 2023)

Para desarrollar este proyecto se propone utilizar la metodología de prototipado o metodología evolutiva, para desarrollar o generar prototipos incrementales, debido a la facilidad que ofrece la metodología para detectar y solucionar errores. La metodología es flexible, permite realizar pruebas y modificando cada vez que sea necesario, además de brindar una forma de trabajar centrada en aspectos visuales más fáciles de detectar. El proyecto será desarrollado en 6 fases, como lo muestra la Figura 2.

Figura 2.

Esquema de metodología prototipado incremental del proyecto



Nota. Adaptado de *El patrón BLoC en Clean Architecture*, por J. S. Fernandez, 2020, XurxoDev

A continuación, un resumen de las fases del proyecto:

3.5.2 Investigación previa y análisis

Considerando las dificultades de los campesinos en el uso de tecnologías de alta complejidad, se busca investigar los diferentes métodos de gamificación, para encontrar las herramientas propicias que ayuden a satisfacer las falencias en el uso de las tecnologías y facilitar la administración de sus activos fijos. Además, se investigará la documentación del estado del arte requerida para profundizar en los conocimientos teórico-prácticos relacionado a los temas a tratar en el proyecto, como lo son la automatización y el control de activos fijos; asimismo el manejo de herramientas tecnológicas tales como Android Studio, Flutter y FireBase.

Definir los requerimientos tomando como base las necesidades que se desea resolver de los campesinos, esto para conocer aquellos requisitos que deberá cumplir el aplicativo. Finalizada esta parte se conocerán los requisitos de sistema operativo y hardware necesarios, a su vez se tendrá en cuenta otros trabajos similares realizados con anterioridad.

3.5.3 Diseño primordial rápido

En la segunda fase se diseñará un prototipo básico y rápido, se pretende hacer un bosquejo que ayude a desarrollar el aplicativo, en este punto el diseño aún no está completo. Con ello se definirá qué datos son necesarios, qué aspectos visuales se deben tener en cuenta para hacer mejor la experiencia, qué capacidad espacio y memoria son requeridas, además de evaluar la viabilidad del uso de las herramientas planteadas. Consiguiendo un diseño visual preliminar del aplicativo móvil. Además, se definirán casos de uso, diagramas, arquitectura, entre otras herramientas necesarias para el análisis de requisitos.

Con este diseño se planificará cuáles serían los diferentes tipos de riesgos que se podrían presentar, para luego idear una serie de posibles soluciones cuando se presenten.

3.5.4 Construcción del prototipo

Con base en la información obtenida de la investigación previa y los diseños del aplicativo, se procederá al desarrollo en conjunto realizando simulaciones a través de emuladoras que permitan determinar la eficiencia del software, integrando elementos de gamificación con una interfaz genérica para las vistas principales; corroborando el aporte hecho al cumplimiento de los objetivos previamente planteados. Por consiguiente, se definirá si es necesario volver a la fase de análisis o diseño de arquitectura, según sea el caso.

3.5.5 Evaluación

Finalizada la construcción del prototipo se hará la validación del prototipo. Se tomará como grupo de pruebas a 5 campesinos, que realizará el testeo del aplicativo, para posteriormente, hacer entrevistas y/o visitas de campo a los tester con el fin de saber sus comentarios y posibles puntos de mejoras tengan al respecto del aplicativo.

En paralelo, se diseñarán pruebas de rendimiento y aceptación, para poder garantizar que se cumplan con los requerimientos propuestos, además de medir el rendimiento del aplicativo y encontrar posibles errores que se presenten.

Como resultado de esta fase se espera tener una realimentación del aplicativo y un informe de correcciones a mejorar en futuras iteraciones del prototipo.

3.5.6 Mejoras del prototipo

Esta fase se realizará tras finalizar la evaluación del prototipo y analizar los resultados de este, para mejorar el aplicativo. Si el campesino informa que no está contento con la aplicación, se toman los datos recolectados de la evaluación para corregir errores lógicos, mejoras visuales y de interactividad con el usuario. Al utilizar un emulador de celular se probarán y corregirán dichos errores constantemente. Este proceso se realizará las veces requeridas hasta que el aplicativo cumpla con todos los requisitos y el usuario esté satisfecho con la herramienta.

Se maneja un sistema de control de versiones para tener respaldos de todas las modificaciones hechas y un registro de cómo va evolucionando la aplicación.

3.5.7 Implementación

Una vez el aplicativo móvil concluya con todas las pruebas y complete todos los requerimientos estipulados, estará lista para ser desplegada. Se organizará y presentará una documentación acerca del desarrollo del proyecto, con el propósito de eventuales correcciones y mantenimiento a futuro. Al finalizar estas fases se obtiene un prototipo final.

3.5.8 Metodología Kanban

La metodología Kanban es un enfoque de gestión visual que se utiliza para organizar y controlar el flujo de trabajo en un proyecto. Kanban, inspirado en los sistemas de producción de Toyota, se basa en tableros físicos o digitales donde se representan las tareas como tarjetas que se

mueven a través de diferentes etapas del proceso. Cada columna en el tablero representa una etapa del flujo de trabajo, como "Por hacer", "En progreso" y "Terminado". Además, Kanban facilita la identificación de cuellos de botella y la resolución de problemas, ya que se pueden ver claramente dónde se encuentran las tareas en el proceso y tomar medidas para eliminar obstáculos. Kanban ayuda a optimizar el rendimiento y entrega de productos de alta calidad de manera más eficiente. (Martins, 2024)

4. Desarrollo

El desarrollo del proyecto abarcó desde la identificación y análisis de los requerimientos iniciales hasta la implementación y despliegue final del aplicativo. Este proceso incluyó varias fases clave, cada una diseñada para asegurar que se cumplieran los objetivos del proyecto de manera eficiente y efectiva. A lo largo del desarrollo, se emplearon las metodologías prototipos y Kanban para garantizar la calidad y funcionalidad del sistema.

4.1 Requerimientos

A continuación, se presenta el listado de requerimientos funcionales y no funcionales:

4.1.1 Requerimientos funcionales

Los requerimientos funcionales son todas aquellas tareas que debe cumplir el sistema para el correcto funcionamiento del software. Los requerimientos funcionales son evidenciados en la Tabla 3.

Tabla 1

Listado de Requerimientos no funcionales.

Id	Nombre Requerimiento	Funcionalidad	Rol Objetivo
1	Registrar usuario	Permitir el registro de los usuarios a la aplicación	Usuario
2	Gestionar sesiones	Permitir el ingreso y cierre de sesión de los usuarios	Usuario / Gestor
3	Gestionar perfil de usuario	Ver o editar información del usuario	Usuario / Gestor
4	Gestionar Activo	Permitir al usuario registrar un activo	Usuario
5	Registrar ingresos y gastos	Sección que permite al usuario registrar gastos e ingresos en un activo específico	Usuario
6	Llevar control de nivel del usuario a partir de misiones completados	El gestor puede observar el nivel de cada usuario a partir de las misiones que vaya avanzado	Gestor
7	Generar informe de avance de nivel	El gestor puede generar informe de todos los usuarios cuando avanzan de nivel	Gestor
8	Generar informe de logros alcanzados	El usuario puede visualizar los logros que haya obtenido individualmente por nivel	Usuario
9	Mostrar el aprovechamiento	El usuario puede visualizar las ganancias de cada activo. El gestor puede ver el aprovechamiento de cada activo del usuario	Usuario / Gestor
10	Generar informe de activos	El usuario puede visualizar el rendimiento de un activo a través de la aplicación.	Usuario
11	Visualizar consejos	El usuario puede visualizar consejos del sistema para sus activos	Usuario

Id	Nombre Requerimiento	Funcionalidad	Rol Objetivo
12	Preguntas frecuentes	Sección donde el usuario puede encontrar soluciones a sus preguntas	Usuario
14	Generar informe de actividades realizadas	El usuario puede revisar su historial. El gestor podrá revisar el historial de cada usuario.	Gestor
15	Descargar informes	Sección que permite al usuario descargar informes de sus activos en formato .pdf	Usuario
16	Aumentar capacidad de activos al subir de nivel	Se aumenta el número de activos por usuario cada vez que suba de nivel	Usuario
17	Comparar activos	Permitir al usuario comparar el aprovechamiento de cada activo	Usuario

4.1.2 Requerimientos no Funcionales

Los requerimientos no funcionales intervienen en las características de la aplicación, como la velocidad, seguridad, integridad de los datos, accesibilidad, etc. En la Tabla 2, se describen los requerimientos no funcionales de la aplicación.

Tabla 2

Requerimientos no Funcionales

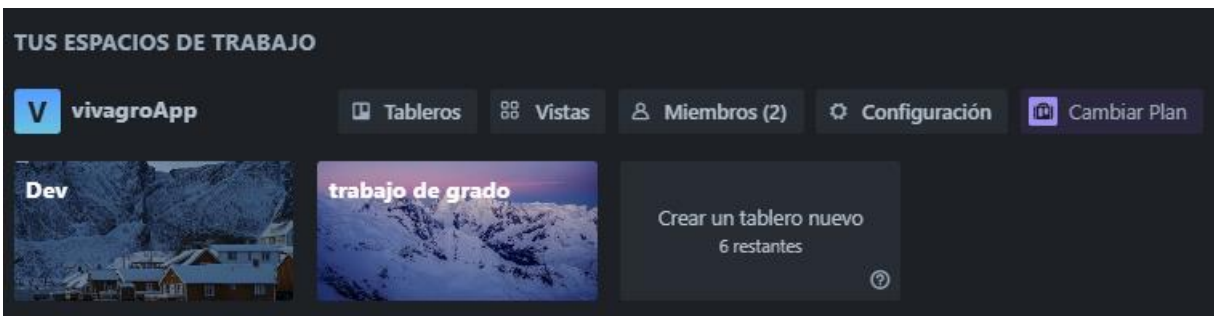
Id	Nombre Requerimiento	Funcionalidad
18	Interfaz de usuario	Se diseñará una interfaz intuitiva con poca saturación de botones para que sea amigable y sencilla de usar para los usuarios
19	Rendimiento	La aplicación tendrá tiempos de respuesta cortos a la interacción del usuario
20	Accesibilidad	La aplicación solo será funcional en aplicativos móviles Android
21	Integridad de los datos	Los datos almacenados estarán completos y sin variaciones, se guardarán localmente si el usuario no tiene internet, de tal manera que sean confiables.

4.2 Plan de desarrollo

Después de haber definido de los requisitos tanto funcionales como no funcionales de la aplicación móvil, se adoptó la metodología Kanban como enfoque para estructurar y administrar las tareas del desarrollo del proyecto. Para mantener una visión clara y actualizada del avance, se empleó Trello como herramienta principal. Esta plataforma permitió la subdivisión del espacio de trabajo en múltiples tableros, cada uno focalizado en una etapa específica del proyecto.

Figura 3

Espacio de trabajo en Trello



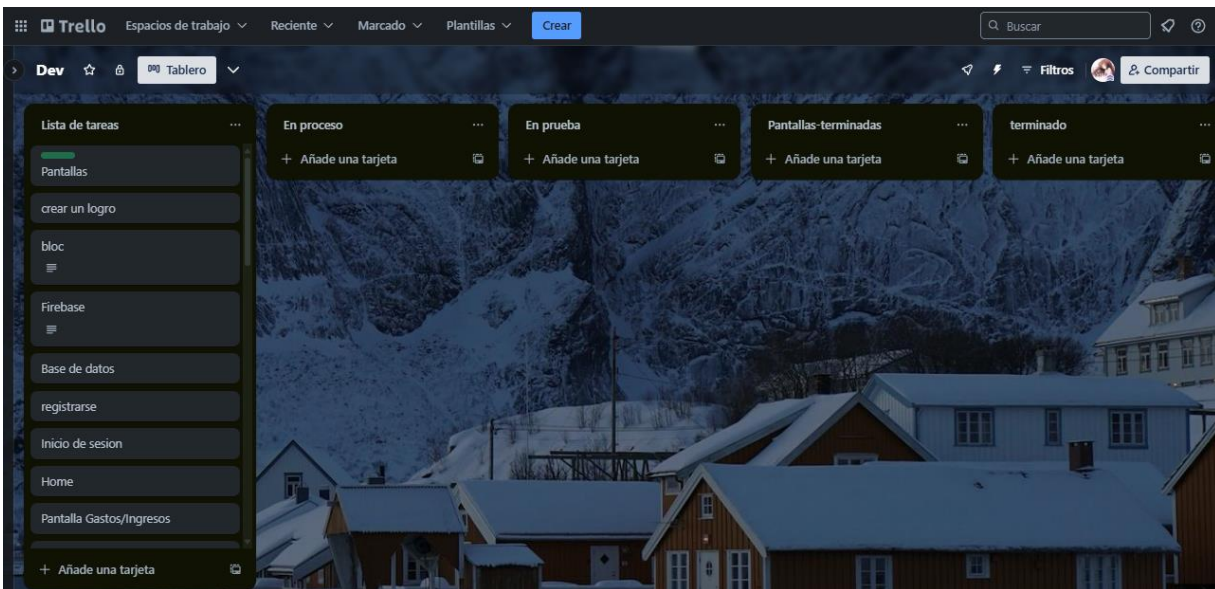
Nota. Adaptado de Trello (s. f.)

4.2.1 Tablero de desarrollo

En el tablero de desarrollo, se avanzaba con la codificación y programación de la aplicación. Este tablero se estructuró en columnas que representaban las distintas fases del desarrollo, tales como 'Lista de tareas'(Figura 4), 'En Proceso'(Figura 5), 'Terminado'(Figura 6), entre otras. Cada tarea específica de programación se representaba con una tarjeta, permitiendo al equipo llevar un seguimiento detallado del progreso técnico de la aplicación.

Figura 4.

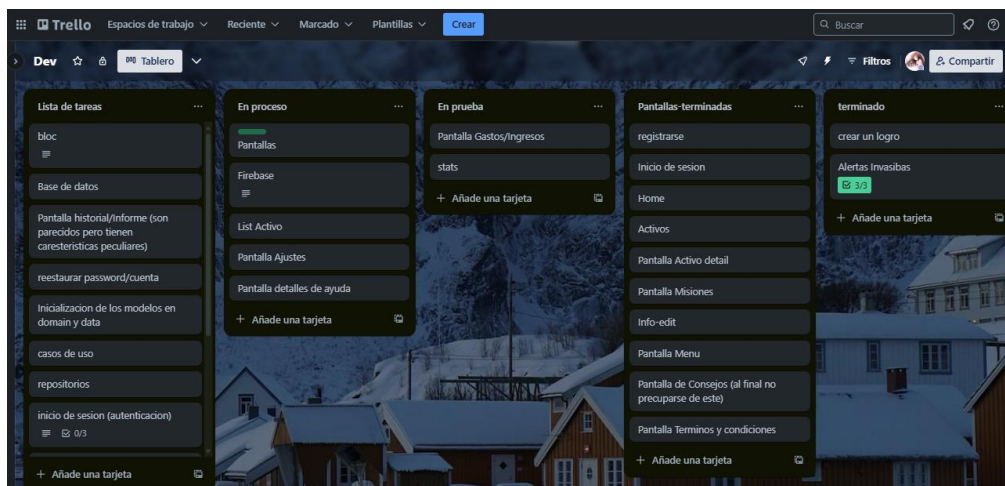
Tablero de la metodología Kanban en el inicio del proyecto



Nota. La figura muestra la lista de tareas a realizar. Adaptado de Trello (s. f.)

Figura 5.

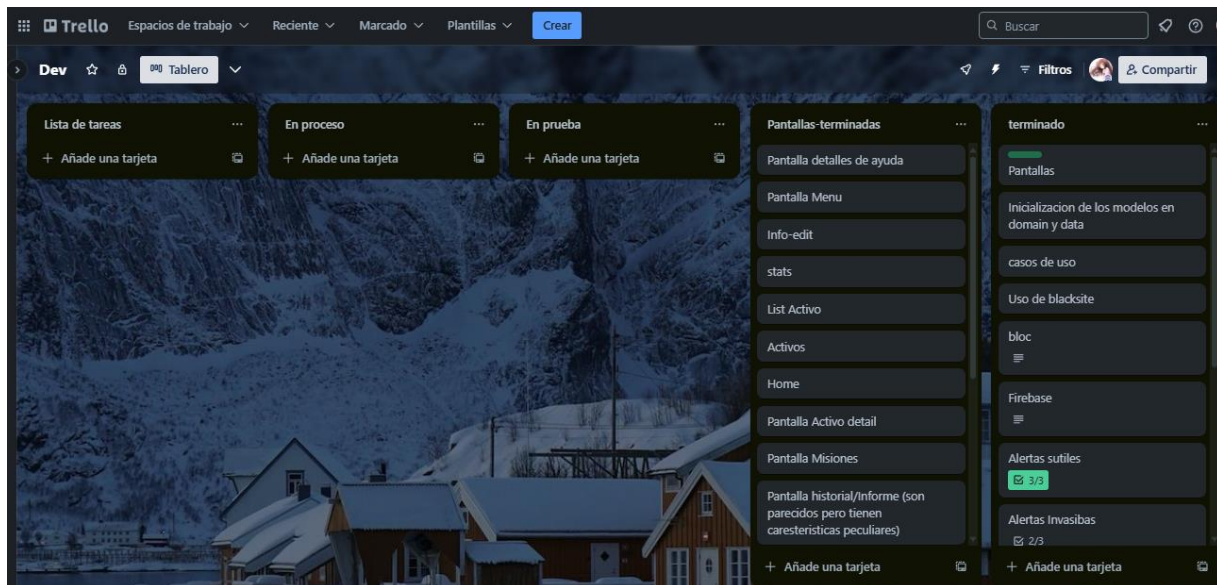
Tablero de la metodología Kanban en el desarrollo del proyecto



Nota. La figura muestra las tareas en progreso, en prueba y tareas terminadas. Adaptado de Trello (s. f.)

Figura 6

Tablero de la metodología Kanban en la finalización del proyecto



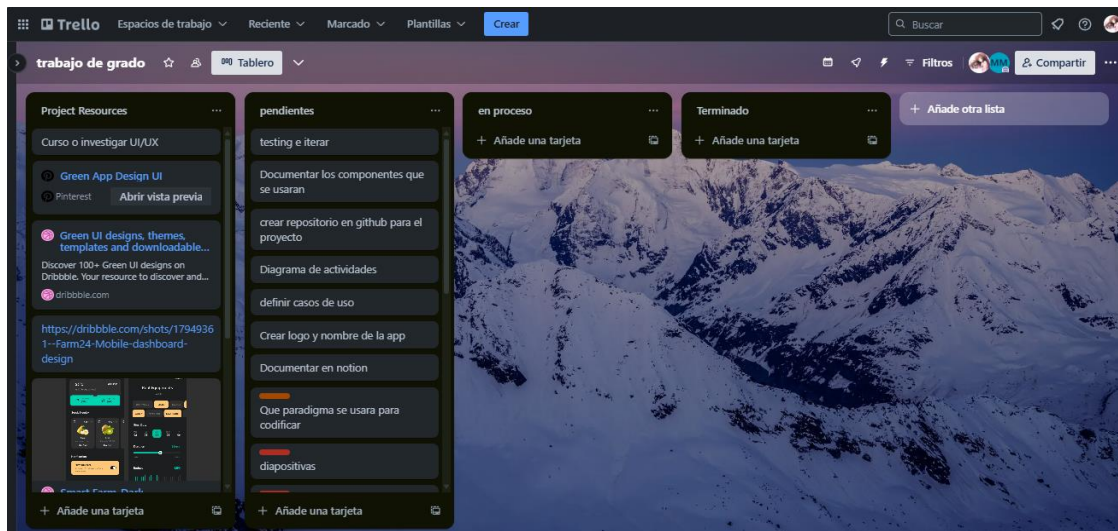
Nota. La figura muestra las tareas terminadas. Adaptado de Trello (s. f.)

4.2.2 Tablero de trabajo de grado

Paralelamente, se utilizó otro tablero dedicado a la documentación y la orquestación general del trabajo de grado. Este tablero incluía tareas relacionadas con la redacción, revisión y consolidación de los documentos necesarios, así como la coordinación de las diferentes partes del proyecto. Las columnas de este tablero facilitaban la gestión de las tareas desde su inicio hasta su finalización, asegurando que cada aspecto del trabajo de grado estuviera adecuadamente documentado y coordinado.

Figura 7.

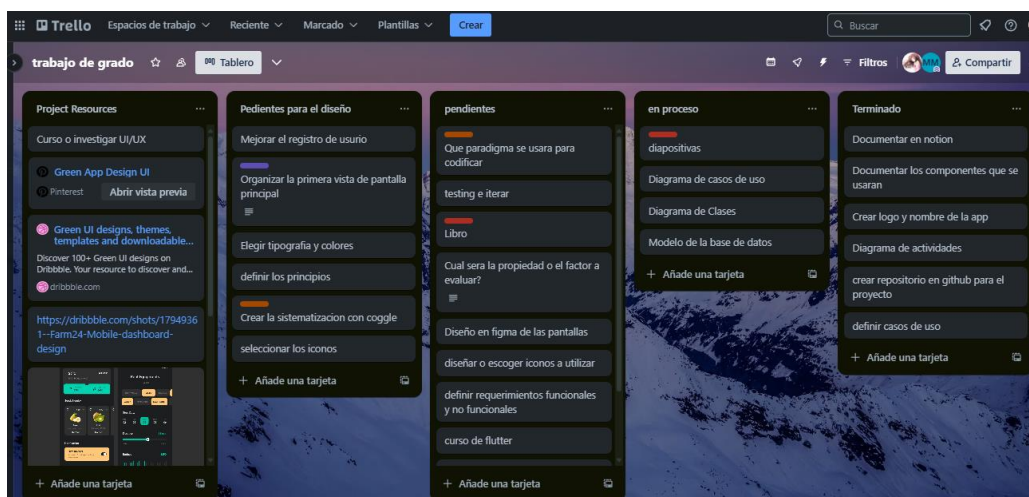
Tablero del trabajo de grado en el inicio del proyecto



Nota. La figura muestra materiales y actividades del proyecto, tales como los recursos del proyecto y la planificación de este. Adaptado de Trello, s. f.

Figura 8.

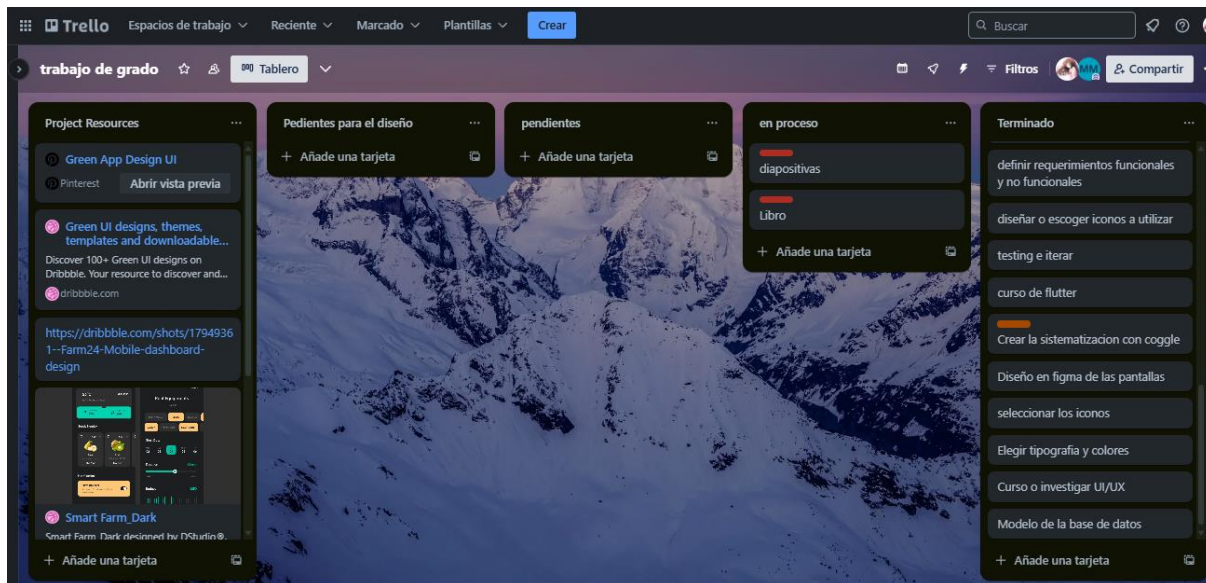
Tablero del trabajo de grado durante el desarrollo del proyecto



Nota. La figura muestra los pendientes para diseño, pendientes, en progreso y terminado de la aplicación

Figura 9.

Tablero de trabajo de grado en la finalización del proyecto



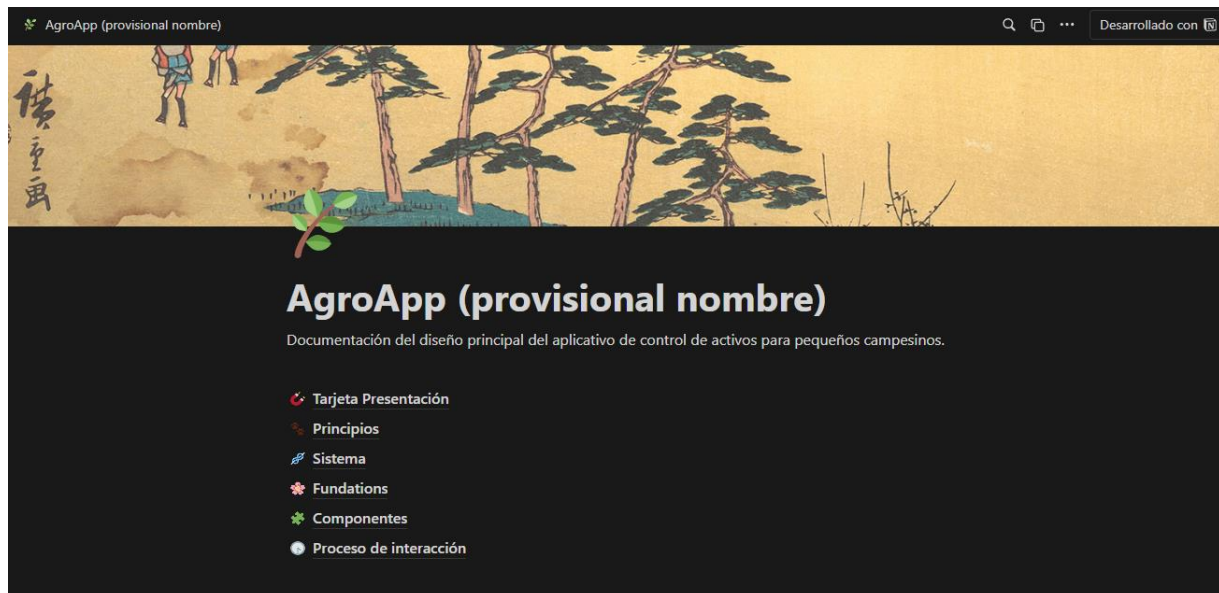
Nota. La figura muestra las actividades de desarrollo terminadas una vez culminado el proceso.

Adaptado de Trello s. f.

Antes de comenzar con la fase de implementación, se creó un espacio en Notion dedicado a establecer directrices clave para el desarrollo. En este espacio se definieron aspectos como iconificación, tipografía, y las pantallas a crear, entre otros. Esta documentación sirvió como referencia constante durante todo el proceso de desarrollo, garantizando la coherencia y la alineación con los objetivos del proyecto.

Figura 10.

Pantalla principal de documentación en Notion.



Nota. La figura muestra el espacio de trabajo creado en Notion, donde se listan la tarjeta de presentación del proyecto, los principios, sistemas, inicios, componentes y el proceso de interacción.

4.3 Diseño

A continuación, es presentado el diseño de la aplicación, desde el punto de vista de los diagramas de Casos de uso, diagrama de clases, diagramas de actividades entre otros.

4.3.1 Diagrama de caso de uso general

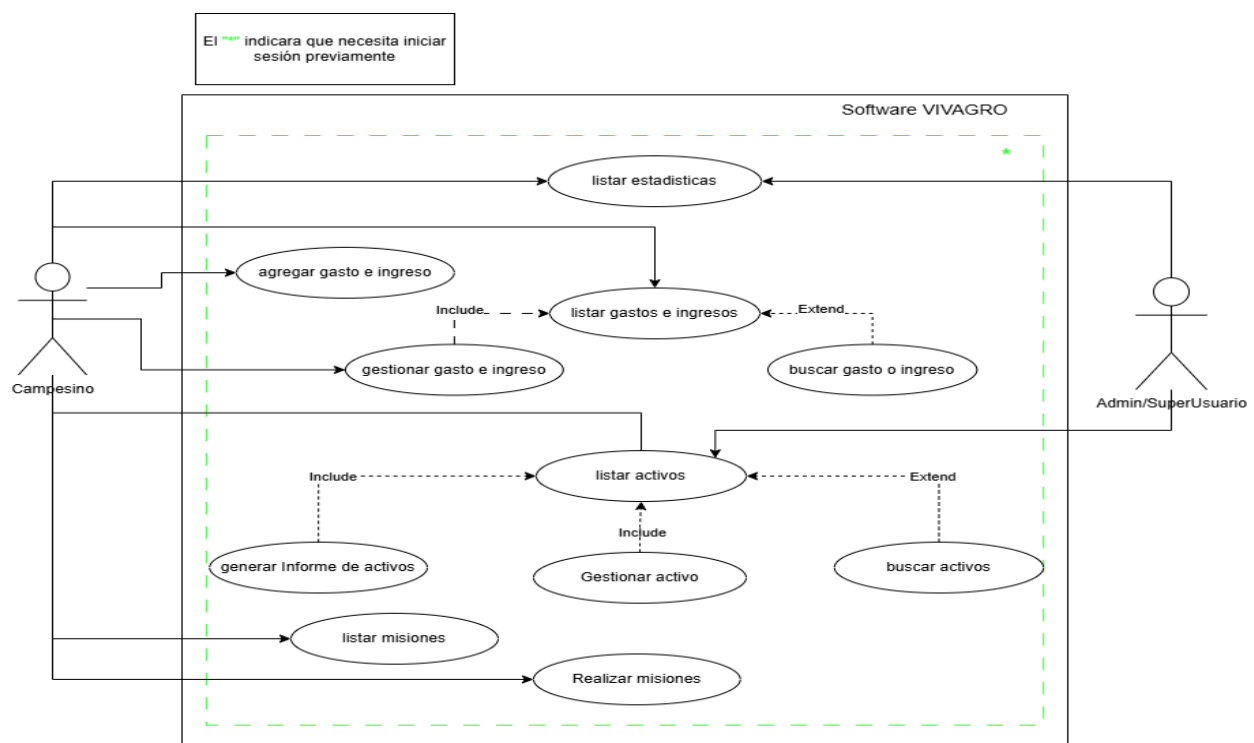
IBM define Los casos de uso como un conjunto de herramientas que buscan refinar un conjunto de requisitos basados en un rol o tarea, buscando agrupar requisitos comunes según el

tipo de rol u objetivo y siendo un reconocimiento de los requisitos que debe cumplir el proyecto. (IBM, 2024)

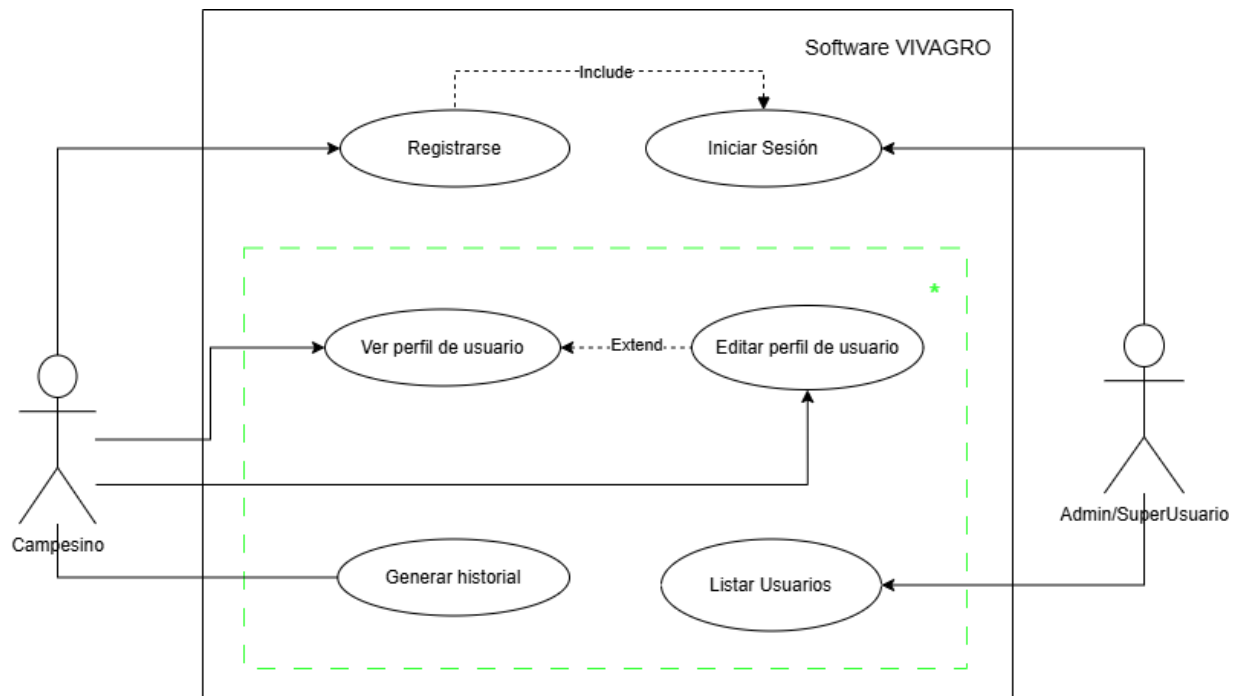
Para este proyecto se realizaron los siguientes casos de uso para representar la funcionalidad del software y los actores que interactúan con el mismo, que pueden ser visualizados en el *Apéndice A*. El diagrama de Caso de uso general que recopila los requerimientos es presentado en la Figura 11 y Figura 12.

Figura 11.

Diagrama de caso de uso general 1



Nota. La figura muestra el diagrama de Caso de uso general de la aplicación, dividido en dos imágenes. (1/2)

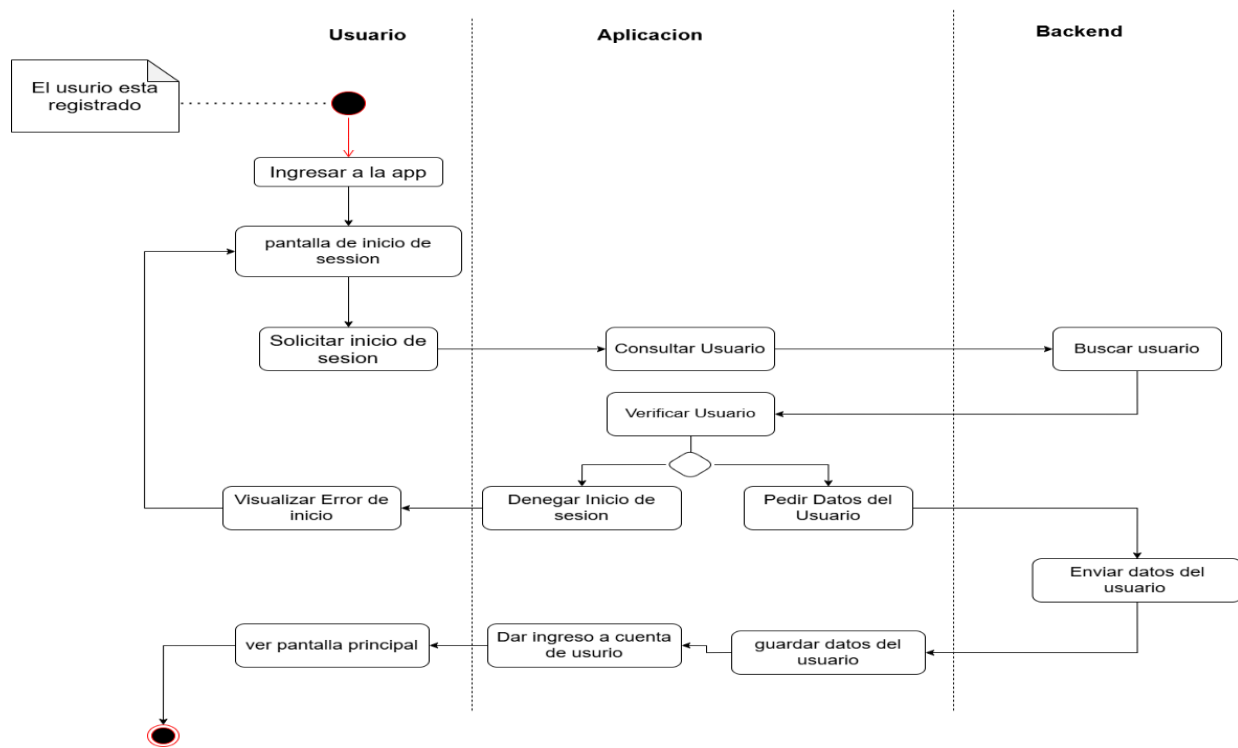
Figura 12.*Diagrama de caso de uso general 2*

Nota. La figura muestra el diagrama de Caso de uso general de la aplicación, dividido en dos imágenes. (2/2)

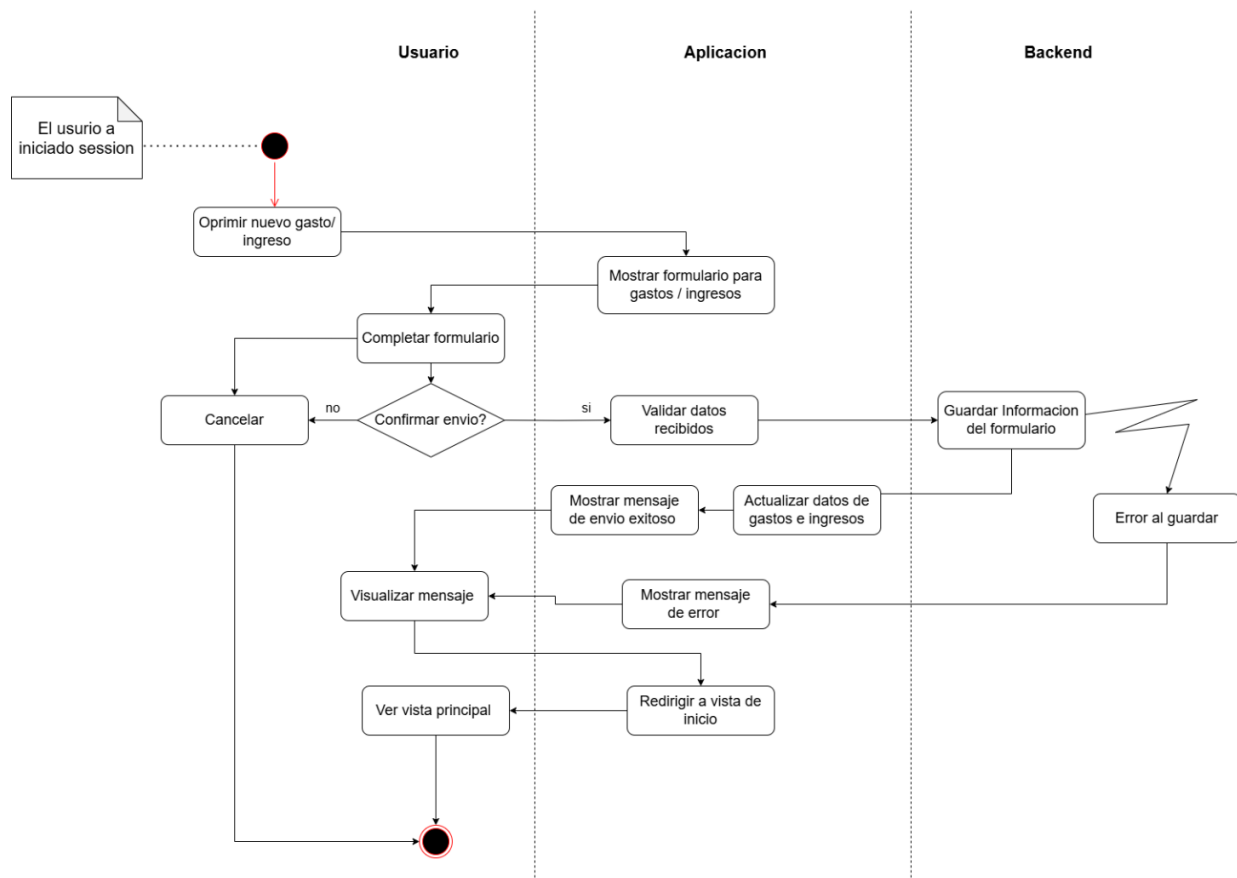
4.3.2 Diagramas de actividades

Los diagramas de actividades son aquellos que proveen un punto de vista del comportamiento de un sistema a través de describir una secuencia de acciones en un proceso. Estos mismo son similares a un diagrama de flujo debido a que se observa entre cada acción en una actividad.

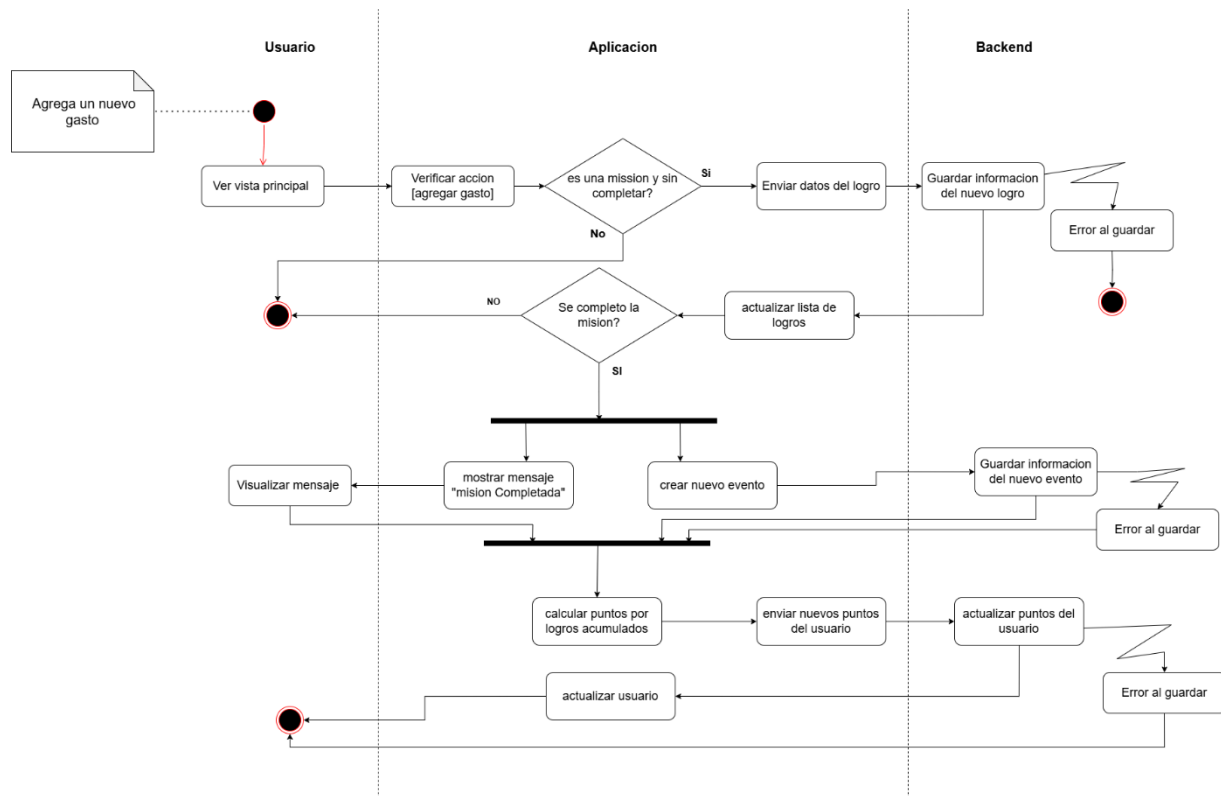
A continuación, son presentados los diagramas de actividades principales de este proyecto:

Figura 13.*Diagrama de actividad inicio de sesión.*

Nota. La figura muestra el diagrama de actividades en el momento en que el usuario se registra a la aplicación

Figura 14.*Diagrama de actividad creación de nuevo gasto/ingreso*

Nota. La figura lista el proceso donde se crea un nuevo gasto o ingreso, y las entidades por las que pasa el usuario a través del proceso (usuario, aplicación, Base de datos).

Figura 15.*Diagrama de actividad creación de un logro*

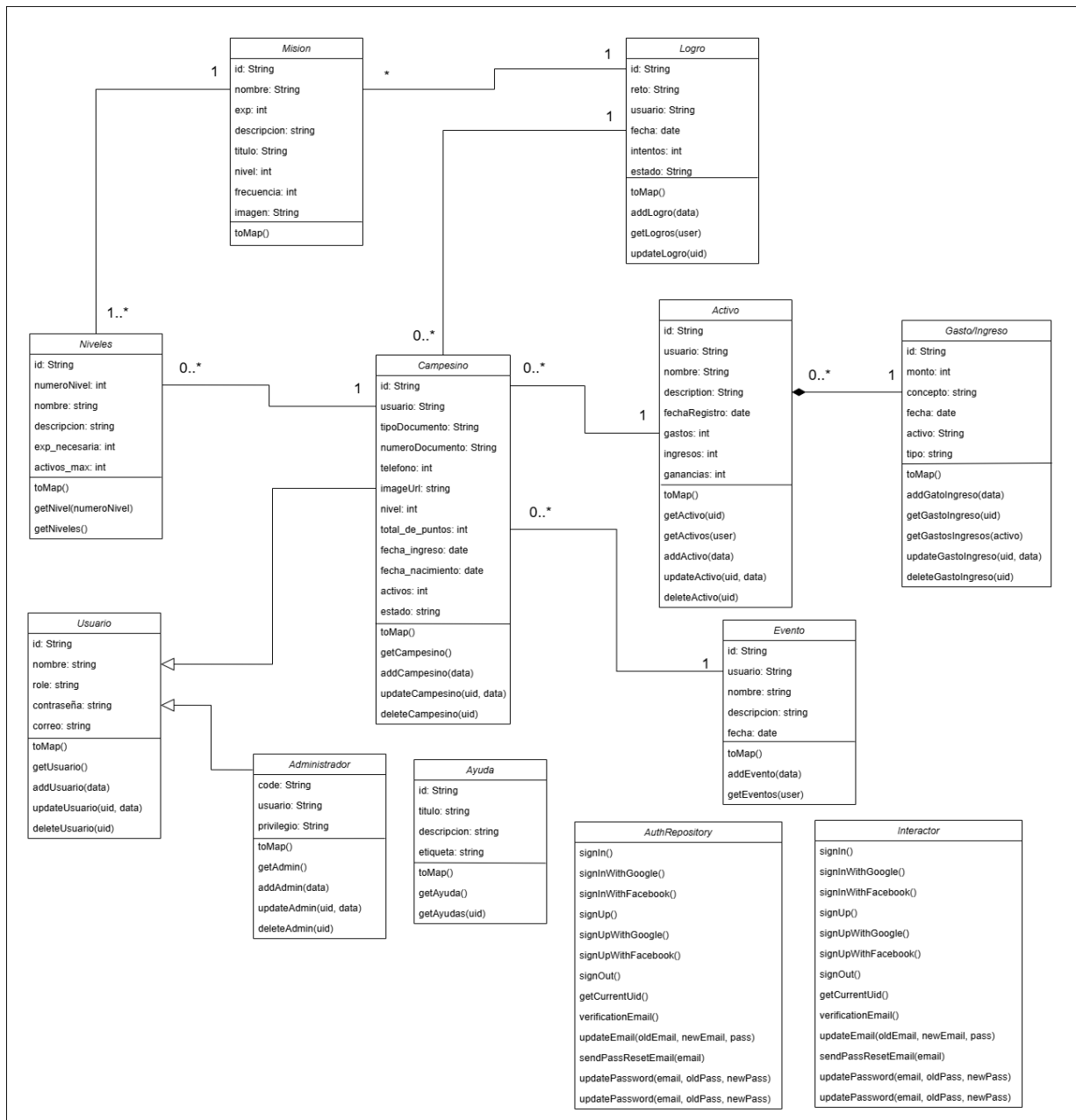
Para observar los diagramas de actividades más detallados dirigirse al **Apéndice B**.

4.3.3 Diagrama de clases

Los diagramas de clases son aquellos que describen la estructura de un sistema, siendo fundamentales para el proceso de descripción de cada objeto relacionado con el sistema, tal y como se presenta en la Figura 16

Figura 16.

Diagrama de clases

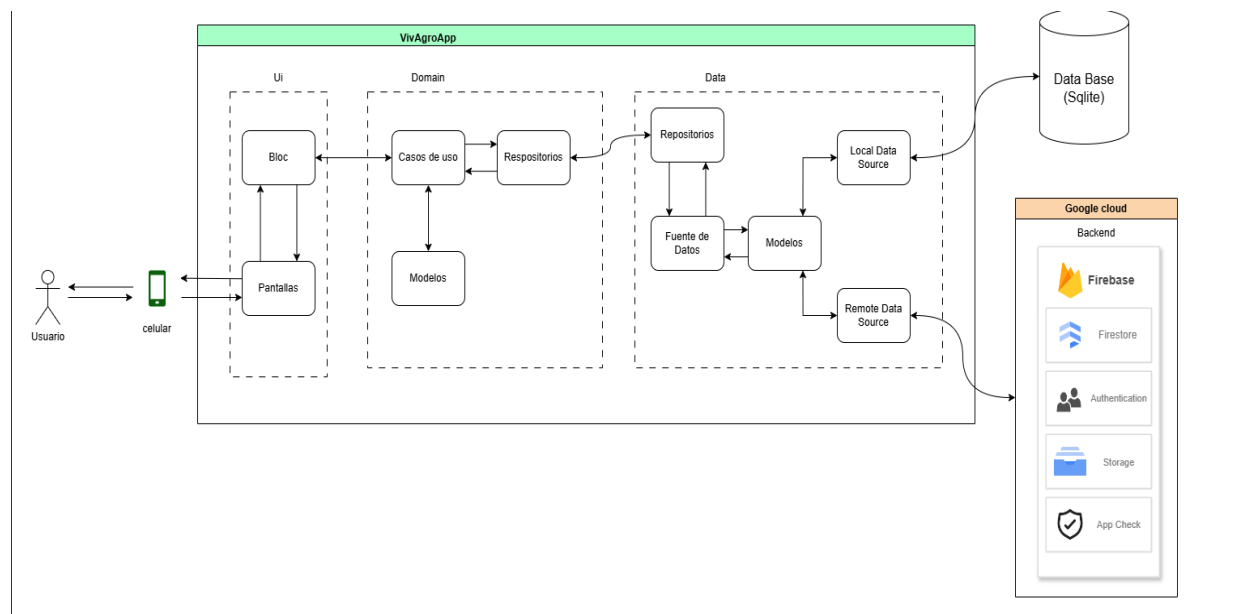


4.3.4 Arquitectura de Software

La figura 6 muestra el diseño de la arquitectura en el desarrollo del proyecto, donde se observan los elementos que componen la plataforma y como se relacionan entre ellos.

El proceso inicia con el usuario cuando interactúa con la interfaz de usuario (UI) de la aplicación, navegando a través de las distintas pantallas y realizando acciones como ingresar datos, presionar botones, etc. Cuando el usuario realiza una acción en la interfaz de usuario, como presionar un botón para iniciar sesión, la interfaz de usuario envía un evento al correspondiente BLoC. El BLoC procesa el evento recibido y solicita una operación específica al Caso de uso correspondiente de la capa de dominio. El Caso de uso, que contiene la lógica de negocio de la aplicación, interactúa con los repositorios para acceder a los datos necesarios para completar la operación solicitada. Los modelos son utilizados por los casos de uso para transferir datos entre las capas y realizar cálculos o validaciones adicionales si es necesario. El repositorio interactúa con las fuentes de datos, realizando peticiones como una API remota o Firebase, donde se utiliza firebase authentication como autenticador, firebase database para base de datos, firebase storage para las imágenes, firebase app Check para autenticar las solicitudes de la aplicación y realizar las operaciones de lectura o escritura necesarias.

Una vez que se completan las operaciones necesarias, los resultados se propagan hacia arriba a través de las capas, donde es observado por el usuario.

Figura 17.*Diagrama de arquitectura del proyecto*

4.3.5 Mockups

En esta sección se encuentran las vistas maquetadas de la aplicación realizadas a lo largo de todo el proyecto. Para ver todas las vistas dirigirse al **Apéndice D**.

4.4 Implementación

En esta sección, se detallan los pasos y procedimientos llevados a cabo para convertir todos los requerimientos, diagramas, diseños y especificaciones definidos del proyecto en un sistema funcional y operativo. La implementación abarca desde la configuración del entorno de desarrollo en Android Studio hasta la integración de los diferentes componentes del sistema; se implementó

la metodología de arquitectura limpia, esto con el fin de un mejor mantenimiento del código y una mayor escalabilidad, dividiendo cada responsabilidad en capas lógicas.

La Figura 18 se observa el archivo main.dart, donde se inicializa las funciones esenciales para el uso de firebase en la aplicación y la inyección de dependencias.

Figura 18.

Archivo main.dart de la aplicación



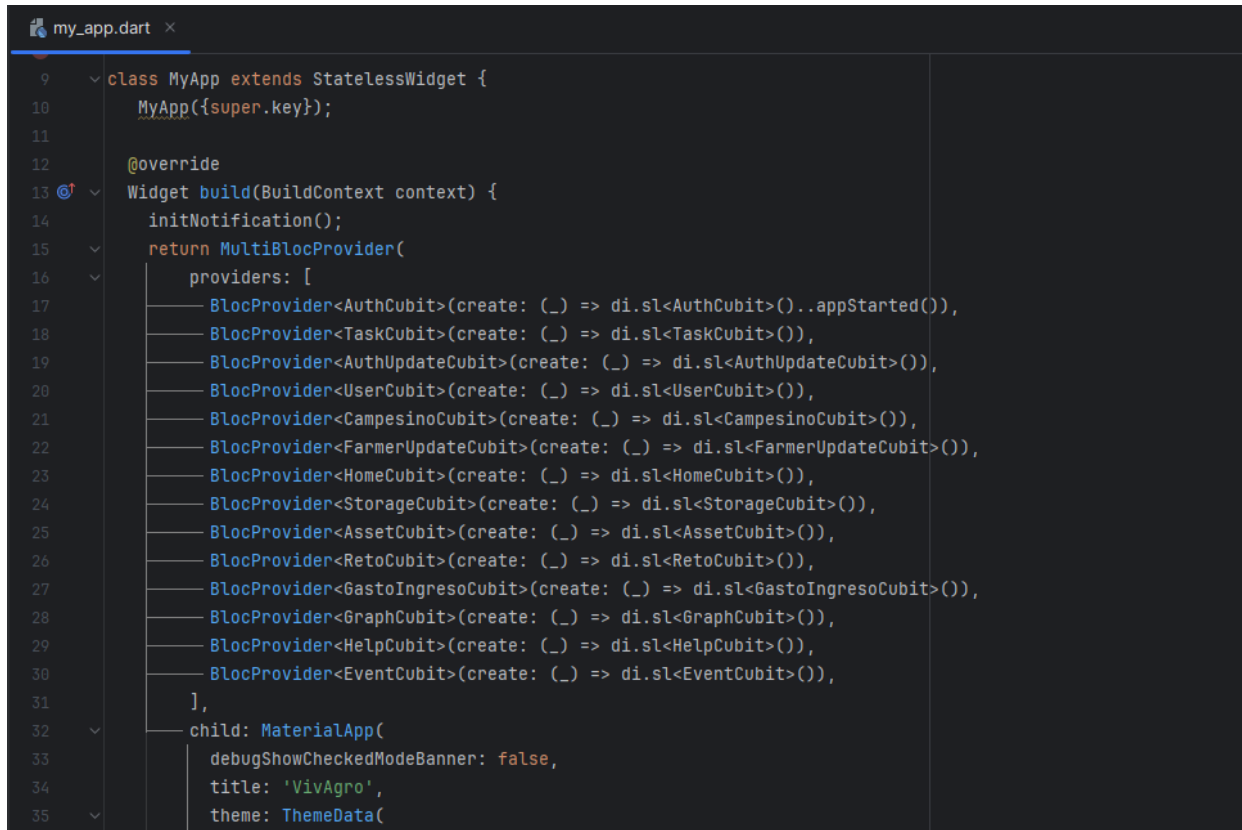
```
main.dart x
1  import 'package:flutter/material.dart';
2  import 'package:gtext/gtext.dart';
3  import 'package:vivagroApp/data/firebase/firebase_connection.dart';
4  import 'package:firebase_app_check/firebase_app_check.dart';
5
6  import 'app/my_app.dart';
7  import 'package:vivagroApp/inject_dependencies.dart' as di;
8
9  void main() async{
10     WidgetsFlutterBinding.ensureInitialized();
11     await FirebaseConnection.initialize();
12     await di.init();
13     GText.init(to: 'es', enableCaching: false);
14     await FirebaseAppCheck.instance.activate(
15         androidProvider: AndroidProvider.debug,
16     );
17
18     runApp(MyApp());
19 }
```

El archivo my app.dart presentado en la Figura 19, se encuentra el widget raíz de la aplicación con los correspondientes providers (línea 15), con el objetivo de un acceso desde cualquier parte del proyecto al estado e información expuesta en los cubits declarados. Además,

en la línea 17 se establece que el cubit de autenticación es el primero en inicializarse en el aplicativo.

Figura 19.

Archivo my_app.dart

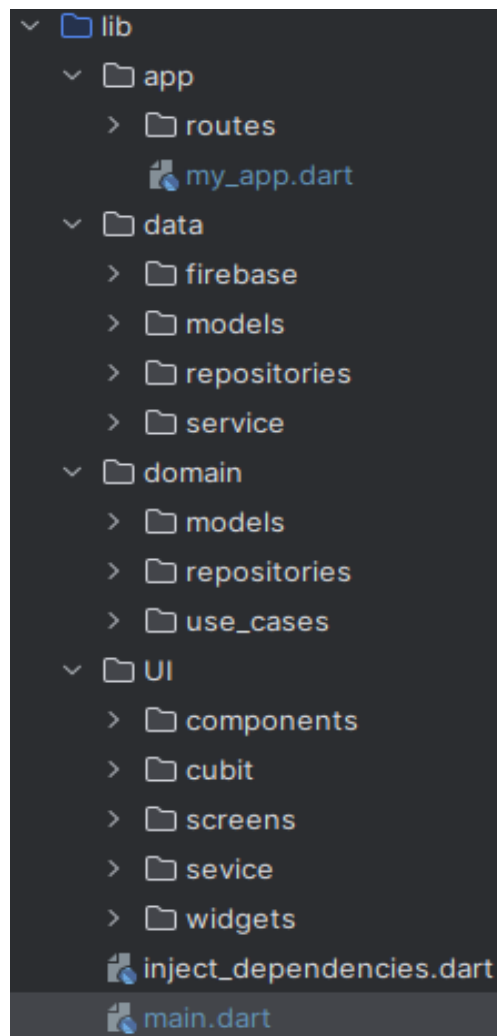


```
9 class MyApp extends StatelessWidget {
10   MyApp({super.key});
11
12   @override
13   Widget build(BuildContext context) {
14     initNotification();
15     return MultiBlocProvider(
16       providers: [
17         BlocProvider<AuthCubit>(create: (_) => di.sl<AuthCubit>()..appStarted()),
18         BlocProvider<TaskCubit>(create: (_) => di.sl<TaskCubit>()),
19         BlocProvider<AuthUpdateCubit>(create: (_) => di.sl<AuthUpdateCubit>()),
20         BlocProvider<UserCubit>(create: (_) => di.sl<UserCubit>()),
21         BlocProvider<CampesinoCubit>(create: (_) => di.sl<CampesinoCubit>()),
22         BlocProvider<FarmerUpdateCubit>(create: (_) => di.sl<FarmerUpdateCubit>()),
23         BlocProvider<HomeCubit>(create: (_) => di.sl<HomeCubit>()),
24         BlocProvider<StorageCubit>(create: (_) => di.sl<StorageCubit>()),
25         BlocProvider<AssetCubit>(create: (_) => di.sl<AssetCubit>()),
26         BlocProvider<RetoCubit>(create: (_) => di.sl<RetoCubit>()),
27         BlocProvider<GastoIngresoCubit>(create: (_) => di.sl<GastoIngresoCubit>()),
28         BlocProvider<GraphCubit>(create: (_) => di.sl<GraphCubit>()),
29         BlocProvider<HelpCubit>(create: (_) => di.sl<HelpCubit>()),
30         BlocProvider<EventCubit>(create: (_) => di.sl<EventCubit>()),
31       ],
32       child: MaterialApp(
33         debugShowCheckedModeBanner: false,
34         title: 'VivAgro',
35         theme: ThemeData(
```

En la Figura 20 se muestra la jerarquía y organización de carpetas utilizada en el proyecto siguiendo los principios de la arquitectura limpia. Dentro de la carpeta 'lib', se encuentran las carpetas que representan las diferentes capas que estructuran el proyecto. Estas carpetas son: 'data', 'domain', 'UI', y 'app'. Al mismo nivel que estas carpetas, también se encuentran los archivos 'main.dart' y el archivo de inyección de dependencias. La estructura está diseñada para promover la separación de responsabilidades y facilitar el mantenimiento del código.

Figura 20.

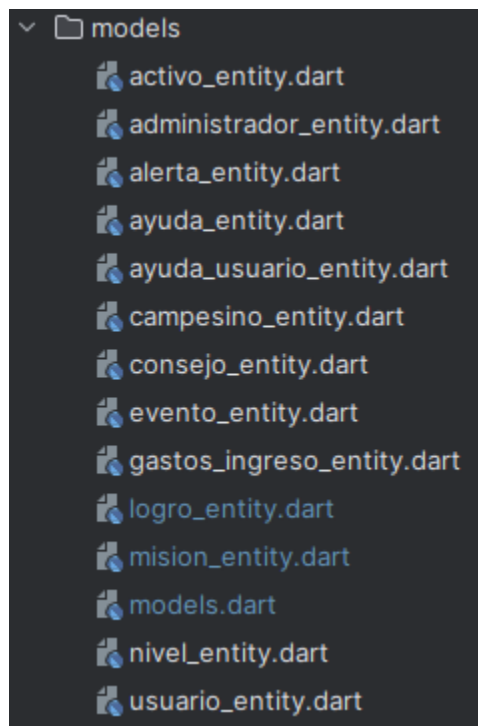
Jerarquía de carpetas siguiendo arquitectura limpia



Modelos propuestos en la capa domain para el proyecto se evidencian en la Figura 21.

Figura 21

Modelos implementados en el aplicativo



Para la verificación de la realización de las misiones, como se muestra en la Figura 22, se implementa una función que se ejecuta en segundo plano. Esta función recibe como parámetro la acción realizada por el usuario. En las líneas 17 y 18, se obtienen las listas de misiones y logros gestionadas por los Cubits. A partir de la línea 19, se verifica cuáles misiones el usuario aún no ha completado de acuerdo con la acción realizada. La función identifica las misiones pendientes que coinciden con el nombre de la acción o que están en proceso, y procede en consecuencia.

Figura 22*Implementación de la función verificar misión*

```
10 Future<void> VerifyMission({required BuildContext context, required String nombre, required String uid }) async{
11   late List<RetoEntity> retosSinEmpezar = [];
12   late List<LogroEntity> logroEnProgreso = [];
13
14   //esto puede en otra parte para no pedir context
15   final missionGoals = BlocProvider.of<RetoCubit>(context).state;
16   if(missionGoals is RetoLoaded){
17     final listaRetos = missionGoals.retos;
18     final listaLogros = missionGoals.logros;
19     retosSinEmpezar = listaRetos.where((reto) =>
20       !listaLogros.any((logro) => logro.reto == reto.id) && reto.nombre.contains(nombre)
21     ).toList();
22     logroEnProgreso = listaLogros.where((logro) =>
23       logro.estado=='En proceso' && listaRetos.any((reto) => reto.id == logro.reto && reto.nombre == nombre)
24     ).toList();
25
26
27     if(logroEnProgreso.isNotEmpty){
28       logroEnProgreso.forEach((logro) async {
29         final reto = listaRetos.firstWhere((reto) => reto.id == logro.reto);
30         Map<String,dynamic> data = {'intentos': logro.intentos+1};
31         if(reto.frecuencia==logro.intentos+1) data['estado'] = 'Completado';
32         await BlocProvider.of<RetoCubit>(context).updateGoal(
33           uid: uid,
34           data: data
35         );
36       });
37     });
38   }
```

En la Figura 23 se muestra la configuración del archivo 'layoutEvent.dart' permaneciendo constantemente a la escucha de los eventos del usuario en toda la aplicación, lo que permite una respuesta continua y consistente a las interacciones del usuario en todo el sistema.

Figura 23.*Implementación de eventos principales en toda la aplicación 1*

```

lib > UI > screens > layout_event.dart > _LayoutEventState > _bodyWidget
18 class _LayoutEventState extends State<LayoutEvent> {
19   @override
20   Widget build(BuildContext context) {
21     return MultiBlocListener(
22       listeners: [
23         BlocListener<RetoCubit, RetoState>(
24           listener: (context, retoState) {
25             if (retoState is RetoCompleted) {
26               var campesinoState = context.read<CampesinoCubit>().state;
27               if (campesinoState is CampesinoDataLoaded) {
28                 BlocProvider.of<EventCubit>(context).newEvent(
29                   data: EventoEntity(
30                     usuario: campesinoState.data.id!,
31                     nombre: 'mission completada',
32                     descripcion: ''
33                   )); // EventoEntity
34               AlertPasive(context: context, backgroundColor: green_light, msg: 'Mission Completada!!');
35             }
36             if (retoState is RetoCompleted || retoState is RetoSuccess) {
37               var campesinoState = context.read<CampesinoCubit>().state;
38               if (campesinoState is CampesinoDataLoaded) {
39                 BlocProvider.of<RetoCubit>(context).getRetos(
40                   uid: campesinoState.data.id!,
41                   numeroNivel: campesinoState.data.nivel
42                 );
43             }
44             if (retoState is RetoLoaded) {
45               var campesinoState = context.read<CampesinoCubit>().state;
46               if (campesinoState is CampesinoDataLoaded) {
47                 MissionHelper().PuntosGanados(context: context, retoState: retoState, campesino: campesinoState.data);
48             }
49           }
50         ), // BlocListener
51         BlocListener<FarmerUpdateCubit, FarmerUpdateState>(
52           listener: (context, farmerState) {
53             if (farmerState is FarmerUpdateSuccess) {
54               BlocProvider.of<CampesinoCubit>(context).getDataCampesino();
55             }
56           }, // BlocListener
57         BlocListener<CampesinoCubit, CampesinoState>(
58           listener: (context, userState) {
59             if (userState is CampesinoDataLoaded) {
60               var retoState = context.read<RetoCubit>();
61               if (retoState.state is! RetoLoaded) {
62                 retoState.getRetos(uid: userState.data.id!, numeroNivel: userState.data.nivel);
63             }
64             BlocProvider.of<HomeCubit>(context).getProgresLevel(uid: userState.data.id!);
65           }
66         ), // BlocListener
67       ],
68       child: _bodyWidget(),
69     ); // MultiBlocListener
70   }
71 }

```

En el archivo 'layoutEvent.dart', como se muestra en la Figura 24, se asegura que siempre esté a la escucha de los eventos. Para lograr esto, la mayoría de las pantallas se envuelven en un 'Navigator', lo que garantiza que todas las pantallas estén actualizadas con la información pertinente.

Figura 24.

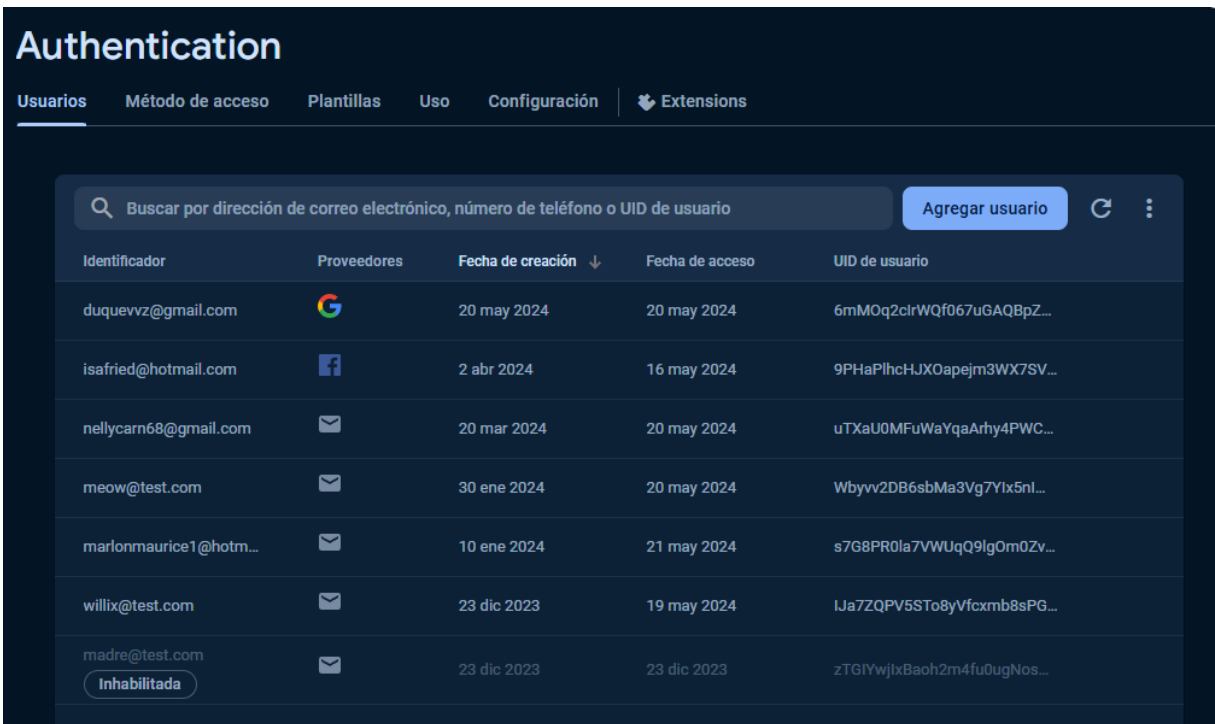
Archivo layoutEvent.dart 2

```
88   _bodyWidget(){
89     return SingleChildScrollView(
90       child: Container(
91         height: MediaQuery.of(context).size.height,
92         child: PopScope(
93           canPop: false,
94           onPopInvoked: (didPop) async {
95             if (didPop) {
96               return;
97             }
98             await navigatorKey.currentState?.maybePop().then((canPop) {
99               if(!canPop){
100                 SystemNavigator.pop();
101               }
102             });
103           },
104           child: Navigator(
105             key: navigatorKey,
106             onGenerateRoute: (settings){
107               return MaterialPageRoute(
108                 builder: (context){
109                   return BottomNavLayout();
110                 }
111               ); // MaterialPageRoute
112             }
113           ), // Navigator
114         ) // PopScope
115       ) // Container
116     ); // SingleChildScrollView
117   }
118 }
```

La función de Firebase Authentication que se implementó en el aplicativo se presenta en la Figura 25, una función de Firebase que valida a los usuarios para el inicio y registro con distintos proveedores.

Figura 25.

Consola de Firebase authentication de Vivagro



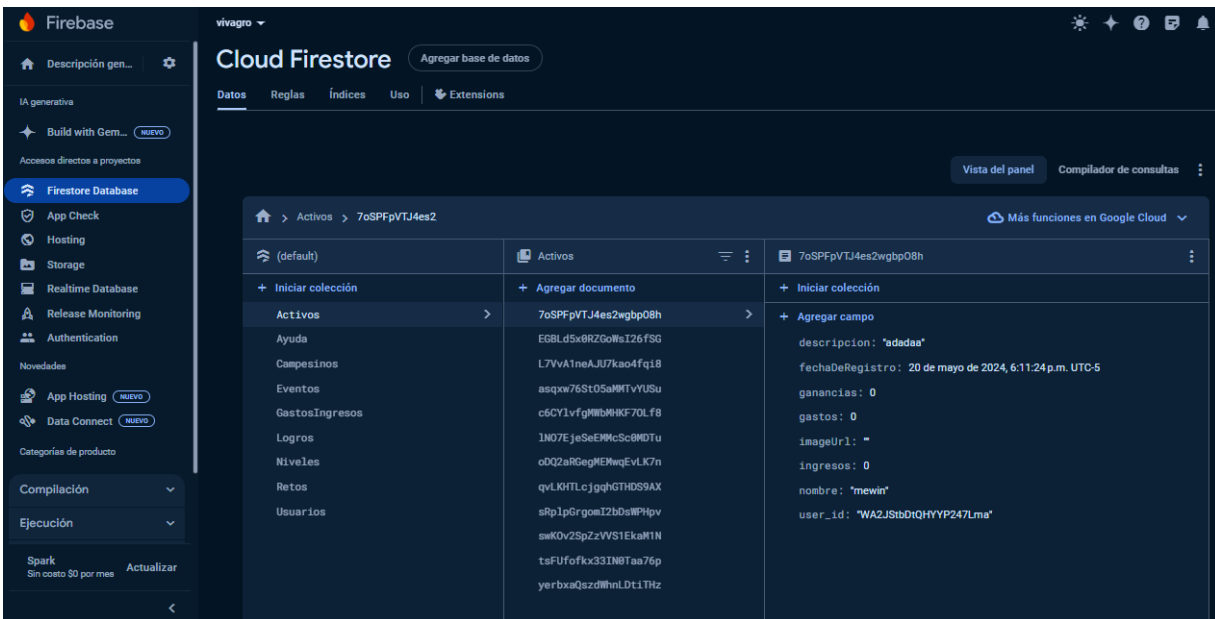
The screenshot shows the 'Authentication' section of the Firebase console. It features a navigation menu with 'Usuarios', 'Método de acceso', 'Plantillas', 'Uso', 'Configuración', and 'Extensions'. Below the menu is a search bar with the text 'Buscar por dirección de correo electrónico, número de teléfono o UID de usuario' and an 'Agregar usuario' button. The main content is a table of users with the following columns: 'Identificador', 'Proveedores', 'Fecha de creación', 'Fecha de acceso', and 'UID de usuario'. The table contains eight rows of user data, with the last row marked as 'Inhabilitada'.

Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario
duquevz@gmail.com		20 may 2024	20 may 2024	6mMOq2clrWQf067uGAQBpZ...
isafried@hotmail.com		2 abr 2024	16 may 2024	9PHaPlhcHJXOapejm3WX7SV...
nellycarn68@gmail.com		20 mar 2024	20 may 2024	uTXaU0MFuWaYqaArhy4PWC...
meow@test.com		30 ene 2024	20 may 2024	Wbyyv2DB6sbMa3Vg7Ylx5nl...
marlonmaurice1@hotm...		10 ene 2024	21 may 2024	s7G8PR0la7VWUqQ9lgOm0Zv...
willix@test.com		23 dic 2023	19 may 2024	IJa7ZQPv5STo8yVfcxmb8sPG...
madre@test.com		23 dic 2023	23 dic 2023	zTGIYwjxBaoh2m4fu0ugNos...

La Figura 26 se presenta la base de datos de documentos NoSQL Cloud Firestore que es usado como base de datos del aplicativo.

Figura 26.

Consola de Cloud Firestore usado en Vivagro



4.5 Prototipo

A continuación, se presenta los prototipos realizados en nuestro proyecto.

4.5.1 Vistas de usuario definitivas para la aplicación

Para visualizar las vistas definitivas de la aplicación realizadas para la aplicación móvil dirigirse al *Apéndice E*.

4.6 Evaluación del funcionamiento del aplicativo

La evaluación de este proyecto se ejecuta bajo la metodología de prototipos, en la cuál uno de los principales pasos es la evaluación del cliente por medio de pruebas. Se realizaron las siguientes pruebas para la evaluación del proyecto:

4.6.1 Pruebas unitarias

Durante el desarrollo del aplicativo, se llevaron a cabo pruebas unitarias de manera continua para mantener el correcto funcionamiento de funcionalidades o evaluar puntos específicos dentro del aplicativo. Estas pruebas fueron las más recurrentes a lo largo del proyecto, lo que permitió identificar y corregir errores en las primeras fases de desarrollo, facilitando la corrección temprana.

4.6.2 Pruebas de compatibilidad

Un elemento crucial del proyecto es la garantía de que el aplicativo sea accesible y funcional en una amplia gama de dispositivos móviles, incluidos aquellos de gama baja o con resolución limitada. Con este objetivo en mente, se llevaron a cabo exhaustivas pruebas de compatibilidad utilizando una variedad de enfoques. Se emplearon emuladores de baja resolución para simular condiciones de dispositivos con recursos limitados, además de realizar pruebas en dispositivos reales de gama baja con diferentes resoluciones de pantalla. Este enfoque permitió identificar y abordar cualquier problema de compatibilidad, asegurando que el aplicativo se

adaptara de manera efectiva a las diversas características de hardware y resolución de pantalla. Además, se corrigieron cualquier error visual que surgiera durante estas pruebas, garantizando una experiencia consistente y óptima para todos los usuarios, independientemente del dispositivo que utilicen, se puede observar con más detalle los resultados obtenidos en el **Apéndice C**.

4.6.3 Prueba de aceptación

Durante el desarrollo del aplicativo, se realizaron pruebas de aceptación con usuarios para evaluar la usabilidad y funcionalidad del sistema. Estas pruebas involucraron a un grupo seleccionado de usuarios que interactuaron con la aplicación y proporcionaron retroalimentación a través de encuestas. La información recopilada permitió identificar áreas de mejora y asegurar que el aplicativo cumpliera con las expectativas y necesidades de los usuarios finales. Estas pruebas fueron esenciales para validar la experiencia del usuario y garantizar un producto final de alta calidad. Los resultados y análisis de estas encuestas se pueden encontrar en el Apéndice D.

5. Conclusiones

Finalizando el desarrollo del aplicativo, se logró cumplir con el objetivo general de presentar una herramienta eficiente y accesible para los usuarios, siendo un 66,7% de los usuarios que consideraron excelente el rendimiento y la velocidad del aplicativo. Esta herramienta facilita

la gestión y el control de activos de pequeños productores agrícolas, mejorando la productividad de los usuarios, basado en las encuestas realizadas a los usuarios.

Mediante un riguroso análisis de los requerimientos, se identificaron y documentaron todas las necesidades esenciales de los usuarios finales, permitiendo diseñar un aplicativo que responde a la enseñanza de la gestión de activos en pequeños sistemas agrícolas, reiterando en que todas las funcionalidades claves se encuentren en la aplicación y sean accesibles de manera intuitiva, con un porcentaje de 77,8% de los usuarios que estuvieron totalmente de acuerdo con el fácil uso de la aplicación y sus funcionalidades.

Haciendo uso de la estructura en UML, los casos de uso, diagramas de clase y actividades, se obtuvo una comprensión clara del alcance y funcionalidades buscadas del aplicativo móvil. Con este enfoque se garantizó una base sólida para la posterior implementación. La planificación meticulosa antes de la fase de codificación fue crucial. Esta fase incluyó la definición detallada de requisitos, la creación de mockups y la elaboración de un plan de implementación estructurado. Esta preparación rigurosa no solo facilitó la implementación sin contratiempos, sino que también contribuyó significativamente a minimizar errores. Durante las pruebas y testeos del sistema, los problemas detectados fueron fácilmente identificables y rápidamente solucionados gracias a la robusta base de diseño establecida previamente.

El desarrollo de este proyecto no solo implicó el aprendizaje del lenguaje de programación Dart y el framework Flutter, sino también implementar la arquitectura limpia en el código e integrar Firebase y SQLite como bases de datos locales. Aunque inicialmente fue un desafío familiarizarse con estas herramientas, una vez superada la curva de aprendizaje, su adopción fue fluida y eficiente.

La evaluación realizada por los pequeños productores agrícolas demostró que el aplicativo móvil cumple con los requerimientos planteados, ofreciendo una solución confiable y efectiva para la gestión de sus activos. El resultado de las encuestas demostró que el aplicativo no solo es funcional, sino también de fácil uso y un incentivo para el uso frecuente de este, al haber sido utilizado diariamente por todos los usuarios encuestados entre 15 y 30 minutos en el transcurso de las pruebas de aceptación.

6. Recomendaciones

Durante el desarrollo de este proyecto, se identificaron diversas áreas que podrían beneficiarse de mejoras adicionales o ajustes para optimizar el rendimiento y la experiencia general del usuario. Estas recomendaciones están diseñadas para abordar específicamente estas necesidades identificadas, con el objetivo de fortalecer la efectividad y eficiencia del sistema implementado. A continuación, se detallan las áreas clave de enfoque y las recomendaciones específicas destinadas a maximizar el impacto positivo del proyecto en su contexto operativo y de usuarios:

- Agregar opción de tutorial en la que se instruya dentro de la aplicación el uso de la aplicación y toda la terminología utilizada.
- Crear un interfaz administrador dentro de la aplicación que permita monitorear las acciones de los usuarios.

- Implementar un informe específico para los gastos e ingresos de un activo que permita ver de manera concreta todos los gastos e ingresos realizados, con la descripción y fecha registrada.
- Permitir descargar el informe en un archivo Excel para un mejor análisis de los datos por parte de los usuarios.
- Estudiar la posibilidad y viabilidad de implementar pagos por medio de la aplicación, de tal manera que un gasto o un ingreso se registre automáticamente en la aplicación.
- Subir la aplicación a Google Play Store facilitará la obtención de un token válido para usar Storage y monitorizar las peticiones con FirebaseAppCheck.
- Hacer que la fuente utilizada en la aplicación sea ajustable para garantizar una visualización clara y legible en diferentes dispositivos móviles.
- Asegurarse que la cedula y el número de teléfono brindados por el usuario sean únicos.
- Revisar y ajustar la terminología utilizada en la aplicación con el fin de ser más comprensible y accesible para todos los usuarios, mejorando así la experiencia de usuario.
- Idear un plan para mantener al usuario usando la aplicación al finalizar los niveles y no tenga más misiones por realizar

Referencias Bibliográficas

- Aguilera Castillo, A., Fúquene Lozano, C. A., & Ríos Pineda, W. F. (2014). Aprende jugando: el uso de técnicas de gamificación en entornos de aprendizaje. <https://cutt.ly/yXb49CM>
- Alonso, A. (2021, junio 25). Flutter y el patrón Provider. DEV Community. <https://dev.to/hextiandro/flutter-y-el-patron-provider-33c3>
- Alonso, N. (2021, 13 octubre). ¿Qué es el software libre? Características y ventajas. Grupo Atico34. Recuperado 23 de marzo de 2024, de <https://cutt.ly/ySJQVtJ>
- Android Developers. (s. f.). Android Studio. Recuperado 14 de julio de 2024, de <https://developer.android.com/studio?hl=es-419>
- Android Developers. (s. f.). Cómo crear y administrar dispositivos virtuales. Recuperado 15 de julio de 2024, de <https://developer.android.com/studio/run/managing-avds?hl=es-419>
- Arjun, M. (2020, marzo 24). *One step ahead into methodologies*. SYD 701 Class Blog - Arjun Madhavankutty. <https://arjunmadhavankuttysyd701.wordpress.com/2020/03/24/topic-4-one-step-ahead-into-methodologies/>
- Barroso, L. (s. f.). Flutter: arquitectura clean usando BLoC. Paradigmadigital.com. Recuperado 14 de julio de 2024, de <https://www.paradigmadigital.com/dev/flutter-arquitectura-clean-usando-bloc/>
- Borrás Gené, O. (2015). Fundamentos de la gamificación. Universidad Politécnica de Madrid. Recuperado 23 de marzo de 2022, de <https://cutt.ly/MSJKGce>
- Campión, R. S. (2015). Gamificación: Cómo motivar a tu alumnado y mejorar el clima en el aula. ResearchGate. Recuperado 22 de marzo de 2022, de <https://cutt.ly/CSJQL4p>

Cloud Firestore. (s. f.). Firebase. Recuperado 14 de julio de 2024, de <https://firebase.google.com/products/firestore?hl=es>

Cloud Storage para. (s. f.). Firebase. Recuperado 14 de julio de 2024, de <https://firebase.google.com/docs/storage?hl=es-419>

DANE. 3er CNA (2014). Censo nacional agropecuario Colombia. <https://cutt.ly/oSJQJq5>

Dart overview. (s. f.). Dart.dev. Recuperado 14 de julio de 2024, de <https://dart.dev/overview>

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, 28 septiembre). From Game Design Elements to Gamefulness: Defining Gamification. ResearchGate. Recuperado 23 de marzo de 2022, de <https://cutt.ly/jSJQX4O>

Fernández, J. S. (2020, abril 27). *El patrón BLoC en Clean Architecture*. XurxoDev. <https://xurxodev.com/el-patron-bloc-en-clean-architecture/>

Firebase Authentication. (s. f.). Firebase. Recuperado 14 de julio de 2024, de <https://firebase.google.com/products/auth?hl=es>

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

Gamificación: el aprendizaje divertido | educativa. (s. f.). Educativa. Recuperado 15 de diciembre de 2021, de <https://cutt.ly/gSJKKUj>

Gestiona los proyectos de tu equipo desde cualquier lugar. (s. f.). Trello.com. Recuperado 14 de julio de 2024, de <https://trello.com/es>

Hernández, E. S. G. (2008). Sistema Automatizado para el Control de Activos Fijos Tangibles. Universidad Central «Marta Abreu» De Las Villas. Recuperado 23 de marzo de 2022, de <https://cutt.ly/ASJQBhe>

- Herrera, M., Méndez, Y., Tobón, G., & Sierra, A. (2016). Ni pequeño productor, ni agricultor familiar, soy campesino. Dime qué paz quieres y te diré qué campo cosechas. Reflexiones sobre lo rural en los diálogos de La Habana, 149-76. <https://cutt.ly/KSJKRAi>
- IBM Product Master 12.0.0. (2024, junio 11). [IBM.com](https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases). <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>
- Jaramillo, A. A., & Correa-Henao, G. J. (2013, 10 junio). Vista de Construcción de un Aplicativo para el Control De Activos Tecnológicos al Interior de una Organización: Beneficios y Experiencias. Lámpsakos. Recuperado 23 de marzo de 2022, de <https://cutt.ly/VSJkPmM>
- Luis, H. (2021, 24 marzo). ¿Qué es una aplicación móvil? Anincubator Website. <https://anincubator.com/que-es-una-aplicacion-movil/>
- Martins, J. (2024, enero 19). ¿Qué es la metodología Kanban y cómo funciona? Asana. <https://asana.com/es/resources/what-is-kanban>
- Medium. (S. f.). Clean Architecture en Flutter ¿Cómo implementarla en el 2023? Recuperado 14 de julio de 2024, de <https://medium.com/bancolombia-tech/clean-architecture-en-flutter-cómo-implementarla-en-el-2023-34f4d28d06b5>
- Montalvo, J., Barriga Ospina, Ó. A., Rojas Miranda, S. C. (2017). Tecnología aplicada en el agro colombiano: asimetrías evidenciadas en la competitividad entre los subsectores café y maíz durante el periodo 2005–2015. <https://cutt.ly/dSJQPbY>
- Oliveira, W., & Bittencourt, I. I. (2019). Tailored gamification to educational technologies (Vol. 10, pp. 978-981). Springer Singapore. <https://doi.org/10.1007/s13278-019-0589-3>
- Provider. (s. f.). Dart Packages. Recuperado 14 de julio de 2024, de <https://pub.dev/packages/provider>

Salazar, A., Zapata-Madrigal, G. D., & García Sierra, R. (2021). Electrofun: una aplicación basada en gamificación para apoyar la implementación del sistema de gestión de activos ISO 55000 en Codensa. Scielo. Recuperado 23 de marzo de 2022, de <http://www.scielo.org.co/pdf/inde/v39n1/2145-9371-inde-39-01-138.pdf>

Sociedad de Agricultores de Colombia. (2019, 7 marzo). TIC para el agro SAC - Sociedad de Agricultores de Colombia. Recuperado 15 de diciembre de 2021 de <https://sac.org.co/tic-para-el-agro/>

Sqflite. (s. f.). Dart Packages. Recuperado 14 de julio de 2024, de <https://pub.dev/packages/sqflite>

Tecno digital. (2023, julio 2). Metodología de Prototipo para el Desarrollo de Software. <https://informatecdigital.com/desarrollo/metodologia-de-prototipo-para-el-desarrollo-de-software/>

Tecnologías de la Información y las Comunicaciones (TIC). (s. f.). Gov.co. Recuperado 14 de julio de 2024, de <https://www.mintic.gov.co/portal/715/w3-article-5755.html>

Trello. (s.f.). *Gestiona los proyectos de tu equipo desde cualquier lugar*. Trello.com. <https://trello.com/es>

Vite Cevallos, H., Montero, K., & Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales: Revista Multidisciplinaria de Investigación. <https://shorturl.at/d4mz0>

Apéndices

Apéndice A. Casos de uso

Se detallarán los casos de uso del aplicativo, los cuales describen las interacciones entre los usuarios y el sistema para cumplir con los diferentes objetivos y funcionalidades del proyecto.

Tabla 3.

Caso de uso registrar usuario #1.

Caso de uso: Registrar usuario por formulario
Descripción: Permite al usuario registrarse mediante un formulario de registro
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener correo para registrarse
Flujo normal: <ol style="list-style-type: none"> 1. El usuario descarga e inicia la aplicación 2. Se presenta el formulario de registro 3. Oprime registrarse en la aplicación 4. Rellena el formulario con sus datos: nombre completo, correo electrónico y contraseña 5. El usuario oprime el botón “registrarse”. 6. Se presenta la pantalla principal al usuario
Postcondición: El usuario se registra e inicia sesión con su correo y contraseña
Flujo alternativo: <ul style="list-style-type: none"> • Si los datos del formulario son inválidos, se muestra un mensaje de error y el usuario permanece en la pantalla de registro

Tabla 4*Caso de uso Registrar Usuario #2.*

Caso de uso: Registrar Usuario (proveedor Google)
Descripción: permite a un usuario registrarse utilizando una cuenta de Google.
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener cuenta Google para registrarse
Flujo normal: <ol style="list-style-type: none"> 1. El usuario descarga e inicia la aplicación 2. Oprime registrarse en la aplicación 3. El usuario selecciona registro con Google <ol style="list-style-type: none"> 3.1. Se dirige a la aplicación de Google y se muestra una lista de cuentas Google 3.2. Selecciona una cuenta Se presenta la pantalla principal al usuario
Postcondición: El usuario se registra e inicia sesión con su cuenta Google
Flujo alternativo: Si la autenticación con Google falla, se muestra un mensaje de error y el usuario permanece en la pantalla de registro

Tabla 5.*Caso de uso Registrar Usuario #3*

Caso de uso: Registrar Usuario (proveedor Facebook)
Descripción: permite a un usuario registrarse utilizando una cuenta de Facebook.
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener cuenta Google para registrarse
Flujo normal: <ol style="list-style-type: none"> 1. El usuario descarga e inicia la aplicación 2. Oprime registrarse en la aplicación 3. El usuario selecciona registro con Facebook <ol style="list-style-type: none"> 3.1. Se dirige a la aplicación de Facebook 3.2. El usuario autoriza el registro 4. Se presenta la pantalla principal al usuario
Postcondición: El usuario se registra e inicia sesión con su cuenta Facebook
Flujo alternativo: <ul style="list-style-type: none"> • Si la autenticación con Facebook falla, se muestra un mensaje de error y el usuario permanece en la pantalla de registro

Tabla 6.*Caso de uso Iniciar Sesión #1*

Caso de uso: Iniciar sesión.
Descripción: El usuario ingresa a la aplicación con sus credenciales
Actores: Usuario (Campesino).
Precondiciones: El usuario debe estar registrado.
Flujo normal: <ol style="list-style-type: none"> 1. Entra en la aplicación. 2. Se visualiza el formulario de inicio de sesión. 3. El usuario digita su correo y la contraseña. 4. Oprime el botón “ingresar”. Se visualiza la pantalla principal.
Postcondición: El usuario inicia sesión con su correo y contraseña
Flujo alternativo: Si los datos del formulario son inválidos, se muestra un mensaje de error y el usuario permanece en pantalla de inicio de sesión.

Tabla 7*Caso de uso Iniciar Sesión #2*

Caso de uso: Iniciar sesión. (proveedor Google)
Descripción: El usuario ingresa a la aplicación con una cuenta Google
Actores: Usuario (Campesino).
Precondiciones: El usuario debe estar registrado.
Flujo normal: <ol style="list-style-type: none"> 1. Entra en la aplicación. 2. Se presenta la pantalla de inicio de sesión. <ol style="list-style-type: none"> 2.1. Oprime Ingreso con Google: <ol style="list-style-type: none"> 2.1.1. Se dirige a la aplicación de Google y se muestra una lista de cuentas Google 2.1.2. Selecciona la cuenta de Google 3. Se visualiza la pantalla principal.
Postcondición: El usuario inicia sesión con su cuenta Google
Flujo alternativo: <ul style="list-style-type: none"> • Si la autenticación con Google falla, se muestra un mensaje de error y el usuario permanece en pantalla de inicio de sesión.

Tabla 8*Caso de uso Iniciar Sesión #3*

Caso de uso: Iniciar sesión. (proveedor Facebook)
Descripción: El usuario ingresa a la aplicación con una cuenta Facebook
Actores: Usuario (Campesino).
Precondiciones: El usuario debe estar registrado.
Flujo normal: <ol style="list-style-type: none"> 1. Entra en la aplicación. 2. Se presenta la pantalla de inicio de sesión. <ol style="list-style-type: none"> 2.1. Oprime Ingreso con Facebook: <ol style="list-style-type: none"> 2.1.1. Se dirige a la aplicación de Facebook 2.1.2. El usuario autoriza el inicio de sesión Se visualiza la pantalla principal.
Postcondición: El usuario inicia sesión con su cuenta Facebook
Flujo alternativo: Si la autenticación con Facebook falla, se muestra un mensaje de error y el usuario permanece en pantalla de inicio de sesión.

Tabla 9*Caso de uso Cerrar sesión*

Caso de uso: Cerrar sesión.
Descripción: Permite a un usuario cerrar sesión en la aplicación.
Actores: Usuario (Campesino).
Precondiciones: El usuario debe estar registrado e iniciar sesión en la aplicación.
Flujo normal: <ol style="list-style-type: none"> 1. Entra en la aplicación. 2. El usuario se dirige a la parte superior izquierda y oprime el botón de menú. 3. Se visualiza las diferentes opciones en el menú. 4. Se dirige a la parte inferior izquierda y oprime el botón salir. 5. Se visualiza la pantalla de inicio de sesión.
Postcondición: La sesión del usuario se ha cerrado y el usuario es redirigido a la pantalla de inicio de sesión.
Flujo alternativo: <ul style="list-style-type: none"> • Si ocurre un error durante el proceso de cierre de sesión, se muestra un mensaje de error.

Tabla 10.*Caso de uso Agregar activo #1.*

Caso de uso: Agregar activo (primera vez).
Descripción: Permite al usuario agregar un activo por primera vez.
Actores: Usuario (Campesino).
Precondiciones: El usuario debe tener la aplicación, estar registrado e iniciar sesión en la aplicación.
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. En la pantalla principal, el usuario oprime el botón en la parte inferior derecha. 3. Se visualiza en pantalla un blank site con un botón para agregar nuevo activo. 4. El usuario oprime el botón “agregar nuevo activo”. 5. Llena los datos del activo (Nombre del activo, descripción del activo y selección la foto(opcional)). 6. El usuario oprime el botón de “Confirmar”. Se presenta en la pantalla de activos
Postcondición: El usuario agrega correctamente su primer activo.
Flujo alternativo: Si ocurre un error agregando el activo, se muestra un mensaje de error.

Tabla 11*Caso de uso Agregar activo #2.*

Caso de uso: Agregar activo.
Descripción: Permite al usuario agregar una activo.
Actores: Usuario (Campesino).
Precondiciones: El usuario debe tener la aplicación, estar registrado e iniciar sesión en la aplicación.
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. En la pantalla principal, el usuario se dirige a la parte inferior derecha y oprime el botón. 3. Se visualiza en pantalla una lista de los activos del usuario. 4. Repite pasos (4-7) del caso de uso agregar activo (primera vez).
Postcondición: El usuario agrega correctamente un activo.
Flujo alternativo: <ul style="list-style-type: none"> • Si ocurre un error agregando el activo, se muestra un mensaje de error.

Tabla 12*Editar datos del activo*

Caso de uso: Editar datos del activo.
Descripción: Permite al usuario agregar y/o editar datos de un activo.
Actores: Usuario (Campesino).
Precondiciones: El usuario debe tener la aplicación, estar registrado, iniciar sesión en la aplicación y haber registrado un activo.
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. Se visualiza la pantalla principal home. 3. Se dirige a la parte inferior derecha y oprime el botón de los activos. 4. Se visualiza la vista de los activos actuales. 5. El usuario escoge el activo a eliminar y selecciona el botón para editarlo(lápiz). 6. se visualizan los datos que puede editar o agregar 7. Selecciona los cambios y oprime el botón “Confirmar”. Se visualiza la vista de los detalles del activo con los cambios realizados.
Postcondición: El usuario actualiza correctamente el activo.
Flujo alternativo: Si ocurre un error actualizando el activo, se muestra un mensaje de error.

Tabla 13*Caso de uso Eliminar activo.*

Caso de uso: Eliminar activo.
Descripción: Permite al usuario eliminar un activo.
Actores: Usuario (Campesino).
Precondiciones: El usuario debe tener la aplicación, estar registrado, iniciar sesión en la aplicación y haber registrado un activo.
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. Se visualiza la pantalla principal Home. 3. Se dirige a la parte inferior derecha y oprime el botón de los activos. 4. Se visualiza la vista de los activos actuales.

Caso de uso: Eliminar activo.
<ol style="list-style-type: none"> 5. El usuario escoge el activo a eliminar y selecciona el botón para editarlo(lápiz). 6. Se visualizan los datos del activo (nombre, descripción y fecha de creación), oprime el botón para eliminarlo(X). 7. Se visualiza una alerta que confirma si se desea eliminar el activo, oprime el botón confirmar. 8. Se visualiza la vista de los activos actuales.
Postcondición: El usuario elimina el activo deseado.
Flujo alternativo:
<ul style="list-style-type: none"> • Si ocurre un error al eliminar el activo, se muestra un mensaje de error.

Tabla 14.*Caso de uso Registrar ingresos por activos*

Caso de uso: Registrar ingresos por activos
Descripción: Permite al usuario registrar un ingreso por activo
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado, iniciar sesión en la aplicación y haber registrado un activo.
Flujo normal:
<ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. Se visualiza la pantalla principal home. 3. El usuario se dirige a la parte central de la pantalla, y oprime el botón “Nuevo Ingreso”. 4. Se visualiza una vista tipo formulario con el nombre “Nuevo Ingreso” 5. Selecciona el activo con el que obtuvo un ingreso, la cantidad gastada, el concepto con la descripción del ingreso y la fecha del ingreso. 6. Se dirige a la parte inferior y oprime el botón “Confirmar”. 7. Se visualiza la vista principal.
Postcondición: El usuario registra sus ingresos correctamente en el activo seleccionado
Flujo alternativo:
<ul style="list-style-type: none"> • Si ocurre un error agregando el ingreso, se muestra un mensaje de error.

Tabla 15*Caso de uso Registrar gastos por activo*

Caso de uso: Registrar gastos por activo
Descripción: Permite al usuario registrar los gastos de un activo
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado, iniciar sesión en la aplicación y haber registrado un activo.
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. Se visualiza la pantalla principal Home. 3. El usuario se dirige a la parte central de la pantalla, y oprime el botón “Nuevo gasto”. 4. Se visualiza una vista tipo formulario con el nombre “Nuevo gasto” 5. Selecciona el activo con el que obtuvo un ingreso, la cantidad gastada, el concepto con la descripción del ingreso y la fecha del gasto. 6. Se dirige a la parte inferior y oprime el botón “Confirmar”. Se visualiza la vista principal.
Postcondición: El usuario registra sus gastos correctamente en el activo seleccionado.
Flujo alternativo: Si ocurre un error agregando el gasto, se muestra un mensaje de error.

Tabla 16*Generar un informe de activos.*

Caso de uso: Generar un informe de activos
Descripción: Permite al usuario crear un informe de cada activo
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado, iniciar sesión en la aplicación y haber registrado un activo.
Flujo normal:

Caso de uso: Generar un informe de activos
<ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. El usuario se dirige a la parte inferior derecha y selecciona el botón de activos. 3. El usuario oprime el botón de ver informes. 4. Se visualiza la pantalla de informe general de todos los activos: <ol style="list-style-type: none"> 4.1. El usuario selecciona la barra búsqueda en la parte superior y escoge el activo deseado. 4.2. El usuario selecciona las fechas que le interesan. 5. Se visualiza la pantalla de informes con los filtros aplicados.
Postcondición: El usuario genera y visualiza el informe de acuerdo con los filtros.
Flujo alternativo: Si ocurre un error generando el informe de activos con filtros, se muestra la tabla de activos con datos sin filtro.

Tabla 17.

Caso de uso descargar informe de activos

Caso de uso: Descargar informe de activos
Descripción: Permite al usuario descargar el informe de los activos
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado, iniciar sesión en la aplicación y haber registrado un activo.
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. El usuario se dirige a la parte inferior derecha y selecciona el botón de activos. 3. El usuario oprime el botón de ver informes. 4. se visualiza la pantalla de informe general de todos los activos. <ol style="list-style-type: none"> 4.1. El usuario selecciona la barra búsqueda en la parte superior y escoge el activo deseado. 4.2. El usuario selecciona las fechas que le interesan. 4.3. Se visualiza la pantalla de informes con los filtros aplicados. 5. El usuario selecciona el botón de “Descargar” en la parte superior de la vista. Se muestra en pantalla una notificación del estado de la descarga.
Postcondición: El usuario descarga el informe del activo correctamente.
Flujo alternativo: Si ocurre un error de descarga, se muestra un mensaje de error.

Tabla 18.*Mostrar consejos.*

Caso de uso: Mostrar consejos
Descripción: Permite al usuario mostrar consejos
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado e iniciar sesión en la aplicación.
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar a la aplicación 2. Visualización de la pantalla principal. 3. El usuario oprime la tarjeta central con la nube. 4. Se observa en pantalla los consejos propuestos.
Postcondición: El usuario visualiza los consejos propuestos

Tabla 19.*Visualizar las misiones.*

Caso de uso: Visualizar las misiones
Descripción: Permite al usuario visualizar las misiones por nivel del usuario
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado e iniciar sesión en la aplicación.
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Visualización de la pantalla principal. 3. El usuario selecciona en la parte inferior central la sección de misiones. 4. Se observa en pantalla las misiones con su respectivo estado.
Postcondición: El usuario observa o busca las misiones actuales.
Flujo alternativo: <ul style="list-style-type: none"> • Si ocurre un error al cargar las misiones, se muestra una pantalla de error y un botón de intentar nuevamente.

Tabla 20.*Visualizar las estadísticas de los activos*

Caso de uso: Visualizar las estadísticas de los activos
Descripción: Permite al usuario visualizar las estadísticas de los activos
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado e iniciar sesión en la aplicación, haber registrado mínimo un activo y agregado mínimo un gasto o ingreso.
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Visualización de la pantalla principal. 3. El usuario oprime en la parte inferior izquierda la sección de estadísticas. 4. Se observa en pantalla las estadísticas de los activos, donde se observa una gráfica con las ganancias diarias, semanales o mensuales, y los activos con mayor o menor gasto, ingreso y ganancias.
Postcondición: El usuario observa las estadísticas de sus activos
Flujo alternativo: <ul style="list-style-type: none"> • Si ocurre un error cargando los activos, se muestra una pantalla de error y un botón de intentar nuevamente. • Si ocurre un error cargando los gastos e ingresos de los activos, se muestra una pantalla de error y un botón de intentar nuevamente.

Tabla 21.*Comparar activos.*

Caso de uso: Comparar activos
Descripción: Permite al usuario comparar activos y ver sus estadísticas
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado e iniciar sesión en la aplicación, haber registrado mínimo un activo y agregado mínimo un gasto o ingreso.
Flujo normal: <ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Visualización de la pantalla principal. 3. El usuario oprime en la parte inferior izquierda la sección de estadísticas.

Caso de uso: Comparar activos
<ol style="list-style-type: none"> 4. Se observa en pantalla las estadísticas de los activos. 5. Oprime el botón de “comparar”. 6. Se presentan una lista de activos para seleccionar 7. El usuario selecciona sus activos y oprime el botón inferior 8. Se presenta la pantalla de comparación, donde se muestra información detallada de los activos seleccionados.
Postcondición: El usuario observa las estadísticas de sus activos seleccionados
Flujo alternativo: <ul style="list-style-type: none"> • Si ocurre un error cargando los gastos e ingresos de los activos seleccionados, se muestra un mensaje de error,

Tabla 22.

Editar datos del usuario.

Caso de uso: Editar datos del usuario
Descripción: Permite al usuario editar sus datos personales
Actores: Usuario (Campesino)
Precondiciones: El usuario debe tener la aplicación, estar registrado, iniciar sesión en la aplicación.
Flujo normal: <ol style="list-style-type: none"> 1. Entrar a la aplicación. 2. Se visualiza la pantalla principal home. 3. El usuario da Tab a las tres líneas en la parte superior izquierda. 4. Oprime el botón de “Tu información”. 5. Se presenta la pantalla perfil con la información del usuario. 6. El usuario oprime el botón inferior de “editar”. 7. El usuario edita sus datos personales 8. Oprime el botón de “confirmar”. 9. Se presenta la información del usuario actualizada y se muestra un mensaje de confirmación.
Postcondición: El usuario actualiza sus datos personales.
Flujo alternativo: <ul style="list-style-type: none"> • Si ocurre un error al actualizar los datos del usuario, se muestra un mensaje de error. • Si el usuario oprime el botón de “cancelar”, se mantienen los datos y se mantiene en la pantalla de perfil.

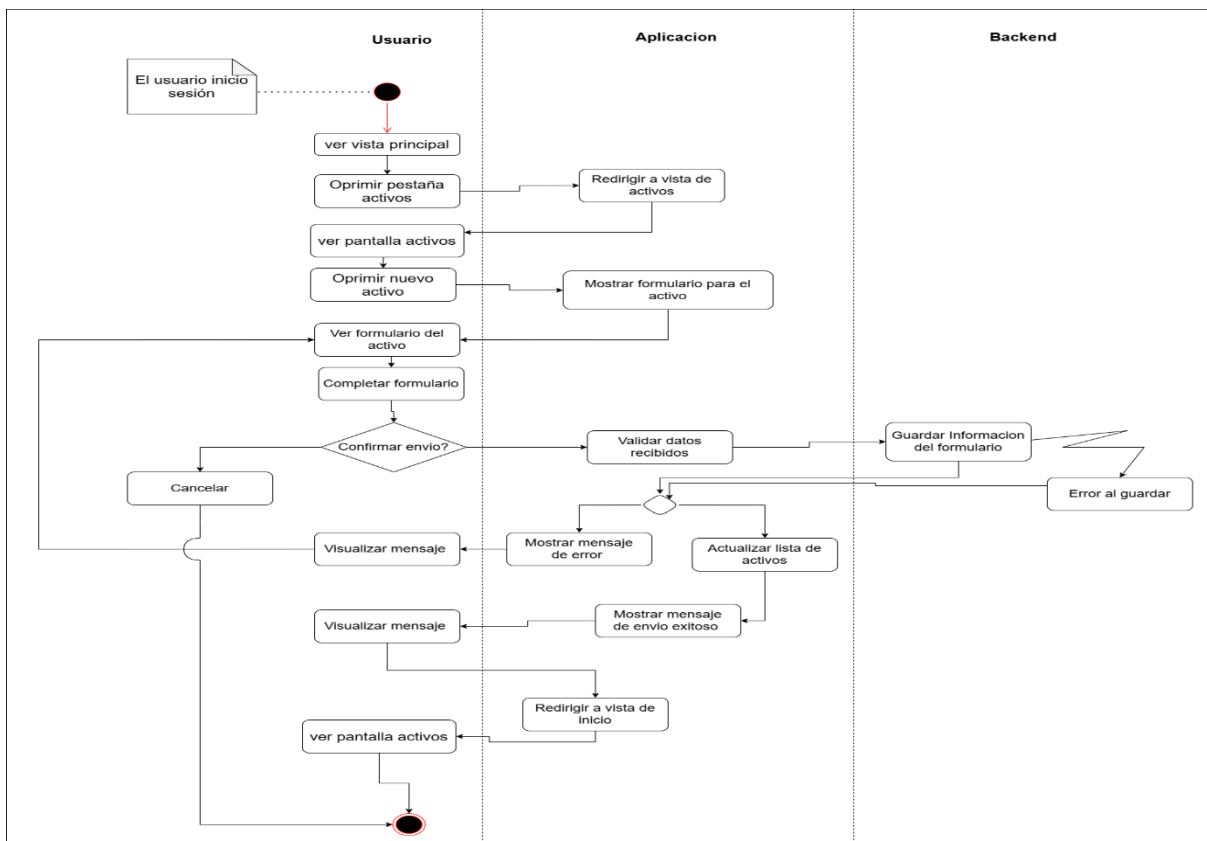
Apéndice B. Diagramas de actividades

En esta sección, se incluyen los diagramas de actividades de la aplicación móvil, correspondientes a los casos de uso 'agregar activo', 'cerrar sesión', 'generar estadísticas', 'descargar historial', 'eliminar activo' y 'actualizar usuario'.

La figura 27 representa el diagrama de actividades para el caso de uso 'agregar activo', que consiste en un formulario que el usuario debe llenar y que el sistema debe validar y guardar en la base de datos.

Figura 27

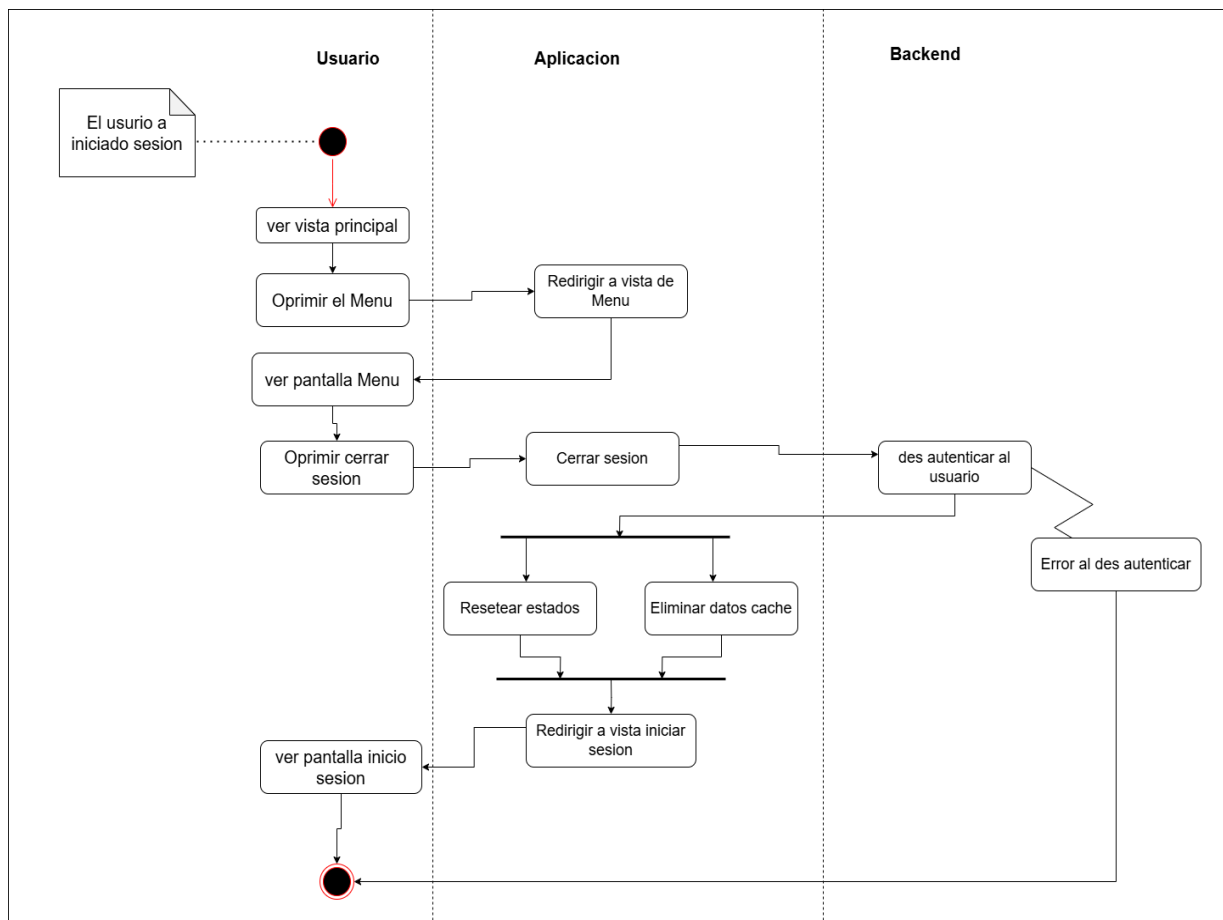
Diagrama de actividad Agregar Activo



En la figura 28 se presenta el diagrama de actividades de 'Cerrar sesión'. Incluye pasos como la navegación del usuario hacia la opción de cierre de sesión, la confirmación de la acción por parte del usuario y la limpieza de datos temporales en el dispositivo.

Figura 28

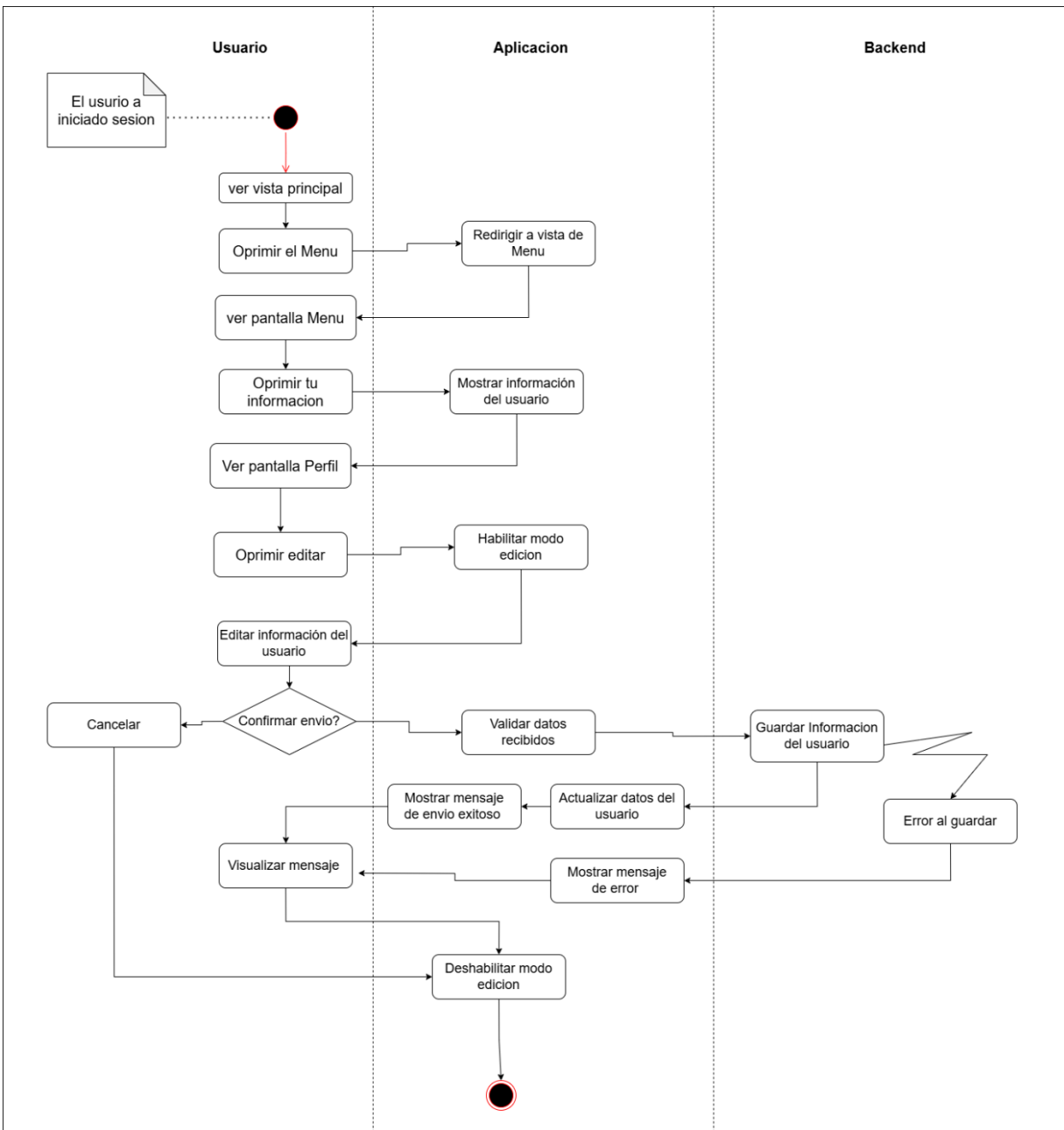
Diagrama de actividad Cerrar Sesión



A continuación, se muestra la figura 29, que representa el diagrama de actividades para la edición de datos del usuario. En este diagrama, se detalla el proceso mediante el cual el usuario realiza modificaciones en el formulario con sus datos personales, asegurándose de ingresar la información de manera precisa y actualizada.

Figura 29

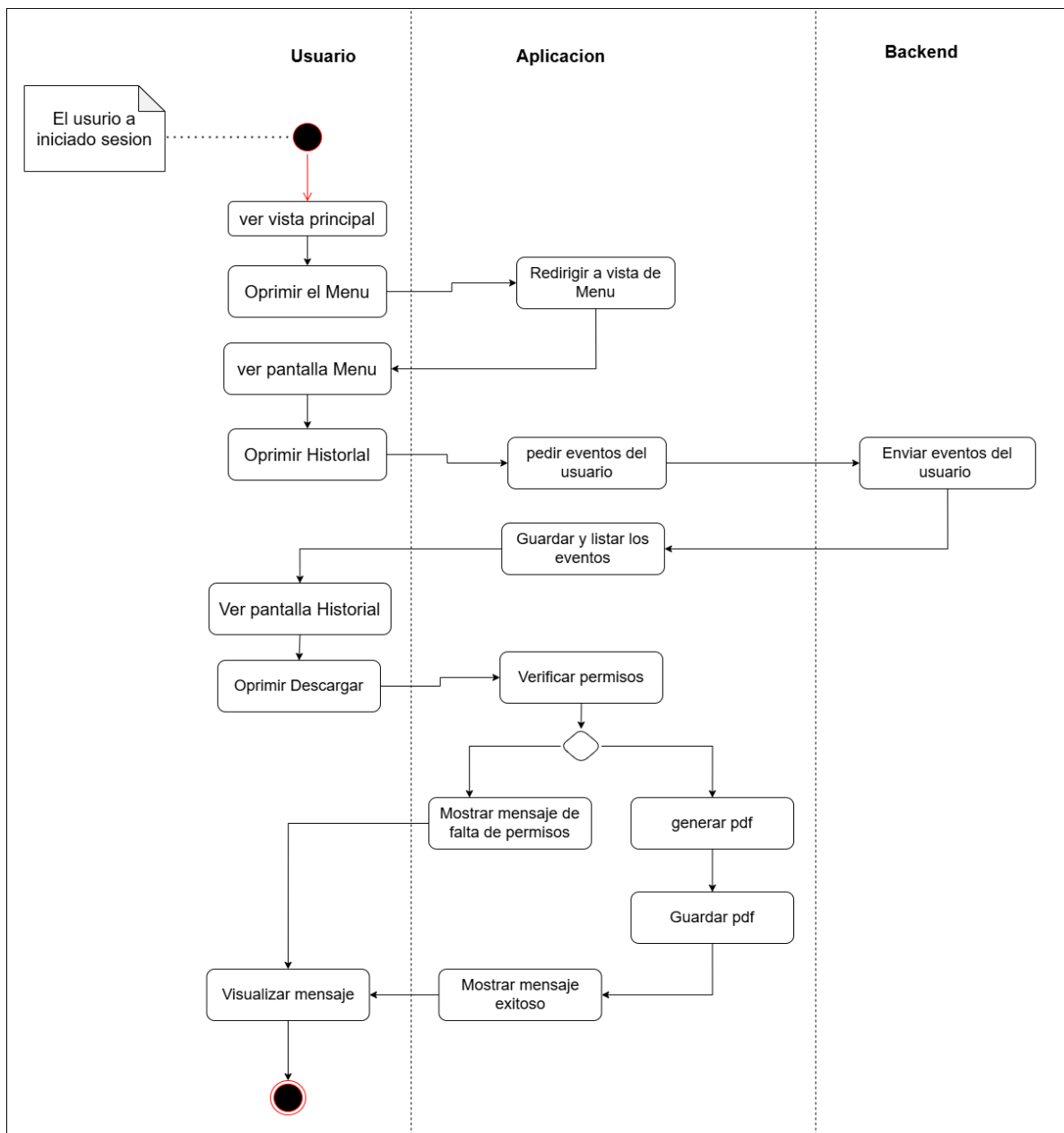
Diagrama de Actividad Editar Información del usuario



La Figura 30 presenta el diagrama de actividades para la descarga del historial. En este proceso, el usuario otorga los permisos necesarios y se genera un archivo PDF que contiene una tabla con todos los eventos registrados por el usuario.

Figura 30

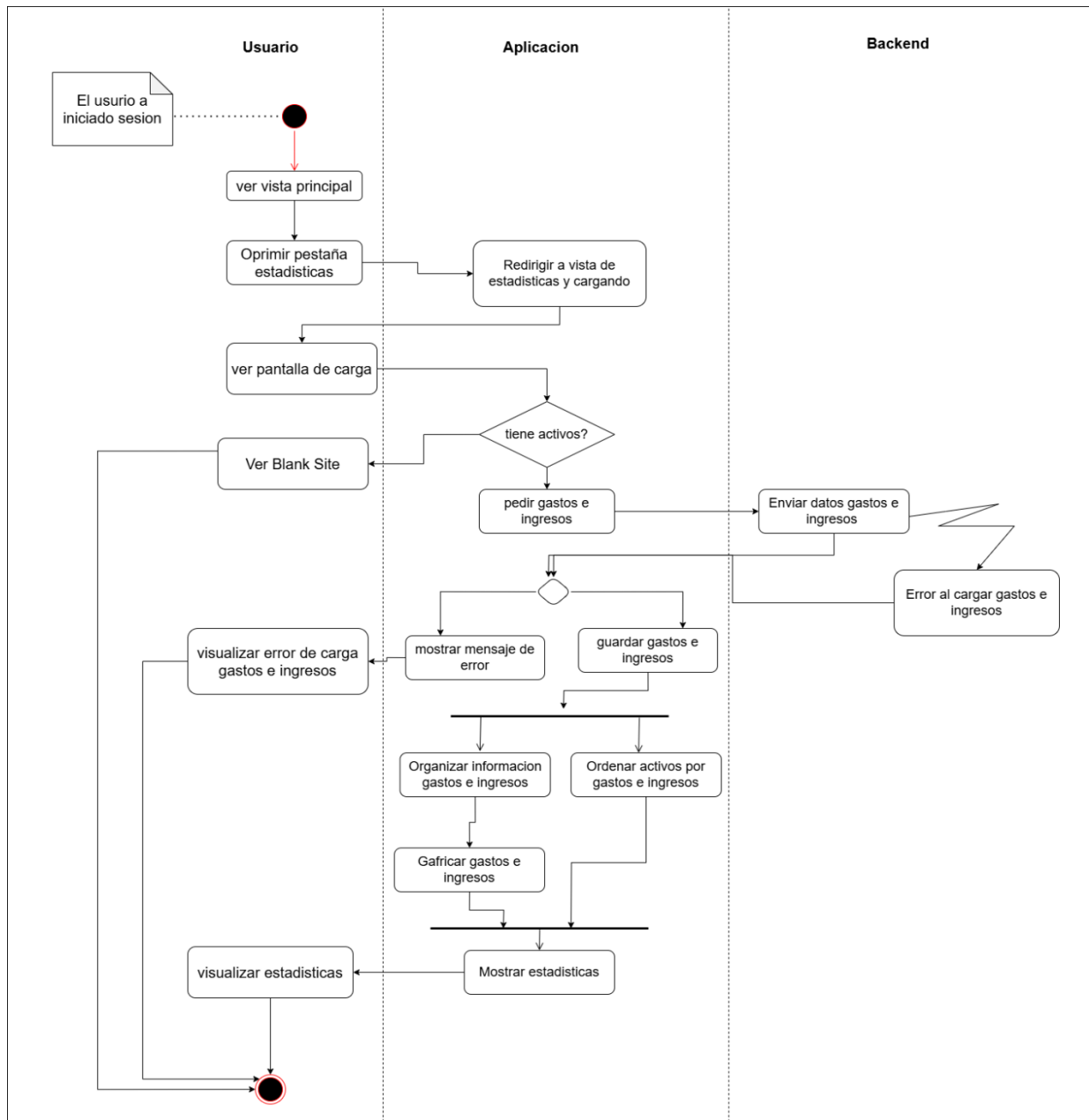
Diagrama de Actividad Descargar Historial



La figura 31 corresponde al diagrama de actividades correspondiente al caso de uso "Visualizar estadísticas". En este proceso, el usuario accede a la sección de estadísticas y el sistema procesa los datos de gastos e ingresos del usuario.

Figura 31

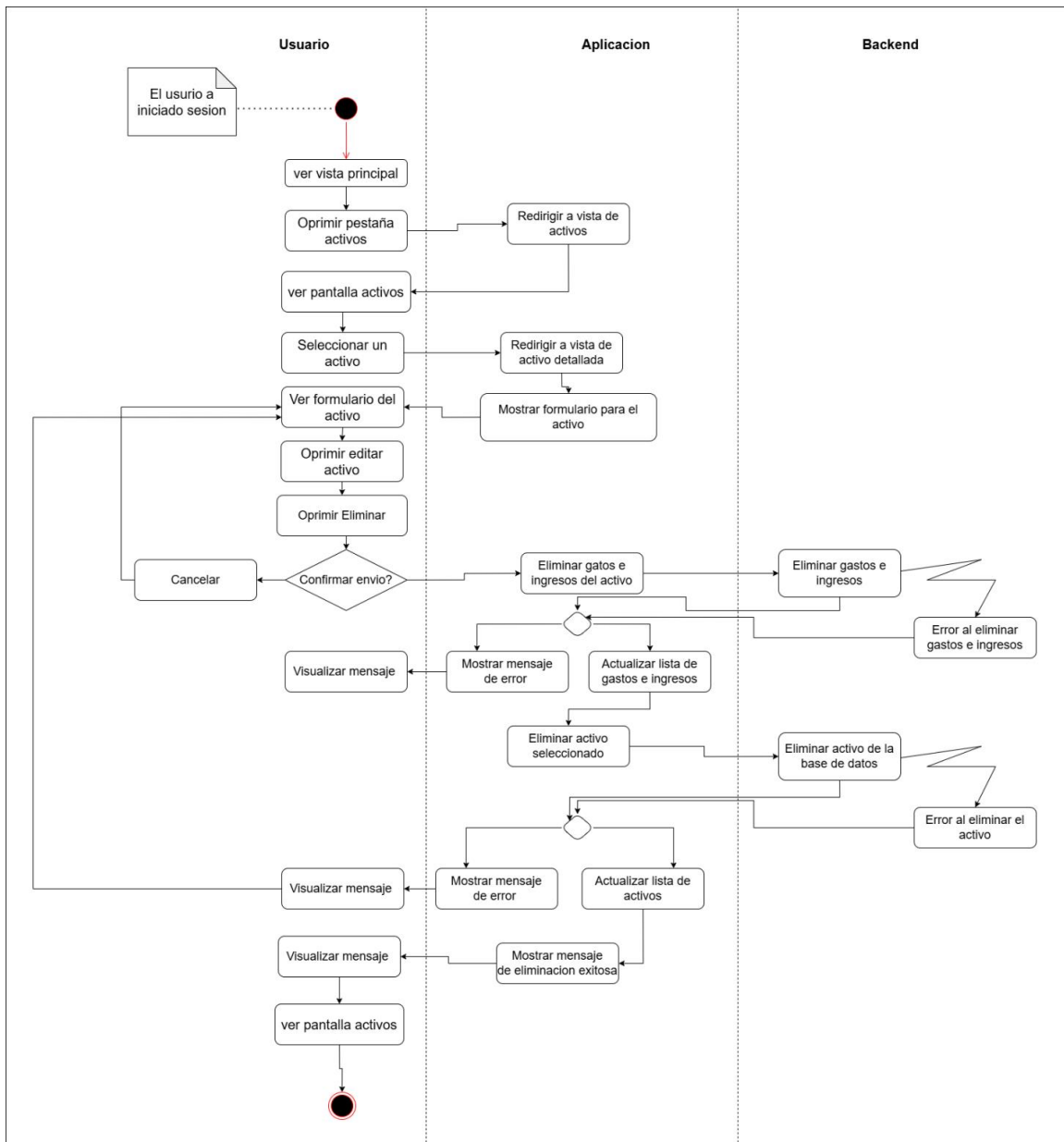
Diagrama de Actividad Visualizar Estadísticas.



El caso de uso "Eliminar activo" se representa en la figura 32, donde el usuario edita el activo y confirma su eliminación, lo cual elimina el activo y, de forma cascada, también elimina sus gastos e ingresos correspondientes asociados.

Figura 32

Diagrama de Actividad Eliminar Activo



Apéndice C. Tablas de pruebas

A continuación, se muestran las evidencias del plan de pruebas para evaluar el funcionamiento del aplicativo.

I. Pruebas de Compatibilidad

Tabla 23

Pruebas de compatibilidad.

Dispositivos	Resolución de pantalla (px)	Sistema operativo	Resultados de la prueba
Motorola G8	1080 x 2280	Android 9.0 Pie	Funcionamiento sin problemas
Xiaomi redmi note 8	1080 x 2340	Android 9.0	Compatibilidad exitosa, con problemas con habilitar permiso de almacenamiento
Píxel XL	1440 x 2560	Android 8.1 oreo	Funcionamiento sin problemas
Samsung Galaxy J7	720 x 1280	Android 5.1	Funcionamiento sin problemas
Samsung Galaxy A02	720 x 1600	Android 10	Problemas menores de visualización en la interfaz

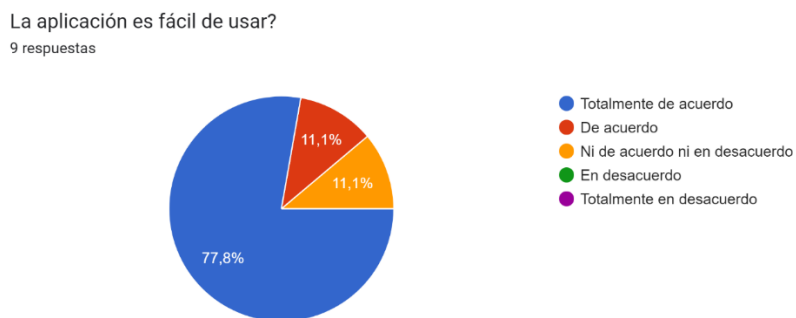
II. Prueba de aceptación.

En esta prueba se realizaron encuestas a usuarios escogidos para que utilizaran la aplicación. Para los resultados de la encuesta se realizó una encuesta por medio de Google forms de 17 preguntas de selección múltiple y fueron contestadas por 9 personas.

La figura 33 fue la primera pregunta de las encuestas “La aplicación es fácil de usar?”, se puede observar que el 77,8% de los usuarios están totalmente de acuerdo, el 11,1% de los usuarios están de acuerdo y a su vez un 11,1% de los usuarios no están ni de acuerdo ni en desacuerdo

Figura 33.

Pregunta #1, encuesta para los usuarios.

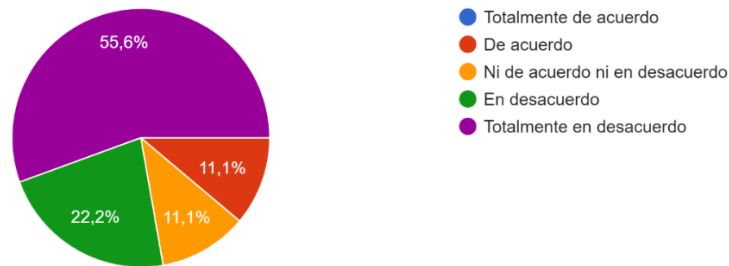


La figura 34 pregunta “los términos utilizados en la aplicación fueron difíciles de entender?”, se observa que un 55,6% y un 22,2% de los usuarios están totalmente en desacuerdo y en desacuerdo respectivamente, solo un 11,1% de los usuarios están de acuerdo con la dificultad en el entendimiento de los términos utilizados en la aplicación.

Figura 34

Pregunta #2, encuesta para los usuarios.

Los términos utilizados en la aplicación fueron difíciles de entender?
9 respuestas

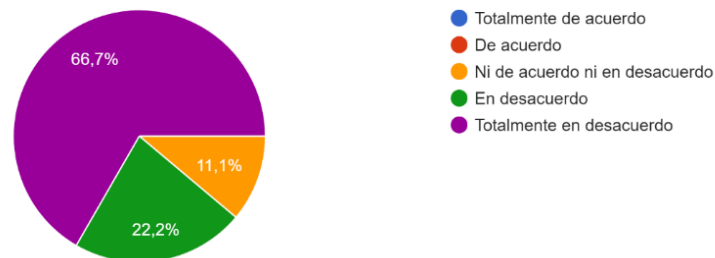


La figura 35 pregunta "fue difícil consultar los ingresos, gastos o activos una vez registrados?" donde se observa que un 66,7% y un 22,2% de los usuarios están totalmente en desacuerdo y en desacuerdo respectivamente, solo un 11,1% de los usuarios indicaron que no estaban de acuerdo ni en desacuerdo.

Figura 35

Pregunta #3, encuesta para los usuarios.

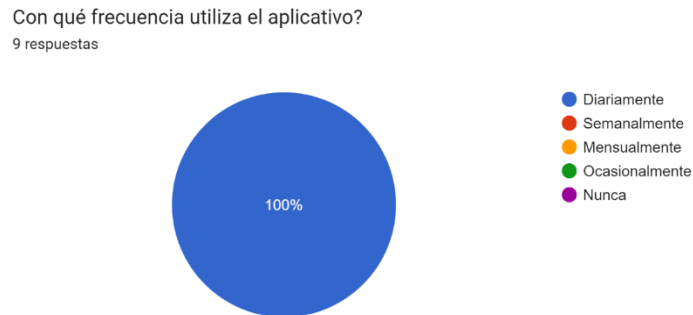
Fue difícil consultar los ingresos, gastos o activos una vez registrados?
9 respuestas



La Figura 36 pregunta “Con qué frecuencia utiliza el aplicativo?”, donde se puede observar que todos los usuarios respondieron positivamente a la pregunta.

Figura 36

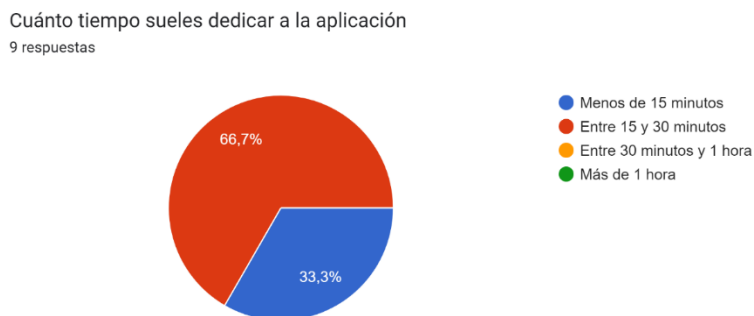
Pregunta #4, encuesta para los usuarios.



La figura 37 pregunta “Cuánto tiempo sueles dedicar a la aplicación?”, donde se puede observar que el 66,7% de los usuarios utilizaron la aplicación entre 15 y 30 minutos, y un 33,3% de los usuarios la utilizaron menos de 15 minutos diariamente.

Figura 37

Pregunta #5, encuesta para los usuarios.

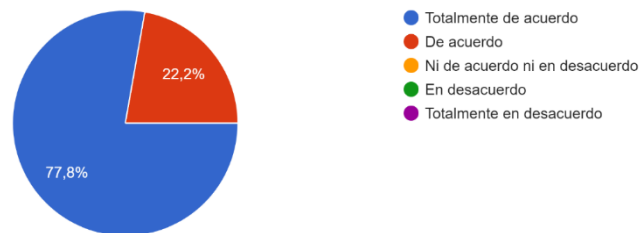


La figura 38 pregunta “Considera usted que los niveles en la aplicación fueron un incentivo para usarla con frecuencia?”, donde se observa que el 77,8% y el 22,2% de los usuarios se mostraron totalmente de acuerdo y de acuerdo respectivamente.

Figura 38

Pregunta #6, encuesta para los usuarios.

Considera usted que los niveles en la aplicación fueron un incentivo para usarla con frecuencia?
9 respuestas



En la figura 39 se presenta la pregunta 7 realizada a los usuarios en la que se pregunta por la facilidad del proceso en el registro de un activo, donde se observa que un 77,8% de los usuarios consideran que no representa una dificultad el registro de los activos.

Figura 39

Pregunta #7, encuesta para los usuarios.



En la figura 40 se presenta la pregunta 8 realizada a los usuarios en las que se le pregunta por el proceso de registro de un gasto o ingreso en la aplicación, donde se observa que un 77,8% de los usuarios consideran que no es una dificultad el registro de un gasto o ingreso en la aplicación.

Figura 40.

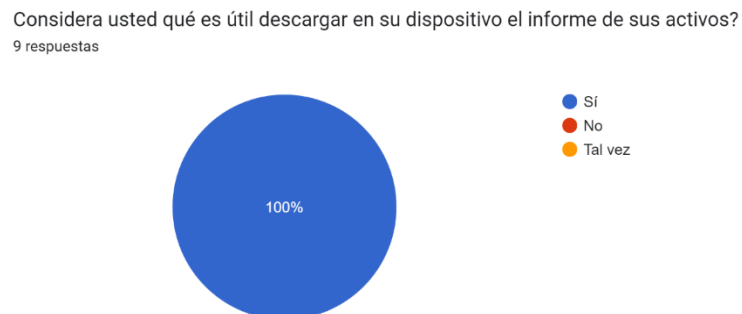
Pregunta #8, encuesta para los usuarios.



La figura 41 pregunta “Considera usted que es útil descargar en su dispositivo el informe de sus activos?”, donde se observa que todos los usuarios afirmaron la utilidad de la descarga de los informes en la aplicación.

Figura 41

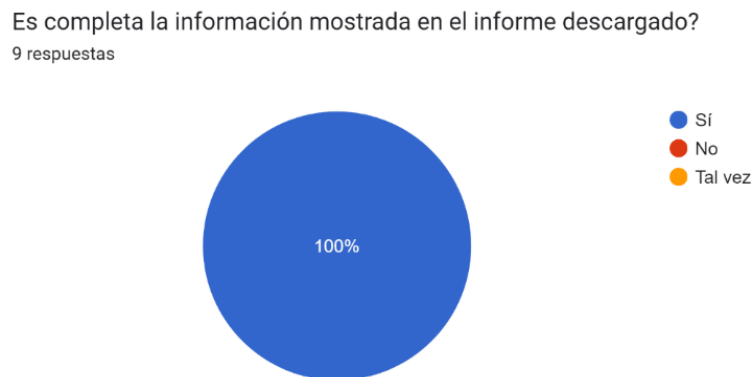
Pregunta #9, encuesta para los usuarios.



La figura 42 pregunta “Es completa la información mostrada en el informe descargado?”, donde se observa que todos los usuarios afirmaron que la información mostrada en el archivo descargado es completa.

Figura 42

Pregunta #10, encuesta para los usuarios.



La figura 43 se presenta la pregunta “Qué tan satisfecho está con la interfaz de usuario del aplicativo móvil?”, donde se observa que un 77,8% de los usuarios se muestran muy satisfechos

con la interfaz de usuario, y un 22,2% de los usuarios indicaron que se sienten satisfechos con la interfaz.

Figura 43

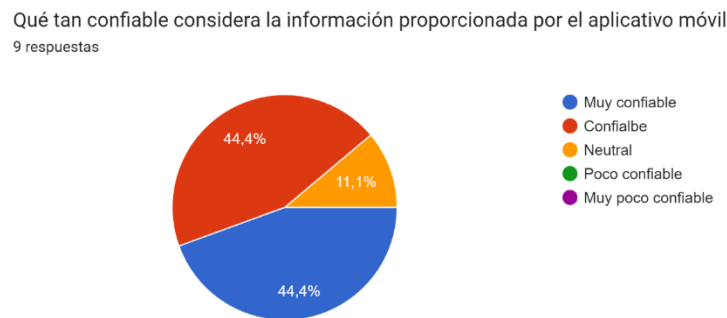
Pregunta #11, encuesta para los usuarios.



La figura 44 se presenta la pregunta “¿Qué tan confiable considera la información proporcionada por el aplicativo móvil?”, donde se observa que un 44,4% de los usuarios indicaron que el aplicativo es muy confiable y confiable respectivamente. Solo un 11,1% de los usuarios se consideraron neutrales a la pregunta.

Figura 44

Pregunta #12, encuesta para los usuarios.

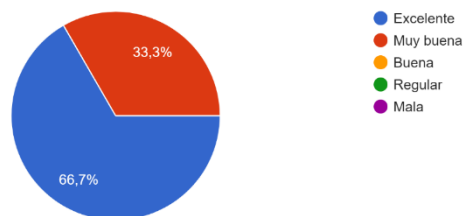


La figura 45 se presenta la pregunta” Cómo califica el rendimiento y la velocidad de la aplicación en términos de respuesta y fluidez mientras utiliza la aplicación?”, donde se observa que un 66.7% de los usuarios consideran excelente y un 33,3% de los usuarios consideraron muy bueno el rendimiento y velocidad del aplicativo en sus teléfonos celulares.

Figura 45

Pregunta #13, encuesta para los usuarios.

¿Cómo califica el rendimiento y la velocidad de la aplicación en términos de respuesta y fluidez mientras utilizas la aplicación?
9 respuestas

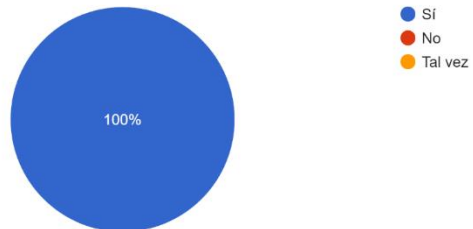


La figura 46 presenta la última pregunta de la encuesta “Considera usted que la aplicación facilita llevar la contabilidad de sus activos”, donde se observa que todas las respuestas de los usuarios fueron afirmativas.

Figura 46

Pregunta #14, encuesta para los usuarios.

Considera usted que la aplicación facilita llevar la contabilidad de sus activos
9 respuestas



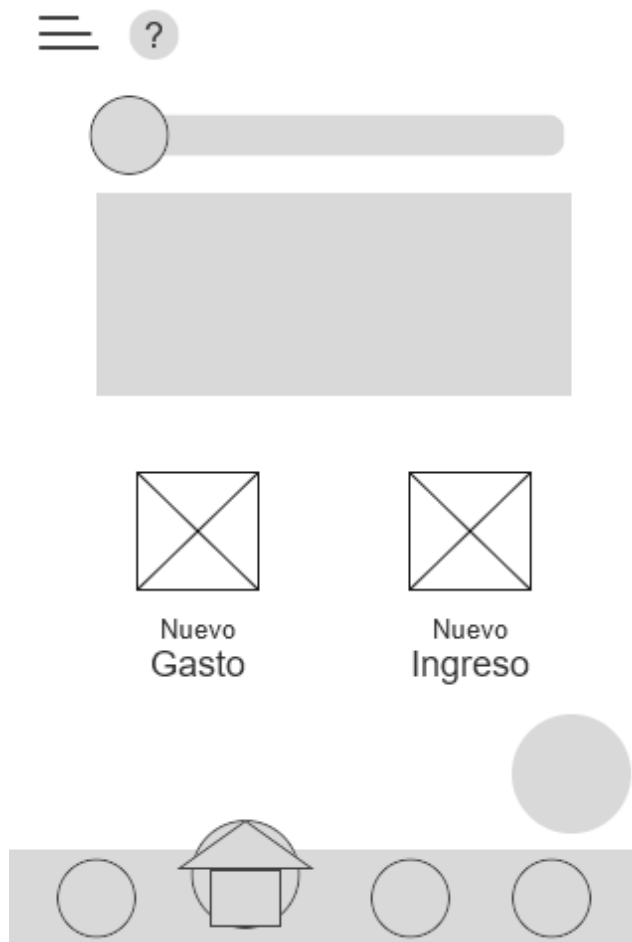
Apéndice D. Mockups y Wireframes

Se presentan en este apéndice los primeros prototipos realizados de la aplicación. En la creación de los prototipos primeramente se realizaron wireframes, estos son una guía visual que nos ayuda a representar de mejor medida la estructura de la aplicación

A continuación, se muestran los Wireframes más importantes de la aplicación:

Figura 47

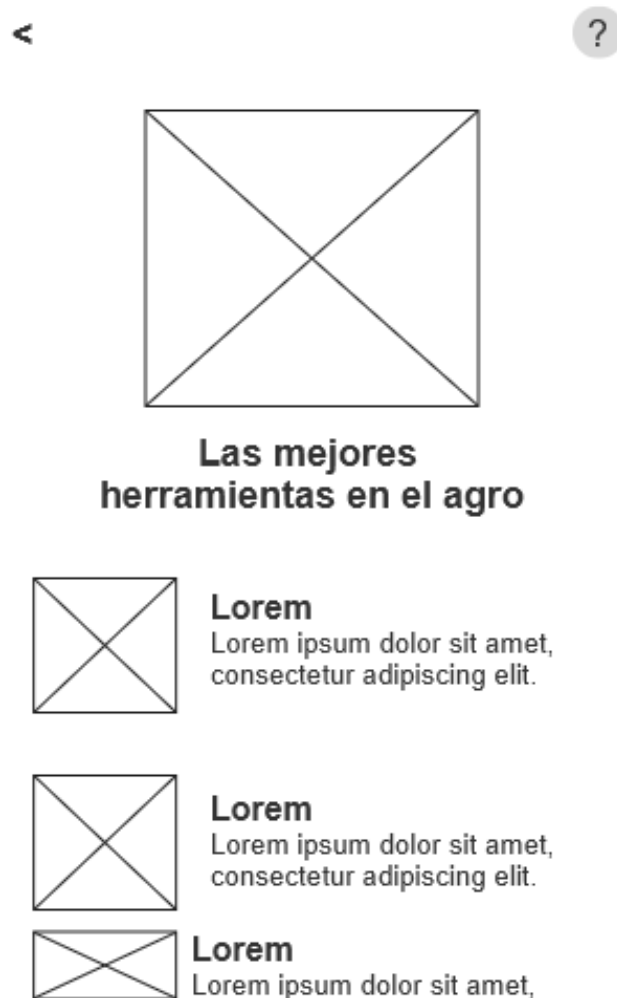
Wireframe Vista Inicio.



La figura 47 es la representación de la vista de usuario que se planteó al momento de iniciar la aplicación.

Figura 48

Wireframe Mostrar consejos.



La figura 48 representa la vista propuesta para los consejos que se listarían hacia los usuarios. Con esta vista se buscaba brindar consejos diarios para que los usuarios pudieran consultarlos y tomar nota de ellos

Figura 49*Wireframe Vista activos.*

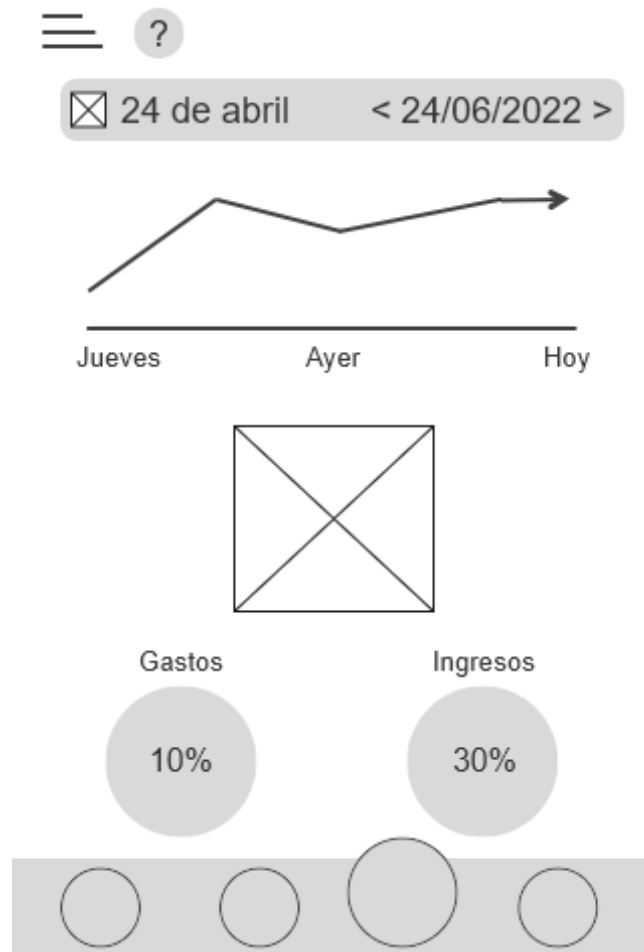
La figura 49 representa la vista de los activos, en la que se buscaba listar los activos que tendría el usuario, la opción de ver informes, así como la opción de buscar un activo en específico.

Figura 50

Wireframe Vista perfil del activo.



La figura 50 representa la vista del perfil de los activos propuesta, en la que se observa la descripción dada por el usuario, la fecha de registro del activo, su usabilidad y sus ganancias

Figura 51*Wireframe Vista Estadísticas.*

La figura 51 representa la vista de estadísticas propuesta, en la que se busca mostrarle a través de un gráfico, las ganancias que tengan todos sus activos, a su vez de un porcentaje de gastos e ingresos.

Figura 52

Wireframe Vista Perfil de usuario.



La figura 52 representa la vista de perfil propuesta, en la que se lista la información básica del usuario, así como los ajustes, historial y términos y condiciones de la aplicación. Se tiene propuesto un botón el cual los puede dirigir a la vista de ayuda, en caso de que tengan alguna pregunta.

Figura 53

Wireframe Vista Actualización de datos del usuario.

<

Tu info

Nombre
Alberto Ramirez Guzman

Celular
3152247250

CC ▾

N.Documento
456078612

Correo
ramirez.g56@gmail.com

Guardar

La figura 53 representa la vista de actualización de datos del usuario propuesta, en la cual se lista la información del usuario, tal como su nombre, celular, número de documento y correo electrónico, se busca que en esta vista propuesta el usuario pueda actualizar esos datos y guardarlos.

Figura 54

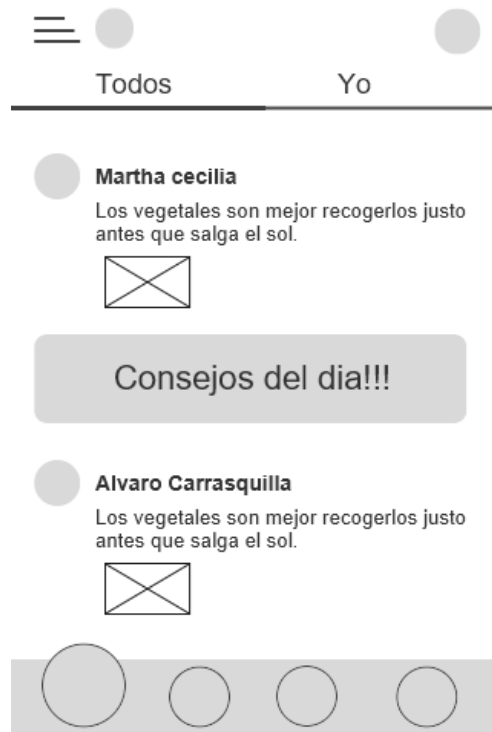
Wireframe Vista Terminos y Condiciones.



La figura 54 representa la vista de términos y condiciones de la aplicación.

Figura 55

Wireframe Vista comunidad propuesta.



La Figura 55 representa la vista de la comunidad propuesta, se contempló la posibilidad de crear una comunidad, en la cual los usuarios se pudieran comunicar los unos con los otros para llevar un registro de sus activos, así como saber cómo van los demás usuarios con sus activos.

Figura 56

Wireframe Vista Historial de las acciones del usuario.

The wireframe shows a mobile application screen for viewing user action history. At the top left is a back arrow, and at the top right is a help icon (question mark). The title 'Historial' is centered. Below the title is a date filter input field with the placeholder 'DD/MM/YYYY'. The main content is a table with two columns: 'Nombre' and 'Fecha'. The table contains seven rows of data. Below the table is a 'Ver mas' button.

Nombre	Fecha
Creacion de cuenta	02/03/2022
Registra 1 activo	15/04/2022
Completa mision	21/04/2022
Comenta	23/04/2022
Subes de nivel	02/05/2022
Modificacion activo	04/05

La figura 56 representa el historial de las acciones del usuario, en la cual puede observar la creación o modificación realizada en la aplicación, a su vez, se tiene un filtro por fecha en la que puede observar las acciones realizadas en un periodo de tiempo.

Figura 57

Wireframe Vista listado de misiones.



La figura 57 representa las misiones del usuario propuestas, en esta vista el usuario puede visualizar las misiones que tiene que realizar para poder progresar con el uso de la aplicación, teniendo misiones diarias y de la comunidad.

Figura 58

Wireframe Vista Gestión de informes de los activos del usuario.

< Informes ?

new value 1 DD/MM/YYYY

Nombre	Gastos	Ingresos	F. Inicial	ganancia
Galpon	\$85.000	\$340.000	12/08/18	\$275.000
Tractor	\$150.000	\$500.000	30/10/21	\$350.000
Tractor	\$150.000	\$500.000	30/10/21	\$350.000
Tractor	\$150.000	\$500.000	30/10/21	\$350.000
Tractor	\$150.000	\$500.000	30/10/21	\$350.000
Tractor	\$150.000	\$500.000	30/10/21	\$350.000

Ver mas

La figura 59 representa la vista de los informes propuesta, en la cual el usuario puede observar los gastos, ingresos, y ganancias de sus activos, a su vez puede filtrar por fecha y por activos para poder descargarlo en un archivo pdf.

Figura 59*Wireframe Vista Registro de gasto.*

< Nuevo gasto ?

Seleccionar activo

Tipo de gasto
new value 1

Cantidad gastada
\$40.0000
valor total \$40.000

Concepto
Reparaciones y cambio de gasolina

Metodo de pago
new value 1

¿Cada cuanto se repite este gasto?
Se registrara automaticamente en tu balance de acuerdo de la frecuencia que escojas

Frecuencia de pago
new value 1

Fecha de pago
Hoy, 20 de junio 2022
Pago unico

Confirmar

La figura 60 representa la vista de registro de un gasto propuesta. En esta vista el usuario tiene que especificar su activo, registrar la cantidad de gasto, la fecha de registro y la descripción del gasto. Se contempló la posibilidad de crear un gasto que se mantuviera mes a mes.

Figura 60*Wireframe Vista Registro de ingreso.*

< Nuevo ingreso ?

Seleccionar activos involucrados

Cantidad gastada
\$40.0000

valor total \$40.000

Concepto
Reparaciones y cambio de gasolina

Metodo de pago

new value 1

Fecha de pago
Hoy, 20 de junio 2022

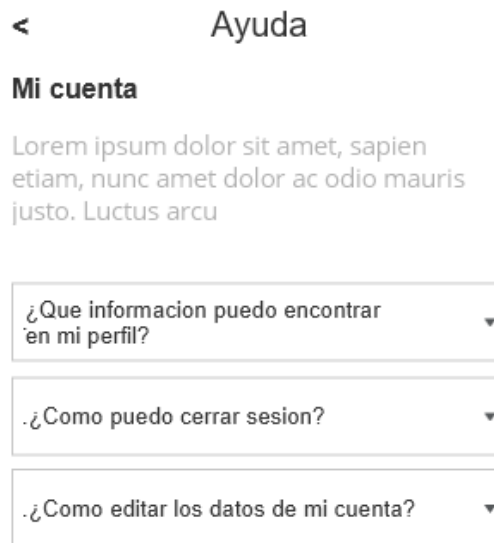
Pago unico

Confirmar

La figura 61 representa la vista de registro de un ingreso propuesta. En esta vista el usuario tiene que especificar el activo del ingreso, la cantidad realizada, la fecha del ingreso y su descripción,

Figura 61

Wireframe Vista Ayuda.



La figura 61 representa la vista de ayuda propuesta. En esta vista el usuario puede revisar las preguntas comunes que pueden tener los usuarios al momento del uso del aplicativo. A su vez, si su pregunta no está formulada en la vista, el usuario puede mandar un correo electrónico para la solución de su pregunta.

- **Mockups**

Se utilizaron mockups para visualizar eficazmente el diseño de la aplicación, esto con el fin de tener una mejor idea al momento de desarrollar la interfaz de usuario, que servirán para el desarrollo del aplicativo. Estos mockups fueron realizados en el editor Figma, y representan un prototipo maquetado de la aplicación

Los mockups de las pantallas más importantes se encuentran a continuación:

Figura 62

Mockup Vista de Inicio.



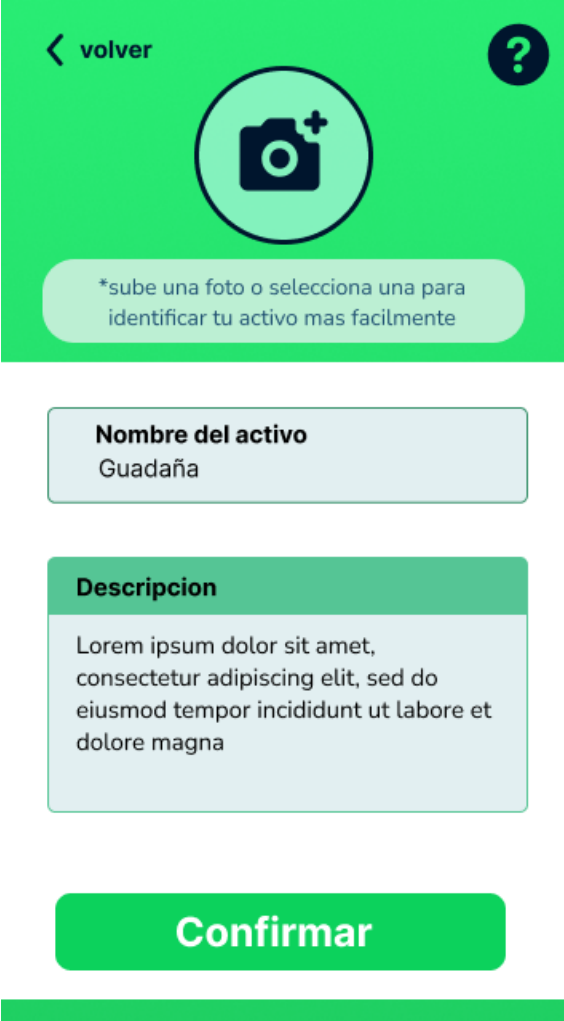
En la figura 62 se observa la vista de usuario de Inicio. En ella el usuario puede crear un nuevo gasto o ingreso, dirigirse a la vista de activos, estadísticas, consejos, misiones y su perfil de usuario. También puede dirigirse a la vista de ayuda con el botón de la parte superior derecha.

Figura 63*Mockup Vista de Activos*

En la figura 63 se observa la vista de los activos, en donde el usuario puede visualizar sus activos, dirigirse a los informes de los activos, añadir un nuevo activo, dirigirse al inicio, a estadísticas, ayuda y perfil.

Figura 64*Mockup Vista comparación de activos*

En la figura 64 se observa la vista de comparación de activos, en la que se comparan dos activos, a su vez, se observa el resultado de la comparación, donde se listan los ingresos, gastos y ganancias de cada activo.

Figura 65*Mockup Creación de activo*

La imagen muestra una pantalla de usuario con un fondo verde. En la parte superior izquierda hay un botón con una flecha hacia atrás y el texto "volver". En la parte superior derecha hay un ícono de una pregunta dentro de un círculo negro. En el centro superior hay un ícono de una cámara con un signo "+" dentro de un círculo blanco. Debajo de este ícono hay un recuadro blanco con el texto: "*sube una foto o selecciona una para identificar tu activo mas facilmente".

Debajo de esto hay un recuadro gris claro con el título "Nombre del activo" y el texto "Guadaña".

Debajo de eso hay un recuadro gris claro con el título "Descripcion" y el texto "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna".

En la parte inferior hay un botón verde con el texto "Confirmar".

En la figura 65 se observa la creación de un activo, en el que se encuentra el nombre del activo a crear y la descripción, opcionalmente se puede subir una imagen del activo en cuestión.

Figura 66

Mockup Vista de Consejos.



En la Figura 66 se observa la vista de usuario que se realizó para la visualización de consejos para el usuario, en esta se lista el clima y los consejos del día.

Figura 67

Mockup de Bienvenida a la Aplicación.



En la Figura 67 se observa la vista diseñada para el primer momento en el cual el usuario inicia la aplicación, solo se mostrará al momento de iniciar la aplicación y si no tiene una cuenta abierta actualmente en el dispositivo

Figura 68

Mockup Inicio de Sesión.



En la Figura 68 se observa el inicio de sesión, donde el usuario debe ingresar el correo electrónico. También pueden ingresar por Google o Facebook.

Figura 69

Mockup Crear Cuenta.



Crea tu cuenta!

Nombre Completo

Correo

Contraseña

Repite - Contraseña

Registrate

¿Ya tienes cuenta? [Inicia sesion](#)

o inicia sesion con

G f

En la Figura 69 se observa la vista para la creación de cuenta, donde se requiere el nombre completo del usuario, su correo electrónico y la contraseña para ingresar en la aplicación.

Figura 70*Mockup Estadísticas.*

En la Figura 70 se observa la vista de estadísticas, en esta se listan las ganancias realizadas por todos los activos, los ingresos y gastos registrados por el usuario en la aplicación, a su vez del botón que lo dirige a la comparación de activos. En esta vista se puede dirigir a el inicio, los activos, el perfil y la vista de ayuda. El usuario también puede filtrar por fecha los gastos e ingresos para observar las ganancias en un periodo de tiempo específico.

Figura 71

Mockup Actualización Información de Perfil.

Mockup of a user profile update screen. The interface includes a navigation bar with a back arrow and the text "volver", a help icon (question mark in a circle), and a large green circular profile icon. Below the icon is the section header "TU INFORMACION". The form contains five input fields:

- Nombre:** Alberto Ramirez Guzman
- Celular:** 3152247250
- CC:** (with a dropdown arrow)
- N° Documento:** 456078612 (with a red warning icon)
- Correo:** ramirez.g56@gmail.com

At the bottom of the form is a large blue button labeled "Guardar".

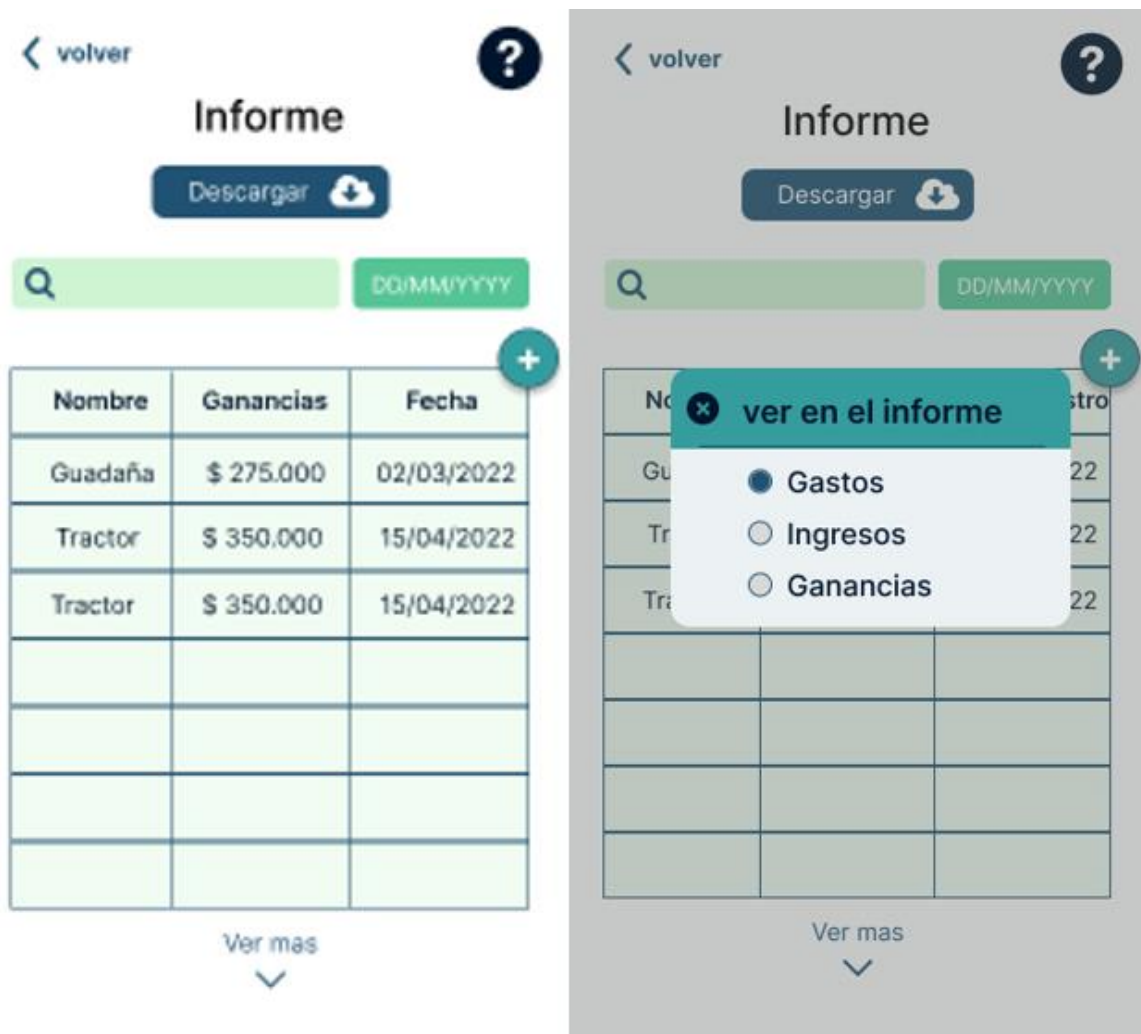
En la Figura 71 se observa el mockup de actualización de datos de perfil del usuario, en el cual puede modificar su nombre, celular, numero de documento y correo electrónico.

Figura 72*Mockup Historial.*

Nombre	Fecha
Creacion de cuenta	02/03/2022
Registrar 1 activo	15/04/2022

Ver mas
↓

En la Figura 72 se observa la vista maquetada del historial de las acciones realizadas por el usuario en la aplicación.

Figura 73*Mockup Informe de Activos.*

En la Figura 73 vista maquetada se observa el informe de activos de la aplicación. En esta vista el usuario puede listar todos sus activos y filtrar si desea ver sus gastos, ingresos o ganancias. A su vez, el usuario puede descargar el documento, que se encontrará en formato de PDF.

Figura 74*Mockup Nuevo Gasto.*

< volver ?

Nuevo Gasto

seleccionar activo

Cantidad gastada
\$ 40.000

valor total	\$40.000
-------------	----------

Concepto
Reparaciones y cambio de gasolina

Fecha de pago

02 / 09 / 2022 ▼

Confirmar

En la Figura 74 se observa la vista maquetada de la creación de un nuevo gasto, donde se encuentra el botón para seleccionar un activo, la cantidad gastada, la descripción del gasto y la fecha en la cual se originó ese gasto.

Figura 75*Mockup Nuevo Ingreso.*

< volver ?

Nuevo Ingreso

seleccionar activo

Cantidad Recibida
\$ 40.000

valor total \$40.000

Concepto
Reparaciones y cambio de gasolina

Fecha de pago
02 / 09 / 2022

Confirmar

En la Figura 75 se observa la vista maquetada de la creación de un nuevo ingreso, donde se lista el botón de seleccionar un activo, la cantidad de dinero recibida, la descripción del ingreso y la fecha en la que se originó ese ingreso.

Figura 76*Mockup Misiones.*

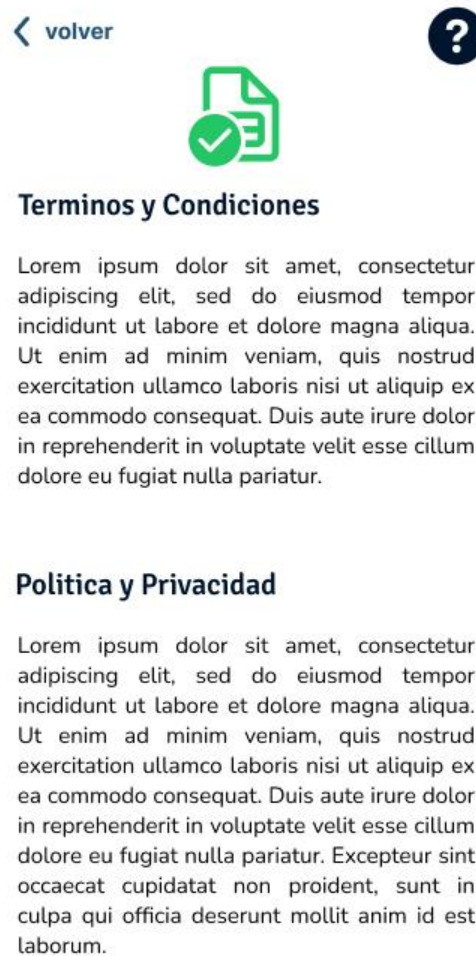
En la Figura 76 se observa la vista maquetada de las misiones, donde se listan las misiones que el usuario deberá realizar para subir de nivel y aumentar el número de activos que tenga en la aplicación.

Figura 77*Mockup Información de Perfil*

En la Figura 77 se observa la vista maquetada de la información de perfil, donde se encuentra listado la foto y el nombre del usuario. A su vez que botones que lo envían a otras vistas, tales como la información del usuario, ajustes de la aplicación, historial del usuario y términos y condiciones y política de privacidad.

Figura 78

Mockup Términos y Condiciones.



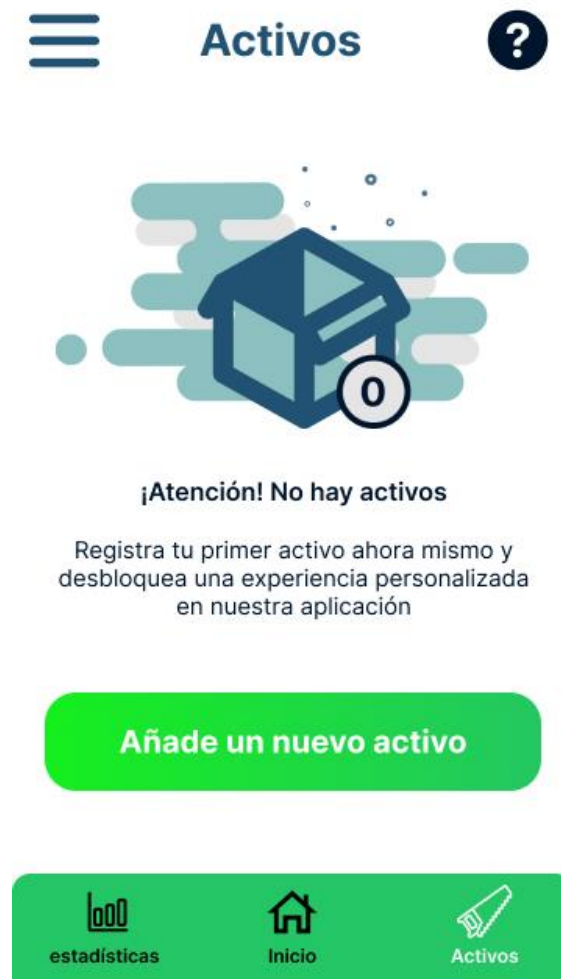
En la Figura 78 se observa la vista maquetada de términos y condiciones, en la cual se encuentra la descripción de los términos y condiciones que los usuarios aceptan una vez comienzan a utilizar la aplicación.

Figura 79

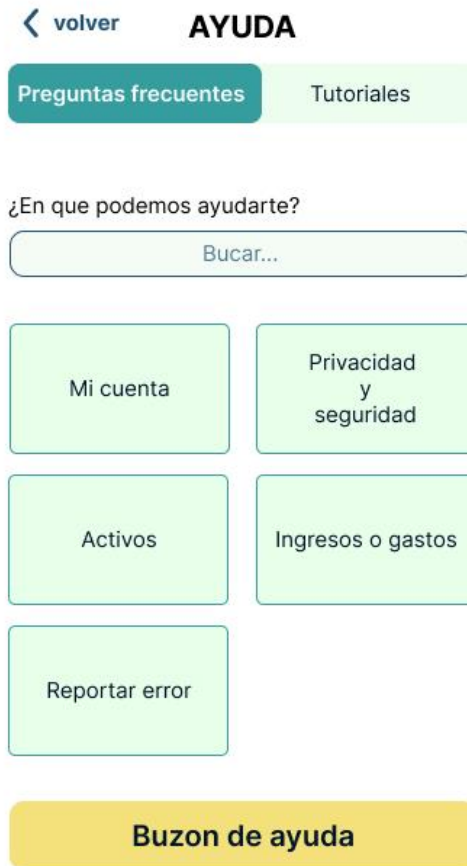
Mockup Información de Activo.



En la Figura 79 se observa la vista maquetada para la información del activo, que lista el nombre del activo, su descripción, fecha de registro, gastos e ingresos registrados en la aplicación

Figura 80*Mockup Activos.*

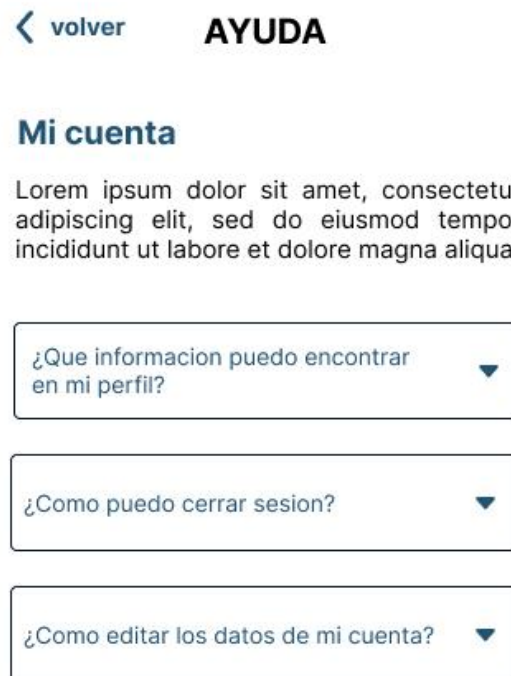
En la Figura 80 se observa la vista maquetada de los activos, en esta vista se destaca la descripción que aparece en la aplicación la primera vez que es usada y no se tiene un activo registrado.

Figura 81*Mockup Ayuda.*

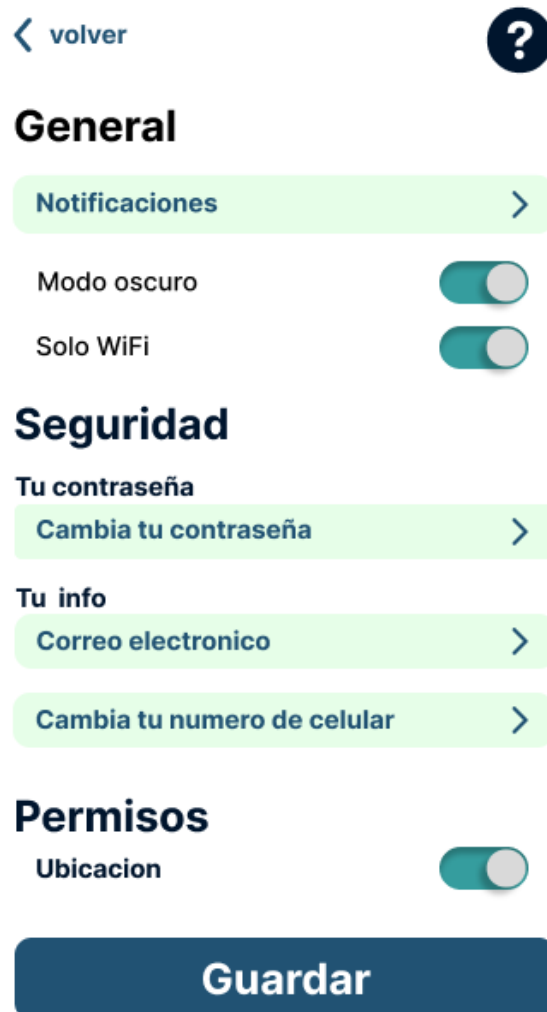
En la Figura 81 se observa la vista maquetada de ayuda, en la cual se lista al usuario las preguntas frecuentes que podrían tener los usuarios la primera vez que utilizan el aplicativo, a su vez de un buzón de ayuda al que se pueden dirigir para enviar sus dudas e inquietudes de la aplicación.

Figura 82

Mockup Ayuda Cuenta.



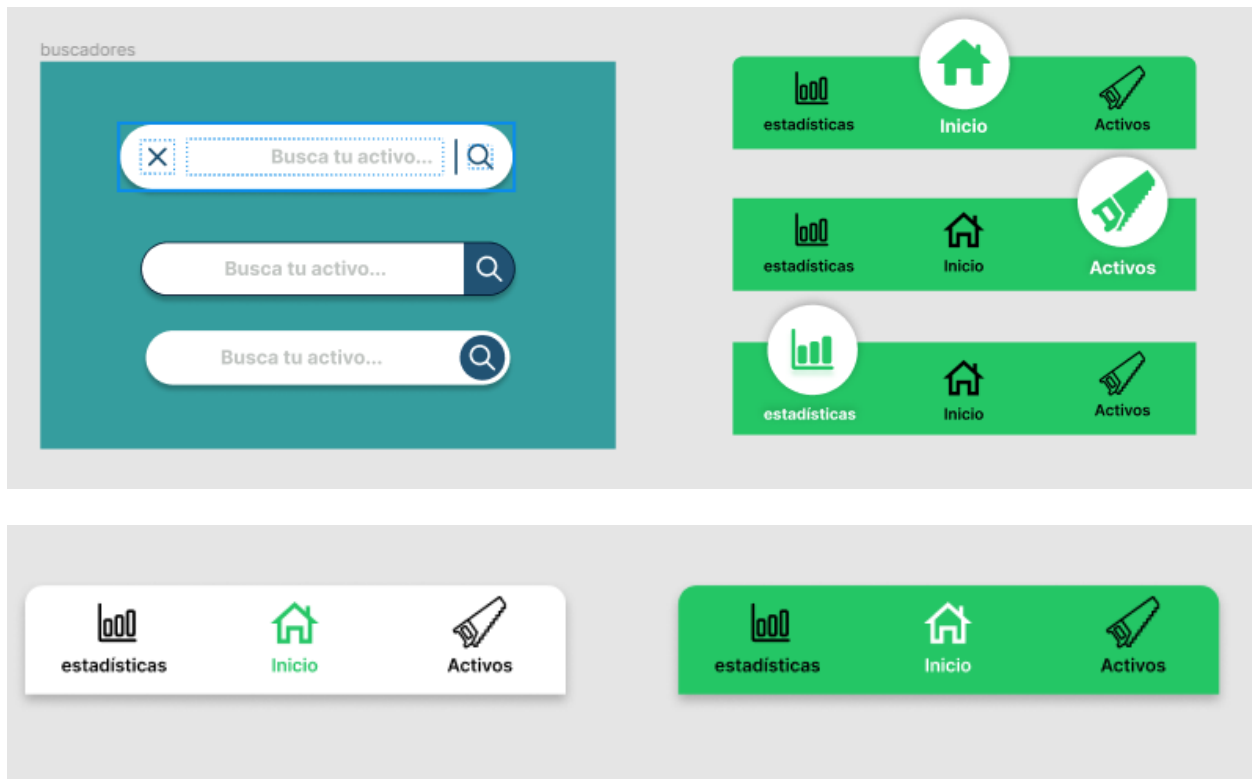
En la Figura 82 se observa la vista maquetada de Ayuda Cuenta; una vez los usuarios entran en cada uno de los cuadros de ayuda, pueden observar las diferentes preguntas frecuentes referentes a la aplicación.

Figura 83*Mockup Configuración.*

En la Figura 83 se observa la vista maquetada de la configuración, donde se lista las configuraciones generales, tales como las notificaciones de la aplicación, el modo oscuro, y Wifi. Las configuraciones de Seguridad y los Permisos de la aplicación.

Figura 84

Mockup NavBar y barras de búsqueda.



En la Figura 84 se observan las distintas ideas realizadas para los NavBar y las barras de búsqueda que tendrían la aplicación

Apéndice E. Vistas de la aplicación

Una vez acabado el proceso de diseño, se realizan las vistas de la aplicación, utilizando la herramienta de Flutter, este es el primer prototipo realizado para los usuarios.


Estas son las vistas del prototipo de la aplicación:

Vista de inicio.

Esta es la vista principal y la vista de inicio una vez se accede a la aplicación, en el AppBar se encuentran los elementos como el menú principal, donde se encuentra la vista de perfil de usuario y le botón de ayuda. En esta vista el usuario puede agregar un ingreso o un gasto, visualizar las misiones, y en el NavBar le permite navegar entre los activos y las estadísticas. Cabe resaltar que el usuario visualizará esta vista una vez haya iniciado sesión en la aplicación.

Figura 85*Vista de Inicio***Registro**

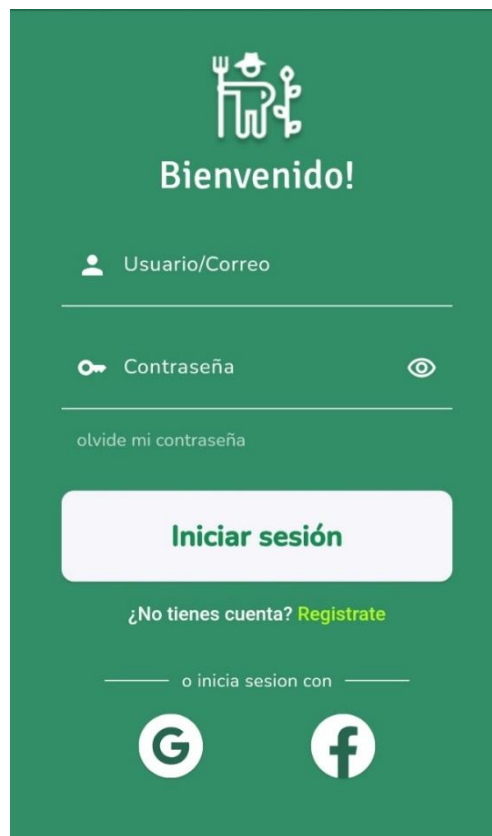
Para registrarse el usuario debe diligenciar los datos listados en la Figura 86, donde tiene que ingresar su nombre, correo electrónico y una contraseña valida

Figura 86*Vista de Registro*

The image shows a registration screen with a green background. At the top, there is a white icon of a person with a hat and a pitchfork. Below the icon, the text "Crea tu cuenta" is displayed. The form consists of four input fields, each with a white underline and a small icon to the left: "Nombre Completo" (person icon), "correo electrónico" (envelope icon), "Contraseña" (key icon), and "Repite la contraseña" (key icon). To the right of the password fields are eye icons for toggling visibility. Below the fields is a white button with rounded corners labeled "Registrarse". Underneath the button, there is a link: "¿Ya tienes cuenta? [Inicia sesión](#)". At the bottom, there is a separator line with the text "o regístrate con" and two circular icons for Google (G) and Facebook (f).

Iniciar sesión.

Para iniciar sesión el usuario necesita su correo y contraseña, a su vez, puede iniciar sesión con Gmail y Facebook. También tiene la opción de “Olvide mi contraseña”, en el momento que lo requiera

Figura 87*Vista Iniciar Sesión.*

Bienvenido!

Usuario/Correo

Contraseña

olvide mi contraseña

Iniciar sesión

¿No tienes cuenta? [Regístrate](#)

o inicia sesion con



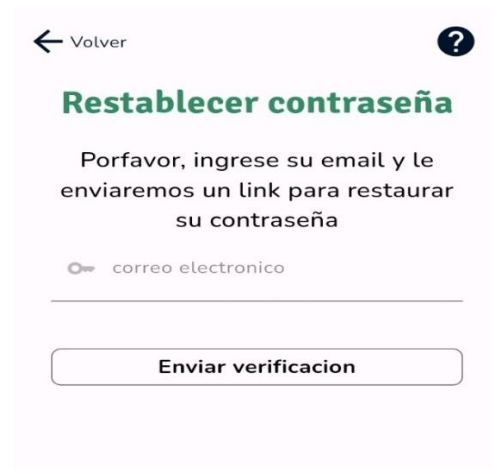

 

Figura 88*Vista Restablecer contraseña*

← Volver 

Restablecer contraseña

Porfavor, ingrese su email y le enviaremos un link para restaurar su contraseña

correo electronico

Enviar verificacion

Ver perfil.

El usuario puede observar la información de su perfil y editarlo. En el momento en que se registra no es necesario subir una imagen. En esta vista el usuario puede añadirla y a su vez, actualizar sus datos y cambiar su contraseña.

Figura 89

Vista ver Perfil.



Figura 90

Vista actualizar Perfil y añadir imagen de Perfil.

← Volver

?

TU INFORMACIÓN

Nombre
nancy cuadros

Celular
3245678914

Documento
seleccionar ▾

Número de Documento
15769284 ⓘ

Correo Electrónico
qola@test.com 🔍

Confirmar **Cancelar**

En la Figura 90 se puede visualizar la vista para la actualización de usuarios, en la cual el usuario puede actualizar sus datos y confirmar sus ajustes a su perfil.

Estadísticas.

En las estadísticas se visualizan las ganancias totales de todos los activos del usuario con su respectivo gasto e ingreso; en esta sección el usuario puede visualizar todos los gastos e ingresos registrados, a su vez puede comparar los activos que tenga actualmente. (Figura 91). En la Figura 92 se seleccionan los activos a comparar para luego visualizar la gráfica de las ganancias que cada activo ha obtenido a lo largo del mes seleccionado por el usuario. (Figura 93).

Figura 91

Vista de Estadísticas.



Figura 92

Vista seleccionar activos a comparar.



Figura 93

Vista comparación de activos.

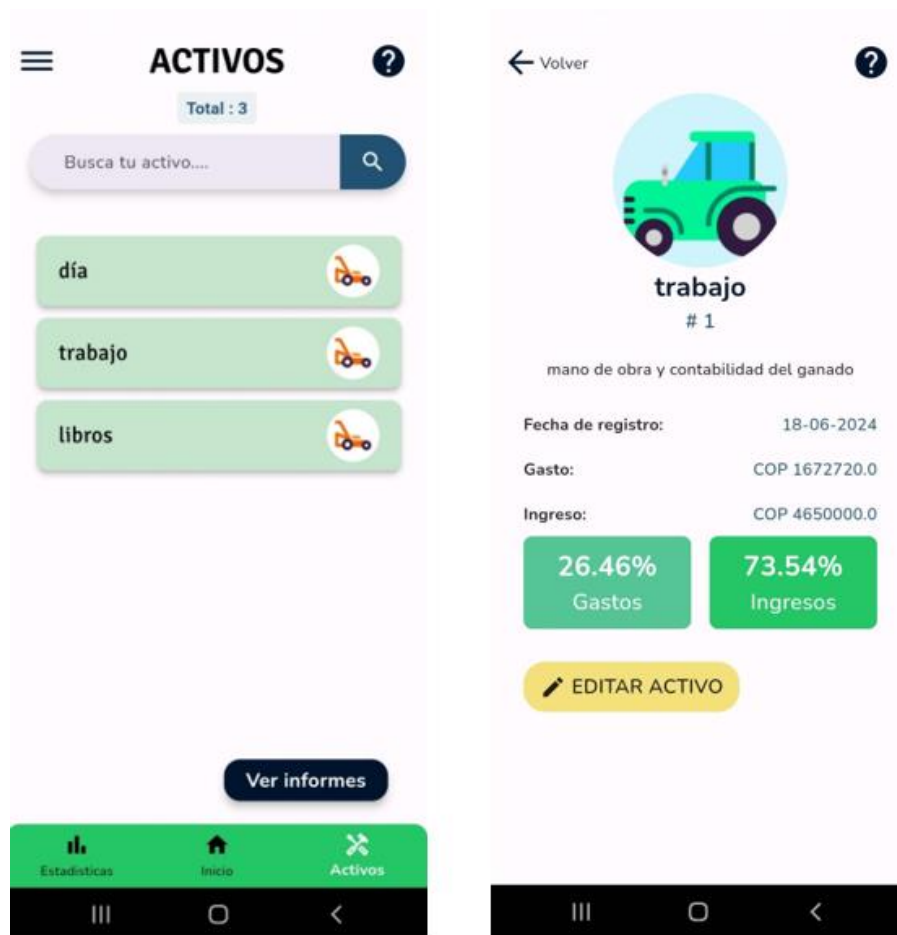


Activos

El usuario puede visualizar todos los activos que tiene registrados en la cuenta, de cada uno puede observar la descripción realizada del activo, la fecha de creación del activo, la cantidad monetaria registrada de gastos e ingresos por el usuario y el porcentaje de este (**figura 94**).

Figura 94

Activo y perfil del Activo.

**Menú.**

El usuario puede acceder al menú desde la vista de inicio, activos o estadística, encontrada en la parte superior izquierda (Figura 85). En esta vista se encuentra el nivel y el nombre de usuario, a su vez que los botones que dirigen a la información, los ajustes e historial del usuario; y los términos y condiciones de la aplicación. (Figura 95)

Figura 95*Vista Menú.***Misiones.**

El usuario puede visualizar las misiones del aplicativo. A medida que el usuario realiza las misiones sube de nivel, y se generan nuevas misiones propias del nivel que el usuario ha conseguido. **(figura 96)**

Figura 96

Misiones del usuario.

**Gastos/Ingresos**

El usuario puede agregar un gasto o ingreso en la aplicación, detallando el activo específico, la cantidad, descripción y fecha realizada de este. (Figura 97) A su vez, en la Figura 98 el usuario puede visualizar y editar el historial de gastos e ingresos registrados en la aplicación en la pestaña de estadísticas. (Figura 91)

Figura 97

Vista final Gastos o Ingresos.

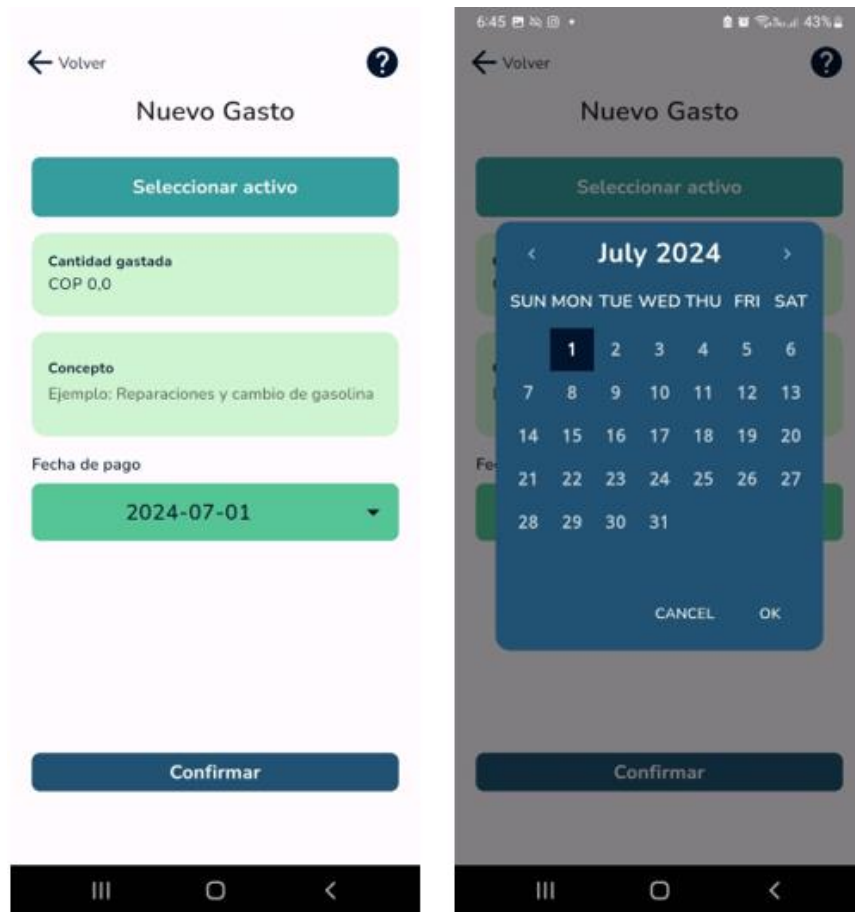
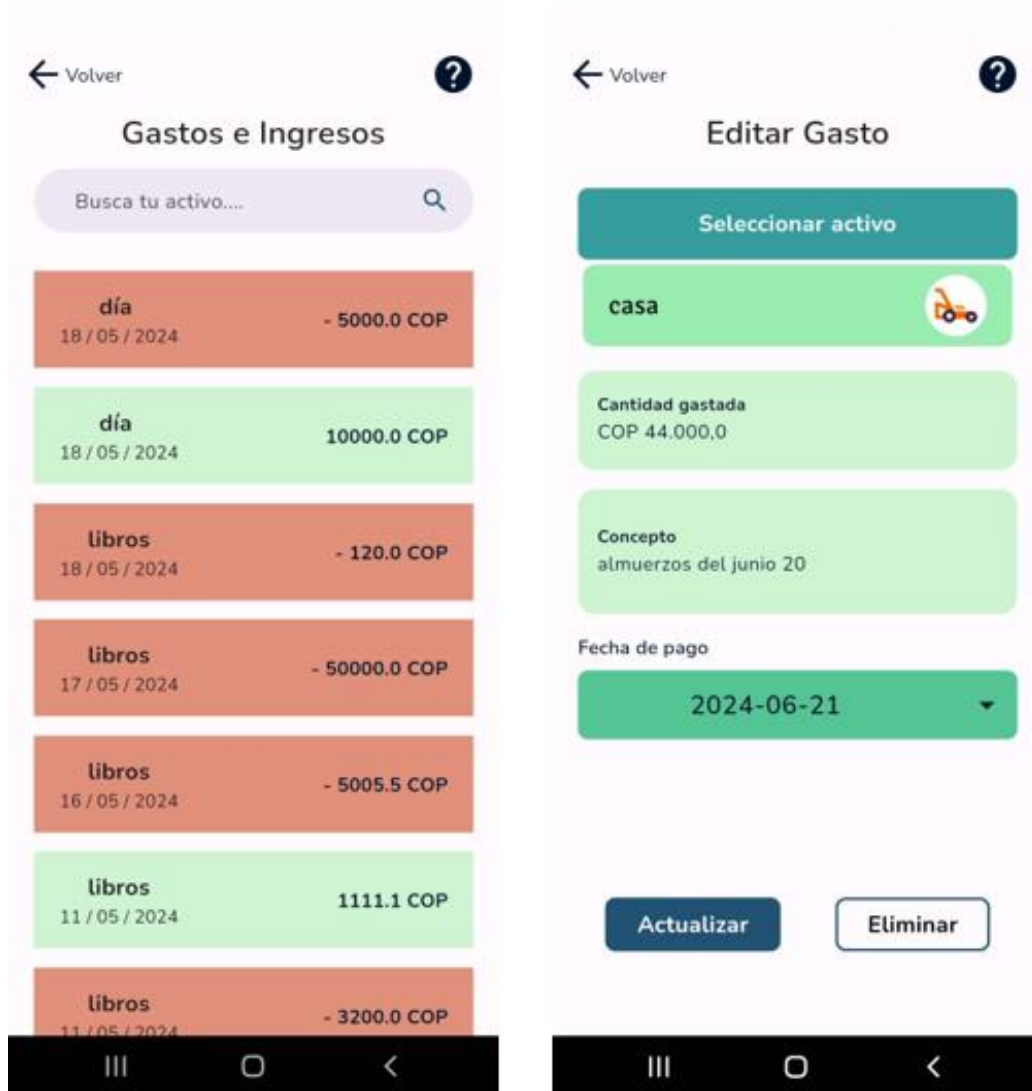


Figura 98

Historial gastos e ingresos y Editar Gasto/Ingreso.



Historial

El usuario puede ingresar al historial de la aplicación a través del Menú de este. (Figura 95). En esta vista el usuario puede ver todas las acciones realizadas dentro de la aplicación, tales como el registro de un activo, un gasto o un ingreso, o propias de la aplicación, como las misiones completadas y el subir de nivel.

Figura 99*Vista Historial.*

sid	Nombre	Fecha
Y0Js	nuevo activo	20-05-2024
SjBj	Subiste de nivel	20-05-2024
UHFo	mission completada	20-05-2024
MUVi	nuevo activo	20-05-2024
	agregar ingreso	02-04-2024
	agregar activo	02-04-2024
	eliminar activo	02-04-2024
	agregar ingreso	02-04-2024
	agregar ingreso	02-04-2024

Informe

El usuario puede ingresar a la vista de informes a través de la sección de activos, en la parte inferior derecha de la pantalla. (Figura 94) En esta vista el usuario puede visualizar el informe de activos desde la aplicación, donde se detalla el nombre, ganancias, gastos, ingresos y fecha de registro del activo. (Figura 100) A su vez, el usuario puede descargar el informe de los activos en

un archivo PDF, donde se le notificará la descarga del informe; al oprimir la notificación lo llevará al documento PDF donde podrá visualizar y compartir el documento. (Figura 101)

Figura 100

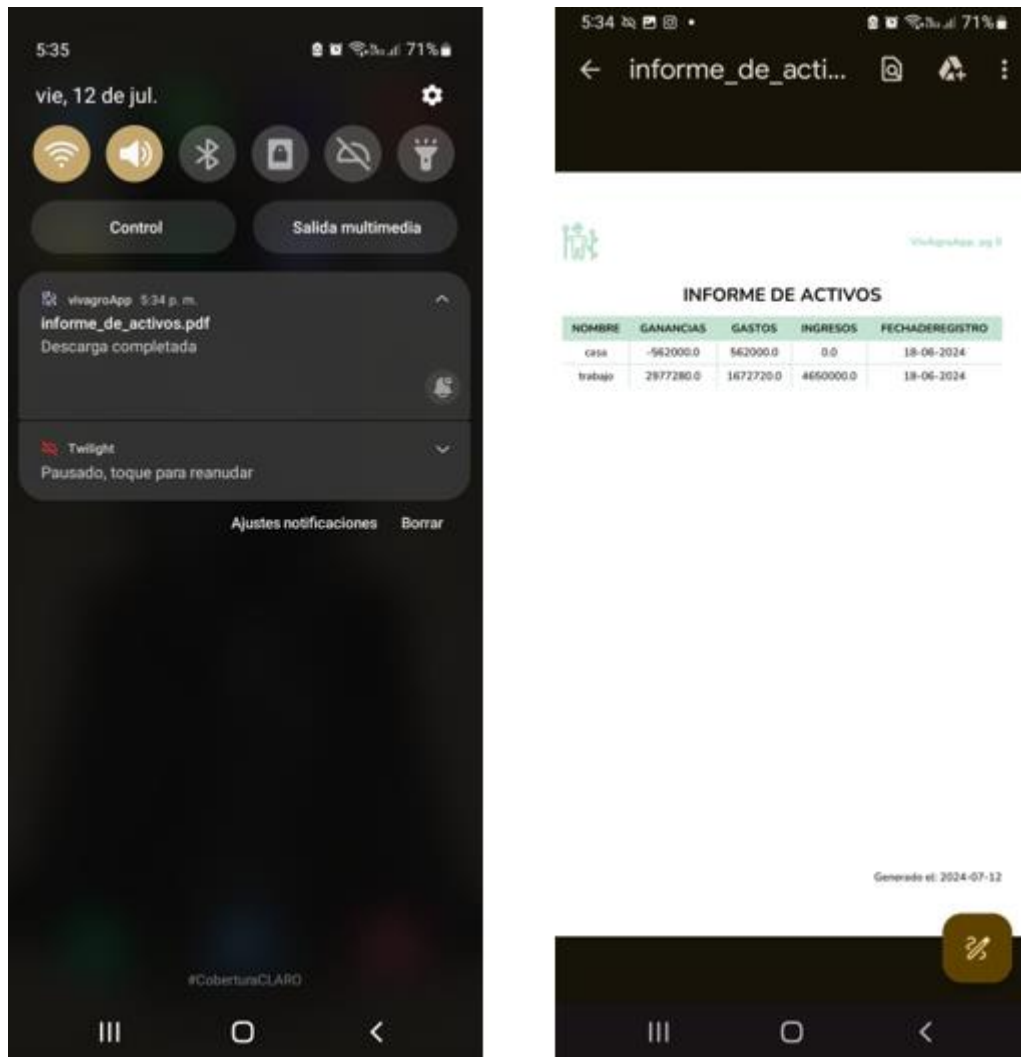
Vista Informe de Activos.



Nombre	Gastos	Ingresos
día	\$ 5000.00	\$ 10000.00
trabajo	\$ 0.00	\$ 0.00

Figura 101

Notificación y visualización del documento descargado.



Ajustes de la aplicación

El usuario puede visualizar los ajustes de la aplicación accediendo al menú de la aplicación. (Figura 95) En esta vista el usuario puede cambiar su contraseña, verificar su correo, administrar los permisos de la aplicación y eliminar su cuenta de usuario de ser necesario.

Figura 102*Vista Ajustes.***Ayuda**

En la figura 103 el usuario puede visualizar las ayudas para la aplicación, a su vez que puede dirigirse al botón de buzón de ayuda en caso de que su pregunta no se encuentre en las preguntas frecuentes.

Figura 103*Vista Ayuda.*