

Aplicación móvil orientada al mejoramiento de la valoración del producto agrícola y la toma de decisiones en la comercialización.

Deisy Katherine Rangel Flórez y Andres Javier Cuadros Sanabria

Trabajo de Grado para optar al Título de Ingeniero de Sistemas

Director

Hugo Hernando Andrade Sosa

Magíster en Ingeniería - Área Informática y Ciencias de la Computación

Codirector

Urbano Eliécer Gómez Prada

Magíster en Ingeniería - Área Informática y Ciencias de la Computación

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2021

Dedicatoria

“A mi Madre Eulalia Sanabria Zamora por su apoyo y confianza que me ha brindado durante toda mi vida. A mis hermanos Jenny, Ludís Marcela y Jhon William Cuadros Sanabria, que de una y otra manera me apoyaron en la carrera.”

Andres Cuadros Sanabria

“A mi familia,
A mi madre Gladys Flórez Rodríguez, mi padre Alirio Rangel Vera, mi hermana Yelithza Rangel Flórez y mi querido sobrino que viene en camino, por su apoyo incondicional y fundamental en mi vida. Este proyecto se lo dedico a ellos por ser el motor que me impulsa todos los días, por acompañarme, aconsejarme y guiarme en los momentos más difíciles.

Mil gracias, los amo.”

Deisy Katherine Rangel Flórez

Agradecimientos

En primer lugar, a Dios quien permitió que se dieran las cosas para obtener el título de ingenieros de Sistemas UIS.

A nuestra amada Universidad industrial de Santander que nos dio la posibilidad de mejorar nuestra vida completamente, de conocer personas increíbles, formarnos como profesionales y por la oportunidad de viajar a otro país.

A los profesores Urbano Eliécer Gómez Prada y Hugo Hernando Andrade Sosa por su tiempo y dedicación para este trabajo de grado.

A los agricultores de la vereda el Aventino en Mutiscua N. de Santander por su colaboración y acogida para realizar las pruebas de la aplicación móvil AgroLibreta.

A nuestros amigos Eduard Caballero, Johan Vinazco y Herny Sánchez por su compañerismo y apoyo durante esta importante etapa de nuestra vida.

Al beneficio de residencias Universitarias UIS por ser un apoyo inmenso que mejoró nuestras condiciones de vida durante nuestro ciclo en la universidad.

Tabla de contenido

Introducción	17
1. Objetivos.....	19
1.1 Objetivo General.....	19
1.2 Objetivos Específicos	19
2 Marco de referencia	20
2.1 Marco Conceptual.....	20
2.1.1 Sistema de Información.....	20
2.1.2 Costo	21
2.1.3 Sistema de Costeo	21
2.1.4 Modelo	22
2.2 Marco Contextual	22
2.2.1 Agricultura en Colombia.....	22
2.2.2 Cultivo y Producción de Arveja.....	23
2.2.3 Modelo de referencia.....	25
2.2.4 Informe general del cultivo	26
2.3 Marco Tecnológico	26
2.3.1 Trello.....	26
2.3.2 Visual Studio Code	27
2.3.3 Flutter	27
2.3.4 Dart.....	28
2.3.5 AVD Manager.....	28
2.3.6 SQLite	29
2.3.7 Google Firebase	29
2.3.8 Firebase Authentication	30
2.3.9 Cloud Firestore.....	30

2.3.10	Firestore Hosting	30
2.3.11	Sqflite.....	31
2.3.12	JSON.....	31
2.3.13	Quicktype	31
2.3.14	Patrón Provider	32
2.3.15	Patrón BLoC	32
2.3.16	GitHub.....	33
2.3.17	Flaticon.....	34
2.3.18	Brackets.....	34
2.3.19	CSS.....	34
2.3.20	HTML5	35
2.3.21	JavaScript	35
2.4	Aplicaciones enfocadas en la agricultura.....	35
2.4.1	Agroptima	36
2.4.2	AgroWin.....	36
2.5	Marco metodológico	36
2.5.1	Metodología de prototipado	37
2.5.2	Metodología Kanban.....	39
3	Resultados.....	41
3.1	Aplicación móvil.....	42
3.1.1	Requerimientos	42
3.1.2	Plan de desarrollo para el prototipo	44
3.1.3	Diagrama de casos de uso	47
3.1.4	Diagramas de actividades.....	49
3.1.5	Diagramas de bases de datos.....	50
3.1.6	Diagrama de arquitectura	53
3.1.7	Implementación.....	54
3.1.8	Vistas de usuario propuestas	68
3.1.9	Vistas de usuario definitivas	68

3.2	Sitio web	68
3.2.1	Requerimientos	69
3.2.2	Diagrama de casos de uso	70
3.2.3	Diagrama de actividades	71
3.2.4	Diagrama de bases de datos	72
3.2.5	Diagrama de arquitectura	72
3.2.6	Implementación.....	72
3.2.7	Vistas de usuario propuestas y definitivas para el sitio web.....	76
3.3	Evaluación del funcionamiento de la aplicación móvil y del sitio web.....	76
3.3.1	Pruebas unitarias	76
3.3.2	Pruebas de integración	76
3.3.3	Pruebas alfa	78
3.3.4	Pruebas beta	79
3.3.5	Pruebas de estrés	79
3.3.6	Pruebas de carga.....	80
4	Conclusiones.....	80
5	Recomendaciones	82
	Referencias bibliográficas.....	85
	Apéndices.....	88

Lista de Tablas

Tabla 1. Porcentajes de los costos de producción aproximados agrupados por conceptos.	25
Tabla 2. Requerimientos funcionales de la aplicación móvil	42
Tabla 3. Requerimientos no funcionales de la aplicación móvil	43
Tabla 4. Requerimientos funcionales del sitio web	69
Tabla 5. Requerimientos no funcionales del sitio web	69
Tabla 6. Prueba de validación para registrar costo	77
Tabla 7. Descripción requerimientos funcionales de la aplicación móvil	88
Tabla 8. Descripción requerimientos no funcionales de la aplicación móvil	90
Tabla 9. Descripción requerimientos funcionales del sitio web	91
Tabla 10. Descripción requerimientos no funcionales del sitio web	92
Tabla 11. Prueba de validación para registrar usuario	142
Tabla 12. Prueba de validación para registrar cultivo.....	143
Tabla 13. Prueba de validación para añadir un registro fotográfico	144
Tabla 14. Prueba de validación para registrar modelo de referencia	145
Tabla 15. Pruebas de validación para consultar costos.....	145
Tabla 16. Prueba de validación para ver informe	146
Tabla 17. Prueba de validación para realizar una copia de seguridad	146
Tabla 18. Prueba de validación para consultar la última copia de seguridad	147

Lista de Figuras

Figura 1. Patrón Provider	32
Figura 2. Patrón BLoC	33
Figura 3. Pasos del prototipado.....	39
Figura 4. Tablero de la metodología Kanban al inicio del proyecto.....	45
Figura 5 Tablero metodología Kanban en el desarrollo del proyecto.....	46
Figura 6 Tablero metodología Kanban finalizando el proyecto	47
Figura 7. Diagrama de casos de usos de la aplicación móvil.....	48
Figura 8. Diagrama de actividades para registrar costo	49
Figura 9. Diagrama de la base de datos local.....	51
Figura 10. Diagrama de la base de datos remota	52
Figura 11. Diagrama de arquitectura del sistema.....	53
Figura 12. Archivo main	54
Figura 13. Archivo routes	55
Figura 14. Clase para acceder a la base de datos local	56
Figura 15. Método para crear la base de datos local.....	57
Figura 16. Clase con el método para crear un registro de un modelo de referencia.....	58
Figura 17. Provider que maneja información de productos o actividades	59
Figura 18. Patrón BLoC	60
Figura 19. Ejemplo del uso de la herramienta Quicktype.....	61
Figura 20. Archivo con la implementación de un modelo en lenguaje Dart	62
Figura 21. Implementación de la vista de usuario home	63

Figura 22. Implementación del body para la vista home	64
Figura 23. Jerarquía de carpetas de la implementación	65
Figura 24. Firebase Authentication usado en AgroLibreta.....	66
Figura 25. Consola del servicio Cloud Firestore para AgroLibreta.....	67
Figura 26. Repositorio de la aplicación en GitHub.....	68
Figura 27. Diagrama de casos de uso del sitio web	70
Figura 28. Diagrama de actividades para consultar fecha de la copia de seguridad.....	71
Figura 29. Archivo HTML index del sitio web	73
Figura 30. Archivo CSS para el estilo del index.....	74
Figura 31. Script main.js (conexión con Firebase)	75
Figura 32. Diagrama de actividades para registrar un cultivo	93
Figura 33. Diagrama de actividades para registrar modelo de referencia.....	94
Figura 34. Diagrama de actividades para registrar fotografía de un evento comercial	95
Figura 35. Archivo main (parte 1)	97
Figura 36. Archivo main (parte 2)	98
Figura 37. Archivo routes	99
Figura 38. Archivo db_provider (parte 1).....	100
Figura 39. Archivo db_provider (parte 2).....	101
Figura 40. Archivo db_provider (parte 3).....	102
Figura 41. Archivo db_provider (parte 4).....	103
Figura 42. Archivo db_provider (parte 5).....	104
Figura 43. Archivo db_provider (parte 6).....	105
Figura 44. Archivo db_provider (parte 7).....	106

Figura 45. Archivo CRUD modelo_referencia_operations (parte 1).....	107
Figura 46. Archivo CRUD modelo_referencia_operations (parte 2).....	108
Figura 47. Registrar cultivo (vista propuesta).....	109
Figura 48. Listar cultivos (vista propuesta)	110
Figura 49. Modelo de referencia por defecto (vista propuesta)	111
Figura 50. Resumen de costos (vista propuesta).....	112
Figura 51. Registro de costos (vista propuesta)	113
Figura 52. Consultar costos (vista propuesta).....	114
Figura 53. Informe general (vista propuesta).....	115
Figura 54. Modelo de referencia específico (vista propuesta)	116
Figura 55. Gestionar registros fotográficos (vista propuesta).....	117
Figura 56. Iniciar sesión y registrar usuario (primer prototipo).....	119
Figura 57. Listar y registrar cultivo (primer prototipo)	120
Figura 58. Resumen del cultivo y registrar costo (primer prototipo)	121
Figura 59. Registrar producto o actividad y unidad de medida (primer prototipo)	122
Figura 60. Consultar costos con filtros (primer prototipo)	123
Figura 61. Editar costo y detalles del costo (primer prototipo)	124
Figura 62. Modelo de referencia por defecto (primer prototipo).....	125
Figura 63. Informe general del cultivo (primer prototipo)	126
Figura 64. Registrar un modelo de referencia (segundo prototipo)	127
Figura 65. Galería y nuevo registro fotográfico (segundo prototipo).....	128
Figura 66. Asociar costos a registro fotográfico y detalle (segundo prototipo).....	129
Figura 67. Comparar costos con el modelo de referencia (prototipo final)	130

Figura 68. Copia de seguridad y editar perfil de usuario (prototipo final)	131
Figura 69. Configuración del cultivo (prototipo final)	132
Figura 70. Editar precio de venta y cambiar estado del cultivo (prototipo final)	133
Figura 71. Utilidades (prototipo final)	134
Figura 72. Página principal (propuesta)	136
Figura 73. Página de instrucciones (propuesta)	137
Figura 74. Página de descargas (propuesta)	137
Figura 75. Página de ingreso (propuesta)	138
Figura 76. Página principal del sitio web	139
Figura 77. Página de instrucciones del sitio web	139
Figura 78. Página de descargas del sitio web	140
Figura 79. Página de ingreso del sitio web	141
Figura 80. Página para consulta del sitio web	141
Figura 81. Algunos usuarios de la prueba alfa	148
Figura 82. Algunos usuarios de la prueba beta	150
Figura 83. Primer usuario para la prueba de estrés	150
Figura 84. Segundo usuario para la prueba de estrés	151
Figura 85. Tercer usuario para la prueba de estrés	151
Figura 86. Cuarto usuario para la prueba de estrés	152
Figura 87. Quinto usuario para la prueba de estrés	153
Figura 88. Carga de 101 cultivos	154
Figura 89. Registro de 20 costos a cada uno de los 101 cultivos	155
Figura 90. Aplicación al tener los 2100 registros vista home	156

Figura 91. Sincronizar más de 2100 registros.....	157
Figura 92. Pregunta # 1 - Encuesta para la prueba alfa	158
Figura 93. Pregunta # 2 - Encuesta para la prueba alfa	159
Figura 94. Pregunta # 3 - Encuesta para la prueba alfa	159
Figura 95. Pregunta # 4 - Encuesta para la prueba alfa	160
Figura 96. Pregunta # 5 - Encuesta para la prueba alfa	160
Figura 97. Pregunta # 6 - Encuesta para la prueba alfa	161
Figura 98. Pregunta # 7 - Encuesta para la prueba alfa	161
Figura 99. Pregunta # 8 - Encuesta para la prueba alfa	162
Figura 100. Pregunta # 9 - Encuesta para la prueba alfa	162
Figura 101. Pregunta # 10 - Encuesta para la prueba alfa	163
Figura 102. Pregunta # 11- Encuesta para la prueba alfa	163
Figura 103. Pregunta # 12 - Encuesta para la prueba alfa	164
Figura 104. Pregunta #13 - Encuesta para la prueba alfa	164
Figura 105. Pregunta #14 - Encuesta para la prueba alfa	165
Figura 106. Pregunta #15 - Encuesta para la prueba alfa	166
Figura 107. Pregunta #16 - Encuesta para la prueba alfa	166
Figura 108. Pregunta # 1 - Encuesta para la prueba beta.....	167
Figura 109. Pregunta # 2 - Encuesta para la prueba beta.....	168
Figura 110. Pregunta # 3 - Encuesta para la prueba beta.....	168
Figura 111. Pregunta # 4 - Encuesta para la prueba beta.....	169
Figura 112. Pregunta # 5 - Encuesta para la prueba beta.....	170
Figura 113. Pregunta # 6 - Encuesta para la prueba beta.....	171

Figura 114. Pregunta # 7 - Encuesta para la prueba beta.....	172
Figura 115. Pregunta # 8 - Encuesta para la prueba beta.....	172
Figura 116. Pregunta # 9 - Encuesta para la prueba beta.....	173
Figura 117. Pregunta # 10 - Encuesta para la prueba beta.....	173
Figura 118. Pregunta # 11 - Encuesta para la prueba beta.....	174
Figura 119. Pregunta # 12 - Encuesta para la prueba beta.....	174
Figura 120. Pregunta # 13 - Encuesta para la prueba beta.....	175
Figura 121. Pregunta #14 - Encuesta para la prueba beta.....	175
Figura 122. Pregunta # 15 - Encuesta para la prueba beta.....	176
Figura 123. Pregunta # 16 - Encuesta para la prueba beta.....	176
Figura 124. Pregunta # 17 - Encuesta para la prueba beta.....	177
Figura 125. Pregunta # 18 - Encuesta para la prueba beta.....	177
Figura 126. Pregunta # 19 - Encuesta para la prueba beta.....	178
Figura 127. Experiencia del primer productor.....	179
Figura 128.Experiencia de la segunda agricultora.....	180
Figura 129. Experiencia del tercer agricultor.....	181
Figura 130. Andres Cuadros Sanabria en un cultivo de arveja.....	182
Figura 131. Deisy Rangel Flórez en un cultivo de arveja.....	183

Lista de Apéndices

Apéndice A. Requerimientos del sistema.	88
Apéndice B. Diagramas de actividades.	93
Apéndice C. Implementación.....	96
Apéndice D. Vistas de usuario propuestas para la aplicación móvil.	109
Apéndice E. Vistas de usuario definitivas de la aplicación móvil.	118
Apéndice F. Vistas de usuario del sitio web.	136
Apéndice G. Pruebas realizadas para la evaluación del funcionamiento.....	142
Apéndice H. Resultados de las encuestas.	158
Apéndice I. Visita a los agricultores	179

Resumen

Título: Aplicación Móvil Orientada al Mejoramiento de la Valoración del Producto Agrícola y la Toma de Decisiones en la Comercialización*

Autor: Deisy Katherine Rangel Flórez, Andres Javier Cuadros Sanabria**

Palabras Clave: Aplicación móvil, sistema de costeo, producción agrícola, toma de decisiones, modelo de referencia, valoración del producto.

Descripción: En la economía Colombiana, la agricultura es un sector con grandes dificultades de rentabilidad, debido a razones como las importaciones de productos agrícolas, la poca inversión por parte del estado y de las entidades privadas del sector, los problemas en las vías de transporte, la poca tecnificación de los procesos, el desconocimiento en el uso de la tecnología en los pequeños productores, el bajo control de las finanzas, la depreciación de los costos que conlleva todo el proceso productivo, entre otras.

En este trabajo de grado se propone una aplicación para dispositivos móviles como solución a la dificultad que tienen los pequeños productores agrícolas de llevar y poder controlar los costos de sus cultivos. Ellos pueden implementar la tecnología a su favor, utilizando una aplicación móvil y un sitio web como herramienta que les permite generar un conocimiento de los costos directos en la producción de los cultivos, contando con información para la toma de decisiones al momento de comercializar sus productos.

La aplicación móvil contiene un modelo de referencia que sirve como guía a las personas que la utilicen, debido a que los orienta en cómo distribuir los costos directos de la producción para que puedan: tomar decisiones con respecto al presupuesto, analizar los resultados obtenidos en sus cultivos y tener un informe al finalizar el ciclo productivo, de manera que permita identificar fallas durante el proceso y así obtener mejores resultados a futuro.

* Trabajo de Grado

** Facultad de Ingeniería Físicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Hugo Hernando Andrade Sosa. Magíster en Ingeniería - Área Informática y Ciencias de la Computación. Codirector: Urbano Eliécer Gómez Prada. Magíster en Ingeniería - Área Informática y Ciencias de la Computación.

Abstract

Title: Mobile application oriented to improve the estimation of the price of the agricultural product and making decision in the commercialization *

Author: Deisy Katherine Rangel Flórez, Andres Javier Cuadros Sanabria **

Key Words: Application mobile, costing system, agricultural production, make decisions, reference model, appraise the product.

Description: In the Colombian economy, agriculture is a sector with great profitability difficulties, due to reasons such as agricultural product importations, low investment by the state and private entities in the sector, problems in the transportation routes, the lack of technology in the processes, the unawareness on the use of technology by many families who work in the countryside, inadequate financial management, the cost depreciation involved in the entire production process, among others.

In this research, an application for mobile devices is proposed as a solution to the difficulty that small farmers have when managing and regulating the costs of their crops. They can implement the technology in their favor, using a mobile application and a website as a tool that allows them to acquire knowledge of direct costs in the crops production, thus, generating information for decision making at the time of trading their products.

The mobile application consists of a reference model that guides users on how to distribute direct production costs so they can: make decisions regarding the budget, analyze the results obtained in their crops and have a report at the end of the production cycle, which is useful in identifying failures during the processes and therefore achieve better results in the future.

* Degree Work

** Physical-Mechanical Engineering Faculty. Informatics and Systems Engineering School. Director: Hugo Hernando Andrade Sosa. Master in Engineering - Informatics and Computer Science Area. Co-director: Urbano Eliécer Gómez Prada. Master in Engineering - Informatics and Computer Science Area.

Introducción

Entre los países que conforman la región Andina, Colombia se distingue por ser principalmente rural, en el cual, los pequeños productores agrícolas son importantes por su participación en la producción agropecuaria y contribuyen con un 67,3% del volumen de la producción agrícola nacional (Reboul & Escobar, 2019).

Según Perfetti *et al.* (2013), los pequeños productores agrícolas presentan los más altos niveles de necesidades básicas insatisfechas y es por eso que tienen limitaciones en términos de acceso a tierra, tecnología, capital y comercialización que les impide acumular bienes, remunerar sus actividades y alcanzar niveles de bienestar deseables.

Arévalo *et al.* (2015) resalta que los pequeños productores, debido a sus arraigos familiares y culturales, tienen una brecha en el uso de las tecnologías de la información y la comunicación (TIC) que conlleva a malas decisiones y a la disminución en sus ingresos generando desinterés en sus procesos productivos y migración hacia las ciudades. Además, Perry (2013) manifiesta que, con las TIC se tiene un apoyo para mejorar su competitividad y sus condiciones de vida.

La situación descrita fue reconocida de una manera cercana hablando con la señora Gladys Flórez quien es administradora de un sistema productivo en el sector de la agricultura en el municipio de Mutiscua (N. de Santander). Mediante una entrevista la señora comentó que en cuanto al proceso de registrar los costos: *“las personas suelen usar una libreta para registrar los gastos y ventas de un cultivo, así al final de la producción tener claro las ganancias o pérdidas para tomar decisiones en futuros cultivos, además las facturas o soportes de dichos registros son*

guardados en físico sin un control contable corriendo el riesgo que estas se puedan perder” (2020).

Con el objetivo de mitigar la problemática se propuso el desarrollo de un aplicativo móvil llamado AgroLibreta dirigido a pequeños productores agrícolas que no cuentan con un reporte de los costos asociados a sus cultivos, cuyo desarrollo siguió el enfoque metodológico de prototipado y la metodología Kanban. El aplicativo móvil facilita el reporte de los costos asociados a los cultivos apoyándose en un modelo de referencia el cual guía la distribución de los costos, AgroLibreta también pone a disposición de los pequeños productores un sitio web en el cual pueden encontrar ayuda y consultar la fecha de su última copia de seguridad.

AgroLibreta se validó con estudiantes de Tecnología Agroindustrial del IPRED y con productores de la región de Mutiscua, municipio de Norte de Santander y según su opinión se considera que el producto puede llegar a mitigar la problemática expresada.

En este documento es presentado el marco de referencia que a su vez está compuesto por el marco conceptual, contextual, tecnológico, metodológico y la descripción de algunas aplicaciones enfocadas a la agricultura, también se encuentran los resultados obtenidos donde se describen las pruebas unitarias, de integración, alfa, beta, de estrés y de carga, algunas de estas realizadas con los usuarios que permitieron evaluar el desarrollo de la aplicación móvil y del sitio web.

1. Objetivos

1.1 Objetivo General

Desarrollar una aplicación móvil que sirva de sistema de costeo para pequeños productores agrícolas, soportada con un sitio web, orientada al mejoramiento de la valoración del producto y la toma de decisiones en la comercialización, usando tecnologías de libre distribución y la metodología de prototipado. Caso: cultivo de arveja en la vereda el Aventino, municipio de Mutiscua, departamento Norte de Santander.

1.2 Objetivos Específicos

1. Desarrollar una aplicación móvil según la metodología de prototipado, que sea operable en dispositivos móviles Android, con las siguientes características generales:

- a. Generar un modelo de referencia para el sistema de costeo según una estructura de costos que tendrá en cuenta los porcentajes de inversión en semilla, abonos y fertilizantes, plaguicidas, empaques, maquinaria, mano de obra, transporte y otros.
- b. Permitir al usuario la construcción del modelo de referencia específico del cultivo, en función de la estructura de costos general.
- c. Registrar los costos en tiempo real de la producción de tal forma que el sistema compare lo obtenido con lo esperado.
- d. Generar informes comparativos entre los costos registrados de la producción y el modelo de referencia propuesto.
- e. Gestionar soportes de costos de la producción apoyada en la opción de captura de imágenes del móvil.

2. Desarrollar un sitio web para el control de copias de seguridad, instalación y reinstalación del aplicativo móvil.

3. Generar una experiencia con al menos tres usuarios para la evaluación del funcionamiento de la aplicación web y móvil.

2 Marco de Referencia

En el marco de referencia de este trabajo de grado se presentan los siguientes títulos: marco conceptual, marco contextual, marco tecnológico, marco metodológico con el fin de soportar teórica y metodológicamente el contenido de este documento; incluyendo también la descripción de algunas aplicaciones enfocadas y en pro de la agricultura. Por medio de lo investigado en este capítulo se adquirió la información que se implementó para la generación de la aplicación móvil y del sitio web.

2.1 Marco Conceptual

En esta sección se presentan los componentes teóricos del proyecto que se deben tener en cuenta para desarrollar el tema central de este trabajo de grado el cual es la generación de una aplicación móvil soportada en un sitio web.

2.1.1 Sistema de Información

Un sistema de información de una organización según Laudon & Laudon (2016) es un subsistema específico y puede definirse técnicamente como un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar datos que a su vez se pueden

distribuir para generar información que las personas procesan mentalmente en contexto, dándoles significado e informándose para apoyar la toma de decisiones.

2.1.2 Costo

“Se define como costo de un bien, servicio o actividad el valor o precio de los conceptos o recursos requeridos para la elaboración o adquisición de un bien, para la prestación de un servicio o en el desarrollo de una actividad; dichos recursos se clasifican en los denominados elementos o componentes del costo”(p.22) “que por las características especiales de cada uno de ellos se denominan como: materiales directos, mano de obra directa, otros costos variables o relacionados con el uso de la maquinaria y equipos de producción y otros costos de producción ya sean indirectos y/o fijos”(p.54) (Flórez, 2004).

2.1.3 Sistema de Costeo

En un sistema de costeo el objetivo principal es determinar el costo de los productos obtenidos, el valor de los inventarios representados en materiales o suministros; también un sistema de costeo debe efectuar los cálculos necesarios para determinar los costos directos de los productos fabricados, debe elaborar los registros contables originados en las operaciones relacionadas con los componentes del costo y con dichos datos suministrados el sistema debe elaborar los informes y estados de costos, para efectuar los análisis correspondientes y tomar decisiones.

En conclusión, un sistema de costeo debe permitirles a las personas fijar precios de venta, tomar decisiones para controlar los costos de la producción, evaluar desempeños y definir la necesidad de comprar o eliminar un bien o un servicio (Flórez, 2004).

2.1.4 Modelo

Un modelo es un bosquejo que representa un conjunto con cierto grado de precisión y en la forma más completa posible, pero sin buscar ser una réplica de lo que existe en la realidad. Los modelos son útiles para describir, explicar o comprender mejor la realidad, cuando es imposible trabajar directamente en la realidad en sí (Wadsworth, 1997).

2.2 Marco Contextual

En el desarrollo de este proyecto fue preciso investigar y describir las siguientes terminologías que se usarán en el transcurso de este trabajo de grado con la finalidad de exponer los temas referentes en el que gira y se desarrolla, para evitar errores en la comprensión del público que desee tener acceso a esta investigación.

2.2.1 Agricultura en Colombia

Las zonas rurales de Colombia han sido altamente azotadas por una ola de violencia generada por grupos guerrilleros, fuerzas militares del estado y grupos paramilitares llevando miseria a las personas campesinas y abandono por parte del estado (Sierra, 2017).

Según Perfetti *et al.* (2013), los pequeños productores agrícolas en Colombia, definidos como aquellos que ejecutan su actividad productiva en predios menores a dos UAF¹ y emplean principalmente mano de obra familiar, representan el 80% de la población rural teniendo en

¹Unidad Agrícola familiar (UAF), la cual fue definida por la Ley 160 de 1994 como “La empresa básica de producción agrícola, pecuaria, acuícola, o forestal cuya extensión, conforme a las condiciones agroecológicas de la zona y con tecnología adecuada, permite a la familia remunerar su trabajo y disponer de un excedente capitalizable que coadyuve a la formación de su patrimonio”.

cuenta a sus familias. Sin embargo, los altos niveles de pobreza rural (46,1% por ingresos y 53,1% por el IPM¹) recaen en mayor proporción sobre los pequeños productores y sus familias.

Debido a lo anteriormente mencionado, este proyecto está enfocado en los pequeños productores agrícolas de la vereda el Aventino en el municipio de Mutiscua del departamento de Norte de Santander para que a pesar de las dificultades que tienen estas familias, la tecnología pueda beneficiarlos indirectamente con más ingresos y por lo tanto mejorar su calidad de vida.

2.2.2 Cultivo y Producción de Arveja

*“La arveja (*Pisum sativum*, L.) es una planta leguminosa de la familia Fabaceae, domesticada por el hombre desde tiempos muy antiguos, según hallazgos arqueológicos realizados en Tailandia, Irak y Suiza que datan entre 10.000 y 3.000 años antes de Cristo.”*

Algunas de las variedades de arveja cultivadas en Colombia son: Piquinegra, Guatecana, Sindamanoy, Obonuco San Isidro, Obonuco Andina, ICA-Tominé, Alcalá, Sureña y Santa Isabel que es la más cultivada (Departamento Administrativo Nacional de Estadística DANE, 2015).

Las condiciones agroclimatológicas de la arveja se basan en la adaptación que tienen entre 2.200 y 3.000 m.s.n.m (metros sobre el nivel del mar), se cultiva en climas fríos y medios, la temperatura óptima es de 13 a 18 °C, con media máxima de 21 °C y mínima de 9 °C. Las flores, las vainas y los granos tiernos son fuertemente afectados cuando se presentan heladas a temperaturas de -1 a -2 °C. Es una planta que alcanza una altura de 1,7 metros, con un hábito de crecimiento indeterminado o voluble, flor blanca, vainas de 4 a 6 centímetros de largas y de 6 a 8 granos por vaina; para sembrar una hectárea se necesita entre 32 a 44 kilos con rendimiento de

¹Índice de pobreza multidimensional.

4.000 a 5.600 kilogramos de vaina verde respectivamente y 900 a 1.100 kilogramos de grano seco por hectárea, sus granos son de color crema quemado, con hilum de color negro y forma redonda. Tiene un ciclo de vida a partir de la siembra, así: floración a los 75 días, cosecha de grano verde a los 120 días y cosecha grano seco a los 170 días (Departamento Administrativo Nacional de Estadística DANE, 2015).

Algunas actividades de este cultivo son:

La planeación: Es donde se encuentra la selección, adecuación, distribución del lote y la ubicación del cultivo, se han de planear las diferentes actividades de producción, la tecnología necesaria, la mano de obra, el transporte y la comercialización.

El establecimiento del cultivo: Es donde se definen los predios o lotes adecuados para el cultivo considerando factores como: altitud, temperatura, humedad relativa y la luminosidad. El tipo de suelo, labores del suelo y de fertilización como la arada primaria y secundaria.

El manejo del cultivo: Se realiza la preparación del suelo donde se crean las zanjas de drenaje, se incorpora al suelo los sustratos necesarios, se realiza la siembra y las actividades de asistencia técnica como el plan de fertilización, el manejo integrado de plagas o enfermedades, el riego y las podas.

La cosecha: Es donde se realizan las actividades de recolección generalmente de forma manual, el secado del producto húmedo y la selección. Cuando los frutos cumplen con la calidad de coloración para la comercialización se procede a empacar.

La pos-cosecha: La arveja se clasifica y empaca de acuerdo con el estado sanitario, la uniformidad en la longitud de la vaina, el color, la turgencia, la rugosidad, el número de granos por vaina y el espesor de la vaina.

La comercialización: se cumplen con los estándares de calidad y requisitos del comprador para su posterior transporte que debería realizarse preferiblemente en vehículos refrigerados para evitar daños en el producto final que será comercializado.

La Tabla 1 muestra una aproximación del porcentaje que ocupan los costos agrupados por conceptos según una investigación realizada por Peñaranda & Molina (2011).

Tabla 1.

Porcentajes de los costos de producción aproximados agrupados por conceptos.

CONCEPTO	Porcentaje
Porcentaje total	100%
Semilla	3,90%
Abonos y fertilizantes	17,70%
Plaguicidas, herbicidas y fungicidas	12,10%
Materiales y Empaques	6,70%
Maquinaria	6,50%
Mano de obra	46,10%
Transporte	1,50%
Otros	5,50%

2.2.3 Modelo de referencia

El modelo de referencia que se desarrolló para la aplicación se basó en la Tabla 1 ya que contiene lo pertinente a los costos de producción del cultivo de arveja. El usuario asigna un presupuesto el cual se distribuye según la estructura de costos presentada en la Tabla 1, para esto se utilizaron los ítems con los porcentajes a los que se les llamó ‘conceptos’ de esta forma el usuario tiene una guía de lo máximo que debe gastar en cada uno para aumentar las posibilidades en el margen de ganancia y minimizar costos, además tan pronto como se van ingresando los costos la aplicación alerta al usuario en caso de que esté llegando a niveles críticos y pueda ver

en qué está fallando, también al finalizar un cultivo puede comparar gráficamente los resultados obtenidos con respecto a los esperados según el modelo.

2.2.4 Informe general del cultivo

El informe permite al usuario obtener de manera resumida y ordenada los datos de tal forma que puedan entender los resultados de la producción; el informe presenta unas gráficas las cuales comparan el estado real de los costos con el modelo de referencia especificado. Al tener uno o más informes el usuario puede analizar para construir el modelo de referencia que mejor se ajuste a sus necesidades y con el paso del tiempo a minimizar sus costos.

2.3 Marco Tecnológico

Las herramientas producto de la tecnología que se utilizaron para el desarrollo de este proyecto se consideran como tecnofactos, estos se clasificaron de dos formas: para la generación del sitio web y para la generación del aplicativo móvil a excepción de Trello que se utilizó para la planeación del desarrollo, se definen a continuación.

2.3.1 Trello

“Gestiona proyectos, organiza tareas y fomenta un trabajo colaborativo, logrando cotas más altas de productividad, creando tableros, listas y tarjetas para asignar tareas, fijar plazos, consultar parámetros de productividad, configurar calendarios para ver mejoras en los flujos de trabajo” (Trello, s.f.).

Los siguientes tecnofactos fueron utilizados para el desarrollo de la aplicación móvil denominada ‘AgroLibreta’, estos se describen a continuación:

2.3.2 *Visual Studio Code*

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un gran ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity).

Para el proyecto se trabajó con este editor de código ya que nos permitía avanzar de una manera rápida gracias a su gran cantidad de extensiones, snippets y atajos que ayudan a organizar y generar código, también el control integrado de Git que permite trabajar en colaboración (Microsoft, 2021).

2.3.3 *Flutter*

Es un conjunto de herramientas de interfaces de usuario creado por Google para realizar aplicaciones, compiladas nativamente, para Android, IOS, web y escritorio desde una única base de código, además es gratis y de código abierto (Dange, 2019).

Se optó por este framework ya que es de fácil aprendizaje y su documentación es explícita, también la compilación es rápida permitiendo ver los cambios guardados al instante y así avanzar más rápido. Trabajar con Flutter fue un reto ya que el aprender este lenguaje y utilizar esta herramienta era algo nuevo lo que conllevó más tiempo.

2.3.4 Dart

“Dart es un lenguaje optimizado para desarrollar aplicaciones rápidas en cualquier plataforma. Su objetivo es ofrecer el lenguaje de programación más productivo para el desarrollo multiplataforma, junto con una plataforma de ejecución flexible. Dart hace parte de la fundación de Flutter y proporciona el lenguaje y los tiempos de ejecución que impulsan las aplicaciones de Flutter, también admite muchas tareas centrales del desarrollador, como formatear, analizar y probar el código. Dart puede ejecutar código en la plataforma nativa y en la web gracias a su tecnología de compilación. En la plataforma nativa para aplicaciones dirigidas a dispositivos móviles y de escritorio, Dart incluye una máquina virtual con compilación Just-In-Time y un compilador anticipado para producir código de máquina y para aplicaciones dirigidas a la web, Dart incluye un compilador de tiempo de desarrollo y un compilador de tiempo de producción. Ambos compiladores traducen Dart a JavaScript” (Dart, s.f.).

2.3.5 AVD Manager

“Un dispositivo virtual de Android (AVD) es una configuración que define las características de un teléfono o una tablet Android, o de un dispositivo Wear OS, Android TV o Automotive OS, que deseas simular en Android Emulator. El Administrador de AVD es una interfaz que puedes iniciar desde Android Studio y te permite generar y administrar los AVD” (Developers, 2021). Para realizar este proyecto se utilizaron los dispositivos móviles virtuales Nexus 5X API 26 y Start 4.7 WXGA API 23 este último se utilizó para ver su funcionamiento en un móvil de menor resolución.

2.3.6 *SQLite*

SQLite es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad. SQLite está integrado en la mayoría de los teléfonos móviles y computadoras, viene incluido dentro de las aplicaciones que la gente usa frecuentemente. El formato de archivo SQLite es estable, multiplataforma y compatible con versiones anteriores, los archivos de base de datos SQLite se utilizan comúnmente como contenedores para transferir contenido enriquecido entre sistemas y como formato de archivo a largo plazo para datos. El código fuente de SQLite es de dominio público y se puede utilizar para cualquier propósito (sqlite.org, s.f.).

2.3.7 *Google Firebase*

“Firebase ayuda a generar y ejecutar aplicaciones exitosas, respaldado por Google y por los equipos de desarrollo de aplicaciones” (Firebase, s.f.). *“Firebase es una plataforma en la nube para el desarrollo de aplicaciones web y móviles. Está disponible para distintas plataformas (iOS, Android y web). Su función esencial es hacer más sencilla la generación de aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad. Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de estas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis”* (Digital 55, 2020).

2.3.8 Firebase Authentication

“Firebase Authentication busca facilitar la generación de sistemas de autenticación seguros, a la vez que mejora la experiencia de integración y acceso para los usuarios finales. Proporciona una solución de identidad de extremo a extremo, compatible con cuentas de correo electrónico y contraseñas, autenticación telefónica, acceso mediante Google, Twitter, Facebook, GitHub, y más” (Firebase, s.f.).

2.3.9 Cloud Firestore

“Cloud Firestore es una base de datos de documentos NoSQL que permite almacenar, sincronizar y consultar fácilmente datos para apps móviles y web a escala global. Usa colecciones y documentos para estructurar los datos con facilidad. Genera jerarquías para almacenar datos relacionados y recuperar los datos que se necesitan mediante consultas expresivas de manera sencilla. Las consultas se escalan con el tamaño del conjunto de resultados (y no con el del conjunto de datos), por lo que cualquier aplicación está lista para escalar desde el primer día” (Firebase, s.f.).

2.3.10 Firebase Hosting

“Firebase Hosting es un servicio de hosting de contenido web con nivel de producción orientado a desarrolladores, proporciona hosting seguro y rápido para las aplicaciones web. Con un solo comando, puede implementar aplicaciones web y entregar contenido dinámico y estático en una CDN (red de distribución de contenidos) global rápidamente. También puede sincronizar Firebase Hosting con Cloud Functions o Cloud Run para compilar y alojar microservicios en Firebase” (Firebase, s.f.).

2.3.11 Sqflite.

Sqflite es un complemento Flutter para SQLite, un motor de base de datos SQL integrado, de alta confiabilidad y autónomo, que soporta los sistemas operativos iOS, Android y MacOS, permite el soporte de transacciones y lotes de datos, gestiona automáticamente las versiones durante la apertura, contiene ayudantes para insertar / consultar / actualizar / eliminar consultas, su operación de base de datos es ejecutada en un hilo en segundo plano para iOS y Android (pub.dev, 2021).

2.3.12 JSON

“JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del lenguaje de programación JavaScript. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidas por programadores de la familia de lenguajes C y otros” (json.org, s.f.).

2.3.13 Quicktype

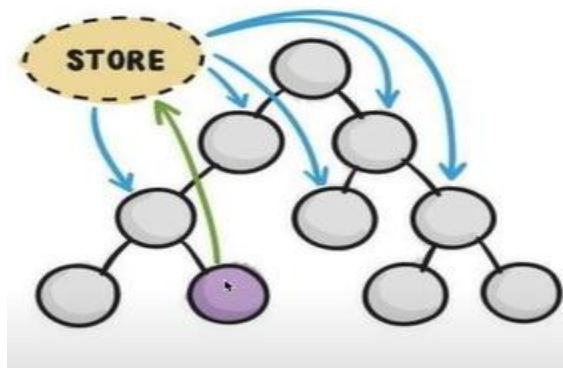
“Quicktype es una plataforma web que genera modelos y serializadores a partir de JSON, esquema y GraphQL para trabajar con datos de forma rápida y segura en cualquier lenguaje de programación, también genera tipos y código para leer, escribir y validar los datos con ello se puede utilizar el código para facilitar el análisis, la seguridad de tipos y el autocompletado” (Quicktype, 2021).

2.3.14 Patrón Provider

Este patrón es la forma más sencilla de gestionar el estado global en una aplicación de Flutter ya que permite que la información llegue de manera directa a cualquier parte del árbol de widgets, este no solo permite exponer un valor sino también crearlo o generarlo, actualizarlo o eliminarlo (Dash-overflow.net, 2021). *“El patrón Provider funciona aislándolo del árbol de widgets a una clase o a un contenedor de información que se le llama ‘Store’, y al momento de que un widget quiera comunicarse con otro el Store será quien guardará la información y después se notificará al widget que esté escuchando, es decir tener un lugar centralizado de información que se encargue de distribuirla a los demás widgets que lo necesitan”* (Herrera, 2019).

Figura 1.

Patrón Provider



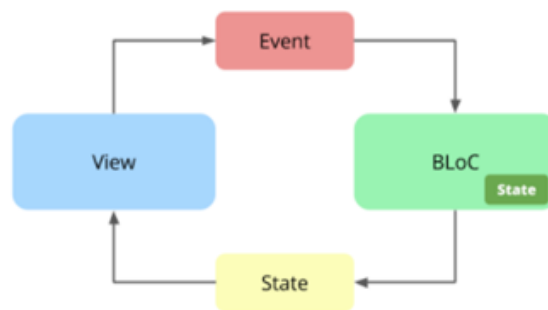
2.3.15 Patrón BLoC

“Un BLoC (Business Logic Components) es un componente intermediario entre las vistas y el modelo, es bastante simple, por lo tanto, más versátil. El patrón BLoC inicialmente, como lo

presentó Google, tiene varios objetivos: centralizar la lógica de negocio, centralizar cambios de estado y mapear al formato que necesita la vista. Por cada vista (componente o widget) o concepto lo suficientemente importante vamos a tener su correspondiente BLoC. En las clases BLoC se van a centralizar los cambios de estado que se producen en la aplicación y serían las encargadas de recibir las acciones o eventos que se producen en la aplicación, y que modifican el estado. Se basa en un flujo de datos unidireccional y para modificar el estado, se hace explícitamente, primero manejando eventos en el componente o widget y después invocando acciones o eventos en el BLoC que modifican su estado” (Sánchez, 2020).

Figura 2.

Patrón BLoC



2.3.16 GitHub

“GitHub es una plataforma que permite la generación de repositorios, es gratis y de código abierto, aunque también tiene funciones de pago, puede alojar y revisar código, administrar proyectos y crear software” (GitHub, 2012).

2.3.17 Flaticon

Fuente de iconos y descarga gratis para cualquier proyecto, iconos SVG y fuentes web de iconos vectoriales para web, iOS y aplicaciones Android, recursos hechos por y para diseñadores. En formatos PNG, SVG, EPS, PSD y BASE 64 (Flaticon, s.f.).

Los siguientes tecnofactos fueron implementados en el desarrollo del sitio web junto a estos también se utilizó Visual Studio Code anteriormente descrito.

2.3.18 Brackets

Es un editor de texto moderno que facilita el diseño en el navegador. Está diseñado desde cero para diseñadores web y desarrolladores front-end. Brinda una conexión en tiempo real a su navegador. Al realizar cambios en CSS y HTML se verán instantáneamente esos cambios en la pantalla (brackets.io, s.f.).

2.3.19 CSS

Es un lenguaje del lado del cliente que permite diseñar y decorar un sitio web, también permite definir los formatos de texto, fuentes y colores. CSS son las siglas de Cascade Style Sheets en español hojas de estilo en cascada, es un lenguaje que permite agregar atributos a los elementos de los documentos realizados en HTML. CSS también permite realizar una separación del diseño (formato y estilos) de los contenidos de las páginas web (Condor & Soria, 2014).

2.3.20 HTML5

Es un estándar que sirve para definir la estructura y el contenido de una página Web. Sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, juegos, entre otros. (Diego, 2019).

2.3.21 JavaScript

“JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado just-in-time (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, también es usado en entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje de programación basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa” (Mozilla developer network , s.f.).

2.4 Aplicaciones enfocadas en la agricultura

Actualmente en internet se encuentran aplicaciones móviles y de escritorio para cualquier necesidad o problema que surja en la vida cotidiana o laboral de las personas, mayormente en el sector de la agricultura en Colombia se identifican una gran variedad de dificultades que fueron anteriormente nombradas y para solucionar algunas de ellas es preciso implementar aplicaciones o herramientas tecnológicas como AgroLibreta, Agroptima, AgroWin, etc. Estas se describen a continuación:

2.4.1 Agroptima

“Es el software agrícola para la gestión de explotaciones agrícolas. Gestiona fincas agrícolas de la forma más sencilla e intuitiva, con la aplicación agrícola se podrá registrar online las actividades del cuaderno de campo y tener el control de los costos de explotación y gestión agrícola de forma fácil y sencilla” (Agroptima, 2017).

2.4.2 AgroWin

Es un sistema especialmente diseñado para ayudarle al empresario agrícola en la gestión, planeación y seguimiento de la empresa y sus recursos, además permite la disminución de los costos, el mejoramiento de los ingresos, el aumento de las utilidades y llevar la contabilidad agropecuaria de manera automática (AgroWin, 2021).

2.5 Marco metodológico

El mecanismo utilizado para el desarrollo de este trabajo de grado fue un conjunto de procedimientos, el cual consistió en implementar la metodología Kanban y el prototipado al mismo tiempo, es decir que estos se aplicaron antes, durante y después del desarrollo. El prototipado permitió avanzar rápidamente en las inconsistencias encontradas y con la metodología Kanban se tuvieron claras las actividades que cada desarrollador debería realizar. A continuación se describe este conjunto de metodologías:

2.5.1 Metodología de prototipado

En el prototipado se busca gestionar un primer prototipo, que es una versión funcional de un sistema de información o una parte del sistema, de manera rápida para evaluarlo con los usuarios y así identificar falencias para que ellos den una mejor idea de los requerimientos de información, posteriormente se hacen las mejoras necesarias y se vuelve a evaluar, este proceso se repite cuantas veces haga falta hasta obtener un sistema lo más operacional posible.

Según Laudon & Laudon, el objetivo del prototipo es *“ser un modelo antecesor. Una vez operacional, el prototipo se refinará gradualmente hasta que cumpla de manera precisa con los requerimientos de los usuarios”* (2016).

Pasos en el prototipado y forma en la que se desarrolló, según la Figura 3:

1. Identificar los requerimientos básicos del usuario: El diseñador del sistema trabaja con el usuario sólo el tiempo suficiente para capturar las necesidades básicas de información. En este paso se realizó la interacción con los usuarios por medio de entrevistas de manera personal a las personas de la comunidad y a personas que han trabajado allí, haciéndoles una serie de preguntas y proponiendo vistas de usuario. De esas entrevistas se identificó la problemática en la tardanza para obtener un total de los costos en tiempo real.

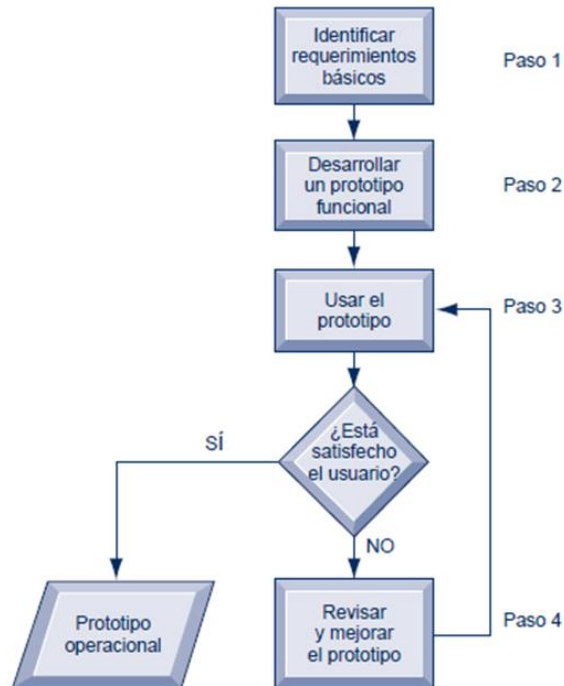
2. Implementar un prototipo inicial. El diseñador del sistema genera rápidamente un prototipo funcional mediante el uso de herramientas para generar software con rapidez. Por medio de diagramas de modelado UML y el desarrollo de un modelo de datos para el aplicativo móvil, se implementó el primer prototipo con las funciones para generar: un modelo de referencia ideal el cual no es modificable por el usuario, el registro de costos y el informe general.

3. Acompañar al usuario en el uso del prototipo. Se anima al usuario a trabajar con el sistema para que haga sugerencias sobre cómo mejorar el prototipo y así determinar qué tan bien cumple con sus necesidades. Para llevar a cabo este paso se explicó a los usuarios el uso de las funcionalidades y los posibles ejemplos en cada una de ellas, al mismo tiempo se estuvo atento a todas las sugerencias o comentarios para identificar lo que se debía mejorar en un segundo prototipo.

4. Revisar y mejorar el prototipo. Se registran los cambios que el usuario solicita, y se refina apropiadamente el prototipo. Después de que se ha revisado el prototipo, el ciclo regresa al paso 3. Los pasos 3 y 4 se repiten hasta que el usuario queda satisfecho. Teniendo claras las modificaciones se procedió a desarrollar el segundo prototipo al cual se le añadieron las funcionalidades para permitir al usuario registrar modelos de referencia de tal forma que lo plantee acorde a sus necesidades, también se añadió la funcionalidad para gestionar soportes de costos, los cuales son fotografías de las facturas que el usuario posea para que pueda asociar a los registros de costos una fotografía y así conocer qué valores están soportados. Se repitió el paso 3 y se procedió al desarrollo del tercer prototipo que incluye, las funciones para: comparar los costos registrados con el modelo de referencia por medio de gráficas en el informe general, realizar copias de seguridad y ajustes en general para un mejor desempeño posible.

Una vez se tuvo un prototipo estable final de la aplicación, se procedió a desarrollar un sitio web para: presentar las instrucciones de uso, la gestión de copias de seguridad y la descarga de la aplicación. Al finalizar el desarrollo con ayuda de un plan de pruebas se validó el sistema en conjunto con la experiencia de algunos pequeños productores y estudiantes de agroindustrial pertenecientes al IPRED.

Figura 3.

Pasos del prototipado**2.5.2 Metodología Kanban**

La metodología Kanban forma parte de las metodologías ágiles para desarrollo de software y su objetivo principal es gestionar la realización de las tareas hasta su finalización, generando un flujo de trabajo visual por medio de un tablero ya sea físico o digital con el fin de planear y hacer un seguimiento a dichas tareas utilizando tarjetas y columnas pero lo más importante es el progreso constante para así evitar la acumulación de trabajo pendiente maximizando la eficiencia, también permite mejorar el flujo y la calidad del trabajo. Uno de los tableros más comunes para implementar esta metodología se compone de las columnas 'Pendientes', 'En proceso' y 'Hecho', pero Kanban se diferencia del resto de metodologías por las siguientes características:

- Visualizar el flujo de trabajo: Para visualizar un proceso se necesita un tablero con tarjetas que representan una tarea del trabajo y columnas que son un paso en el flujo de trabajo de tal manera que cualquier persona identifique qué debe hacer y al realizar una tarea esta se debe mover por las columnas del tablero. Para el desarrollo de la aplicación con ayuda de la herramienta web Trello se crearon las tarjetas cada una con un requerimiento funcional y no funcional estas fueron la lista de tareas, las cuales se movieron en el transcurso del tiempo por las columnas 'En proceso', 'Pendientes de revisión' y 'Hecho', con el fin de tener un flujo de trabajo eficaz y organizar adecuadamente el trabajo en equipo.
- En proceso: Lo cual quiere decir que es un sistema que mejora para el desarrollo de proyectos basados en los objetivos que se quiere conseguir. En esta etapa se podía ver cuales tarjetas estaban siendo realizadas por cada integrante del grupo, lo cual hizo más fácil el desarrollo de la aplicación porque ya cada uno se hacía responsable de dicha tarea pasándola por cada columna hasta que llegara a la columna Hecho y así se finalizó con las tarjetas.
- Priorización según importancia y urgencia: Esta característica hace que en el método se haga lo justo y necesario, es decir que prima la calidad de cada tarea no la rapidez, lo cual permitió desarrollar las tareas según la importancia que tenían para el proyecto, gestionándolas mejor para no caer en actividades secundarias o no tan necesarias y así llegar a tener el menor margen de error al día de las pruebas.
- Seguimiento del tiempo: Las tareas acumuladas o pendientes indican que se deben realizar en su momento, pero también hay otras que se pueden priorizar en función a sus necesidades, es decir que si se lleva un control o se mide el tiempo cada tarea puede

llegar a su fin, pero cumpliendo con lo planeado. Esta fue de gran ayuda ya que se tuvo un control del tiempo invertido para el desarrollo de cada tarea lo que es fundamental para que se pueda entregar a tiempo la app y cumplir con las fechas establecidas (APD, 2021).

3 Resultados

En este capítulo se presentan los resultados de acuerdo con cada uno de los objetivos específicos, según la metodología de prototipado y Kanban. Esta sección se encuentra acompañada de tres videos los cuales se realizaron con el fin de evidenciar el resultado obtenido en cada uno de los objetivos que fueron propuestos. Estos se presentan a continuación:

- El primero correspondiente al numeral 3.1 para explicar el producto final de la aplicación móvil, se encuentra en esta URL <https://youtu.be/JsD68gFJ4ic> .
- El segundo para describir el producto final del sitio web correspondiente al numeral 3.2 en la siguiente URL: <https://youtu.be/LcY7UZm8NHI>.
- El tercero corresponde al numeral 3.3, para demostrar en Firebase los usuarios que utilizaron la aplicación, se puede encontrar en la siguiente URL: <https://youtu.be/IXGUr8hSn7s>.

También se realizaron otros videos que tienen como objetivo ubicar al lector en la evaluación del funcionamiento de la aplicación móvil y del sitio web. En esta introducción se describen todos los videos correspondientes a este capítulo.

Para realizar las pruebas beta que se describen en el numeral 3.3.4 se realizaron tres videos con el fin de complementar y facilitar la experiencia de los usuarios al utilizar la aplicación móvil AgroLibreta, estos son los siguientes:

- En el primero se presenta una introducción a AgroLibreta, el cual se puede consultar en la URL <https://youtu.be/o0RUVGA8-No> .
- En el segundo se presenta una explicación de donde surgió la idea y algunos conceptos o términos para una mayor comprensión, lo puede encontrar en la siguiente URL: <https://youtu.be/O5NRQTObc2g> .
- En el tercero se les propone un paso a paso de cómo utilizar AgroLibreta, lo puede encontrar en la siguiente URL: <https://youtu.be/vpjVZLpiBEo> .

3.1 Aplicación móvil

En este numeral se describe lo correspondiente al primer objetivo específico de este trabajo de grado, el detalle del proceso de desarrollo es presentado a continuación.

Una demostración del producto final de AgroLibreta se encuentra en la primera URL de la introducción anterior correspondiente a un video explicativo de un paso a paso para guiar a los usuarios en su uso.

3.1.1 Requerimientos

En la Tabla 2 se listan los requerimientos funcionales y en la Tabla 3 se listan los requerimientos no funcionales de la aplicación móvil. Para ver las especificaciones de cada requerimiento revisar la sección de *Apéndice A*, Tabla 7 y Tabla 8.

Tabla 2.

Requerimientos funcionales de la aplicación móvil

IDENTIFICADOR	NOMBRE
RF00	Iniciar sesión
RF01	Cerrar sesión
RF02	Registra cultivo
RF03	Listar cultivos
RF04	Registrar un modelo de referencia
RF05	Generar un modelo de referencia por defecto
RF06	Registrar, editar y eliminar costo
RF07	Consultar costos con filtros
RF08	Registrar y eliminar fotografía de un evento comercial
RF09	Asociar costos a registros fotográficos
RF10	Ver costos asociados a registros fotográficos
RF11	Ver informe general del cultivo
RF12	Registrar producto o actividad
RF13	Ver gráfica comparativa con el modelo de referencia
RF14	Generar un PDF del informe general del cultivo
RF15	Registrar Ubicación
RF16	Generar precio de venta sugerido
RF17	Visualizar el perfil de usuario y editarlo
RF18	Cambiar estado del cultivo
RF19	Generar y restaurar copia de seguridad

Tabla 3.

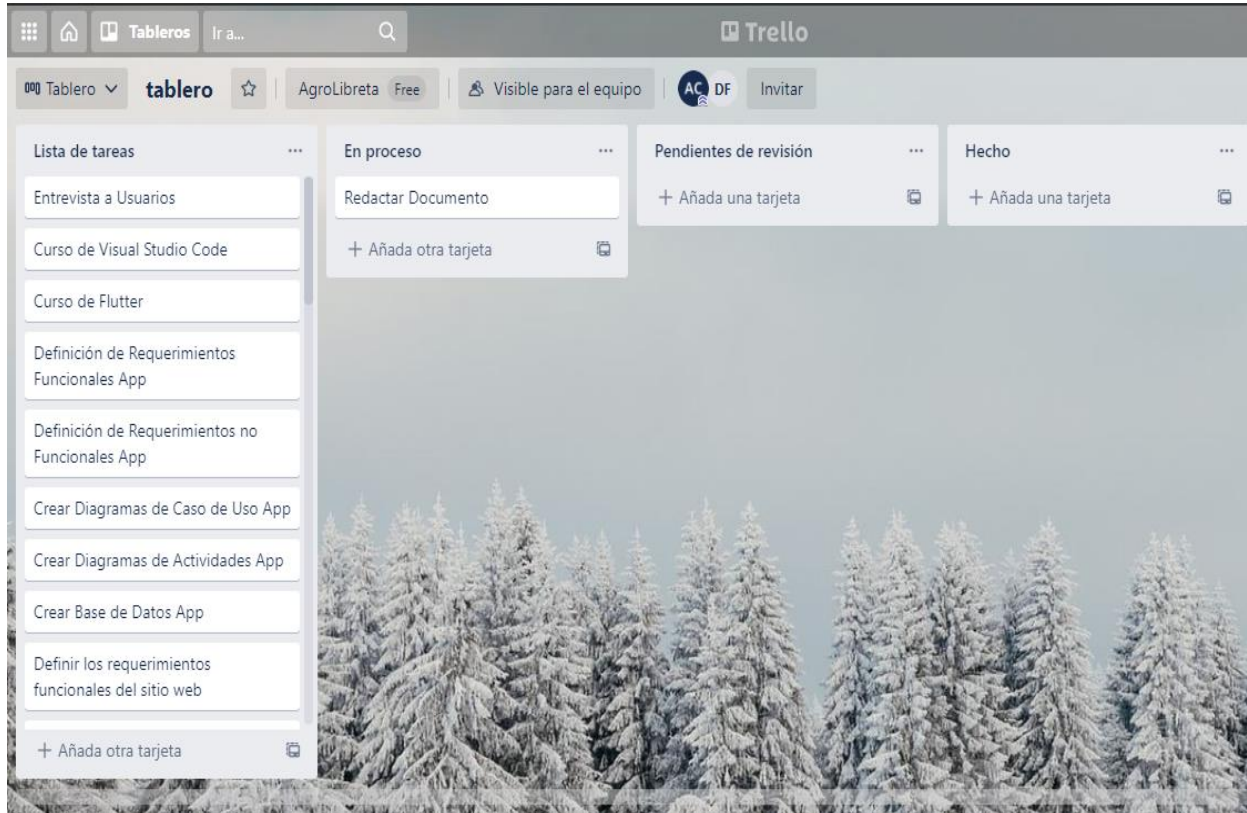
Requerimientos no funcionales de la aplicación móvil

IDENTIFICADOR	NOMBRE
RNF01	Diseñar una interfaz gráfica de usuario
RNF02	Facilidad de mantenimiento y actualización

IDENTIFICADOR	NOMBRE
RNF03	Actualizar la información en tiempo real en la base de datos
RNF04	Seguridad de los datos
RNF05	Rapidez al momento de generar informes
RNF06	Compatible con versiones de Android 5.0 o superior

3.1.2 *Plan de desarrollo para el prototipo*

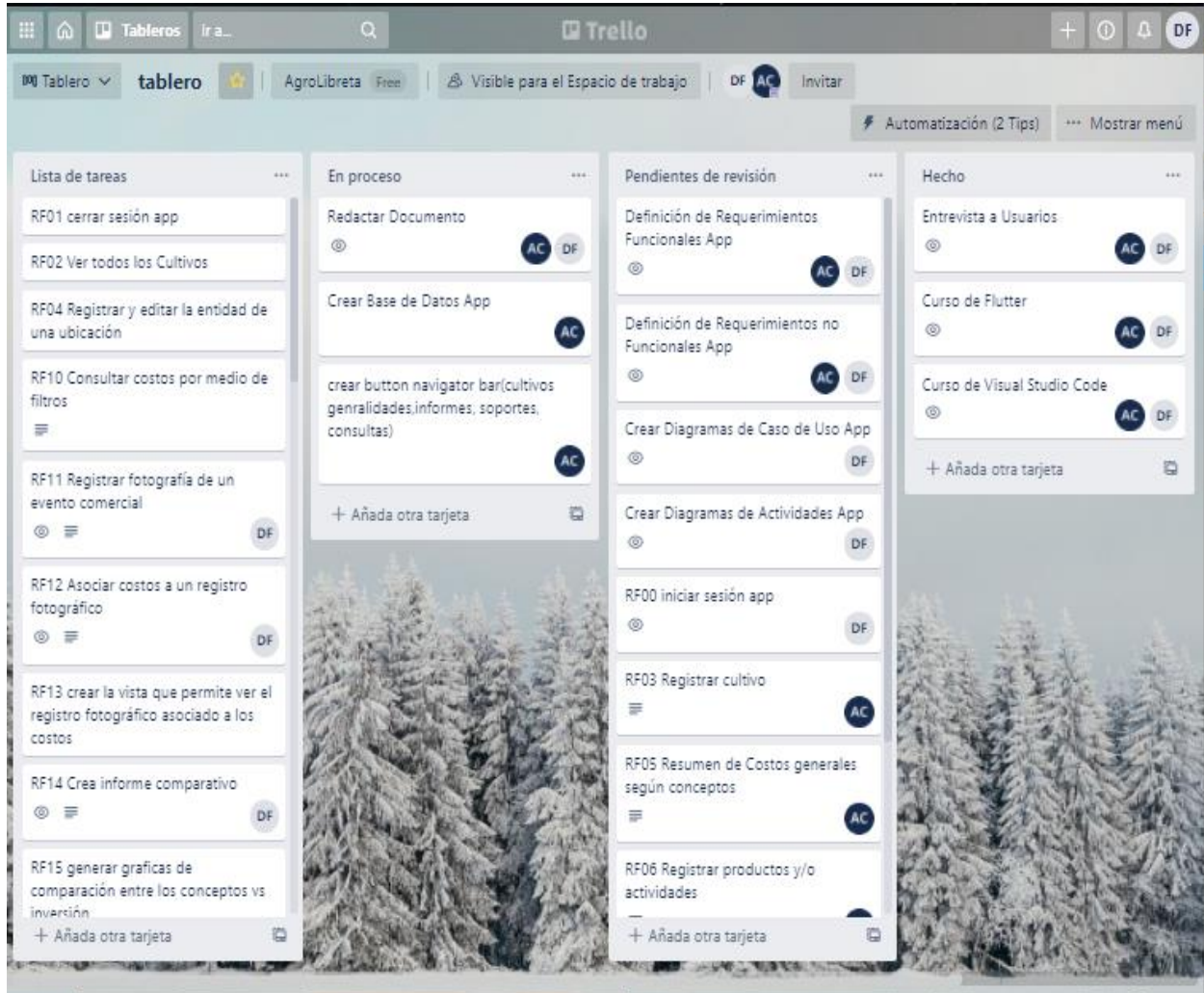
Una vez definidos los requerimientos funcionales y no funcionales de la aplicación móvil se implementó la metodología Kanban para organizar por tareas el desarrollo de los prototipos. A través de la herramienta Trello se generaron las tarjetas mostradas a continuación que permitían llevar un control del seguimiento para el desarrollo de la aplicación móvil y del sitio web. La Figura 4 representa el tablero de Trello al inicio del proyecto, en el cual se pueden observar que la mayoría de las tarjetas se encuentran en la columna ‘Lista de tareas’ a excepción de la tarjeta ‘Redactar documento’ que desde el inicio del proyecto ha estado en la columna ‘En proceso’.

*Figura 4.**Tablero de la metodología Kanban al inicio del proyecto*

La Figura 5 representa el tablero de Trello durante el proceso de desarrollo del proyecto, en ella se observa que las tarjetas ya se encuentran distribuidas en las columnas ‘En proceso’, ‘Pendiente de revisión’ y ‘Hecho’.

Figura 5

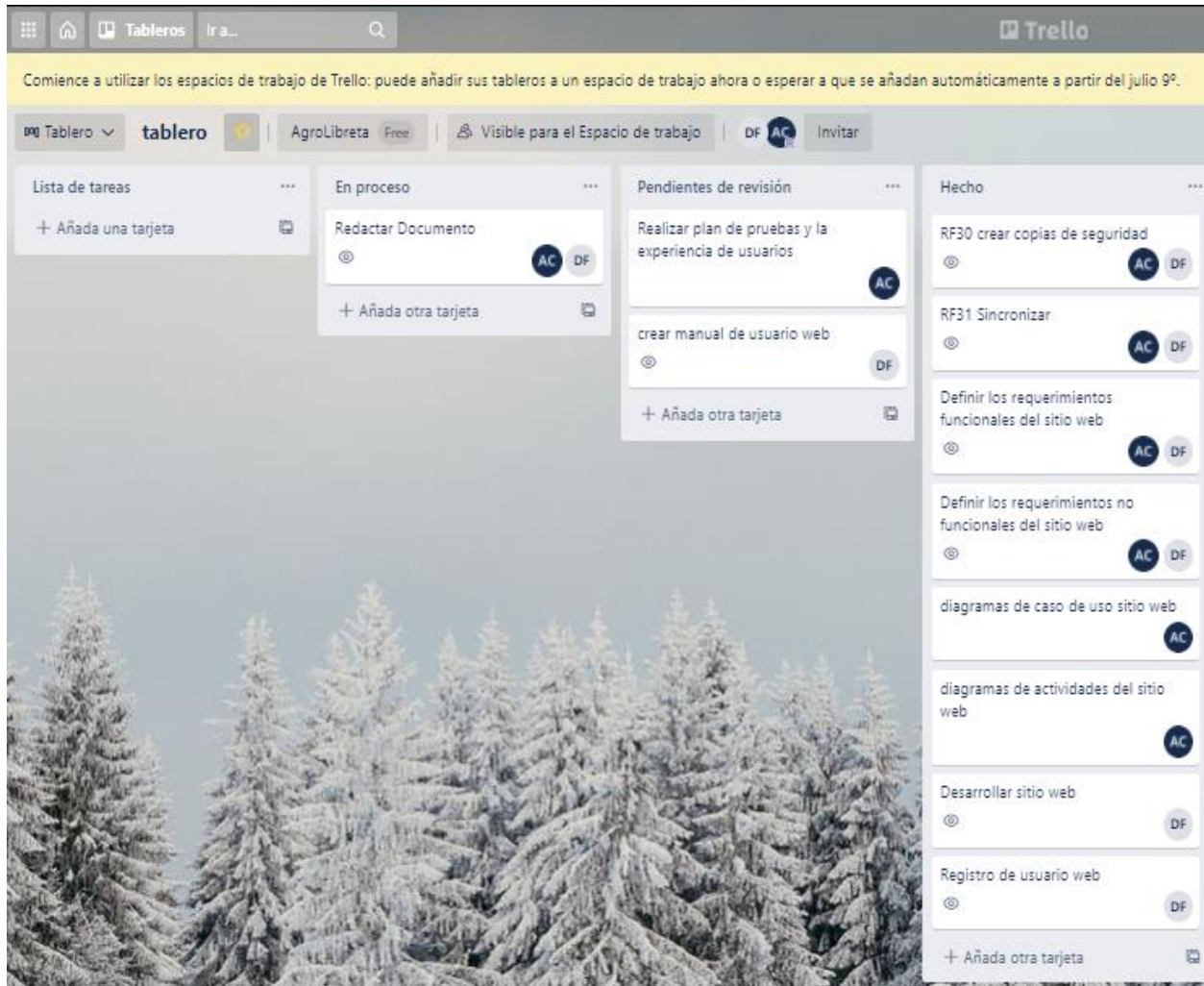
Tablero metodología Kanban en el desarrollo del proyecto



La Figura 6 representa el tablero de Trello cuando finalizó el desarrollo de la aplicación móvil y se observa que falta solo la redacción de este documento.

Figura 6

Tablero metodología Kanban finalizando el proyecto



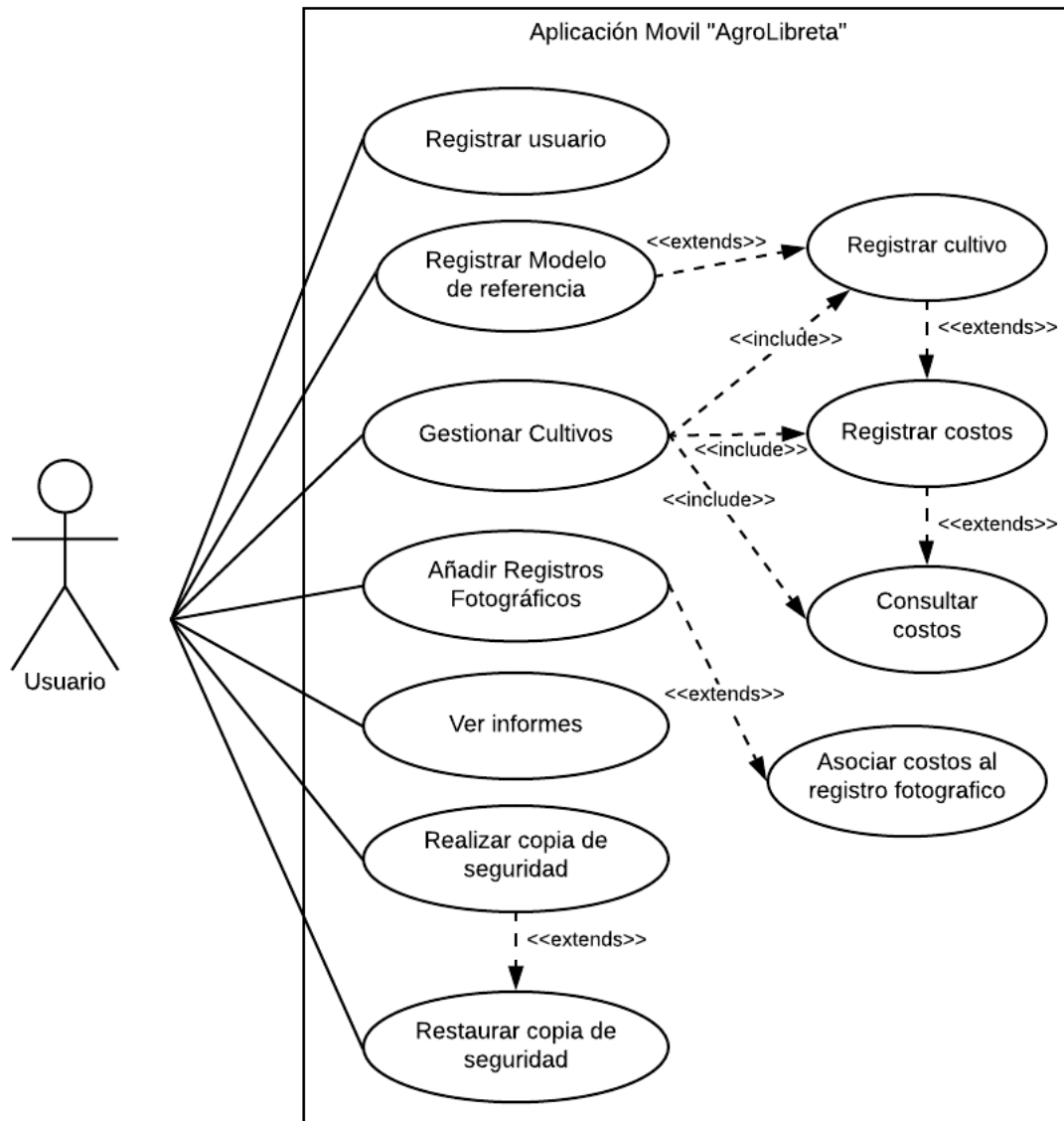
3.1.3 Diagrama de casos de uso

Según los requerimientos funcionales mencionados se generó el diagrama de casos de uso donde se presentan en la Figura 7 los que se consideraron más relevantes para el usuario, se omitieron por motivos de sencillez y fácil comprensión los casos de uso 'Iniciar sesión' el cual estaría enlazado a los demás casos y el de 'Editar perfil' el cual estaría enlazado a registrar

usuario. Además, el caso de uso ver informe se refiere a que el usuario puede disponer de la información generada durante el ciclo de producción de un cultivo.

Figura 7.

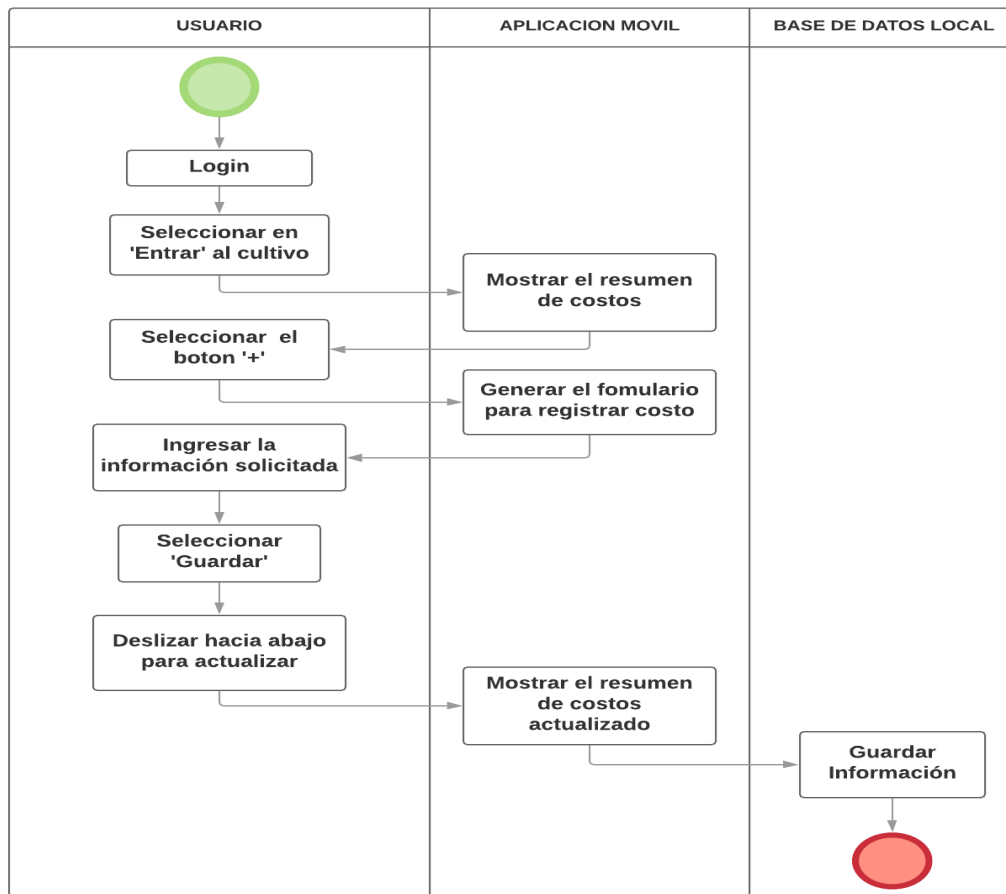
Diagrama de casos de usos de la aplicación móvil



3.1.4 Diagramas de actividades

Los diagramas de actividades que se generaron para los requerimientos de esta aplicación móvil son: 'Registrar un cultivo', 'Registrar un modelo de referencia' y 'Registrar fotografía de un evento comercial', a continuación se presenta el diagrama de actividades del requerimiento 'Registrar costo' (Figura 8), se puede observar que estos diagramas representan las actividades que hace tanto el usuario como la aplicación móvil y como la base de datos local o el storage. Los demás diagramas se pueden consultar en el *Apéndice B*, en la Figura 32, Figura 33 y Figura 34.

Figura 8.
Diagrama de actividades para registrar costo



3.1.5 *Diagramas de bases de datos*

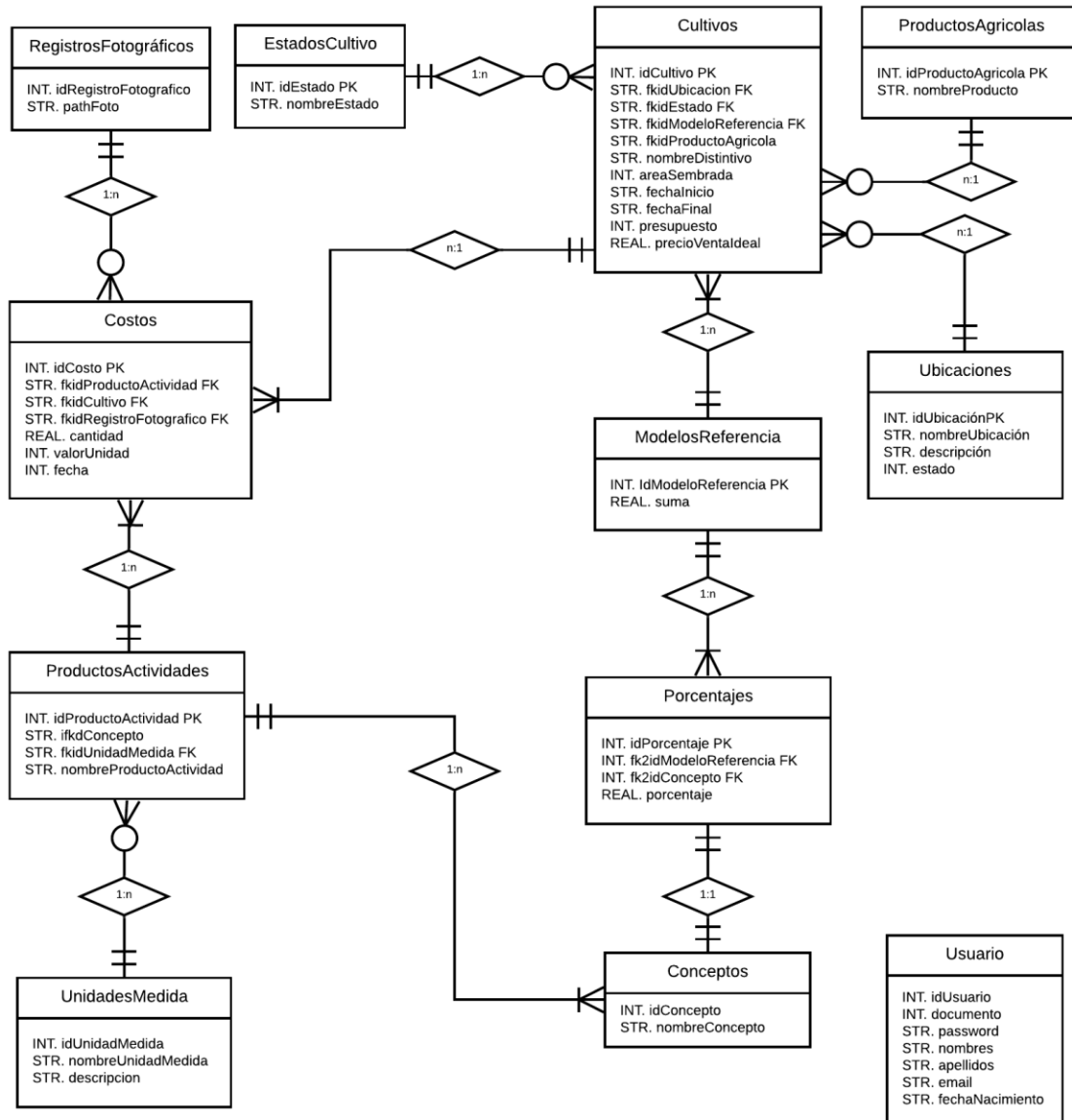
Existen dos bases de datos para cada usuario, una que almacena de manera local en el dispositivo móvil con la estructura que se presenta en la Figura 9 y otra en Firebase que almacena los datos de manera remota representada en la Figura 10. En la base de datos local se utiliza SQLite, es decir, es relacional y para la remota que es NoSQL se utiliza una estructura de colecciones y documentos.

La base de datos para almacenar la copia de seguridad de forma remota está en Cloud Firestore que es un servicio de Firebase, tiene una colección principal llamada 'users', esta colección contiene documentos identificados por el correo electrónico del usuario, cada documento tiene asociados un conjunto de colecciones (los nombres de esas colecciones son 'Cultivos', 'Costos', 'ProductosActividades', 'Porcentajes', 'UnidadesMedida', 'Usuario', 'ModelosReferencia' y 'Ubicaciones') finalmente las colecciones tienen asociados n documentos identificados por un número entero único que contienen los registros generados por el usuario.

La tabla productos agrícolas solo cumple la función de indicar cómo se debe relacionar si en futuras versiones la aplicación se adapta para cultivos de diferentes productos agrícolas.

Figura 9.

Diagrama de la base de datos local

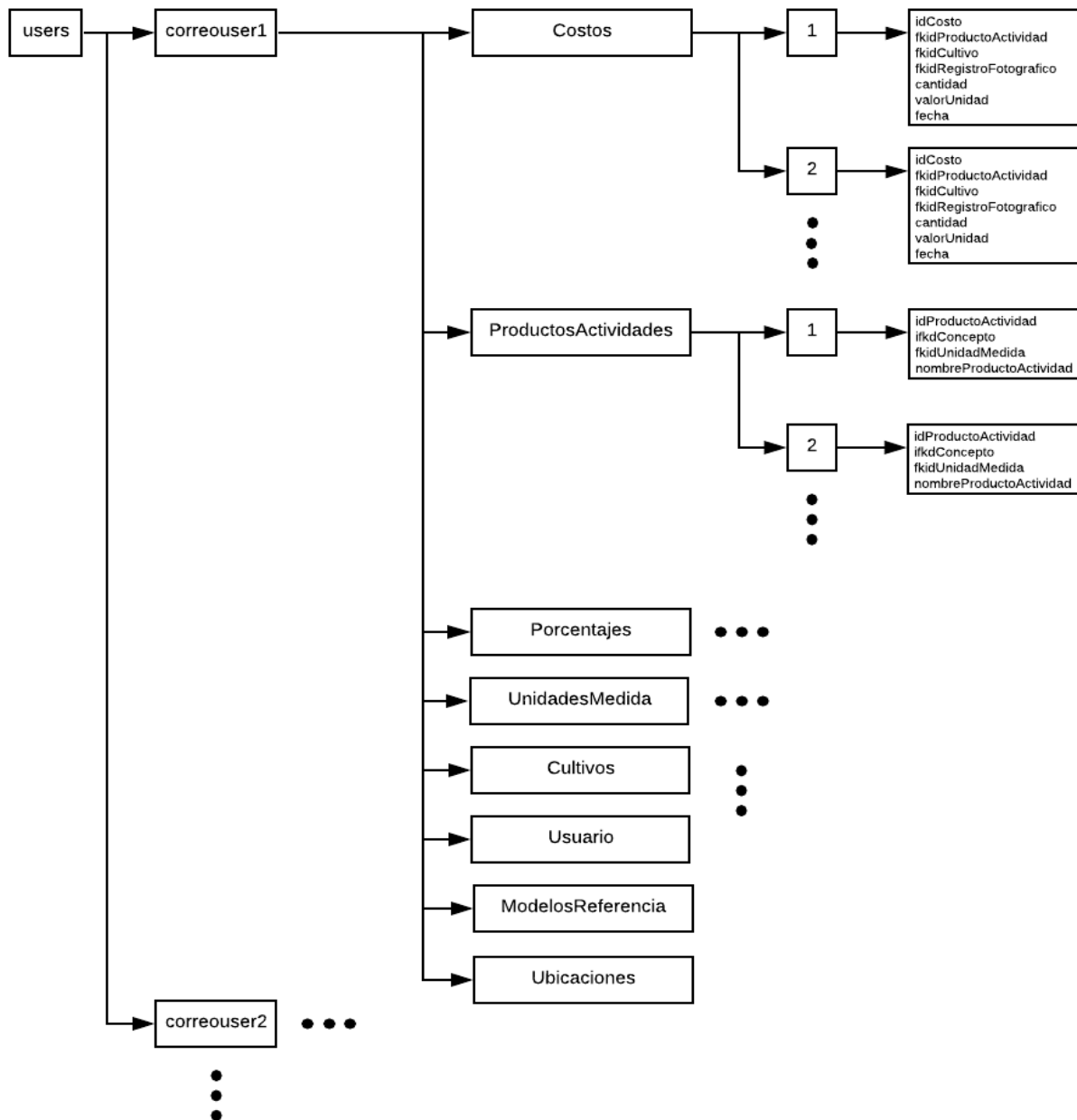


Por el tamaño y complejidad del diagrama de la base de datos remota se ponen puntos suspensivos que indican continuación, los atributos que no están en el diagrama tienen el mismo nombre que los de su correspondiente tabla en la base de datos local. Las tablas

'ProductosAgricola', 'EstadosCultivo', 'RegistrosFotográficos y 'Conceptos' no tienen una colección correspondiente porque no era necesario o por limitaciones del alcance del proyecto.

Figura 10.

Diagrama de la base de datos remota



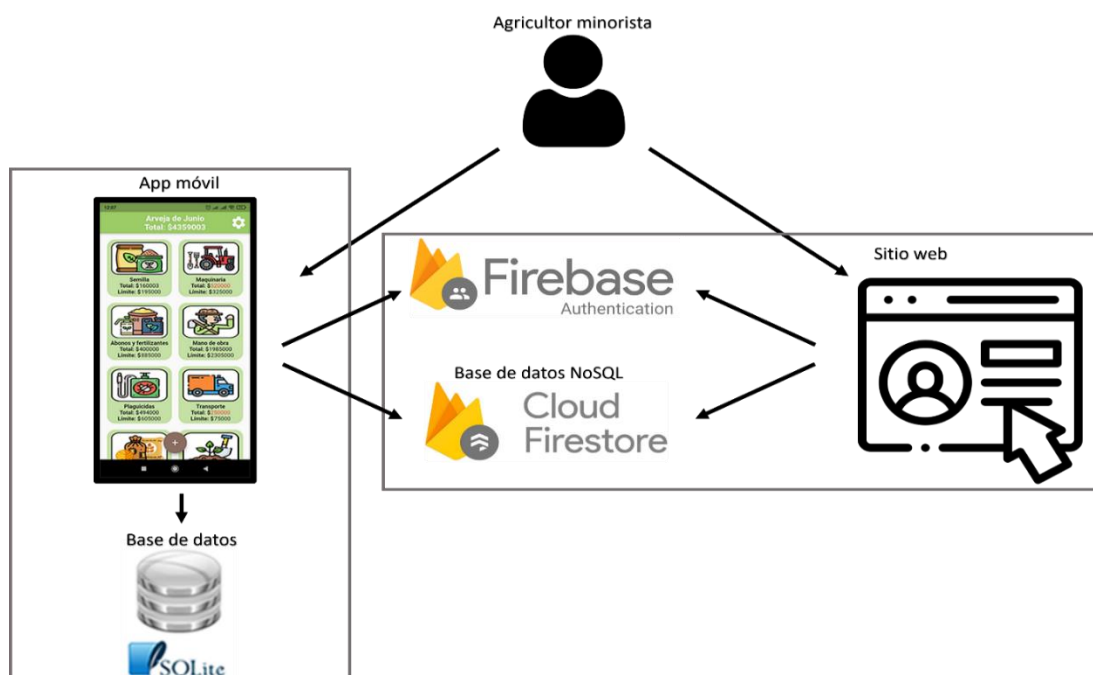
3.1.6 Diagrama de arquitectura

En el diagrama de arquitectura se agrupan la aplicación móvil y la base de datos local porque ambas se encuentran en el dispositivo móvil. Cloud Firestore, Firebase Authentication y el sitio web están alojados en Firebase.

La Figura 11 representa que el usuario puede acceder tanto a la aplicación móvil como al sitio web, desde el sitio web se puede autenticar en Firebase Authentication y consultar la fecha de la última copia de seguridad que está en Cloud Firestore. Desde la aplicación móvil el usuario puede registrarse y autenticarse en Firebase Authentication además de subir y restaurar una copia de seguridad en Cloud Firestore. La base de datos local permite almacenar datos desde la aplicación sin tener conexión a internet.

Figura 11.

Diagrama de arquitectura del sistema



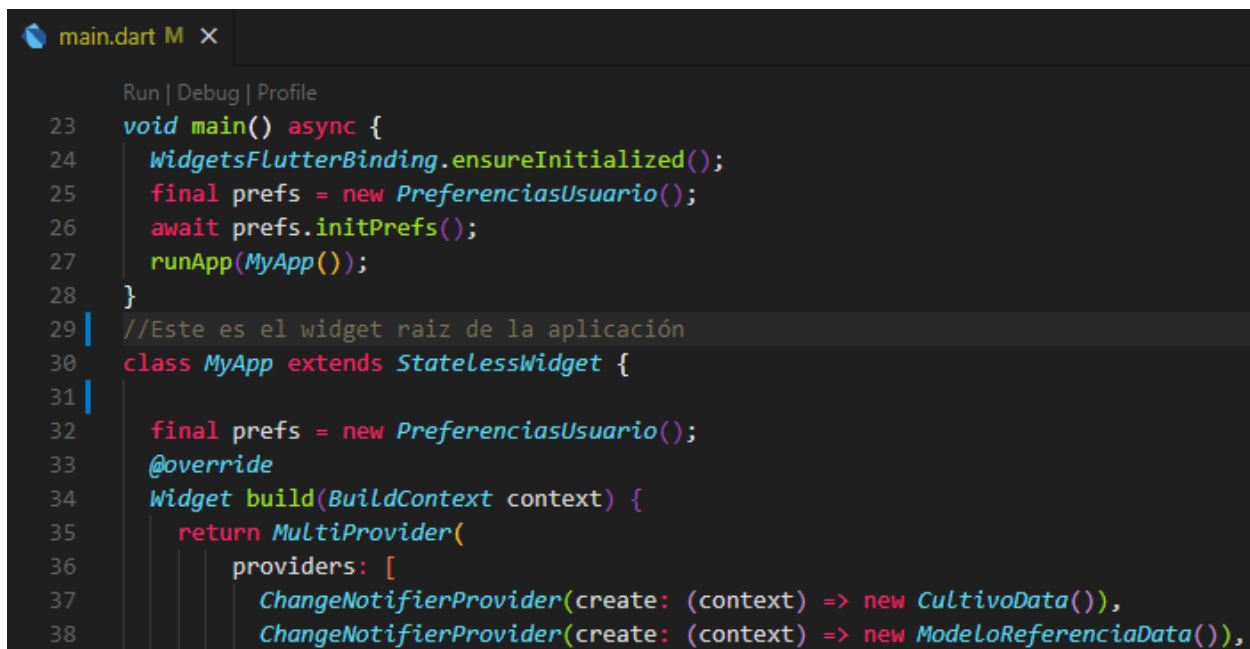
3.1.7 Implementación

En este numeral se presentan algunas descripciones y figuras para demostrar de manera general la implementación de la aplicación, realizada en el editor de código fuente ligero Visual Studio Code y el uso de las tecnologías relacionadas con el desarrollo de la aplicación. Para una visión más amplia consultar en la sección de *Apéndice C*.

El archivo principal 'main.dart' presentado en la Figura 12 es donde se declara el widget raíz en la línea 30 y se crean los Provider donde cualquier widget puede acceder a la información que se disponga justo en la línea 35.

Figura 12.

Archivo main



```
main.dart M X
Run | Debug | Profile
23 void main() async {
24   WidgetsFlutterBinding.ensureInitialized();
25   final prefs = new PreferenciasUsuario();
26   await prefs.initPrefs();
27   runApp(MyApp());
28 }
29 //Este es el widget raíz de la aplicación
30 class MyApp extends StatelessWidget {
31
32   final prefs = new PreferenciasUsuario();
33   @override
34   Widget build(BuildContext context) {
35     return MultiProvider(
36       providers: [
37         ChangeNotifierProvider(create: (context) => new CultivoData()),
38         ChangeNotifierProvider(create: (context) => new ModeloReferenciaData()),
```

Para facilitar la navegación entre pantallas se usó un archivo de rutas llamado 'routes.dart' en la Figura 13. El archivo contiene una función que devuelve un mapa, este genera las rutas de la aplicación, justo en la línea 32 y además en las líneas 27, 28, y 29 se puede evidenciar algunas utilidades que se importaron para este archivo.

Figura 13.

Archivo routes



```
routes.dart M x
27 import 'package:agrolibreta_v2/src/pages/utilidades_folder/producto_
28 import 'package:agrolibreta_v2/src/pages/utilidades_folder/ubicacion
29 import 'package:agrolibreta_v2/src/pages/utilidades_folder/unidad_me
30
31 //función que devuelve el mapa de rutas
32 Map<String, WidgetBuilder> getApplicationRoutes() {
33   return <String, WidgetBuilder>{
34     'taps': (BuildContext context) => TapsPage(),
35     'home': (BuildContext context) => HomePage(),
36     'utilidades': (BuildContext context) => UtilidadesPage(),
37     'informe': (BuildContext context) => InformeCultivoPage(),
38     'costos': (BuildContext context) => CostosPage(),
39     'galeriaRegistrosFoto': (BuildContext context) => GaleriaRegistr
40     'resumenCostos': (BuildContext context) => ResumencostosPage(),
41     'perfilUsuario': (BuildContext context) => PerfilUsuarioPage(),
```

Para la implementación de la base de datos local se utilizó la librería llamada 'sqlite', en la Figura 14 se puede observar una parte del archivo 'db_provider.dart' donde se encuentra la declaración de la clase 'DBProvider' que se encarga de generar o inicializar la base de datos, en la línea 7 se puede observar la instancia de 'Database' llamada '_database' la cual se inicializa si no existe base de datos y también se puede observar otra instancia de la clase 'DBProvider' llamada 'db' que se utiliza para generar las tablas en ese mismo archivo.

Figura 14.

Clase para acceder a la base de datos local

```
db_provider.dart M x
6 class DBProvider {
7   static Database _database;
8   static final DBProvider db = DBProvider._();
9   DBProvider._();
10  //Si la base de datos ya existe se accede a e
11  Future<Database> get database async {
12    if (_database != null) return _database;
13    //si la base de datos no existe debe crearse
14    _database = await initDB();
15
16    return _database;
17  }
```

En la Figura 15 en la línea 24 se observa un ejemplo del método en el que se generan o se declaran las entidades de la base de datos cuando se instala y se ingresa por primera vez. En esta imagen solo se aprecia el ejemplo para la entidad 'Cultivos' con sus respectivos atributos, pero en el archivo original 'db_provider.dart' se generan las entidades de la base de datos local con la misma estructura de este ejemplo, su diagrama lo puede encontrar en el numeral 3.1.5

Figura 15.

Método para crear la base de datos local

```
db_provider.dart M x
18 //Función para crear la base de datos
19 Future<Database> initDB() async {
20   // Path de donde almacenaremos la base de datos
21   Directory documentsDirectory = await getApplicationDocumentsDirectory();
22   final path = join(documentsDirectory.path, 'Agrolibreta.db');
23   // Crear base de datos
24   return await openDatabase(path, version: 1, onOpen: (db) {},
25     onCreate: (Database db, int version) async {
26     await db.execute('''
27       CREATE TABLE Cultivos(
28         idCultivo INTEGER PRIMARY KEY AUTOINCREMENT,
29         fkidUbicacion STRING NOT NULL,
30         fkidEstado STRING NOT NULL,
31         fkidModeloReferencia STRING NOT NULL,
32         fkidProductoAgricola STRING NOT NULL,
33         nombreDistintivo STRING NOT NULL,
34         areaSembrada INTEGER NOT NULL,
35         fechaInicio STRING NOT NULL,
36         fechaFinal STRING ,
37         presupuesto INTEGER NOT NULL,
38         precioVentaIdeal REAL ,
39         FOREIGN KEY (fkidUbicacion) REFERENCES Ubicaciones (idUbicacion),
40         FOREIGN KEY (fkidEstado) REFERENCES EstadosCultivo (idEstado),
41         FOREIGN KEY (fkidModeloReferencia) REFERENCES ModelosReferencia (id
42         FOREIGN KEY (fkidProductoAgricola) REFERENCES ProductosAgricolas (id
43       )
44     ''');
```

Para realizar las consultas se generó un archivo por cada tabla en el cual se define una clase con los métodos para crear o generar, leer, actualizar o borrar según sea requerido. En la Figura 16 se presenta el archivo llamado 'modelos_referencia_operations.dart' que presenta la implementación de la clase 'ModelosReferenciaOperations' con un método para generar un registro de modelo de referencia en la base de datos justo en la línea 10 de la figura, para esto es necesario indicar el nombre de la tabla y pasar el modelo JSON.

Figura 16.

Clase con el método para crear un registro de un modelo de referencia

```
modelos_referencia_operations.dart M X
5  class ModelosReferenciaOperations {
6      ModelosReferenciaOperations modelosReferenciaOperations;
7      //referencia a la base de datos
8      final dbProvider = DBProvider.db;
9      //C - crear
10     Future<int> nuevoModeloReferencia(
11         ModeloReferenciaModel nuevoModeloReferencia) async {
12         final db = await dbProvider.database;
13         final res =
14             await db.insert('ModelosReferencia', nuevoModeloReferencia.toJson());
15         // Es el ID del último registro insertado;
16         return res;
17     }
```

En la Figura 17 se presenta como es la implementación de un Provider, estos se definen como cualquier otra clase con la diferencia que agregan la clase 'ChangeNotifier' la cual permite tener la información de manera global y utilizar la función 'NotifyListeners' que se utiliza para notificar a cualquier widget del árbol si hubo cambios en el estado de los datos. El Provider hace uso de la clase que contiene las operaciones CRUD y por medio de esta acceder a los datos, vemos que esta instancia es llamada '_proActOper' y utilizada en la clase 'ProductoActividadData' generada en la línea 9.

Figura 17.

Provider que maneja información de productos o actividades

```
productos_actividades_data_provider.dart X
1  import 'package:flutter/material.dart';
2  import 'package:agrolibreta_v2/src/modelos/producto_actividad_model.dart';
3  import 'package:agrolibreta_v2/src/data/producto_actividad_operations.dart';
4
5  //Clase que realiza las operaciones CRUD de los productos o actividades en la base de datos
6  final ProductoActividadOperations _proActOper = new ProductoActividadOperations();
7
8  //provider para manejar el estado de los datos de los productos o actividades
9  class ProductoActividadData with ChangeNotifier {
10     //Lista para almacenar los productos o actividades consultados a la base de datos
11     List<ProductoActividadModel> productosActividades = [];
12     //constructor que llama la función getProductoActividad para cargar los datos
13     ProductoActividadData() {
14         this.getProductoActividad();
15     }
16     //Función que accede a los registro de los productos o actividades por medio de la clase Pr
17     //declarada al inicio como _proActOper
18     getProductoActividad() async {
19         final resp = await _proActOper.consultarProductosActividadesOrder();
20         this.productosActividades = [...resp];
21         notifyListeners();
22     }
23 }
```

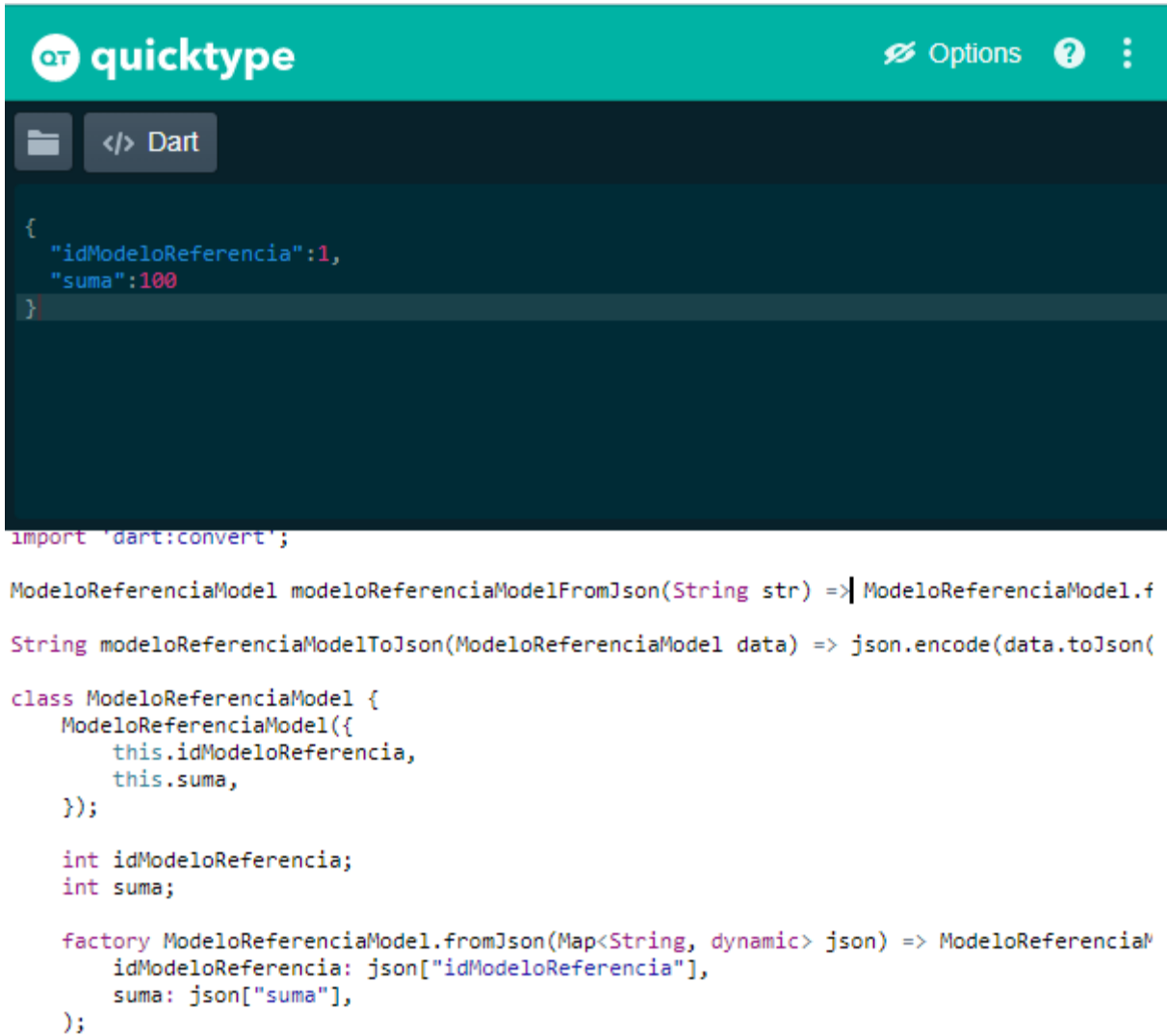
En la Figura 18 se presenta la implementación de un BLoC en la clase llamada 'Login RegistroBloc', este patrón se usó para el manejo del estado del correo electrónico y la contraseña a la hora del registro de usuarios en la aplicación y también para el ingreso ya que es más sencillo utilizarlo para funciones de validación, se puede observar a partir de la línea 6.

Figura 18.

Patrón BLoC

```
6  class LoginRegistroBloc with Validators{
7
8  final _emailController    = BehaviorSubject<String>();
9  final _passwordController = BehaviorSubject<String>();
10
11 //Insertar valores al Stream (entradas)
12 Function(String) get changeEmail    => _emailController.sink.add;
13 Function(String) get changePassword => _passwordController.sink.add;
14
15
16 //Recuperar los datos del Stream (salidas)
17 Stream <String> get emailStream    => _emailController.stream.transform(validarEmail);
18 Stream <String> get passwordStream => _passwordController.stream.transform(validarPassword);
19
20 //si el fomulario es valido activar el boton
21 Stream<bool> get formValidStream =>
22   Rx.combineLatest2(emailStream, passwordStream, (e, p) => true);
23
24 //Obtenr el ultimo valor ingresado a los streams
25 String get email    => _emailController.value;
26 String get password => _passwordController.value;
27
28
29 dispose(){
30   _emailController?.close();
31   _passwordController?.close();
32
33 }
34 }
```

Con ayuda de la herramienta Quicktype se generaron los modelos JSON de las entidades, los cuales permiten manejar de forma sencilla los datos de la base de datos. Para generar un modelo es necesario escribir un JSON válido con los atributos que requiere el modelo asignándoles un valor, en la Figura 19 se presenta cómo se genera el modelo en dicha herramienta para la entidad ‘Modelos de Referencia’.

*Figura 19.**Ejemplo del uso de la herramienta Quicktype*

The screenshot shows the Quicktype application interface. At the top, there is a teal header with the 'quicktype' logo on the left and 'Options' with a question mark icon on the right. Below the header, there is a dark grey bar with a folder icon and a '</> Dart' button. The main area is a dark blue editor showing a JSON object: `{ "idModeloReferencia":1, "suma":100 }`. Below the editor, the generated Dart code is displayed:

```
import 'dart:convert';

ModeloReferenciaModel modeloReferenciaModelFromJson(String str) => ModeloReferenciaModel.f

String modeloReferenciaModelToJson(ModeloReferenciaModel data) => json.encode(data.toJson(

class ModeloReferenciaModel {
  ModeloReferenciaModel({
    this.idModeloReferencia,
    this.suma,
  });

  int idModeloReferencia;
  int suma;

  factory ModeloReferenciaModel.fromJson(Map<String, dynamic> json) => ModeloReferenciaM
    idModeloReferencia: json["idModeloReferencia"],
    suma: json["suma"],
  );
}
```

En la Figura 20 se presenta el modelo generado en la figura anterior para lenguaje Dart implementado en el archivo 'modelo_Referencia_model.dart' en donde se observan dos funciones que convierten un JSON en un modelo y a un modelo en un Map en las líneas 22 y 28.

Figura 20.

Archivo con la implementación de un modelo en lenguaje Dart

```
5 import 'dart:convert';
6
7 ModeloReferenciaModel modeloReferenciaModelFromJson(String str)
8 => ModeloReferenciaModel.fromJson(json.decode(str));
9
10 String modeloReferenciaModelToJson(ModeloReferenciaModel data)
11 => json.encode(data.toJson());
12
13 class ModeloReferenciaModel {
14     ModeloReferenciaModel({
15         this.idModeloReferencia,
16         this.suma,
17     });
18 //atributos del modelo
19 int idModeloReferencia;
20 double suma;
21 //funcion que convierte un JSON en un modelo
22 factory ModeloReferenciaModel.fromJson(Map<String, dynamic> json)
23 => ModeloReferenciaModel(
24     idModeloReferencia: json["idModeloReferencia"],
25     suma: json["suma"].toDouble(),
26 );
27 //funcion que convierte a un modelo en un map
28 Map<String, dynamic> toJson() => {
29     "idModeloReferencia": idModeloReferencia,
30     "suma": suma,
31 };
32 }
```

Para implementar las vistas de usuario siempre se define una clase que extiende de un StatelessWidget o de un StatefulWidget, solo se hace uso del segundo cuando la pantalla necesita redibujarse o cambiar de estado en tiempo real, en la Figura 21 se presenta el ejemplo de la implementación de la vista home, la estructura general que se usó fue un Scaffold (se puede observar en la línea 28) el cual contiene un AppBar, un body y un FloatingActionButton y cada

una de esas partes se pueden definir como un método que retorna un widget para facilitar la comprensión del código, así como cualquier otro widget de menor jerarquía.

Figura 21.

Implementación de la vista de usuario home

```
home_page.dart M X
16 class HomePage extends StatelessWidget {
17   //referencia a la clase que accede al CRUD de las ubicaciones y los
18   final UbicacionesOperations _ubiOper = new UbicacionesOperations();
19   final EstadosOperations _estOper = new EstadosOperations();
20   final prefs = new PreferenciasUsuario();
21   @override
22   Widget build(BuildContext context) {
23     prefs.ultimaPagina = 'taps';
24     final cosData = Provider.of<CostosData>(context);
25     final List<CultivoModel> cultivos = cosData.cultivos;
26     cosData.obtenerCostosByConceptos();
27     //widget padre de la pagina home
28     return Scaffold(
29       appBar: AppBar( // AppBar ...
44       body: _crearListaDeCultivo(context, cultivos),
45       floatingActionButton: _agregarCultivo(context),
46     ); // Scaffold
47   }
```

En la Figura 22 se presenta como ejemplo la implementación del widget del body de la vista home en la cual se puede observar a partir de la línea 52 que se generaron unas cards o tarjetas según la cantidad de cultivos existentes, en estas se observa la información general de los cultivos.

Figura 22.

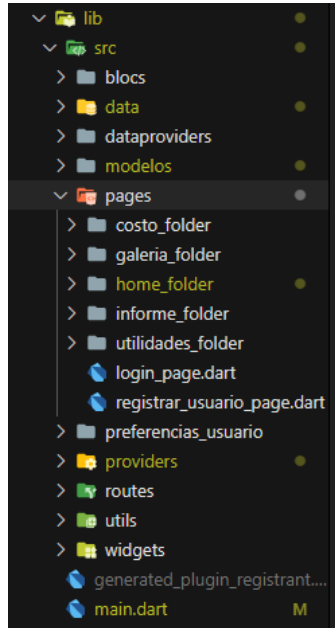
Implementación del body para la vista home

```
home_page.dart M X
48 | //body de la vista de usuario home
49 |   Widget _crearListaDeCultivo(
50 |     BuildContext context, List<CultivoModel> cultivos) {
51 |
52 |     return ListView.builder(
53 |       padding: EdgeInsets.all(5.0),
54 |       itemCount: cultivos.length,
55 |       itemBuilder: (context, index) {
56 |         return _crearCards(
57 |           context,
58 |           cultivos[index],
59 |         );
60 |       },
61 |     ); // ListView.builder
62 |   }
63 | }
```

En la Figura 23 se presenta la jerarquía de carpetas utilizadas en la implementación. En la carpeta llamada 'lib' se puede observar que se encuentra el archivo 'main.dart' y la carpeta 'src', dentro de esta se encuentran una serie de carpetas que se generaron con el objetivo de organizar los archivos del proyecto. Las carpetas con más archivos son: 'dataproviders', 'modelos', 'data', 'providers', 'widgets' y 'pages', en esta última se observa que también se tuvieron que organizar en seis carpetas más 'costo_folder', 'galeria_folder', 'home_folder', 'informe_folder', 'utilidades_folder' y 'preferencias_folder'.

Figura 23.

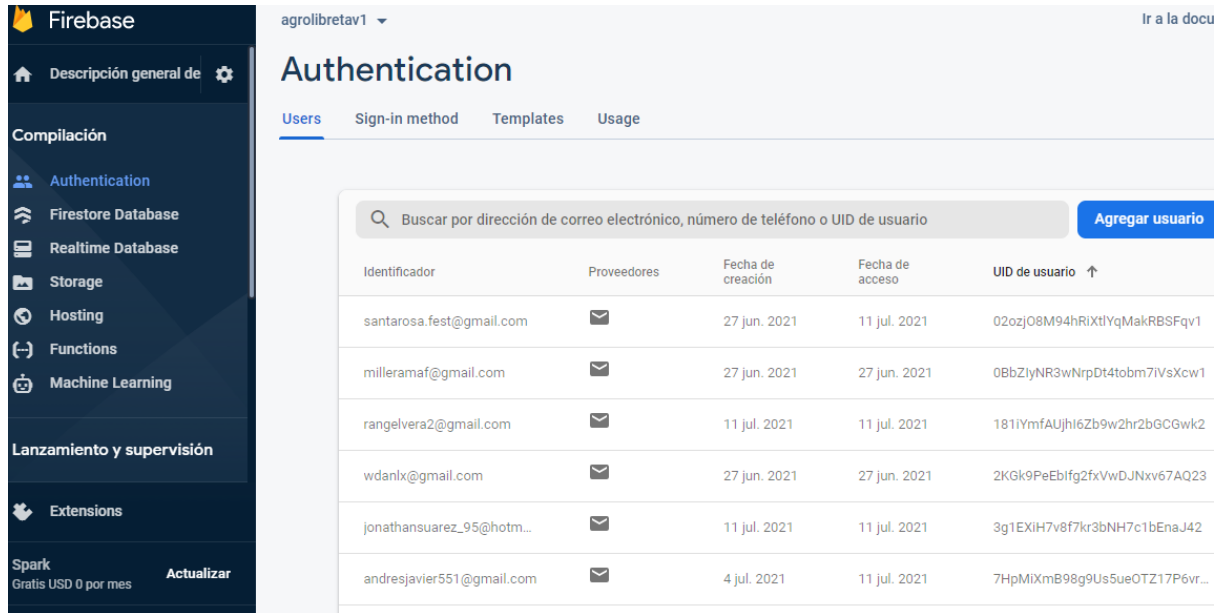
Jerarquía de carpetas de la implementación



La Figura 24 evidencia el uso de Firebase Authentication para almacenar los datos de ingreso de los usuarios y validar la autenticación ya sea en la aplicación o en el sitio web.

Figura 24.

Firestore Authentication usado en AgroLibreta



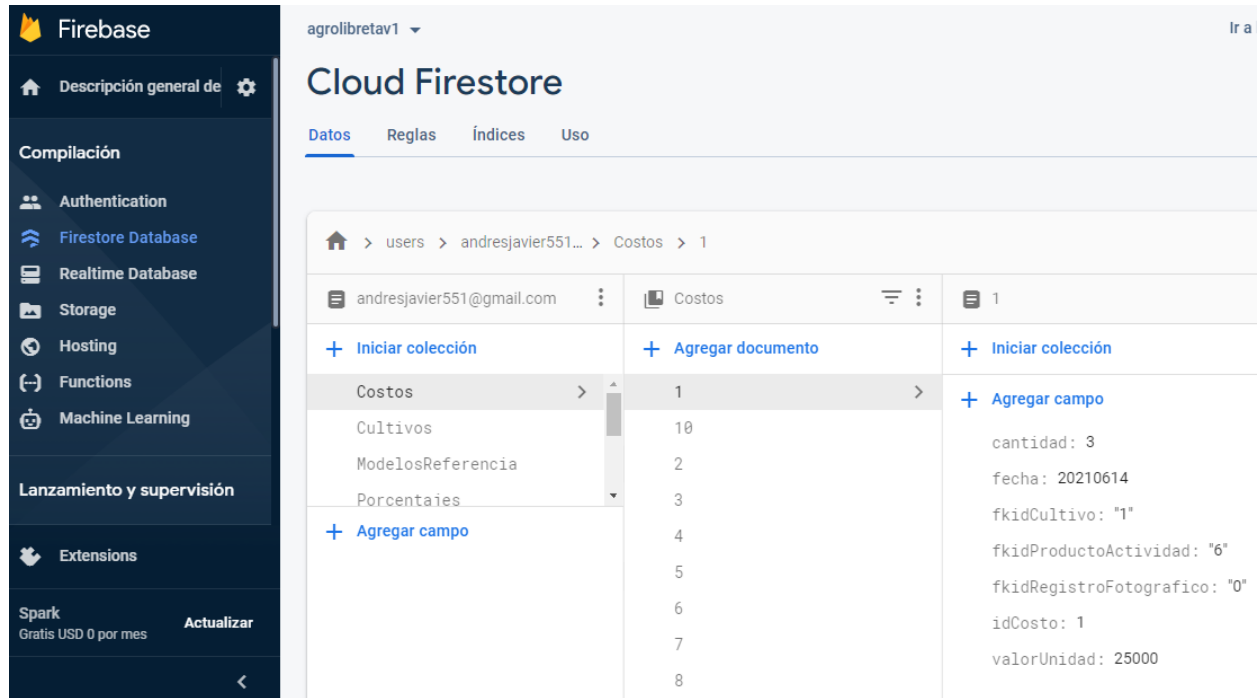
The screenshot displays the Firebase Authentication console for the project 'agrolibretav1'. The left sidebar shows the navigation menu with options like 'Compilación', 'Lanzamiento y supervisión', and 'Extensions'. The main content area is titled 'Authentication' and includes a search bar and a table of users.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario ↑
santarosa.fest@gmail.com	✉	27 jun. 2021	11 jul. 2021	02ozjO8M94hRiXtIYqMakRBSFqv1
milleramaf@gmail.com	✉	27 jun. 2021	27 jun. 2021	0BbZlyNR3wNrpDt4tobm7iVsXcw1
rangelvera2@gmail.com	✉	11 jul. 2021	11 jul. 2021	181YmfAUjh6Zb9w2hr2bGCGwk2
wdanlx@gmail.com	✉	27 jun. 2021	27 jun. 2021	2KGk9PeEblfg2fxVwDJNxxv67AQ23
jonathansuarez_95@hotmail.com	✉	11 jul. 2021	11 jul. 2021	3g1EXIH7v8f7kr3bNH7c1bEaJ42
andresjavier551@gmail.com	✉	4 jul. 2021	11 jul. 2021	7HpMIXmB98g9Us5ueOTZ17P6vr...

En la Figura 25 se evidencia el uso de Cloud Firestore para almacenar los datos de las copias de seguridad de los usuarios que se sincronizan en la base de datos remota.

Figura 25.

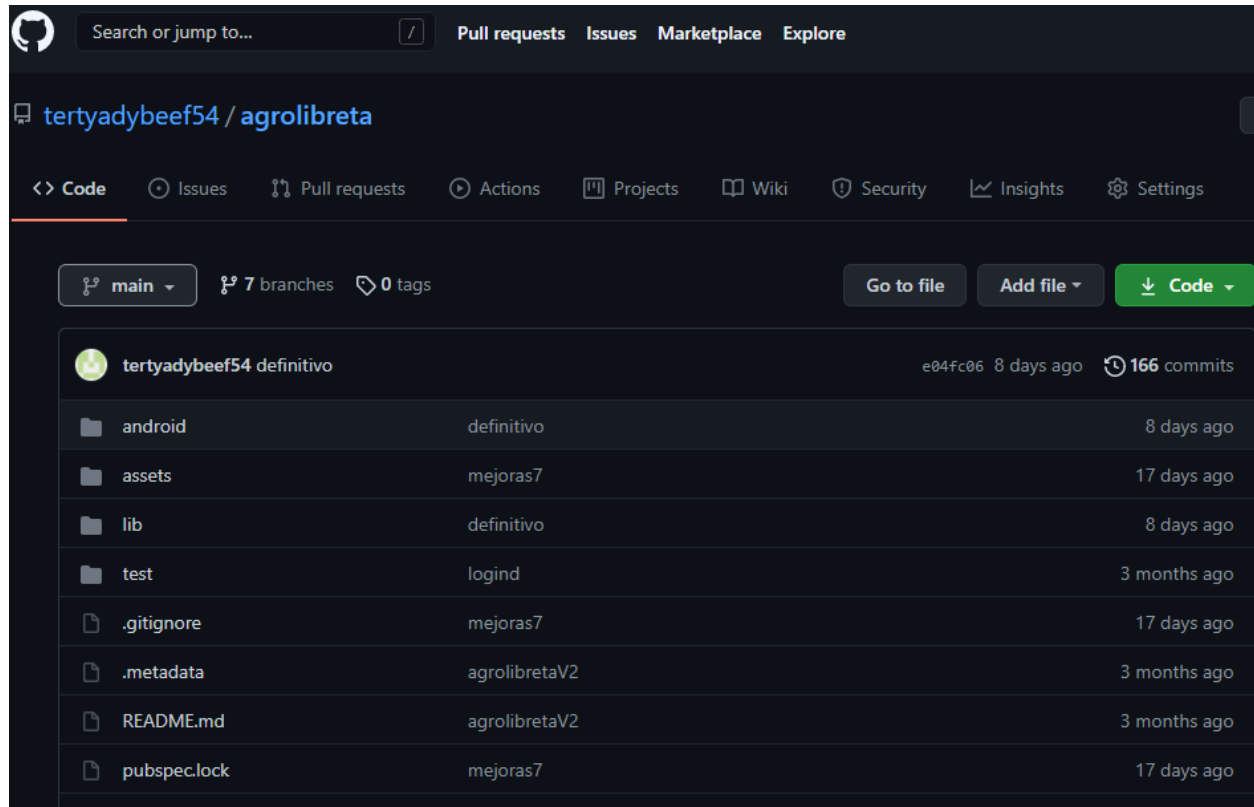
Consola del servicio Cloud Firestore para AgroLibreta



En la Figura 26 se presenta el repositorio para el control de versiones de la aplicación utilizando GitHub, se puede evidenciar que se realizaron 166 commits hasta ese momento.

Figura 26.

Repositorio de la aplicación en GitHub



3.1.8 Vistas de usuario propuestas

Las vistas propuestas para la aplicación móvil son presentadas en el *Apéndice D*.

3.1.9 Vistas de usuario definitivas

Las vistas del prototipo final para la aplicación móvil son presentadas en el *Apéndice E*.

3.2 Sitio web

El segundo objetivo específico define el desarrollo del sitio web y en este numeral se describe en detalle el proceso para ello, un video que presenta el resultado poder ser accedido en

la introducción de este capítulo en la segunda URL presentada correspondiente al numeral 3.2.

Se puede ingresar directamente al sitio web en la siguiente URL: <https://agrolibretav1.web.app/>.

Dicho proceso es presentado a continuación:

3.2.1 *Requerimientos*

En la Tabla 4 se listan los requerimientos funcionales y en la Tabla 5 se listan los requerimientos no funcionales del sitio web. Para ver las especificaciones de cada requerimiento revisar en la sección de *Apéndice A la Tabla 9* y la Tabla 10.

Tabla 4.

Requerimientos funcionales del sitio web

IDENTIFICADOR	NOMBRE
RF01	Iniciar sesión
RF02	Cerrar sesión
RF03	Presentar fecha de la última copia de seguridad
RF04	Presentar instructivo de la aplicación móvil
RF05	Permitir la descarga de la aplicación móvil

Tabla 5.

Requerimientos no funcionales del sitio web

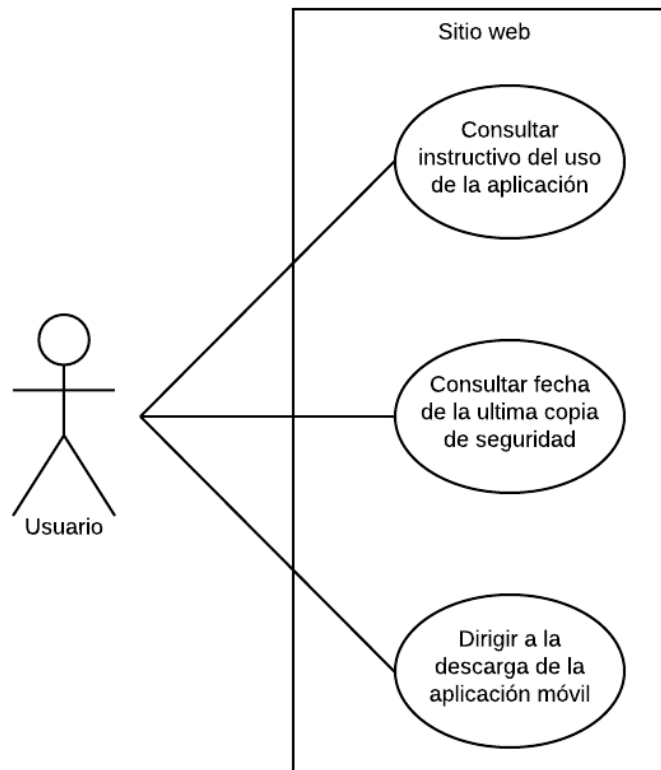
IDENTIFICADOR	NOMBRE
RNF01	Diseñar una interfaz gráfica de usuario
RNF02	Facilidad de mantenimiento y actualización

3.2.2 Diagrama de casos de uso

Según los requerimientos funcionales mencionados se definió el diagrama de casos de uso donde se muestran en la Figura 27 los más relevantes para el usuario, por motivo de sencillez se omitió el caso de uso ‘Iniciar sesión’ que va enlazado al caso de uso consultar la fecha de la última copia de seguridad.

Figura 27.

Diagrama de casos de uso del sitio web



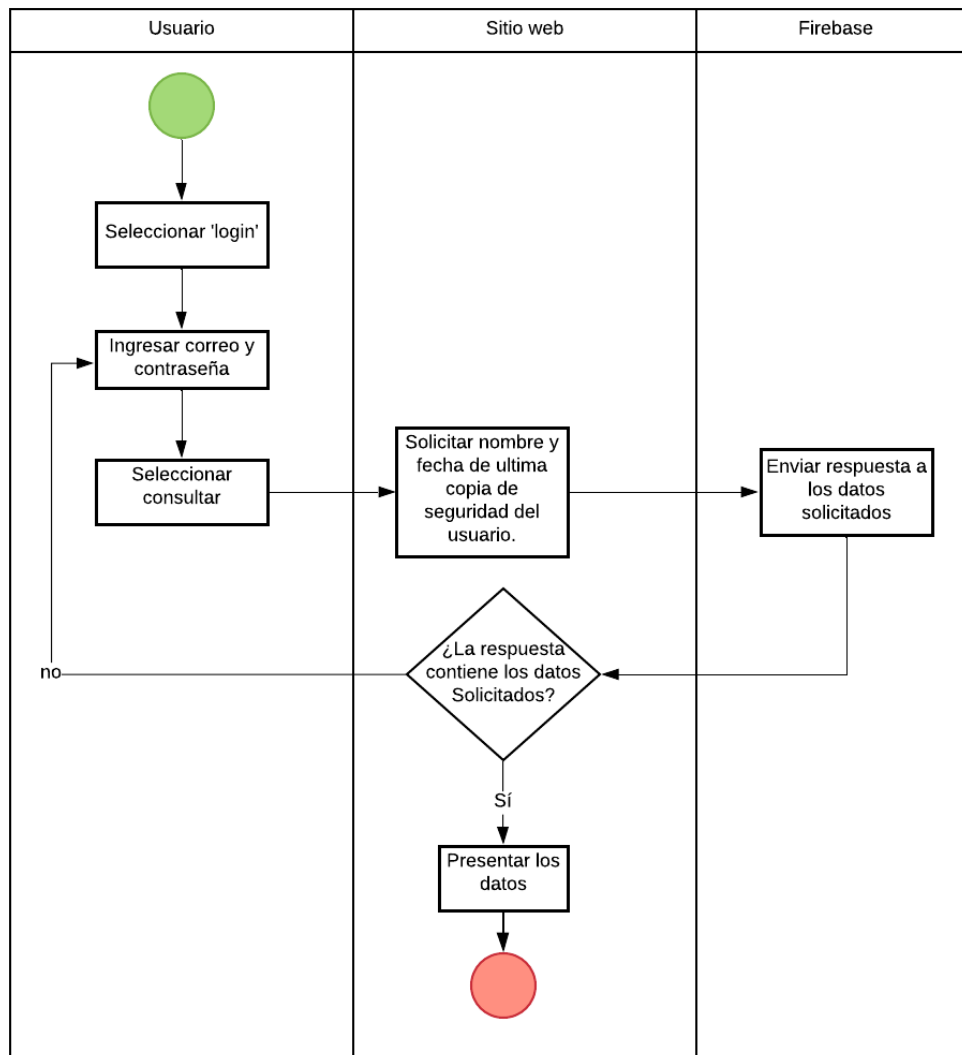
3.2.3 Diagrama de actividades

En este numeral se presenta el diagrama de actividades para el requerimiento ‘Consultar fecha de la copia de seguridad’ mostrado en la

Figura 28, solo se realizó diagrama de actividades al requerimiento más complejo.

Figura 28.

Diagrama de actividades para consultar fecha de la copia de seguridad



3.2.4 Diagrama de bases de datos

La base de datos utilizada por el sitio web es la misma NoSQL que utiliza una estructura de colecciones y documentos también descrita en el numeral 3.1.5 y en la Figura 10 (se utiliza la

misma para el sitio web y para la aplicación móvil), Firebase Hosting accede y consulta en Cloud Firestore para obtener la información.

3.2.5 *Diagrama de arquitectura*

El diagrama de arquitectura se encuentra en la Figura 11 descrito en el numeral 3.1.6 contempla tanto al sitio web como a la aplicación móvil.

3.2.6 *Implementación*

La implementación del sitio web fue desarrollada en el editor de código Brackets y Visual Studio Code con ayuda de HTML5, CSS y JavaScript, este consta de cinco páginas ‘index’, ‘instrucciones’, ‘descargas’, ‘login’ y ‘usuario’ por motivos de optimización sólo se evidencian los archivos correspondientes a la página ‘index’.

La estructura de las páginas y el contenido de ellas fueron desarrollado con HTML5, la Figura 29 corresponde al contenido de la página de inicio.

Figura 29.

Archivo HTML index del sitio web



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head> ... </head>
11 <body>
12 <header> ... </header>
25
26 <div class="slider">
27 <ul> ... </ul>
33 </div>
34 <main>
35 <div class="titulo">
36 <h3>Aplicación móvil orientada al mejoramiento de la valoración del
    producto agrícola y la toma de decisiones en la comercialización.</h3>
37 </div>
38 <div class="contenido1">
39 <div class="columna1">
40 <h2>¿Que es AgroLibreta?</h2>
41 <iframe width="560" height="315"
    src="https://www.youtube.com/embed/o0RUVGA8-No" title="YouTube video
    player" frameborder="0" allow="accelerometer; autoplay; clipboard-
    write; encrypted-media; gyroscope; picture-in-picture"
    allowfullscreen></iframe>
42 <br>
```

Las hojas de estilo CSS como la mostrada en la Figura 30 permitieron definir las fuentes, los colores y los formatos de texto, en dicha figura el archivo ‘estilos_1.css’ corresponde a los estilos implementados a la página ‘index’.

Figura 30.

Archivo CSS para el estilo del index

```
← estilos_1.css
1 - body{
2   font-size: 15px;
3   font-family: Verdana, Arial, Helvetica, sans-serif;
4   margin: 0px;
5   background: #F5F5F5;
6   box-sizing: border-box;
7 }
8
9 - .slider{
10  width: 100%;
11  top: 154px;
12  overflow: hidden;
13  margin-top: 140px;
14 }
15 - .slider ul{
16  display: flex;
17  padding: 0px;
18  margin: 0px;
19  width: 400%;
20
21  animation: cambio 20s infinite;
22  animation-direction: alternate;
23 }
24 - .slider li{
25  list-style: none;
26  width: 100%;
27  top: 150px;
28 }
29 - .slider img{
30  width: 100%;
31  height: 450px;
32 }
33
34 - @keyframes cambio{
35   0%{margin-left: 0;}
36   20%{margin-left: 0;}
37 }
```

Línea 66, Columna 15 — 259 líneas

Para conectar el sitio web con Firebase y que este permitiera realizar consultas se implementó un script en JavaScript para acceder a Firebase como se muestra en la Figura 31.

Figura 31.

Script main.js (conexión con Firebase)

```
main.js X
1  const singinForm = document.querySelector('#login-form');
2
3  singinForm.addEventListener('submit', (e)=>{
4    e.preventDefault();
5    const email = document.querySelector('#email').value;
6    const password = document.querySelector('#password').value;
7    auth
8      .signInWithEmailAndPassword(email, password)
9      .then(userCredential=>{
10     })
11  })
12
13  // events
14  // list for auth state changes
15  auth.onAuthStateChanged((user) => {
16    if (user) {
17      console.log("sign in");
18
19      docRef = fs.collection("users")
20        .doc(`${user.email}`)
21        .collection("Usuario")
22        .doc("1");
23
24      docRef.get().then((doc) => {
25        if (doc.exists) {
26          location.href = "usuario.html";
27        } else {
28          console.log("No such document!");
29        }
30      }).catch((error) => {
31        console.log("Error getting document:", error);
32      });
33    }
34  });
```

3.2.7 Vistas de usuario propuestas y definitivas para el sitio web

Para observar las vistas propuestas al plantear el proyecto y las vistas definitivas al desarrollar el sitio web puede ingresar al

Apéndice F.

3.3 Evaluación del funcionamiento de la aplicación móvil y del sitio web

Se describen los tipos de pruebas que se realizaron para evaluar el cumplimiento de los requerimientos y el funcionamiento de la aplicación móvil y del sitio web. Se realizó un plan de pruebas y una encuesta para conocer la experiencia de usuario la cual se puede ver en detalle junto con los resultados en la sección de *Apéndice G* y *Apéndice H*.

En la tercera URL presentada en la introducción de este capítulo se puede acceder a un video que presenta al servicio brindado por Firebase, la evidencia de los usuarios que realizaron las encuestas y algunos registros de ellos en AgroLibreta.

3.3.1 Pruebas unitarias

Las pruebas unitarias se hicieron a medida que se desarrolló cada requerimiento por parte de los desarrolladores, las cuales permitieron encontrar la mayoría de los errores que tenía cada prototipo y se iban resolviendo una vez se identificaban.

3.3.2 Pruebas de integración

Fueron realizadas para verificar el funcionamiento de las partes del sistema ya integradas. La Tabla 6 representa la prueba de validación para registrar costo, en ella se observa que al utilizar un ejemplo se pueden seguir detalladamente los pasos que intervienen en el proceso desde el usuario, la aplicación, los datos a tener en cuenta y lo que el usuario debería revisar, con ello se pudieron identificar errores los cuales se describen al finalizar cada tabla, en donde se puede observar su fecha de corrección. En la Tabla 11, Tabla 12, Tabla 13, Tabla 14, Tabla 15, Tabla 16, Tabla 17 y Tabla 18 encontrará la prueba de validación que se le realizó a los requerimientos más relevantes que comprende el sistema, estas se llevaron a cabo antes de pasar a las pruebas con usuarios. El total de las tablas se pueden encontrar en la sección de *Apéndice G*.

También se realizó una prueba llamada ‘alfa’ llevada a cabo con 10 usuarios de los cuales 8 contestaron la encuesta y se realizó otra prueba con 12 usuarios llamada ‘beta’ de los cuales también 8 contestaron la encuesta, en total fueron 22 usuarios. Es preciso aclarar que la población de usuarios fue diferente en cada prueba, es decir no hubo usuario repetido en estas pruebas.

Tabla 6.

Prueba de validación para registrar costo

#	Usuario	Aplicación	Datos	Revisar
1	-	Mostrar campos a diligenciar	-	-
2	Seleccionar un nombre de producto o actividad	Consultar y permitir seleccionar productos o actividades	Triple 15	-

#	Usuario	Aplicación	Datos	Revisar
3	Ingresar cantidad de unidades o de jornales	-	10	Que no ingrese letras
4	Ingresar valor total en pesos	Divide el valor total ingresado con la cantidad de unidades o jornales	800000	Que no ingrese letras
5	Ingresar fecha de compra	Mostrar el calendario	20-01-2021	-
#	Errores- Descripción		Corregido	Fecha de corrección
1	Ingresar fecha de compra posterior al día presente		si	02/06/2021

3.3.3 Pruebas alfa

Para estas pruebas participó el profesor Urbano Eliécer Gómez Prada y algunos familiares a los cuales se les explicó de forma verbal como utilizar la aplicación y se les mostró con un ejemplo cómo registrar un usuario, un cultivo, un costo, un registro fotográfico y un modelo de referencia. También se les instruyó como consultar costos, ver un informe general de un cultivo y como crear la copia de seguridad. Además, se probó con un profesor del IPRED a quien solo se le presentó la aplicación sin las respectivas explicaciones de cómo utilizarla para conocer su experiencia de usuario.

La ventaja que se tuvo con esta población al acompañarlos ya sea de manera presencial o virtual fue resolverles las dudas con ejemplos en tiempo real. Por medio de estas pruebas se pudo identificar varios errores los cuales fueron posteriormente corregidos.

Con la prueba realizada al profesor del IPRED se pudo llegar a la conclusión de que la experiencia de usuario era poco intuitiva si no se cuenta con las respectivas instrucciones de cómo utilizarla, posteriormente se realizaron tres videos, estos se encuentran descritos en la

introducción a los resultados en el capítulo 3, en ellos se puede encontrar de donde surgió la idea, una introducción a AgroLibreta y un paso a paso de cómo utilizarla.

Uno de los problemas más relevantes identificados está relacionado con el tamaño de fuente de los dispositivos ya que en algunos afectaba considerablemente la experiencia de usuario al impedir ver el texto de manera ordenada en varias partes de la aplicación, además el motorola g30 no permitió el acceso a la cámara desde la aplicación.

Para conocer la experiencia de usuario de esta población que participó se diseñó una encuesta, los resultados obtenidos se pueden observar con detalle en la sección de *Apéndice H*.

3.3.4 Pruebas beta

Para realizar las pruebas beta se tomó como población un grupo de estudiantes pertenecientes al programa Tecnología Agroindustrial del IPRED y pequeños productores de la región. El proceso que se siguió fue el siguiente:

1. Enviar un correo electrónico con indicaciones a seguir para el desarrollo del proceso (puede consultar el contenido del correo en la sección de *Apéndice G*).
2. Dar soporte por medio de WhatsApp y correo electrónico.
3. Enviar una encuesta diseñada a partir de la realizada en las pruebas alfa, pero con varias modificaciones.

Cabe recalcar que para la realización de las pruebas beta ya se contaba con los vídeos mencionados en las pruebas alfa. Los resultados obtenidos de la encuesta se pueden ver con detalle en la sección de *Apéndice H*, también puede encontrar en la sección de *Apéndice I* algunas fotografías tomadas en la visita a los productores.

3.3.5 Pruebas de estrés

La aplicación móvil AgroLibreta está desarrollada para pequeños productores, se tiene pensado que para esta versión los cultivos de un usuario van a ser administrados por una sola persona desde un único dispositivo, si el usuario cambia de móvil debe volver a instalar la aplicación y restaurar la copia de seguridad por lo tanto se mantiene que solo podrá haber un usuario a la vez que ingresará datos a los cultivos. Esta prueba se llevó a cabo con los usuarios que pudieron realizar una sincronización de sus datos a una misma hora para verificar si se presentaba algún inconveniente o error. Los resultados de las pruebas arrojaron que no hay ningún problema con que varios usuarios sincronicen al mismo momento pues no se presentaron errores y se puede observar en la sección de *Apéndice G*.

3.3.6 Pruebas de carga

Para la prueba de carga se simuló un usuario que ingresó más de 2000 registros. Se pudo observar que la navegación entre pantallas no se afectó, pero algunas funcionalidades específicas se vieron afectadas como lo fue la consulta de costos, la sincronización, visibilización del nombre de la ubicación y del estado de los cultivos en el home.

Algunas figuras que evidencian cómo se realizó esta prueba y los resultados que se obtuvieron se pueden consultar en la sección de *Apéndice G*.

4 Conclusiones

En este proyecto se cumplió con el objetivo general y con cada uno de los objetivos específicos planteados, dando como resultado en el primero la aplicación móvil AgroLibreta y para la cual fue implementado cada uno de los ítems que comprendía a manera de requerimientos funcionales y no funcionales, en el segundo se desarrolló el sitio web que brinda soporte al usuario en el uso de la aplicación y en el control de las copias de seguridad, en el tercero se realizó la evaluación del funcionamiento de la aplicación móvil y del sitio web con más de tres usuarios.

El desarrollo de AgroLibreta requirió un modelo de referencia para llevar el control de los costos, modelo que facilita su clasificación de acuerdo a los recursos utilizados en el cultivo asignados a un valor porcentual que permite la comparación con los costos reales del cultivo. Durante el desarrollo se identificó una falta de conocimiento por parte de los usuarios, con respecto a la forma de obtener el valor porcentual estimado, falencia que lleva a que hagan poco uso de un modelo de referencia específico para cada cultivo. El rechazo a usar el modelo es un ejemplo del arraigo cultural que este proyecto intenta mitigar, por ende, AgroLibreta permite generar un modelo de referencia básico para empezar el uso de la aplicación.

El desarrollo de AgroLibreta permitió aprender el lenguaje Dart y el framework Flutter, el cual se hizo en un plazo de alrededor de tres meses, dicho aprendizaje se fue reforzando e implementando a medida que se desarrollaba la aplicación móvil. El proyecto requirió más tiempo del presupuestado, pero se pudo implementar según el diseño y la metodología planteados.

Flutter cuenta con herramientas documentadas y apoyadas en una comunidad que facilita el desarrollo de las funcionalidades con ayuda de las librerías disponibles junto con la suite de servicios de Firebase que permitió el alojamiento del sitio web y por medio de dichas librerías las funcionalidades de login y copia de seguridad de la aplicación se pudieron desarrollar de una manera más ágil y limpia.

Según las pruebas realizadas con los estudiantes de Tecnología Agroindustrial del IPRED y con los pequeños productores de la región de Mutiscua municipio de Norte de Santander, es posible afirmar que AgroLibreta es una herramienta para llevar el control de los costos en los cultivos. Se espera contar con recursos para aumentar la cantidad de usuarios y así haya una adopción de la herramienta que mejore la información en los procesos financieros de los pequeños productores y por ende una valoración más apropiada de sus productos.

5 Recomendaciones

Estas son recomendaciones que los usuarios sugirieron en la encuesta realizada y algunas que se identificaron para que AgroLibreta tenga un mejor desempeño en futuras versiones. Se clasifican en tres categorías, una para mejorar la experiencia de usuario, otra para añadir nuevas funcionalidades y por último para nuevas versiones.

Las siguientes son recomendaciones identificadas para mejorar la experiencia de usuario:

1. Agregar un paso a paso a modo de instructivo cuando el usuario ingresa por primera vez a cada funcionalidad.

2. Cambiar la palabra plaguicidas por manejo integrado de plagas (MIT).
3. Implementar que al momento de registrar un cultivo tanto el modelo de referencia como la ubicación, se debería seleccionar automáticamente un valor por defecto y permitirle cambiarlos.
4. Implementar una advertencia cuando el usuario decida eliminar un modelo de referencia ya que su eliminación provocará que los cultivos que lo usan no generen un informe y se presenten fallos.
5. Implementar que al momento de registrar un costo lo primero que se seleccione sea el concepto y este lo lleve al formulario de registro que permite seleccionar productos o actividades que pertenecen únicamente a ese concepto.
6. Asignar un número que represente a cada concepto en la gráfica de barras para mejorar la visibilidad en el informe.
7. Mostrar en negrilla los costos asociados a un registro fotográfico en el informe del PDF.
8. Optimizar la visualización de los registros fotográficos en la galería.
9. Permitir el registro de varios usuarios desde un mismo dispositivo sin necesidad de reiniciar los datos de la aplicación.

Las siguientes son recomendaciones identificadas para futuras funcionalidades:

1. Permitir al usuario editar modelos de referencia, productos o actividades y ubicaciones previamente generados.
2. Implementar la funcionalidad de notificaciones cuando el usuario tenga registros sin sincronizar y conexión a internet.

3. Habilitar la confirmación del registro de usuario por medio de correo electrónico para que el usuario se asegure de que tiene acceso a ese correo.
4. Desarrollar la funcionalidad para restaurar la contraseña desde la aplicación móvil.

Las siguientes son recomendaciones identificadas para evitar posibles errores y para el desarrollo de nuevas versiones que dependerá de la escalabilidad de la aplicación, principalmente en la cantidad de usuarios:

1. Hacer obligatorio ingresar un presupuesto estimado ya que no registrarlo evita el buen funcionamiento de la aplicación.
2. Que el tamaño de fuente se ajuste de tal forma que en la mayoría de los dispositivos se pueda ver correctamente el contenido.
3. Migrar la base de datos remota hacia un servidor local para no depender del servicio brindado por Firebase.
4. Permitir el respaldo de los registros fotográficos al momento de crear la copia de seguridad.

Referencias Bibliográficas

- AgroWin. (2021). *AgroWin sistema de gestión total para el agro*. Obtenido de <http://www.agrowin.com/>
- APD, R. (2021). *Metodología Kanban*. Obtenido de <https://www.apd.es/metodologia-kanban/>
- Arévalo Ascanio, J. G., Bayona Trillos, R. A., & Rico Bautista, D. W. (2015). El problema de la brecha tecnológica: Un asunto de cultura. *Sinapsis*.
- Brackets.io. (s.f.). *brackets.io*. Obtenido de: <http://brackets.io/>
- Condor, E., & Soria, I. (2014). *Programación web con CSS, JavaScript, PHP y AJAX*.
- Dange, L. (2019). *Flutter for cross-plataform App and SDK development*. Obtenido de: <https://tinyurl.com/yybxlt24>
- Dart. (s.f.). *Paint your UI to life*. Obtenido de: <https://dart.dev/overview>
- Dash-overflow.net. (2021). *Provider 5.0.0*. Obtenido de <https://pub.dev/packages/provider>
- Departamento Administrativo Nacional de Estadística DANE. (2015). *Boletín mensual Insumos y Factores Asociados a la Producción Agropecuaria*.
- Developers. (2021). *Cómo crear y administrar dispositivos virtuales*. Recuperado de: <https://developer.android.com/studio/run/managing-avds?hl=es-419>
- Diego, P. J. (2019). *¿Qué es HTML5?* Obtenido de <https://openwebinars.net/blog/que-es-html5/>
- Digital 55. (2020). *Firestore: qué es, para qué sirve, funcionalidades y ventajas*. Obtenido de: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- Emprendepyme.net. (2017). *emprendepyme.net*. Obtenido de: <https://www.emprendepyme.net/estructura-de-costes.html>

- Firestore. (s.f.). *Acceso multiplataforma simple y gratuito*. Obtenido de <https://firebase.google.com/products/auth?hl=es>
- Firestore. (s.f.). *Almacena y sincroniza los datos de tu app a escala global*. Obtenido de <https://firebase.google.com/products/firestore?hl=es>
- Firestore. (s.f.). *firebase .google.com*. Obtenido de <https://firebase.google.com/docs/hosting>
- Firestore. (s.f.). *Firestore te ayuda a construir y ejecutar aplicaciones exitosas*. Obtenido de <https://firebase.google.com/>
- Flaticon. (s.f.). *Flaticon.es*. Obtenido de <https://www.flaticon.es/>
- Flórez Rodríguez, G. (2020). (D. K. Rangel Florez, & A. J. Cuadros Sanabria, Entrevistadores)
- Flórez, J. G. (2004). *Sistema de costeo: La asignación del costo total a productos y servicios*.
- GitHub. (2012). *Guía sobre Git GitHub y metodología de Desarrollo Software*. Obtenido de <https://tinyurl.com/y3uwo2f6>
- Herrera, F. (2019). *Flutter: Provider, una alternativa al BLoC*. Obtenido de <https://www.youtube.com/watch?v=-KX2rH0qdKA>
- Json.org. (s.f.). *Introducción a JSON*. Obtenido de <https://www.json.org/json-es.html>
- Laudon, K. C., & Laudon, J. P. (2016). *Sistemas de Información Gerencial*.
- Microsoft. (2021). *Visual Studio Code*. Obtenido de: <https://code.visualstudio.com/docs>
- Mozilla developer network . (s.f.). *JavaScript*. Recuperado de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Peñaranda Cáceres, G., & Molina Gelves, D. Y. (2011). La producción de arveja (*Pisum Sativum*) en la vereda Monteadentro, Provincia de Pamplona, Norte de Santander. *Face*.
- Perfetti, J. J., Hernández, A., Leibovich, J., & Balcázar, Á. (2013). *Políticas para el desarrollo de la agricultura en Colombia*.

- Perry, S. (2013). Agricultura familiar y seguridad alimentaria en los países andinos. *Grupo Diálogo Rural, Conocimiento y Cambio en Pobreza Rural y Desarrollo*.
- Pub.dev. (2021). *Sqflite 2.0.0*. Obtenido de: <https://pub.dev/packages/sqflite>
- Quicktype. (2021). *Quicktype*. Obtenido de <https://quicktype.io/>
- Reboul, D., & Escobar, G. (2019). Determinantes de la diversificación de las estrategias de vida de la Agricultura Familiar en Colombia. *Eutopía*.
- Sánchez, J. (2020). *Introducción al patrón bloc*. Obtenido de: <http://xurxodev.com/introduccion-al-patron-bloc/>
- Sierra, J. R. (2017). Determinantes geográfico-políticos de la acción violenta guerrillera: un análisis de la concurrencia regional. *Revista Española de Ciencia Política*.
- Sqlite.org. (s.f.). *sqlite.org*. Obtenido de <https://www.sqlite.org/index.html>
- Trello. (s.f.). *Trello.com*. Obtenido de <https://trello.com/es>
- Wadsworth, J. (1997). *Análisis de Sistemas de Producción Animal - Tomo 2: las Herramientas Básicas*.

Apéndices

Apéndice A. Requerimientos del sistema.

En este apéndice se presentan las tablas con las respectivas especificaciones de los requerimientos funcionales y no funcionales de la aplicación móvil y del sitio web.

Tabla 7. Descripción requerimientos funcionales de la aplicación móvil

Identificador	Nombre	Prioridad	Descripción
RF00	Iniciar sesión	Alta	Los usuarios deben iniciar sesión para poder ingresar, usando su correo y contraseña establecidos en el registro.
RF01	Cerrar sesión	Media	Los usuarios pueden cerrar sesión o mantenerla abierta cuando lo deseen.
RF02	Registrar y editar cultivo	Alta	Al crear un cultivo el usuario puede registrar los siguientes datos: nombre del cultivo, ubicación, área, fecha de inicio de actividades y presupuesto, así mismo después de creado puede editarlo.
RF03	Listar cultivos	Media	Los usuarios pueden ver la lista de cultivos creados, a la vez que datos distintivos del mismo de tal forma que le permita diferenciarlos.
RF04	Registrar un modelo de referencia	Alta	El usuario puede distribuir los valores de los porcentajes que definen el modelo de referencia de tal forma que se adecúe a sus preferencias creando así su propio modelo.

Identificador	Nombre	Prioridad	Descripción
RF05	Crear un modelo de referencia por defecto	Alta	El sistema genera de forma automática un modelo de referencia para que sea usado por el usuario en los cultivos que él desee.
RF06	Registrar, editar y eliminar costo	Alta	El usuario podrá registrar costos para cada cultivo con los siguientes datos: tipo de costo, cantidad, fecha, unidad de medida y valor de la unidad, después de registrar el costo puede eliminarlo o editarlo.
RF07	Consultar costos con filtros	Media	El usuario puede ver todos los costos en cualquier momento y filtrarlos por, cultivo, fecha, producto o actividad y por concepto (los conceptos son semilla, fertilizantes, plaguicidas, transporte, etc.)
RF08	Registrar y eliminar una fotografía de un evento comercial	Alta	El usuario puede acceder a la cámara del dispositivo para tomar fotografías que respalden los costos, después de esto podrá eliminarla.
RF09	Asociar costos a registros fotográficos	Alta	El usuario puede asociar los costos registrados a los registros fotográficos guardados.
RF10	Ver costos asociados a registros fotográficos	Media	El usuario puede ver cuales costos están asociados a un registro fotográfico y cuáles no.
RF11	Ver informe general del cultivo	Alta	El usuario puede ver un informe del cultivo que desee donde se muestran los costos, los datos y gráficas relacionadas con este.
RF12	Registrar producto o actividad	Alta	El usuario puede parametrizar productos o actividades con los siguientes datos: unidad nombre y concepto.

Identificador	Nombre	Prioridad	Descripción
RF13	Ver gráfica comparativa con el modelo de referencia	Media	El usuario puede ver por medio de una gráfica en cuáles conceptos del cultivo está sobrepasando los límites indicados por el modelo de referencia.
RF14	Generar un PDF del informe general del cultivo	Media	El usuario podrá descargar un PDF del informe general del cultivo.
RF15	Registrar ubicación	Media	El usuario puede registrar las ubicaciones con los siguientes datos: nombre de la ubicación y descripción.
RF16	Generar precio de venta sugerido	Alta	El usuario puede ingresar la cantidad de unidades y el porcentaje de ganancia para generar un precio de venta sugerido.
RF17	Visualizar el perfil de usuario y editarlo	Media	El usuario puede ver sus datos personales registrados y también puede editarlos cuando lo desee.
RF18	Cambiar estado del cultivo	Media	El usuario puede cambiar el estado de sus cultivos en cualquier momento, estos son: activo, inactivo o perdido.
RF19	Crear y restaurar copia de seguridad	Alta	El usuario puede realizar copias de seguridad que serán almacenadas en Firebase y cuando sea necesario restaurarla.

Tabla 8. Descripción requerimientos no funcionales de la aplicación móvil

Identificador	Nombre	Prioridad	Descripción
RNF00	Diseñar una interfaz gráfica de usuario	Alta	Crear una interfaz gráfica agradable y de fácil navegación que le permita al usuario interactuar con la aplicación.

Identificador	Nombre	Prioridad	Descripción
RNF01	Facilidad de mantenimiento y actualización	Alta	El código debe estar organizado y documentado de tal forma que no haya dificultades y sea rápido a la hora de implementar mejoras.
RNF02	Actualizar la información en tiempo real en la base de datos	Alta	La aplicación deberá responder de forma casi instantánea al momento de consultar información o hacer registros y actualizaciones en la base de datos
RNF03	Seguridad de los datos	Alta	Toda la información suministrada podrá tener un respaldo.
RNF04	Rapidez al momento de generar informes	Media	Se debe generar el informe en formato PDF lo más rápido posible
RNF05	Compatible con versiones de Android 5.0 o superior	Media	La aplicación se podrá usar en cualquier dispositivo con sistema operativo Android con versión 5.0 o superior

Tabla 9. Descripción requerimientos funcionales del sitio web

Identificador	Nombre	Prioridad	Descripción
RF00	Iniciar sesión	Alta	Los usuarios pueden iniciar sesión para poder ingresar, usando su correo y contraseña establecidos en el registro en la aplicación.
RF01	Cerrar sesión	Alta	Los usuarios pueden cerrar sesión una vez finalizada su consulta.

Identificador	Nombre	Prioridad	Descripción
RF02	Presentar fecha de la última copia de seguridad	Alta	Los usuarios pueden consultar la fecha de la última copia de seguridad para saber desde qué día no han sincronizado los datos.
RF03	Presentar instructivo de la aplicación móvil	Alta	Los usuarios pueden consultar un instructivo con los pasos que pueden seguir para hacer uso de la aplicación móvil.
RF04	Permitir la descarga de la aplicación móvil	Media	Los usuarios pueden dirigirse a un enlace de descarga del cual obtendrá el archivo '.apk' para instalar en sus dispositivos.

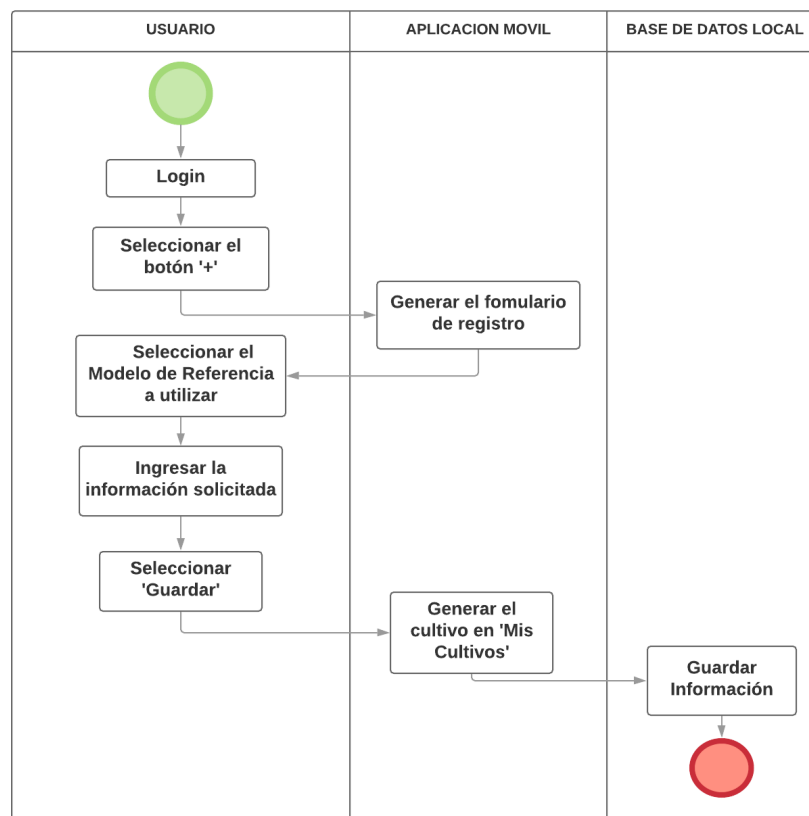
Tabla 10. Descripción requerimientos no funcionales del sitio web

Identificador	Nombre	Prioridad	Descripción
RNF00	Diseñar una interfaz gráfica de usuario.	Alta	Crear una interfaz gráfica que sea intuitiva, agradable y de fácil navegación que le permita al usuario interactuar con el sitio web.
RNF01	Facilidad de mantenimiento y actualización.	Alta	El código debe estar organizado y documentado de tal forma que no haya dificultades y sea rápido a la hora de implementar mejoras.

Apéndice B. Diagramas de actividades.

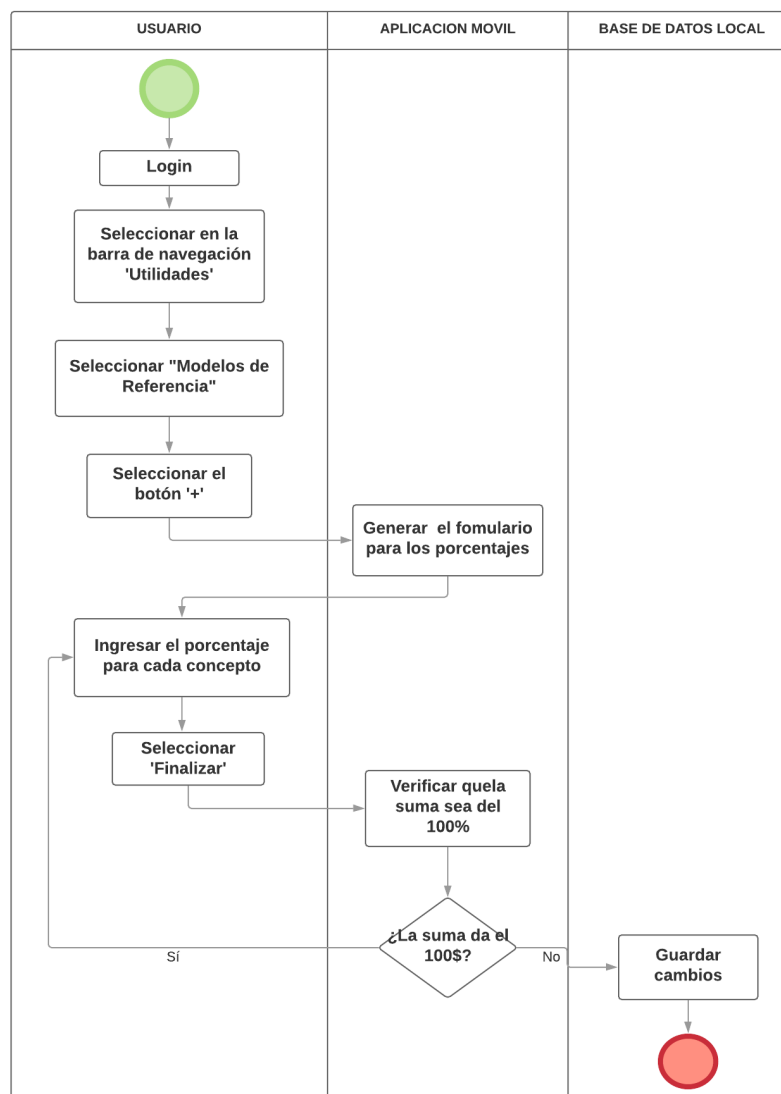
Se presentan los diagramas de actividades de la aplicación móvil para los requerimientos de 'registrar cultivo', 'registrar modelo de referencia' y 'registrar fotografía de un evento comercial'. La Figura 32 representa el diagrama de actividades del requerimiento 'registrar cultivo' el cual es un formulario que el usuario debe diligenciar según el cultivo que desea registrar.

Figura 32. Diagrama de actividades para registrar un cultivo



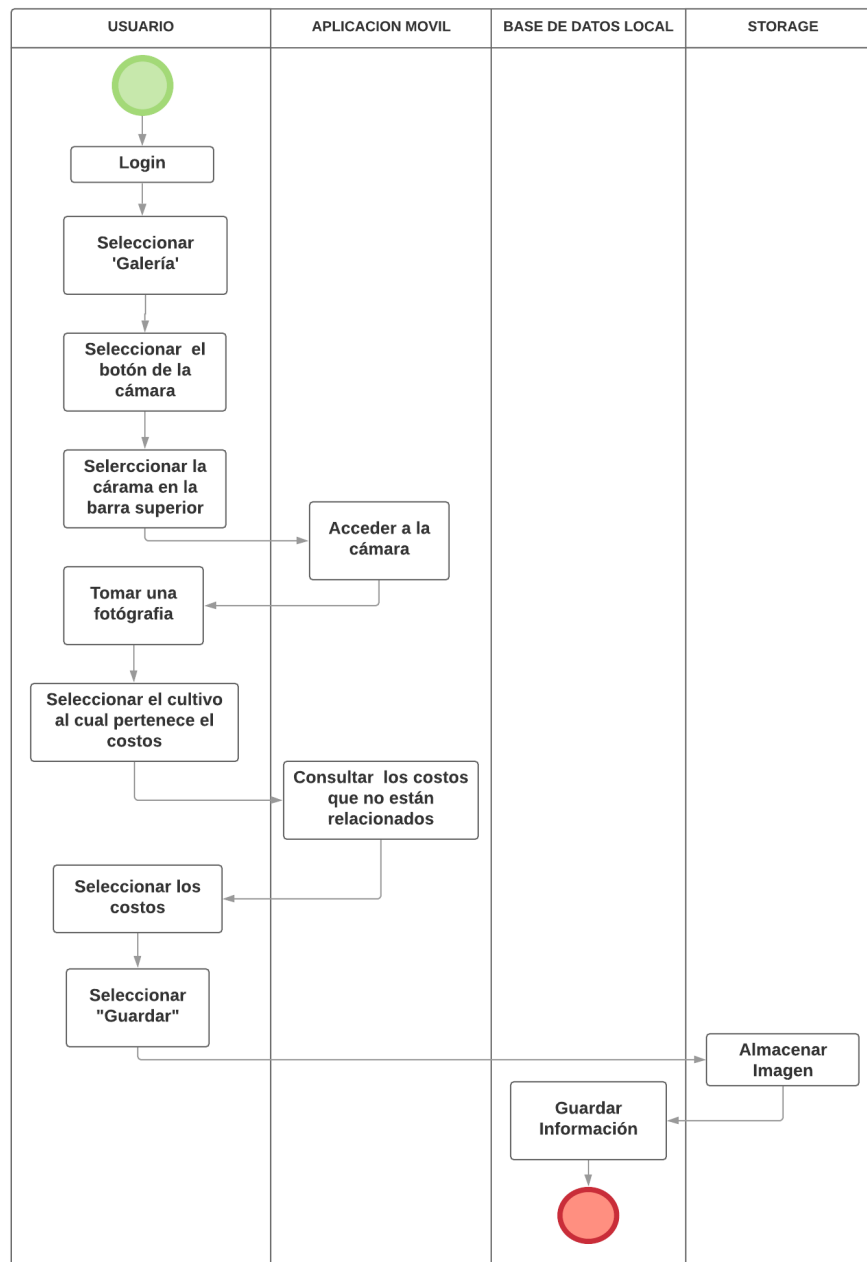
La Figura 33 representa el diagrama de actividades para el requerimiento 'registrar modelo de referencia' el cual es un formulario que el usuario debe diligenciar según los porcentajes que desee para crear un nuevo modelo de referencia que se ajuste a sus necesidades, la suma de estos porcentajes debe ser igual a 100%.

Figura 33. Diagrama de actividades para registrar modelo de referencia



La Figura 34 representa el diagrama de actividades para el requerimiento 'registrar fotografía de un evento comercial en el cual es necesario tomar una fotografía con la cámara del dispositivo para asociarla a los costos que el usuario desee.

Figura 34. Diagrama de actividades para registrar fotografía de un evento comercial



Nota: El Storage corresponde al almacenamiento interno del dispositivo móvil, lo que quiere decir que la imagen primero se guarda en el almacenamiento del móvil y después se guarda la ubicación de esta en la base de datos local.

Apéndice C. Implementación.

En este apéndice se presentan por completo los archivos mostrados en el numeral 3.1.7 para una visión más amplia de lo que fue la implementación. La Figura 35 corresponde a la primera parte del archivo 'main.dart' y en ella se pueden observar todos los providers que se tuvieron que realizar a partir de la línea 36 y también en la línea 25 y 26 la instancia y la inicialización de las preferencias de usuario que permiten mantener la última página vista por el usuario y que la sesión no expire sino hasta que el usuario cierre sesión.

Figura 35. Archivo main (parte 1)

```
23 void main() async {
24   WidgetsFlutterBinding.ensureInitialized();
25   final prefs = new PreferenciasUsuario();
26   await prefs.initPrefs();
27   runApp(MyApp());
28 }
29
30 class MyApp extends StatelessWidget {
31   // This widget is the root of your application.
32   final prefs = new PreferenciasUsuario();
33   @override
34   Widget build(BuildContext context) {
35     return MultiProvider(
36       providers: [
37         ChangeNotifierProvider(create: (context)
38           => new CultivoData()),
39         ChangeNotifierProvider(create: (context)
40           => new ModeloReferenciaData()),
41         ChangeNotifierProvider(create: (context)
42           => new PorcentajeData()),
43         ChangeNotifierProvider(create: (context)
44           => new UbicacionesData()),
45         ChangeNotifierProvider(create: (context)
46           => new UnidadesMedidaData()),
47         ChangeNotifierProvider(create: (context)
48           => new ProductoActividadData()),
49         ChangeNotifierProvider(create: (context)
50           => new CostosData()),
51         ChangeNotifierProvider(create: (context)
52           => new UsuarioProvider()),
53         ChangeNotifierProvider(create: (context)
54           => new RegistrosFotograficosData()),
55         ChangeNotifierProvider(create: (context)
56           => new FiltrosCostosData() ),
57         ChangeNotifierProvider(create: (context)
58           => new PieData() ),
```

La Figura 36 muestra la segunda parte del archivo 'main.dart' en ella se puede observar el widget de 'MaterialApp' que permite incluir más widgets para el diseño de la aplicación, como en las líneas 62 a la 69 las cuales permiten la configuración regional y los diseños para el idioma, también permite incluir el título de aplicación en la línea 71, la especificación de rutas a partir de la líneas 72 a la 76 y el tema del diseño los colores y las fuentes de texto de la aplicación móvil.

Figura 36. Archivo main (parte 2)

```
60 child: MaterialApp(  
61   debugShowCheckedModeBanner: false,  
62   localizationsDelegates: [  
63     GlobalMaterialLocalizations.delegate,  
64     GlobalWidgetsLocalizations.delegate,  
65     GlobalCupertinoLocalizations.delegate,  
66   ],  
67   supportedLocales: [  
68     const Locale('en', ''), // English, no country code  
69     const Locale('es', 'ES'), // *See Advanced Locales below*  
70   ],  
71   title: 'AgroLibreta',  
72   initialRoute: prefs.ultimaPagina,  
73   routes: getApplicationRoutes(),  
74   onGenerateRoute: (RouteSettings settings) {  
75     return MaterialPageRoute(  
76       builder: (BuildContext context) => HomePage  
77         ()),  
78     );  
79   },  
80   theme: ThemeData(  
81     primaryColor: Color(0xff9ccc65),  
82     primaryColorLight: Color(0xffcfff95),  
83     primaryColorDark: Color(0xff6b9b37),  
84     accentColor: Color(0xff8c6d62),  
85     iconTheme: IconThemeData(color: Color(0xff1b5e20)),  
86     textButtonTheme: TextButtonThemeData(style: ButtonStyle(  
87       foregroundColor: MaterialStateProperty.all<Color>(Color(0xff1b5e20))),  
88     ),  
89     elevatedButtonTheme: ElevatedButtonThemeData(style: ButtonStyle(  
90       backgroundColor: MaterialStateProperty.all<Color>(Color(0xff8c6d62))),  
91     ),  
92     timePickerTheme: TimePickerThemeData(backgroundColor: Colors.orange),  
93     appBarTheme: AppBarTheme(  
94       iconTheme: IconThemeData(color: Colors.white),  
95       textTheme: TextTheme(headline6: TextStyle(  
96         color: Colors.white,  
97         fontSize: 20.0,  
98         fontWeight: FontWeight.bold  
99       )),  
100   )),  
101 )  
102 );  
103 }  
104
```

En la Figura 37 se pueden observar todas las rutas que se utilizaron en la aplicación móvil AgroLibreta, éstas se incluyeron con una colección 'Map' donde sus elementos son de tipo 'String' y 'WidgetBuilder'.

Figura 37. Archivo routes

```
33  Map<String, WidgetBuilder> getApplicationRoutes() {
34      return <String, WidgetBuilder>{
35          'taps': (BuildContext context) => TapsPage(),
36          'home': (BuildContext context) => HomePage(),
37          'perfilUsuario': (BuildContext context) => PerfilUsuarioPage(),
38          'configCultivo': (BuildContext context) => ConfigCultivoPage(),
39          'infoCultivo': (BuildContext context) => InformacionCultivo(),
40          'login': (BuildContext context) => LoginPage(),
41          'crearCultivo': (BuildContext context) => CrearCultivoPage(),
42          'resumenCostos': (BuildContext context) => ResumencostosPage(),
43          'crearCosto': (BuildContext context) => CrearCostoPage(),
44          'crearModeloReferencia': (BuildContext context) => CrearModeloReferencia(),
45          'costos': (BuildContext context) => CostosPage(),
46          'informe': (BuildContext context) => InformeCultivoPage(),
47          'utilidades': (BuildContext context) => UtilidadesPage(),
48          'galeriaRegistrosFoto': (BuildContext context) =>
49              GaleriaRegistrosFotograficosPage(),
50          'nuevoRegistroFoto': (BuildContext context) =>
51              NuevoRegistroFotograficoPage(),
52          'detalleRegistroFoto': (BuildContext context) =>
53              DetalleRegistroFotograficoPage(),
54          'restaurarPassword': (BuildContext context) => RestaurarPassword(),
55          'verModelo': (BuildContext context) => VerModeloReferencia(),
56          'registrarUsuario' : (BuildContext context) => RegistrarUsuario(),
57          'modeloUtil': (BuildContext context) => ModeloReferencialList(),
58          'productoUtil':(BuildContext context) => ProductoActividadList(),
59          'ubicacionUtil':(BuildContext context) => UbicacionList(),
60          'unidadMedidaUtil': (BuildContext context) => UnidadMedidaList(),
61          'editarRegistro': (BuildContext context) => EditarRegFotPage(),
62          'verCosto': (BuildContext context) => VerCostoPage(),
63          'EditarCosto': (BuildContext context) => EditarCostoPage(),
64
65      };
66  }
```

La Figura 38 corresponde a la primera parte del archivo 'db_provider' en el cual se inicializa la base de datos local y se generan todas las entidades con sus atributos a medida que el usuario vaya realizando registros de sus cultivos.

Figura 38. Archivo `db_provider` (parte 1)

```

7  class DBProvider {
8      static Database _database;
9      static final DBProvider db = DBProvider.();
10     DBProvider.();
11
12     Future<Database> get database async {
13         if (_database != null) return _database;
14
15         _database = await initDB();
16
17         return _database;
18     }
19
20     Future<Database> initDB() async {
21         // Path de donde almacenaremos la base de datos
22         Directory documentsDirectory = await getApplicationDocumentsDirectory();
23         final path = join(documentsDirectory.path, 'Agrolibreta.db');
24         print(path);
25         // Crear base de datos
26         return await openDatabase(path, version: 1, onOpen: (db) {},
27             onCreate: (Database db, int version) async {
28             print('crear base');
29             await db.execute('''
30                 CREATE TABLE Cultivos(
31                     idCultivo INTEGER PRIMARY KEY AUTOINCREMENT,
32                     fkidUbicacion STRING NOT NULL,
33                     fkidEstado STRING NOT NULL,
34                     fkidModeloReferencia STRING NOT NULL,
35                     fkidProductoAgricola STRING NOT NULL,
36                     nombreDistintivo STRING NOT NULL,
37                     areaSembrada INTEGER NOT NULL,
38                     fechaInicio STRING NOT NULL,
39                     fechaFinal STRING ,
40                     presupuesto INTEGER NOT NULL,
41                     precioVentaIdeal REAL ,
42                     FOREIGN KEY (fkidUbicacion) REFERENCES Ubicaciones (idUbicacion),

```

En la Figura 39 se pueden ver las entidades de ‘ModelosReferencia’, ‘Ubicaciones’, ‘EstadosCultivo’ y ‘ProductosAgricolas’ pertenecientes a la base de datos local.

Figura 39. Archivo db_provider (parte 2)

```
42         FOREIGN KEY (fkidUbicacion) REFERENCES Ubicaciones (idUbicacion),
43         FOREIGN KEY (fkidEstado) REFERENCES EstadosCultivo (idEstado),
44         FOREIGN KEY (fkidModeloReferencia) REFERENCES ModelosReferencia (idModeloReferencia),
45         FOREIGN KEY (fkidProductoAgricola) REFERENCES ProductosAgricolas (idProductoAgricola)
46     )
47     ');
48     await db.execute(`
49     CREATE TABLE ModelosReferencia(
50         idModeloReferencia INTEGER PRIMARY KEY,
51         suma REAL NOT NULL
52     )
53     `);
54     await db.execute(`
55     CREATE TABLE Ubicaciones(
56         idUbicacion INTEGER PRIMARY KEY,
57         nombreUbicacion STRING NOT NULL,
58         descripcion STRING NOT NULL,
59         estado INTEGER NOT NULL
60     )
61     `);
62     await db.execute(`
63     CREATE TABLE EstadosCultivo(
64         idEstado INTEGER PRIMARY KEY,
65         nombreEstado STRING NOT NULL
66     )
67     `);
68     await db.execute(`
69     CREATE TABLE ProductosAgricolas(
70         idProductoAgricola INTEGER PRIMARY KEY,
71         nombreProducto STRING NOT NULL
72     )
73     `);
```

En la Figura 40 se puede ver la inicialización de las entidades ‘Conceptos’, ‘UnidadesMedida’, ‘ProductosActividades’, ‘RegistrosFotográficos’ y ‘Costos’ pertenecientes a la base de datos local.

Figura 40. Archivo `db_provider` (parte 3)

```
73     ''');
74     await db.execute('''
75         CREATE TABLE Conceptos(
76             idConcepto INTEGER PRIMARY KEY,
77             nombreConcepto STRING NOT NULL
78         )
79     ''');
80     await db.execute('''
81         CREATE TABLE UnidadesMedida(
82             idUnidadMedida INTEGER PRIMARY KEY,
83             nombreUnidadMedida STRING NOT NULL,
84             descripcion STRING NOT NULL
85         )
86     ''');
87     await db.execute('''
88         CREATE TABLE ProductosActividades(
89             idProductoActividad INTEGER PRIMARY KEY,
90             fkidConcepto STRING NOT NULL,
91             fkidUnidadMedida STRING NOT NULL,
92             nombreProductoActividad STRING NOT NULL,
93             FOREIGN KEY (fkidConcepto) REFERENCES Conceptos (idConcepto),
94             FOREIGN KEY (fkidUnidadMedida) REFERENCES UnidadesMedida (idUnidadMedida)
95         )
96     ''');
97     await db.execute('''
98         CREATE TABLE RegistrosFotograficos(
99             idRegistroFotografico INTEGER PRIMARY KEY,
100            pathFoto TEXT NOT NULL
101        )
102    ''');
103    await db.execute('''
104        CREATE TABLE Costos(
105            idCosto INTEGER PRIMARY KEY,
106            fkidProductoActividad STRING NOT NULL,
107            fkidCultivo STRING NOT NULL,
108            fkidRegistroFotografico STRING NOT NULL,
```

En la Figura 41 se puede observar la inicialización de las instancias ‘Porcentajes’ y ‘Usuario’, también en la línea 141 se realizó una inserción correspondiente a la entidad ‘ModelosReferencia’.

Figura 41. Archivo `db_provider` (parte 4)

```

108     fkidRegistroFotografico STRING NOT NULL,
109     cantidad REAL NOT NULL,
110     valorUnidad INTEGER NOT NULL,
111     fecha INTEGER NOT NULL,
112     FOREIGN KEY (fkidProductoActividad) REFERENCES ProductosActividades (idProductoActividad),
113     FOREIGN KEY (fkidCultivo) REFERENCES Cultivos (idCultivo),
114     FOREIGN KEY (fkidRegistroFotografico) REFERENCES RegistrosFotograficos (idRegistroFotografico)
115   )
116   ');
117   await db.execute('''
118     CREATE TABLE Porcentajes(
119       idPorcentaje INTEGER PRIMARY KEY,
120       fk2idModeloReferencia STRING NOT NULL,
121       fk2idConcepto STRING NOT NULL,
122       porcentaje REAL NOT NULL,
123       FOREIGN KEY (fk2idModeloReferencia) REFERENCES ModelosReferencia (idModeloReferencia),
124       FOREIGN KEY (fk2idConcepto) REFERENCES Conceptos (idConcepto)
125     )
126   ');
127   await db.execute('''
128     CREATE TABLE Usuario(
129       idUsuario INTEGER PRIMARY KEY,
130       documento INTEGER NOT NULL,
131       password STRING NOT NULL,
132       nombres STRING NOT NULL,
133       apellidos STRING NOT NULL,
134       email STRING NOT NULL,
135       fechaNacimiento STRING NOT NULL,
136       fechaUltimaSincro STRING NOT NULL
137     )
138   ');
139   //##### modelo de referencia
140   await db.rawInsert('''
141     INSERT INTO ModelosReferencia(suma) VALUES(100)
142   ');

```

La Figura 42 corresponde a la parte cinco del archivo `db_provider.dart` en la cual se observan los ocho conceptos del modelo de referencia a los cuales el usuario debe asignarles un porcentaje, también se encuentran los estados de un cultivo 'Activo' 'Inactivo' o 'Perdido'.

Figura 42. Archivo *db_provider* (parte 5)

```
143 //##### conceptos
144   await db.rawQuery('''
145     INSERT INTO Conceptos(nombreConcepto) VALUES("Semilla")
146     ''');
147   await db.rawQuery('''
148     INSERT INTO Conceptos(nombreConcepto) VALUES("Abonos y fertilizantes")
149     ''');
150   await db.rawQuery('''
151     INSERT INTO Conceptos(nombreConcepto) VALUES("Plaguicidas")
152     ''');
153   await db.rawQuery('''
154     INSERT INTO Conceptos(nombreConcepto) VALUES("Materiales y Empaques")
155     ''');
156   await db.rawQuery('''
157     INSERT INTO Conceptos(nombreConcepto) VALUES("Maquinaria")
158     ''');
159   await db.rawQuery('''
160     INSERT INTO Conceptos(nombreConcepto) VALUES("Mano de obra")
161     ''');
162   await db.rawQuery('''
163     INSERT INTO Conceptos(nombreConcepto) VALUES("Transporte")
164     ''');
165   await db.rawQuery('''
166     INSERT INTO Conceptos(nombreConcepto) VALUES("Otros")
167     ''');
168 //##### estados del cultivo
169   await db.rawQuery('''
170     INSERT INTO EstadosCultivo(nombreEstado) VALUES("Activo")
171     ''');
172   await db.rawQuery('''
173     INSERT INTO EstadosCultivo(nombreEstado) VALUES("Inactivo")
174     ''');
175   await db.rawQuery('''
176     INSERT INTO EstadosCultivo(nombreEstado) VALUES("Inactivo")
177     ''');
```

La Figura 43 corresponde a la parte 6 del archivo 'db_provider.dart' en la cual se observan las inserciones directas a la base de datos local para la entidad unidades de medida y los porcentajes para el modelo de referencia, esto se aprecia en la aplicación móvil como valores

por defecto, es decir que el usuario no podrá editarlos y le ayudan a realizar sus registros más rápido.

Figura 43. Archivo *db_provider* (parte 6)

```
177     ''');
178     await db.rawQuery('
179     INSERT INTO EstadosCultivo(nombreEstado) VALUES("Perdido")
180     ''');
181     //##### unidades de medida
182     await db.rawQuery('
183     INSERT INTO UnidadesMedida(nombreUnidadMedida, descripcion) VALUES("kg", "Kilogramo")
184     ''');
185     await db.rawQuery('
186     INSERT INTO UnidadesMedida(nombreUnidadMedida, descripcion) VALUES("Bultos", "Bultos")
187     ''');
188     await db.rawQuery('
189     INSERT INTO UnidadesMedida(nombreUnidadMedida, descripcion) VALUES("125 gr", "Papeleta de 125 gr")
190     ''');
191     await db.rawQuery('
192     INSERT INTO UnidadesMedida(nombreUnidadMedida, descripcion) VALUES("Jornal", "8 horas")
193     ''');
194     await db.rawQuery('
195     INSERT INTO UnidadesMedida(nombreUnidadMedida, descripcion) VALUES("Rollo", "Rollo n metros")
196     ''');
197     //##### porcentajes del MR modelo de referencia
198     await db.rawQuery('
199     INSERT INTO Porcentajes(fk2idModeloReferencia, fk2idConcepto, porcentaje) VALUES("1", "1", 3.9)
200     ''');
201     await db.rawQuery('
202     INSERT INTO Porcentajes(fk2idModeloReferencia, fk2idConcepto, porcentaje) VALUES("1", "2", 17.7)
203     ''');
204     await db.rawQuery('
205     INSERT INTO Porcentajes(fk2idModeloReferencia, fk2idConcepto, porcentaje) VALUES("1", "3", 12.1)
206     ''');
207     await db.rawQuery('
208     INSERT INTO Porcentajes(fk2idModeloReferencia, fk2idConcepto, porcentaje) VALUES("1", "4", 6.7)
209     ''');
210     await db.rawQuery('
211     INSERT INTO Porcentajes(fk2idModeloReferencia, fk2idConcepto, porcentaje) VALUES("1", "5", 6.5)
212     ''');
```

En la Figura 44 se observan los valores por defecto que se insertaron en la base de datos local para las entidades ‘ProductosAgrícolas’ y ‘ProductosActividades’.

Figura 44. Archivo `db_provider` (parte 7)

```

222 //##### productos agricolas
223   await db.rawQuery('''
224     INSERT INTO ProductosAgricolas(nombreProducto) VALUES("Arveja")
225   ''');
226 //##### productos actividades
227   await db.rawQuery('''
228     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("1", "1", "Rabo de gallo")
229   ''');
230   await db.rawQuery('''
231     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("2", "2", "Gallinaza")
232   ''');
233   await db.rawQuery('''
234     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("2", "2", "Triple 15")
235   ''');
236   await db.rawQuery('''
237     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("3", "3", "Metarex")
238   ''');
239   await db.rawQuery('''
240     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("3", "1", "Dinate")
241   ''');
242   await db.rawQuery('''
243     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("4", "5", "Alambre")
244   ''');
245   await db.rawQuery('''
246     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("4", "5", "Cabuya")
247   ''');
248   await db.rawQuery('''
249     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("6", "4", "Arar")
250   ''');
251   await db.rawQuery('''
252     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("6", "4", "Tutorar")
253   ''');
254   await db.rawQuery('''
255     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("6", "4", "Fumigar")
256   ''');
257   await db.rawQuery('''
258     INSERT INTO ProductosActividades(fkidConcepto, fkidUnidadMedida, nombreProductoActividad) VALUES("6", "4", "Recolectar")
259   ''');
260
261   print('base creada');
262 });
263 }
264 }

```

La Figura 45 muestra la primera parte del archivo `modelos_referencia_operations.dart`, se puede observar la clase `ModelosReferenciaOperations` en la cual se generan la función para crear o registrar en este caso un modelo de referencia, esta es llamada `nuevoModeloReferencia` se puede observar en la línea 11 y la función para consultar o leer creada en la línea 23 y llamada `consultarModelosReferencia`.

Figura 45. Archivo CRUD modelo_referencia_operations (parte 1)

```
5 class ModelosReferenciaOperations {
6   ModelosReferenciaOperations modelosReferenciaOperations;
7
8   final dbProvider = DBProvider.db;
9
10  //C - crear
11  Future<int> nuevoModeloReferencia(
12    ModeloReferenciaModel nuevoModeloReferencia) async {
13    final db = await dbProvider.database;
14    final res =
15      await db.insert('ModelosReferencia', nuevoModeloReferencia.toJson());
16    // Es el ID del último registro insertado;
17    print('MR');
18    print(res);
19    return res;
20  }
21
22  //R - leer
23  Future<List<ModeloReferenciaModel>> consultarModelosReferencia() async {
24    final db = await dbProvider.database;
25    final res = await db.query('ModelosReferencia');
26
27    return res.isNotEmpty
28      ? res.map((s) => ModeloReferenciaModel.fromJson(s)).toList()
29      : [];
30  }
```

La Figura 46 representa la segunda parte del archivo 'modelo_referencia_operations.dart' se puede observar la función para actualizar llamada 'updateModelosReferencia' en la línea 33, la función para borrar un registro en la línea 42 llamada 'deleteModelosReferencia' y una función para obtener el id del modelo de referencia llamada 'getModeloById' generada en la línea 52.

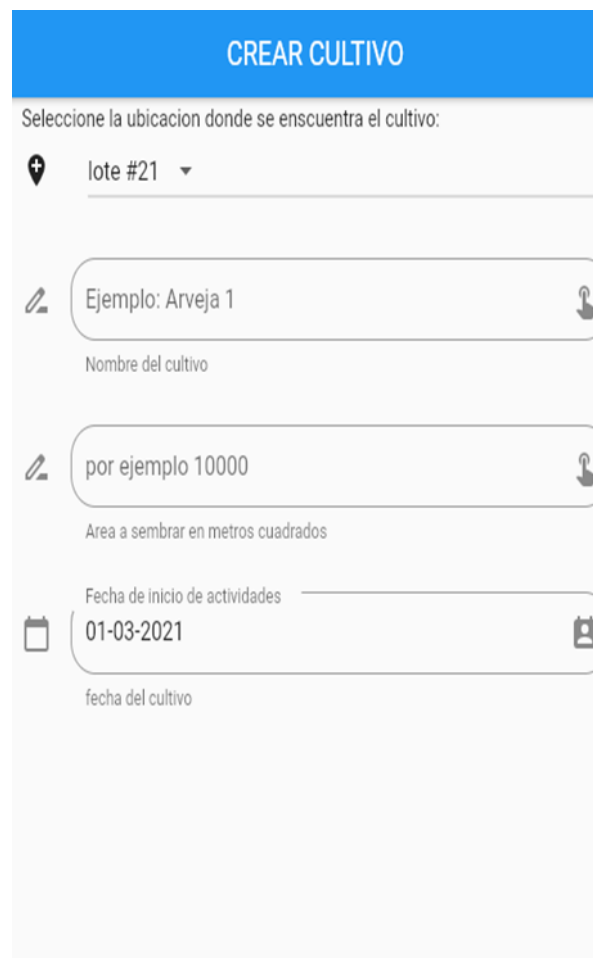
Figura 46. Archivo CRUD modelo_referencia_operations (parte 2)

```
32 //U - actualizar
33 Future<int> updateModelosReferencia(
34     ModeloReferenciaModel nuevoModeloReferencia) async {
35     final db = await dbProvider.database;
36     final res = await db.update(
37         'ModelosReferencia', nuevoModeloReferencia.toJson(),
38         where: 'idModeloReferencia = ?', whereArgs: [nuevoModeloReferencia.idModeloReferencia]);
39     return res;
40 }
41
42 //D - borrar un registro
43 Future<int> deleteModeloReferencia(int id) async {
44     final db = await dbProvider.database;
45     final res = await db.delete('ModelosReferencia',
46         where: 'idModeloReferencia = ?', whereArgs: [id]);
47     print('eliminado');
48     return res;
49 }
50
51 //otros R otras consultas
52 Future<ModeloReferenciaModel> getModeloById(int id) async {
53     final db = await dbProvider.database;
54     final res = await db
55         .query('ModelosReferencia', where: 'idModeloReferencia = ?', whereArgs: [id]);
56
57     return res.isNotEmpty ? ModeloReferenciaModel.fromJson(res.first) : null;
58 }
59 }
```

Apéndice D. Vistas de usuario propuestas para la aplicación móvil.

Se presentan las vistas de usuario realizadas al plantear el proyecto estas permitieron una fácil comprensión de cómo sería la interfaz de usuario y el diseño de esta. Se realizaron por medio de una herramienta web llamada moqups.com. La Figura 47 presenta el formulario para el registro o la creación de un nuevo cultivo, en esta se puede observar que se pregunta por la ubicación y el nombre del cultivo, el área a sembrar y la fecha de inicio de actividades.

Figura 47. Registrar cultivo (vista propuesta)

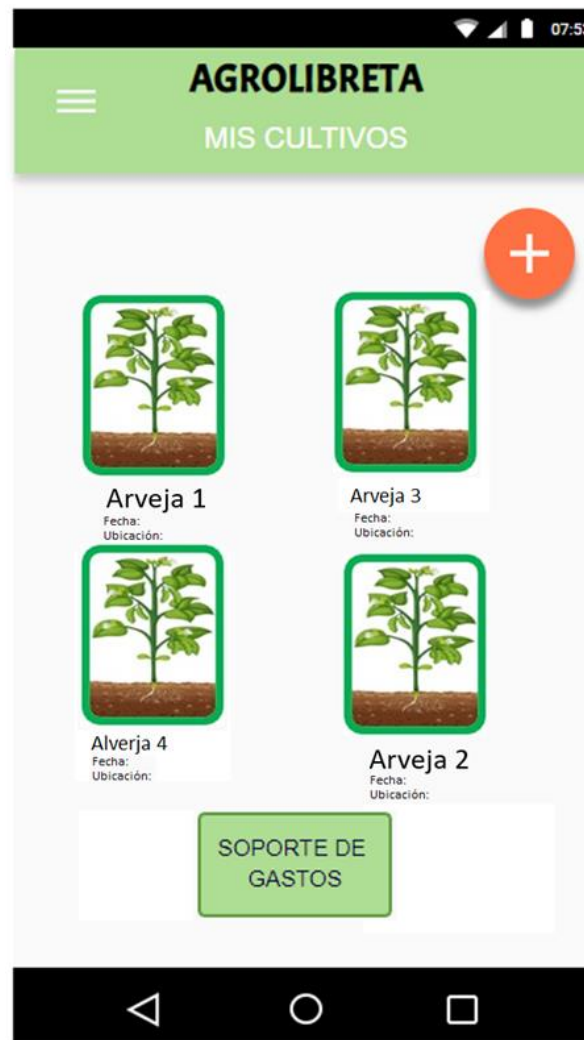


The image shows a mobile application interface for creating a new crop. The title bar is blue and contains the text "CREAR CULTIVO". Below the title bar, there is a prompt: "Seleccione la ubicación donde se encuentra el cultivo:". The form consists of four main input fields, each with a small icon on the left and a small icon on the right:

- The first field has a location pin icon on the left and a dropdown arrow on the right. The text inside is "lote #21".
- The second field has a pencil icon on the left and a hand icon on the right. The text inside is "Ejemplo: Arveja 1". Below this field is the label "Nombre del cultivo".
- The third field has a pencil icon on the left and a hand icon on the right. The text inside is "por ejemplo 10000". Below this field is the label "Area a sembrar en metros cuadrados".
- The fourth field has a calendar icon on the left and a calendar icon on the right. The text inside is "01-03-2021". Below this field is the label "fecha del cultivo".

La Figura 48 representa lo que es para la aplicación AgroLibreta el home, en el cual se listan los cultivos registrados o generados y también sus respectivos botones para consultar costos y para un nuevo cultivo.

Figura 48. Listar cultivos (vista propuesta)



En la Figura 49 se observa cómo se planteó la vista de usuario para la funcionalidad del modelo de referencia por defecto.

Figura 49. Modelo de referencia por defecto (vista propuesta)



Conceptos	Porcentajes
Semilla	3.9%
Ferlizantes	17.7%
Plaguicidas	12.1%
Maquinaria	6.5%
Mano de obra	41.8%
Empaques	6.7%
Transporte	5.6%
Otros	5.7%
TOTAL	100%

En la Figura 50 se observa la vista de usuario que se realizó para la funcionalidad que le permite al usuario saber cómo van sus costos con respecto al presupuesto planeado, esto es según el modelo de referencia que tenga asociado el cultivo.

Figura 50. Resumen de costos (vista propuesta)



La Figura 51 representa la vista de usuario que se planteó para la funcionalidad de registrar costos, aquí se observa que el usuario debe diligenciar todos los campos del formulario y al finalizar debe seleccionar el botón para guardar.

Figura 51. Registro de costos (vista propuesta)

The screenshot displays the 'NUEVO COSTO' (New Cost) screen of an application. At the top, there is a blue header with the text 'NUEVO COSTO'. Below this, a green bar contains the text 'Arveja_1'. The form consists of several fields:

- Concepto:** A dropdown menu currently showing 'Abono y Fertilizantes'.
- Cantidad:** A text input field with a dollar sign icon on the left and a placeholder 'ingrese un número'. Below the field is the label 'cantidad'.
- Producto o Actividad:** A dropdown menu currently showing 'gallinaza'.
- Unidad:** A dropdown menu currently showing 'kg'.
- Valor de la unidad:** A text input field with a dollar sign icon on the left and a placeholder 'ingrese un número'. Below the field is the label 'valor de la unidad'.
- Fecha en que se efectuó el ...:** A date selection field with a calendar icon on the left and a placeholder 'Fecha en que se efectuó el ...'. Below the field is the label 'fecha del gasto'.

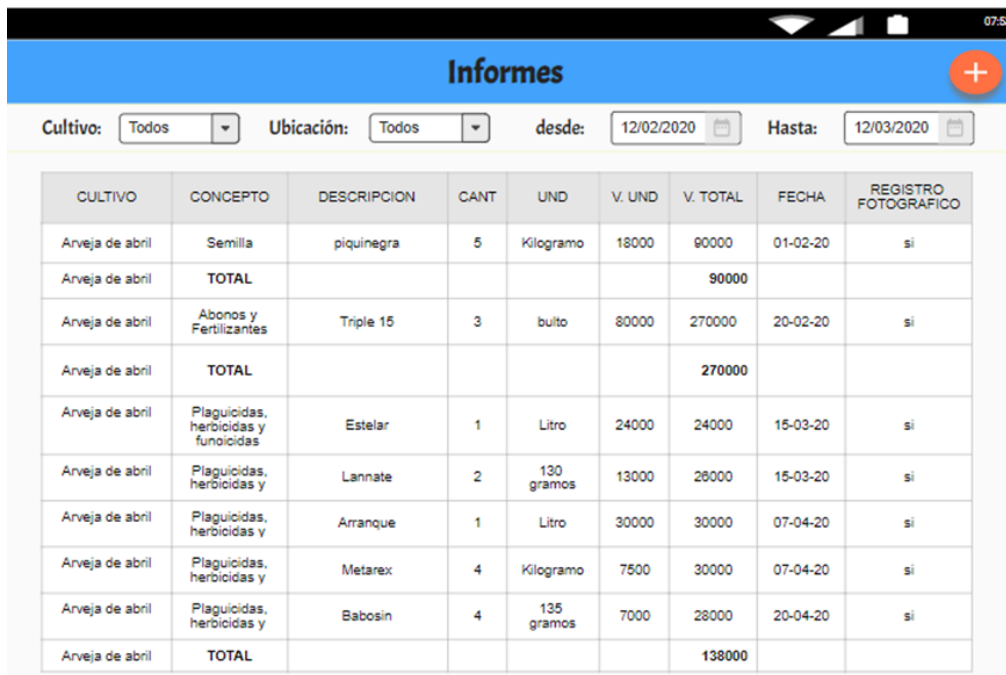
La Figura 52 es la vista de usuario para la funcionalidad de consultar costos, esta es una tabla con filtros de búsqueda para obtener una mayor precisión.

Figura 52. Consultar costos (vista propuesta)

FECHA	CANT.	UNIDAD	DESCRIPCIÓN	V. UND	V. TOTAL	
21/02/2020	2	litros	Fungicida	50.000	100.000	<input checked="" type="checkbox"/>
03/03/2020	3	bultos	Abono	80.000	240.000	<input checked="" type="checkbox"/>
10/03/2020	2	rollo	cabuya	30.000	60.000	<input type="checkbox"/>
29/02/20	3					
Total: \$400.000						

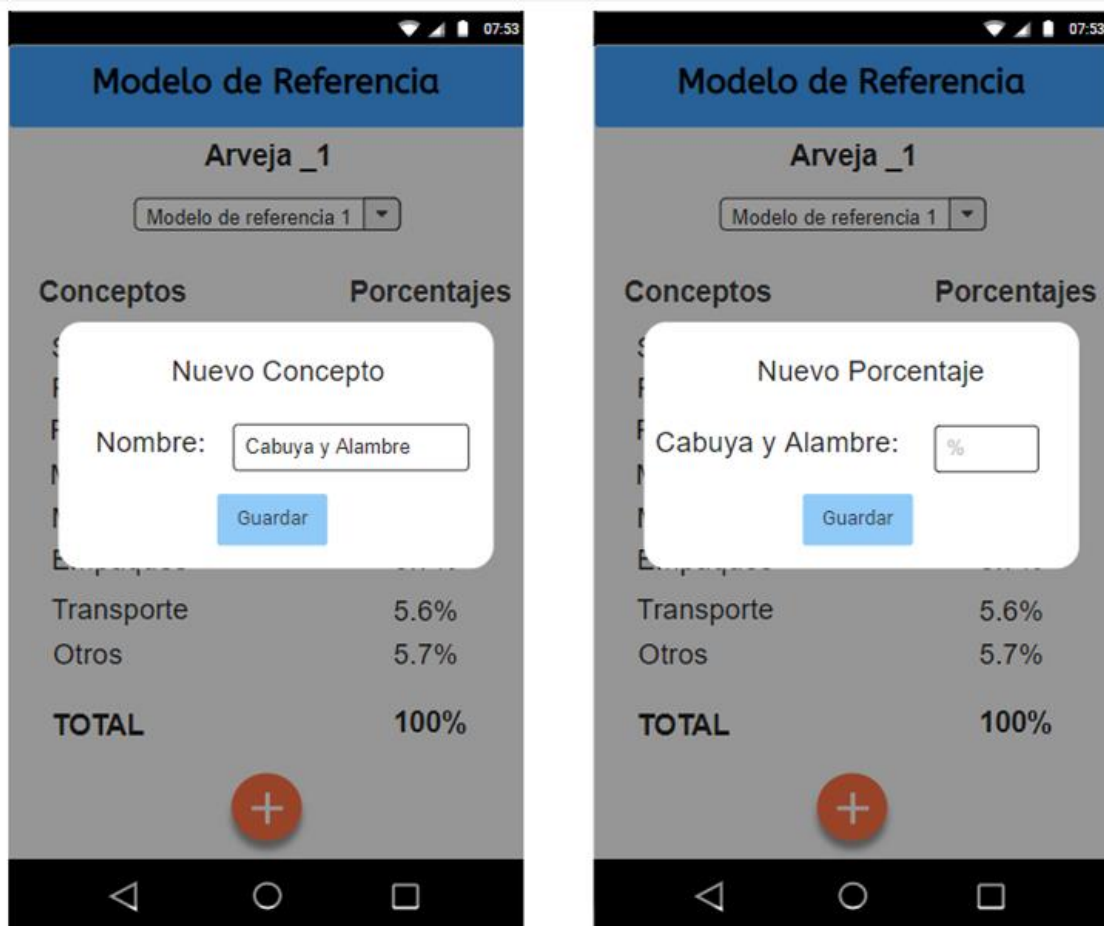
La Figura 53 muestra el informe que se planeó en las vistas de usuario para la funcionalidad de informe general del cultivo, en el cual también hay filtros para una mejor búsqueda.

Figura 53. Informe general (vista propuesta)

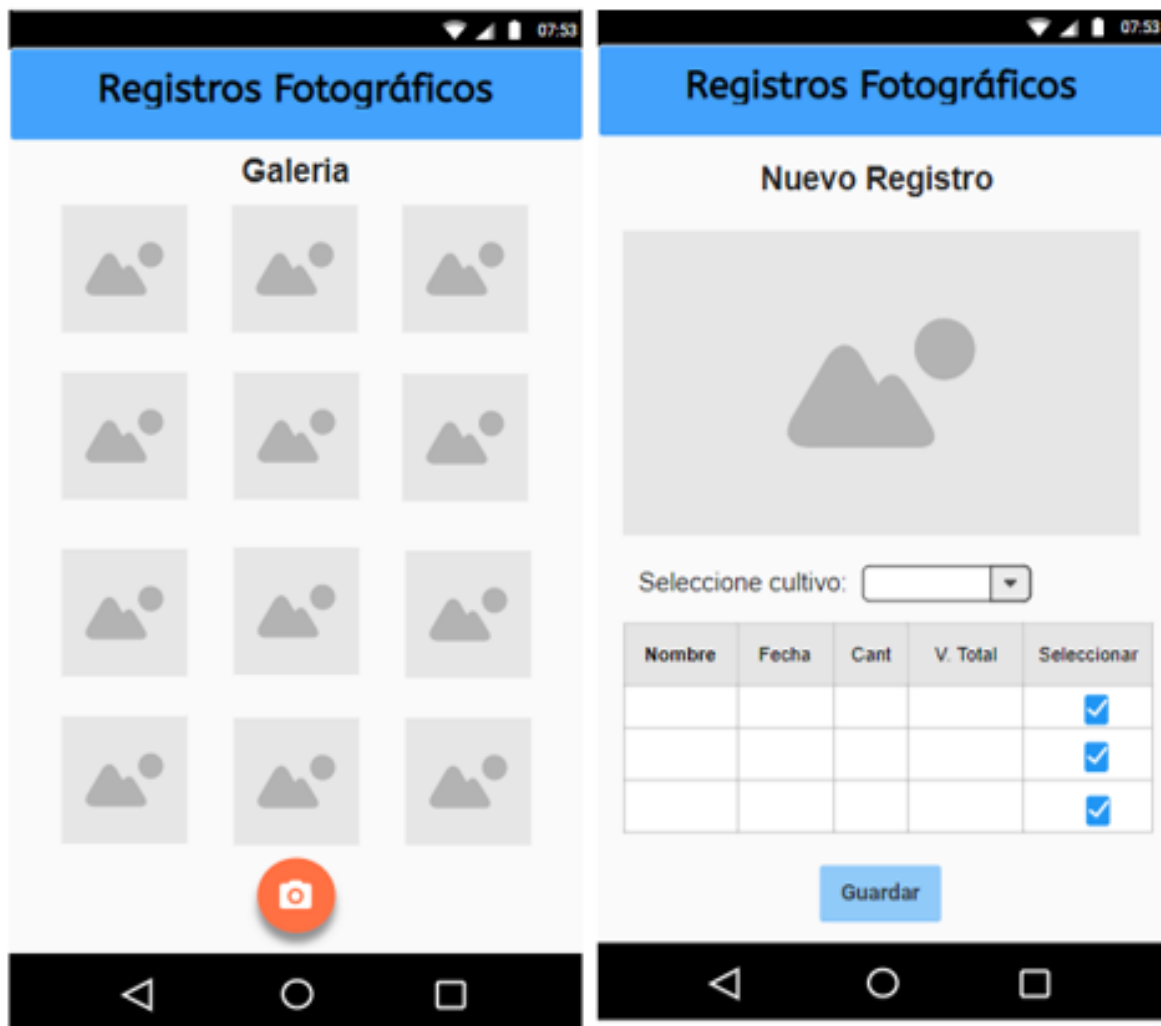


CULTIVO	CONCEPTO	DESCRIPCION	CANT	UND	V. UND	V. TOTAL	FECHA	REGISTRO FOTOGRAFICO
Arveja de abril	Semilla	piquinegra	5	Kilogramo	18000	90000	01-02-20	si
Arveja de abril	TOTAL					90000		
Arveja de abril	Abonos y Fertilizantes	Triple 15	3	bulto	80000	270000	20-02-20	si
Arveja de abril	TOTAL					270000		
Arveja de abril	Plaguicidas, herbicidas y fungicidas	Estelar	1	Litro	24000	24000	15-03-20	si
Arveja de abril	Plaguicidas, herbicidas y	Lannate	2	130 gramos	13000	26000	15-03-20	si
Arveja de abril	Plaguicidas, herbicidas y	Arranque	1	Litro	30000	30000	07-04-20	si
Arveja de abril	Plaguicidas, herbicidas y	Metarex	4	Kilogramo	7500	30000	07-04-20	si
Arveja de abril	Plaguicidas, herbicidas y	Babosin	4	135 gramos	7000	28000	20-04-20	si
Arveja de abril	TOTAL					138000		

La Figura 54 representa las vistas de usuario que se plantearon para registrar o crear un modelo de referencia, aquí se observa que se generaba un concepto con su respectivo porcentaje.

Figura 54. Modelo de referencia específico (vista propuesta)

En la Figura 55 se observan las vistas de usuario planteadas antes de desarrollar la aplicación móvil para la funcionalidad de ‘añadir registros fotográficos’, en estas se encuentra una vista para la galería y otra para tomar la fotografía a la que se le asocian los costos previamente registrados.

Figura 55. Gestionar registros fotográficos (vista propuesta)

Apéndice E. Vistas de usuario definitivas de la aplicación móvil.

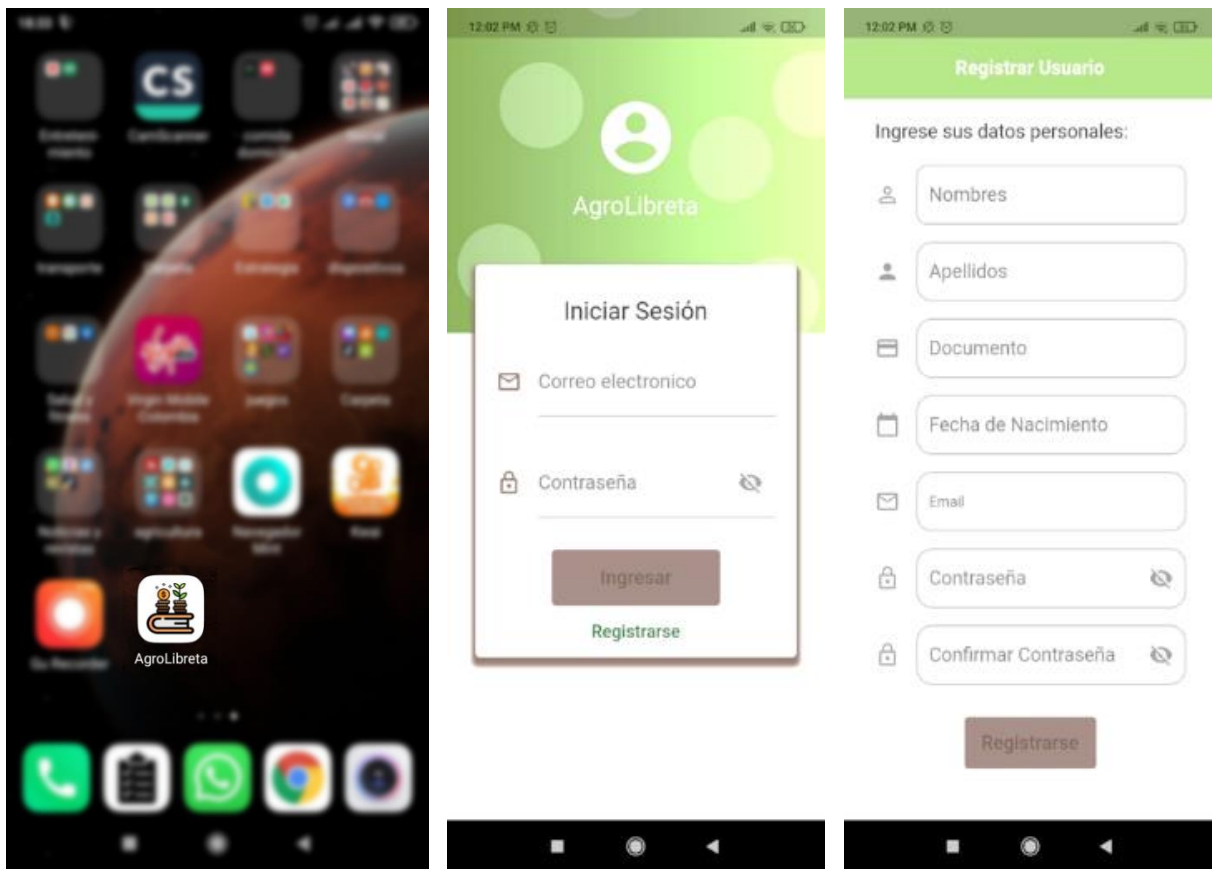
Seguidamente se presentan las vistas de usuario de la aplicación móvil presentadas de acuerdo con la metodología de prototipado, así como se planteó el proyecto este se desarrolló en tres prototipos siendo el último el producto final.

1. Primer prototipo

En este prototipo básicamente se implementaron los requerimientos para modelo de referencia ideal o por defecto, el cual no es modificado por el usuario, registrar cultivos, registrar / editar / consultar / eliminar costos y por último el informe general’.

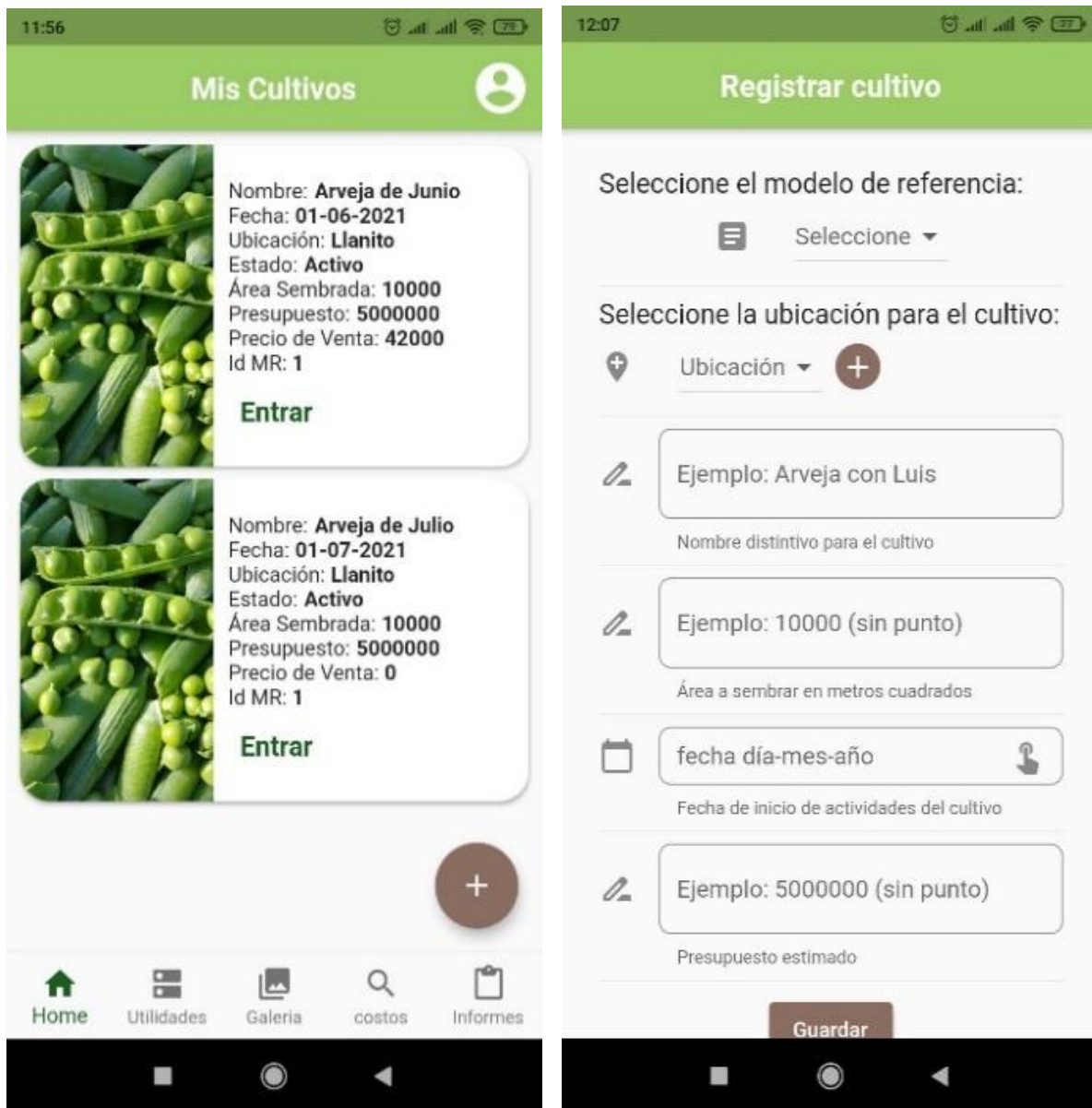
En la Figura 56 se pueden observar las vistas de la aplicación móvil AgroLibreta para el registro de usuarios, antes de llegar a ello debe ingresar al icono de la aplicación en su dispositivo, seleccionar ‘Registrarse’ y diligenciar el formulario.

Figura 56. Iniciar sesión y registrar usuario (primer prototipo)



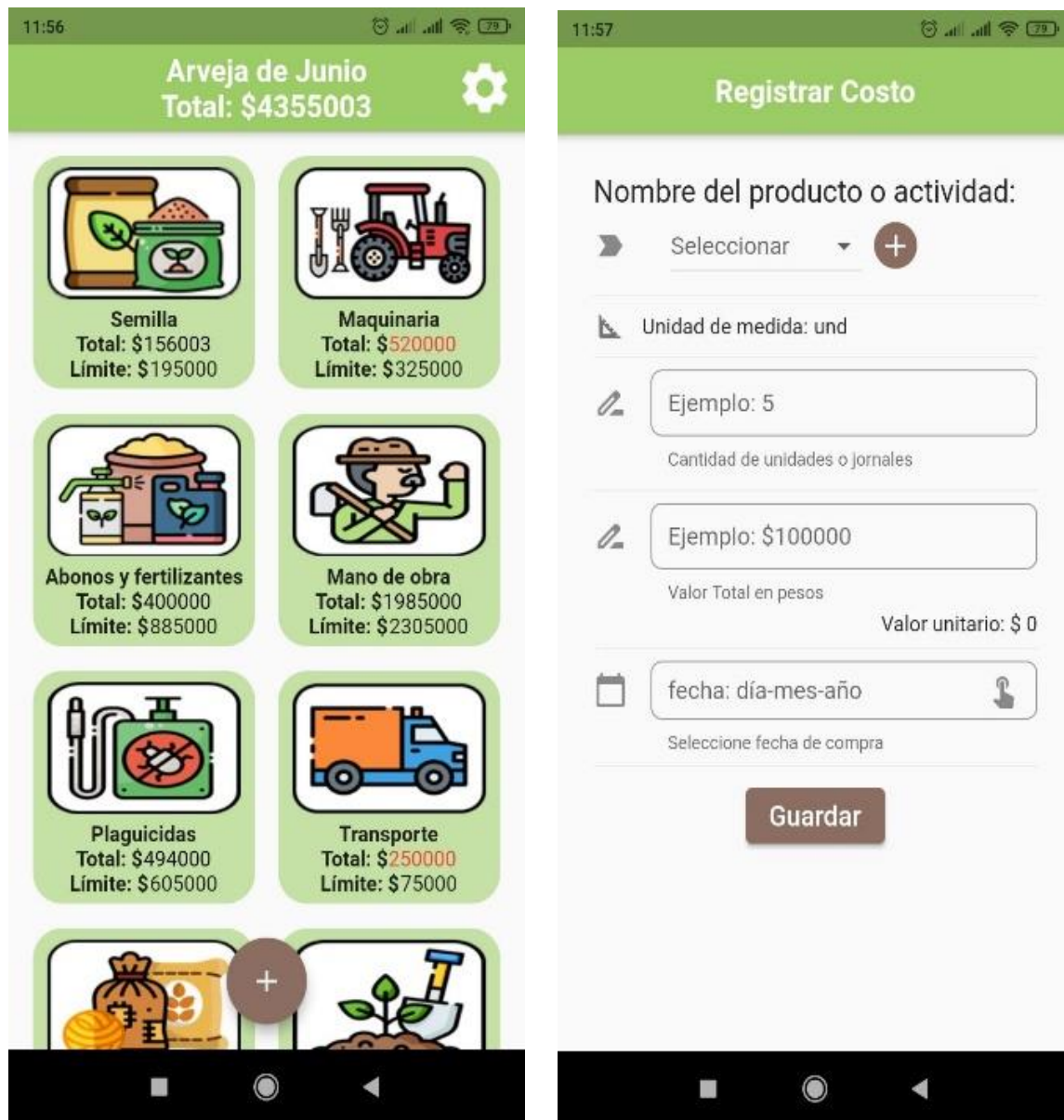
En la Figura 57 se puede observar a la izquierda la vista de home en la cual se listan los cultivos existentes y el botón para registrar uno nuevo, en la vista de la derecha se observa el formulario para ello.

Figura 57. Listar y registrar cultivo (primer prototipo)



En la Figura 58 se observa la vista de usuario para el resumen del cultivo en la cual se muestran los costos actualizados de los conceptos del modelo de referencia y sus límites, también se puede observar la vista del formulario para registrar un nuevo costo.

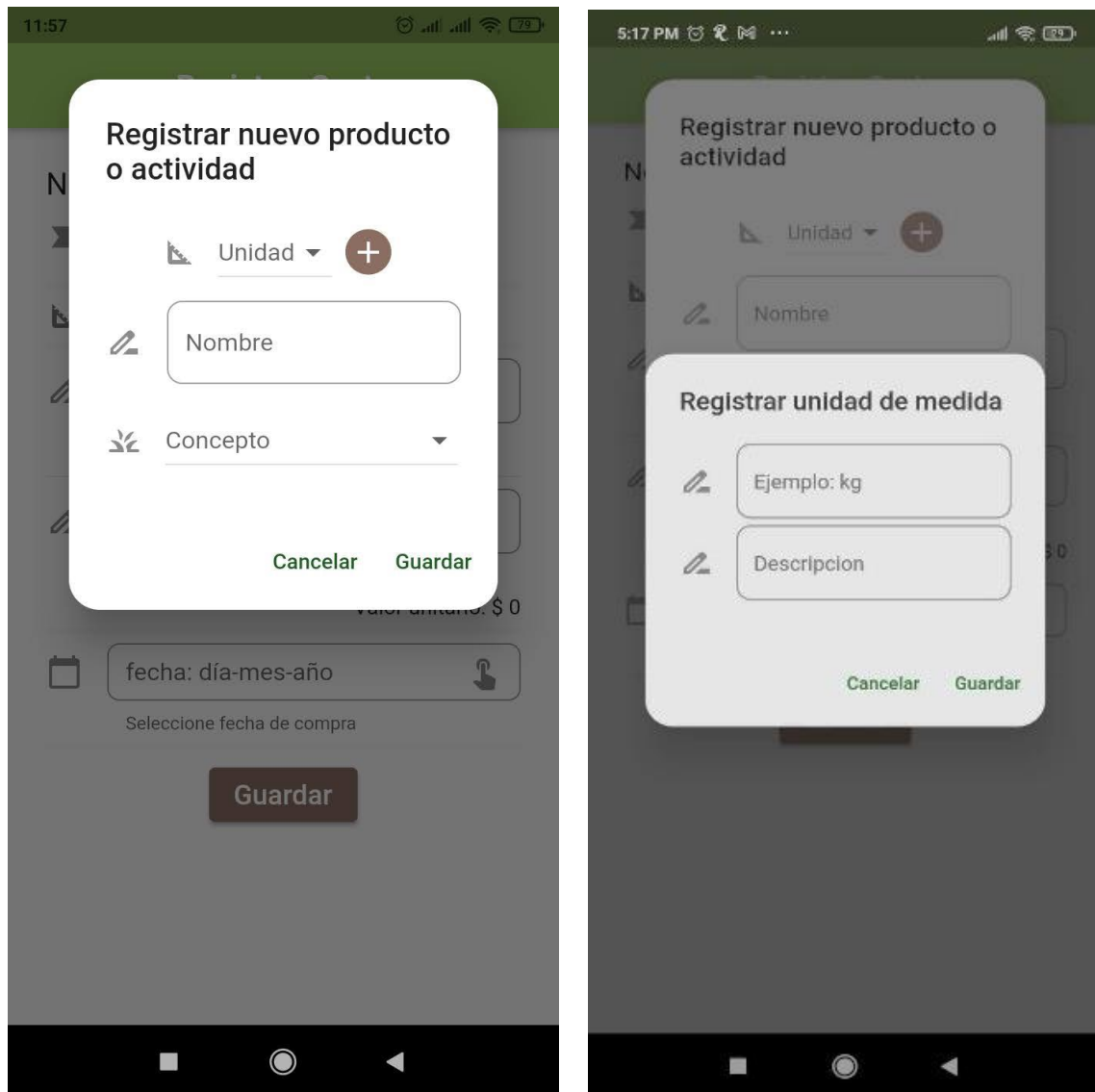
Figura 58. Resumen del cultivo y registrar costo (primer prototipo)



En la Figura 59 se observa la vista de usuario para la funcionalidad de registrar un nuevo producto o actividad la cual se despliega al oprimir el botón de '+' la vista 'registrar costo', también se puede observar la vista de usuario para la funcionalidad de registrar unidad de

medida, esta se despliega al oprimir el botón de ‘+’ de la vista ‘registrar nuevo producto o actividad’.

Figura 59. Registrar producto o actividad y unidad de medida (primer prototipo)



La Figura 60 corresponde a la vista de usuario para la funcionalidad de consultar costos con filtros, se observa que se puede consultar por un rango de fechas, por el nombre del producto

o la actividad y por concepto del modelo de referencia, también se pueden distinguir los costos que se encuentran asociados a un registro fotográfico por que se encuentran en negrilla.

Figura 60. Consultar costos con filtros (primer prototipo)

15:47

Buscar costos por:

Cultivo: Arveja de enero ▾

Desde: hasta:

Producto o actividad: Seleccionar ▾

Concepto: Concepto ▾

Fecha	Cnt	Und.	Nombre	V.und	V.total
01-06-21	5.0	kg	Rabo de gallo	5000	25000

Home Utilidades Galería **costos** Informes

15:45

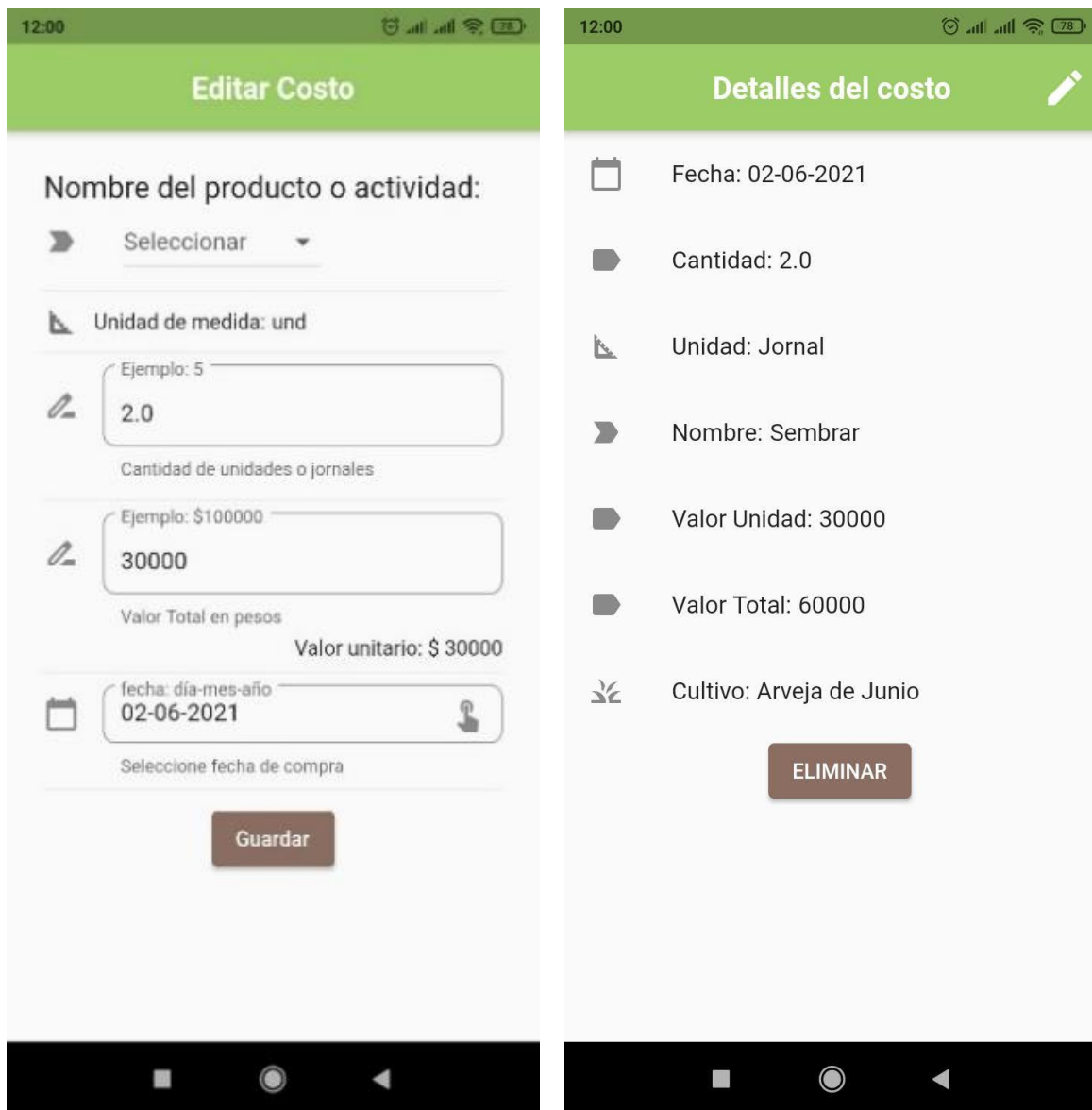
Buscar costos por:

Fecha	Cnt	Und.	Nombre	V.und	V.total
01-06-21	5.0	kg	Rabo de gallo	5000	25000
10-06-21	5.0	kg	Rabo de gallo	10000	50000
14-06-21	3.0	Rollo	Alambre	25000	75000
04-07-21	100.0	Jornal	Deshierbar	30000	3000000
04-07-21	20.0	kg	Dithane	100000	2000000
04-07-21	100.0	Bultos	Triple 15	30000	3000000

Home Utilidades Galería **costos** Informes

En la Figura 61 se observan las vistas de la aplicación móvil AgroLibreta para editar costo en la cual se pueden modificar sus valores y la vista para detalles del costo, en la cual se pueden observar atributos como la fecha, el valor total, el cultivo al cual pertenece y demás.

Figura 61. Editar costo y detalles del costo (primer prototipo)



En la Figura 62 se puede observar el modelo de referencia por defecto que tiene la aplicación móvil si los usuarios no desean registrar uno nuevo para su cultivo, para ingresar a este se debe seleccionar en la barra de navegación ‘Utilidades’.

Figura 62. Modelo de referencia por defecto (primer prototipo)



La Figura 63 representa la vista de la aplicación móvil para la funcionalidad del informe general del cultivo, en este se puede observar el costo total y los costos correspondientes a un cultivo

Figura 63. Informe general del cultivo (primer prototipo)



2. Segundo prototipo

En este prototipo se implementó las funcionalidades para generar el modelo de referencia específico, gestionar soportes de costos, estos son fotografías de las facturas o eventos comerciales que el usuario posea, de tal forma que puede asociar a los registros de costos una fotografía y así conocer qué valores están soportados o sustentados.

En la Figura 64 se observa la vista de usuario para la funcionalidad ‘registrar un modelo de referencia específico’ en el cual el usuario puede ingresar los porcentajes que desee para su nuevo modelo de referencia, teniendo en cuenta que la suma de todos debe ser del 100%.

Figura 64. Registrar un modelo de referencia (segundo prototipo)

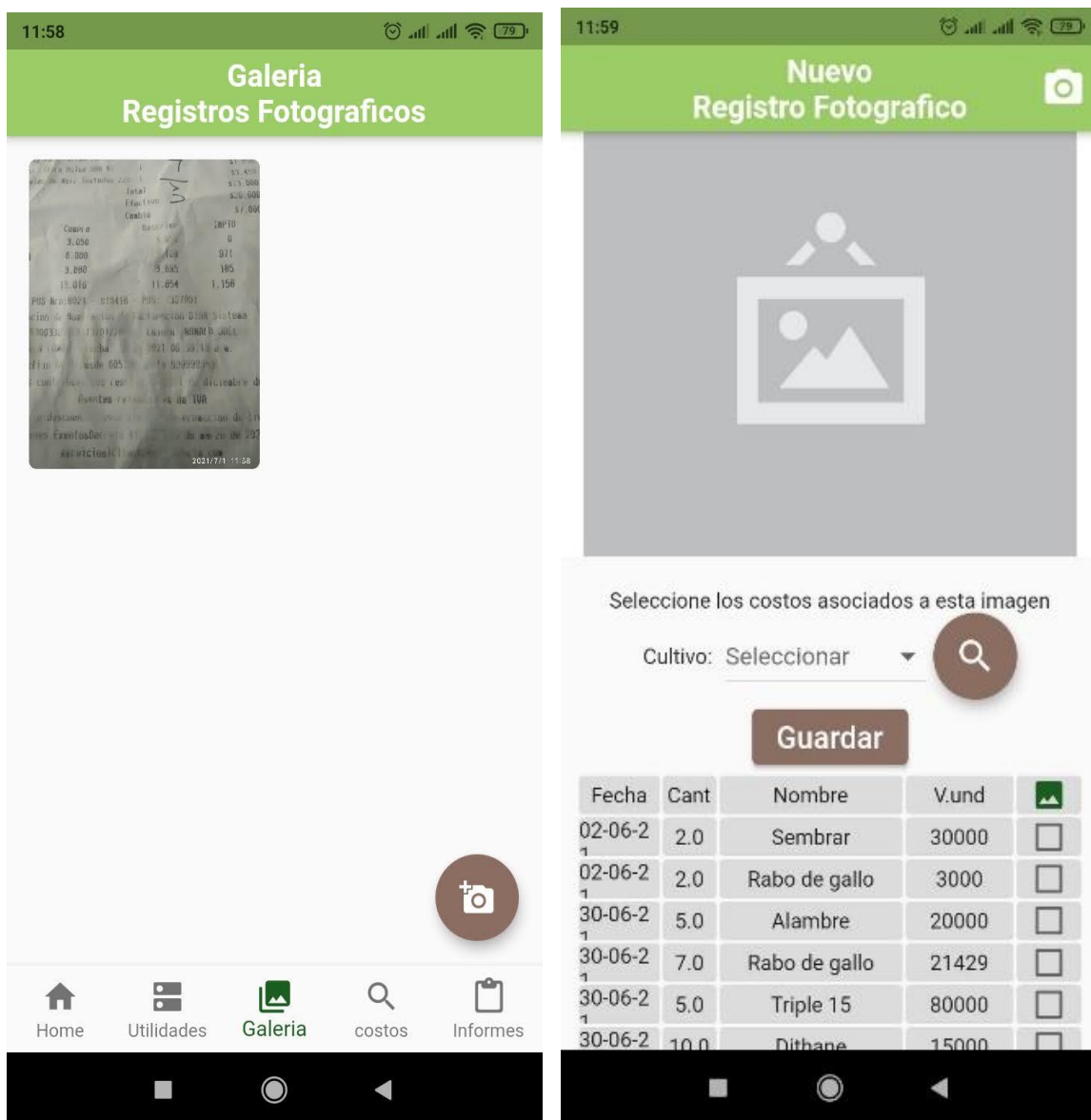
Categoría	Ejemplo de Valor (%)
Semilla:	Ej: 3.9 %
abonos y fertilizantes:	Ej: 17.7 %
Plaguicidas:	Ej: 12.1 %
Materiales y empaques	Ej: 6.7 %
Maquinaria	Ej: 6.5 %
Mano de obra	Ej: 46.1 %
Transporte	Ej: 1.5 %
Otros	Ej: 5.5 %

debe ser 100 y lleva: 0.0 %

Finalizar

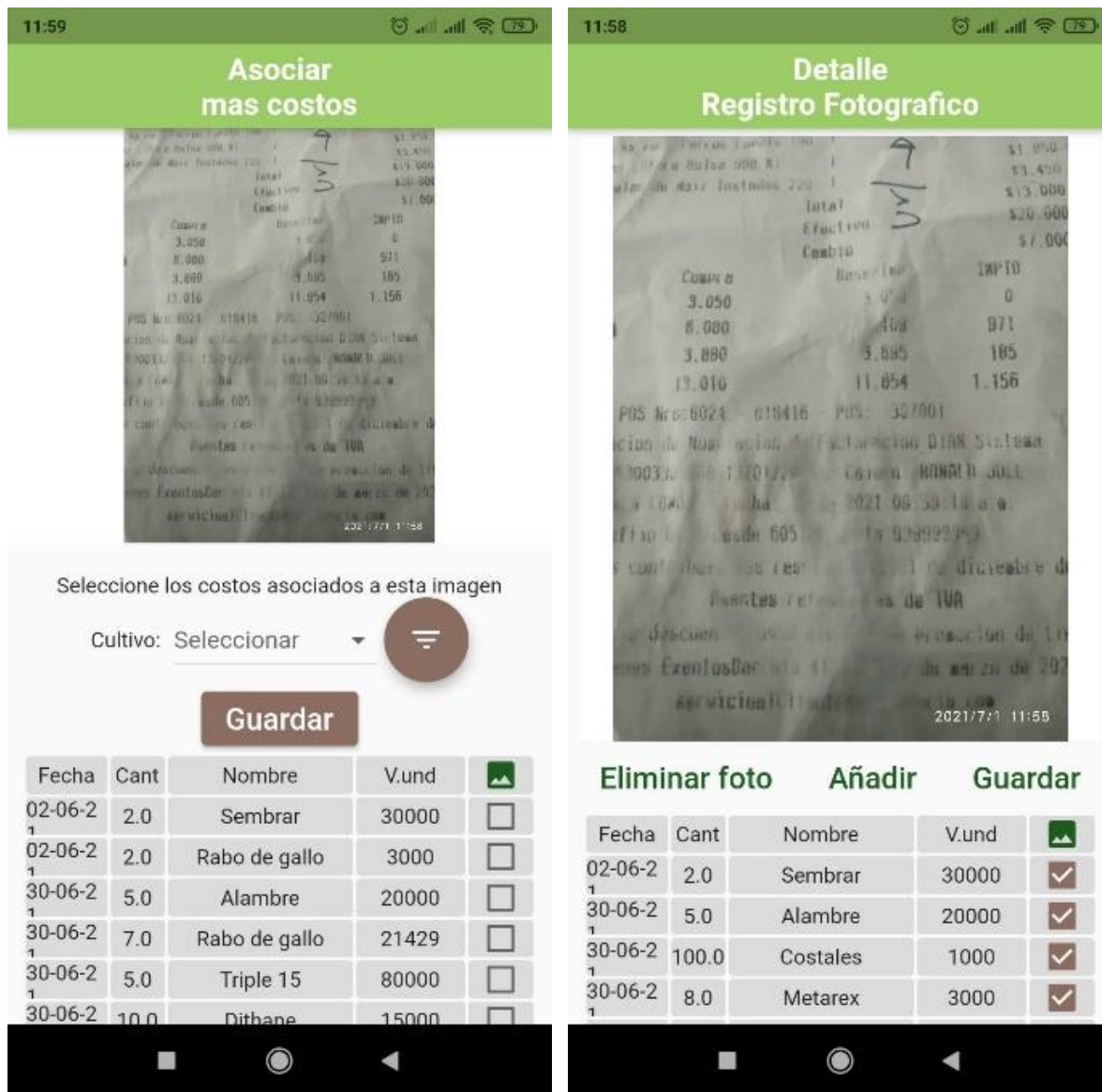
En la Figura 65 se observan las vistas para galería de registros fotográficos y la vista para añadir uno de estos, se debe oprimir el icono de la cámara para tomar desde la aplicación la fotografía y después seleccionar los costos que el usuario desee asociar.

Figura 65. Galería y nuevo registro fotográfico (segundo prototipo)



La Figura 66 representa las vistas de usuario para asociar más costos a un registro fotográfico y para ver el detalle de estos. Para ingresar a la vista ‘asociar más costos’ es necesario seleccionar la fotografía desde la galería

Figura 66. Asociar costos a registro fotográfico y detalle (segundo prototipo)

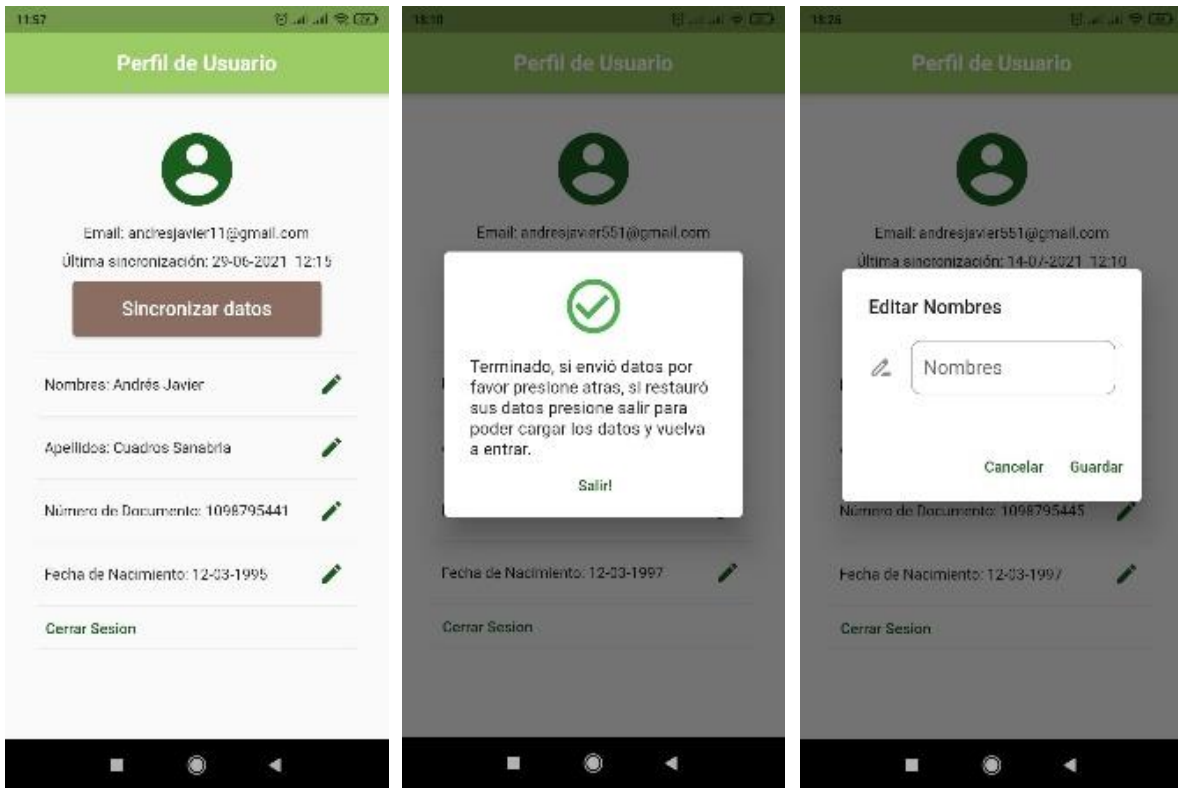


3. Tercer prototipo

Con este prototipo se terminó la aplicación móvil implementando las funcionalidades para comparar los costos registrados con el modelo de referencia por medio de gráficas en el informe general, copias de seguridad, editar perfil de usuario y configuración del cultivo.

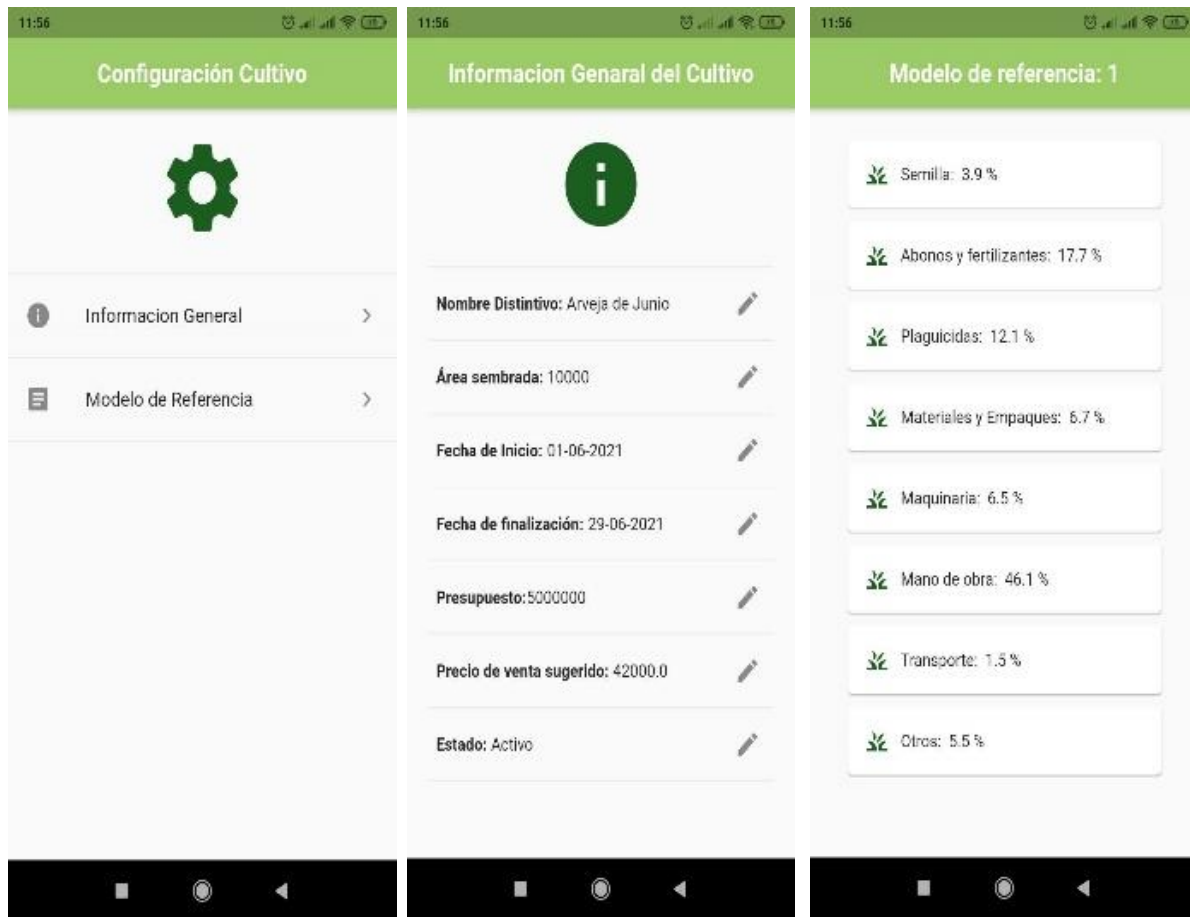
La Figura 67 es la vista de la otra funcionalidad del informe general en la cual se grafican los costos del cultivo y los límites de los conceptos del modelo de referencia, esto con el fin de comparar visualmente los costos y sus límites con respecto al presupuesto estimado, también se observa la funcionalidad del informe general para generar un documento PDF que le permite al usuario tener un historial documentado de todos sus cultivos.

Figura 67. Comparar costos con el modelo de referencia (prototipo final)



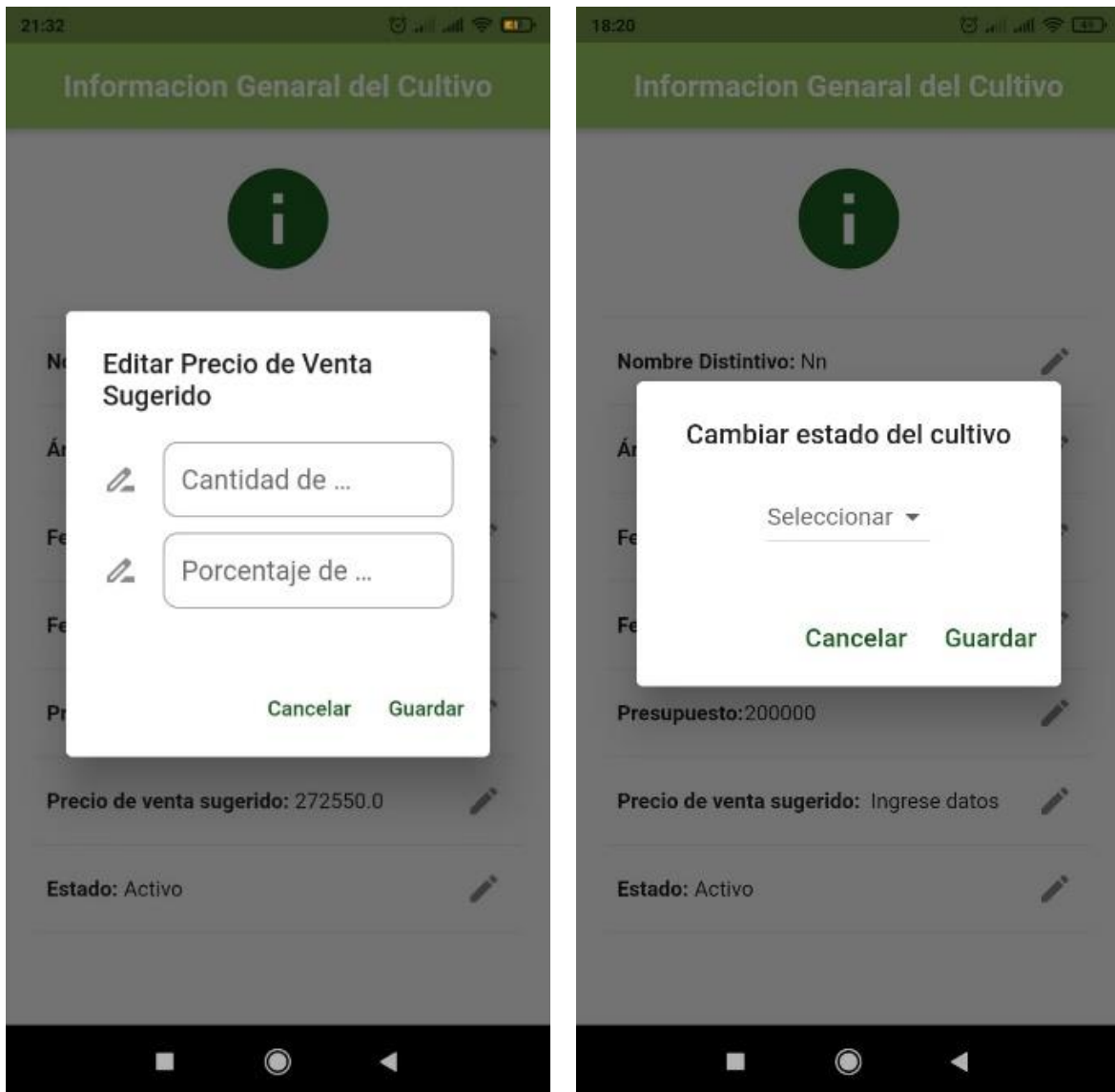
La Figura 69 muestra la vista de la aplicación móvil AgroLibreta para la configuración de cultivos, si desea ingresar a esta opción debe estar dentro del cultivo al cual quiere editar y seleccionar el icono de configuración, encontrará dos opciones una para observar el modelo de referencia que se está utilizando y otra para ingresar a la información general.

Figura 69. Configuración del cultivo (prototipo final)



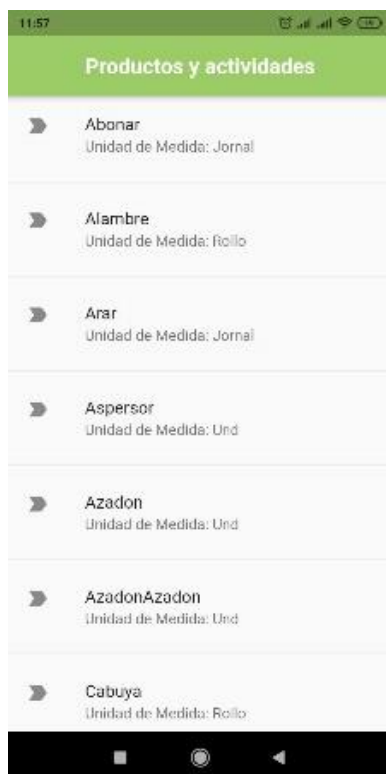
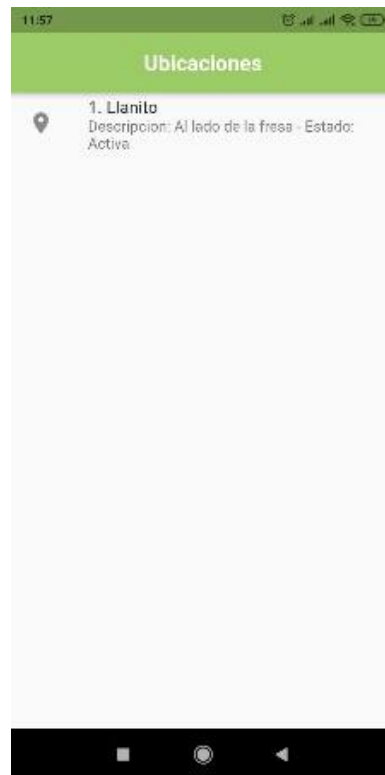
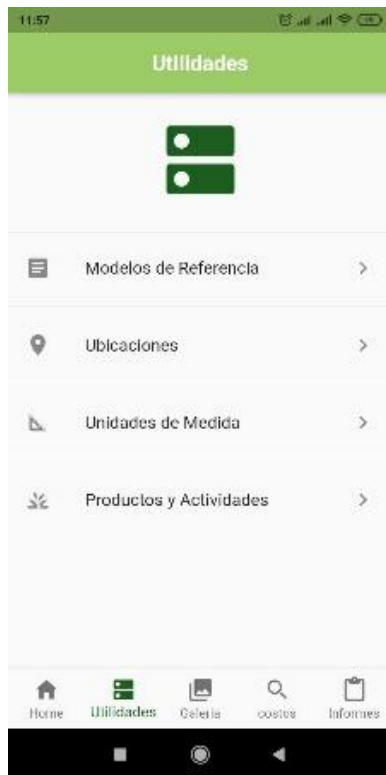
En la Figura 70 se puede observar que dentro de la funcionalidad de configuración del cultivo evidenciada en la figura anterior se encuentra la opción de editar precio de venta la cual le sugiere un valor según el porcentaje de ganancia que desea obtener y la opción de cambiar el estado del cultivo, el cual tiene tres opciones 'Activo' 'Inactivo' y 'Perdido'.

Figura 70. Editar precio de venta y cambiar estado del cultivo (prototipo final)



La Figura 71 muestra la función que recopila todos los modelos de referencia, las ubicaciones, las unidades de medida y los productos o actividades para poder visualizarlos y tenerlos en cuenta, esta opción se llamó ‘Utilidades’ y se encuentra en la barra de navegación.

Figura 71. Utilidades (prototipo final)



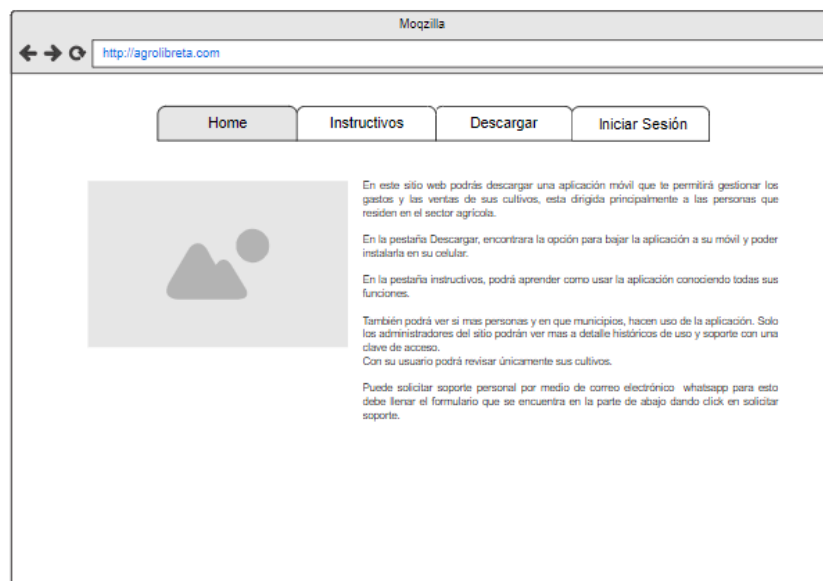
Apéndice F. Vistas de usuario del sitio web.

En este apéndice se presentan las vistas de usuario propuestas del sitio web, realizadas al plantear el proyecto y las vistas definitivas al subirlo en el hosting.

1. Vistas de usuario propuestas

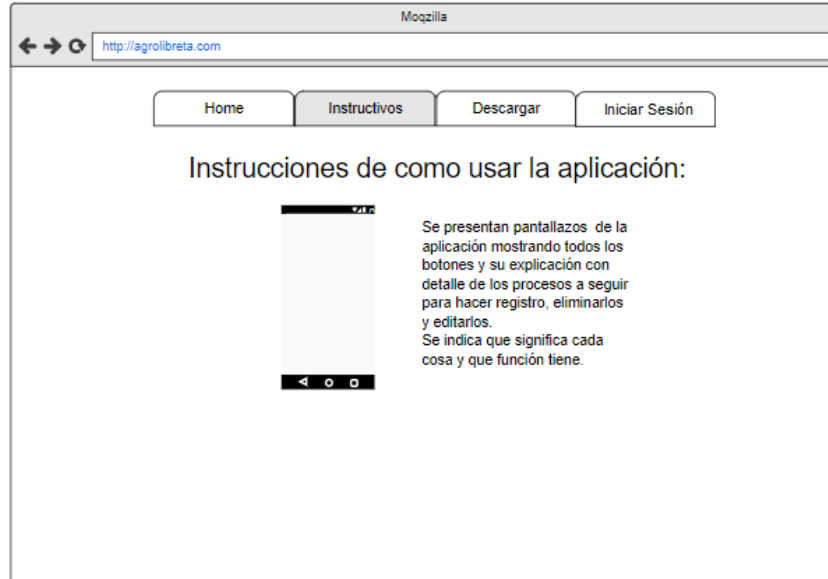
Estas vistas se desarrollaron previo a la aprobación del plan por medio de la herramienta moqups.com, en ella también se realizaron las vistas propuestas de la aplicación móvil. En la Figura 72 se observa la vista de usuario planteada antes de desarrollar la aplicación móvil, esta representa la página de inicio en la cual se describe el proyecto.

Figura 72. Página principal (propuesta)



La Figura 73 se muestra la página de instrucciones planteada antes de desarrollar el proyecto para las vistas de usuario, en ella se planteó mostrar un instructivo para el uso de la aplicación móvil Agrolibreta.

Figura 73. Página de instrucciones (propuesta)



La Figura 74 muestra la página para descargas que se tenía planteada para el desarrollo de la aplicación móvil en ella se debía tener un botón o un enlace para la descarga de esta.

Figura 74. Página de descargas (propuesta)



La Figura 75 representa la página que se planteó para el login o inicio de sesión en la cual el usuario debe ingresar su correo y contraseña.

Figura 75. Página de ingreso (propuesta)



The image shows a screenshot of a web browser window titled "Mozilla" with the address bar displaying "http://agrolibreta.com". The page layout includes a navigation menu at the top with buttons for "Home", "Instructivos", "Descargar", and "Iniciar Sesión". Below the menu, there is a large placeholder image on the left and a login form on the right. The login form consists of two input fields labeled "USUARIO:" and "CONTRASEÑA:", both containing "Placeholder" text, and a button labeled "INGRESAR" below them.

2. Vistas de usuario definitivas

Estas vistas son las implementadas en el sitio web ya subido en un hosting, puede observar que se desarrollaron tal cual a las vistas propuestas. Para ingresar al sitio web seleccione la siguiente URL <https://agrolibretav1.web.app/> . La Figura 76 representa la vista de la página principal o de inicio del sitio web al subirlo en el hosting, se puede observar que contiene una explicación de lo que es la aplicación móvil AgroLibreta utilizando video y algunas definiciones.

Figura 76. Página principal del sitio web



La Figura 77 muestra la página de instrucciones definitiva del sitio web en la cual se describe un paso a paso para utilizar la aplicación móvil.

Figura 77. Página de instrucciones del sitio web



En la Figura 78 se evidencia el resultado final de la página de descargas, en ella se puede descargar el archivo ‘.apk’ de la aplicación.

Figura 78. Página de descargas del sitio web



La Figura 79 muestra como resultado la página de ingreso en el sitio web una vez subido al hosting para que el usuario consulte su última sincronización.

Figura 79. Página de ingreso del sitio web



En la Figura 80 se observa la página a la que el usuario ingresa una vez que oprima el botón consultar de la página anterior, su función es que el usuario vea cuándo fue la última vez que realizó una copia de seguridad.

Figura 80. Página para consulta del sitio web



Apéndice G. Pruebas realizadas para la evaluación del funcionamiento.

En seguida se presentan las evidencias del plan de pruebas para la evaluación del funcionamiento de la aplicación móvil, estas figuras se encuentran clasificadas por los tipos de pruebas descritos en el numeral 3.3.

1. Pruebas de integración.

Estas son las tablas realizadas para las pruebas de integración también descritas en el numeral 3.3.2 en las cuales se describe un ejemplo del funcionamiento y los errores encontrados por los desarrolladores al integrar las partes, dichos errores fueron resueltos en su totalidad. Se realizaron las tablas correspondientes a los requerimientos de mayor complejidad.

Tabla 11. Prueba de validación para registrar usuario

#	Usuario	Aplicación	Datos	Revisar
1	-	Mostrar campos a diligenciar	-	-
2	Ingresar nombres	-	Gladys	Que ingrese más de tres caracteres
3	Ingresar apellidos	-	Flórez Rodríguez	Que ingrese más de tres caracteres
4	Ingresar número de documento	-	6023456	Que no ingrese letras
5	Ingresar fecha de nacimiento	Mostrar calendario	5-6-1965	Que no ingrese una fecha futura
6	Ingresar email	-	Galdysfr64@hotmail.com	Que cumpla con la estructura válida
7	Ingresar	-	G123456F	Que ingrese más

	contraseña			de seis caracteres
8	Confirmar contraseña	-	G123456F	Que sea igual a la contraseña
9	Registrarse	Mostrar 'Usuario Registrado' y la página de inicio de sesión	-	-
#	Errores- Descripción		Corrección	Fecha corrección
1	Al ingresar valores alfanuméricos en el campo contraseña no permitía continuar.		sí	07/04/2021

Tabla 12. Prueba de validación para registrar cultivo

#	Usuario	Aplicación	Datos	Revisar
1	-	Mostrar campos a diligenciar	-	-
2	Seleccionar un modelo de referencia	Consultar y permitir seleccionar un modelo de referencia	1	-
3	Seleccionar una ubicación	Consultar y permitir seleccionar una ubicación	una piñuela	-
4	Ingresar un nombre distintivo para el cultivo	-	Arveja de enero	-
5	Ingresar el área a sembrar	-	1000	Que no ingrese letras
6	Ingresar fecha de inicio de actividades del cultivo	Mostrar el calendario	15-01-2021	Que la fecha no sea anterior a 01-01-2021
7	Ingresar presupuesto	-	3000000	Que no ingrese letras
8	Guardar	-	-	Que se haya seleccionado un modelo de referencia y una ubicación
9	-	Mostrar la página de 'Mis cultivos' con el cultivo creado	-	-

#	Usuario	Aplicación	Datos	Revisar
10				*Observaciones: Si no existe la ubicación debe crearla
#	Errores- Descripción		Corregido	Fecha de corrección
1	Las unidades de medida no estaban especificadas		sí	25/06/2021

Tabla 13. Prueba de validación para añadir un registro fotográfico

#	Usuario	Aplicación	Datos	Revisar
1	Seleccionar icono de la cámara	-	-	-
2	-	Acceder a la cámara	-	-
3	Tomar una fotografía al soporte del evento comercial	-	-	-
4	Seleccionar un cultivo	Mostrar la fotografía tomada, consultar cultivos y permitir seleccionar uno	Arveja de enero	-
5	Presionar el botón de la lupa	Mostrar costos del cultivo seleccionado	-	Que los costos asociados a otros registros no se muestren
5	Seleccionar costos asociados	Mostrar checkbox seleccionado	20-01-21/10/Triple 15/80000	-
6	Guardar	-	-	Que se haya tomado una fotografía
7	-	Mostrar página 'Galería Registros fotográficos' con la imagen guardada	-	-
#	Errores - Descripción		Corregido	Fecha de corrección
1	No permitía cargar más de 3 imágenes para mostrar en la galería		sí	11/05/2021

Tabla 14. Prueba de validación para registrar modelo de referencia

#	Usuario	Aplicación	Datos	Revisar
1	-	Mostrar campos a diligenciar	-	-
2	Ingresar un porcentaje en cada concepto	Mostrar suma de porcentajes ingresados	2 - 15 - 14 - 5.5 - 6 - 48 - 5 - 4.5	-
3	Finalizar	-	-	Que la suma de los porcentajes sea del 100%
4	-	Mostrar la página 'Modelos de Referencia' con el modelo creado	-	-
#	Errores - Descripción		Corregido	Fecha de corrección
1	Permitía almacenar conceptos con porcentajes sin restricciones		sí	21/06/2021

Tabla 15. Pruebas de validación para consultar costos

#	Usuario	Aplicación	Datos	Revisar
1	Seleccionar un cultivo	Consultar los cultivos y permitir seleccionar uno	Arveja de enero	-
2	Seleccionar un producto o actividad	Consultar y permitir seleccionar un producto o actividad	Triple 15	-
3	Seleccionar el ícono de la lupa	Presentar los costos que cumplan las condiciones	-	Que los costos con registro fotográfico estén en negrilla
4				Observaciones: Se debe seleccionar dos veces seguidas el ícono de la lupa al momento

#	Usuario	Aplicación	Datos	Revisar
				de consultar
#	Errores - Descripción		Corregido	Fecha de corrección
1	No permitía filtrar por fechas debido al formato		sí	07/06/2021

Tabla 16. Prueba de validación para ver informe

#	Usuario	Aplicación	Datos	Revisar
1	Seleccionar un cultivo	Consultar cultivos y permitir seleccionar uno	Arveja de enero	-
2	Presionar ícono de la lupa	Listar datos relacionados al cultivo	-	-
3	Seleccionar ícono de 'PDF'	Mostrar vista previa del informe en PDF	-	-
#	Errores - Descripción		Corregido	Fecha de corrección
1	No fácil ver si los resultados eran favorables o desfavorables		sí	30/06/2021

Tabla 17. Prueba de validación para realizar una copia de seguridad

#	Usuario	Aplicación	Datos	Revisar
1	Seleccionar botón 'sincronizar datos'	Presentar alerta que indica proceso en ejecución	-	Que tenga conexión
2	-	Presentar ícono verde, indicaciones y botón para salir	-	-
3	-	-	-	*Observaciones: Sólo si está recuperando datos debe presionar el botón salir

#	Usuario	Aplicación	Datos	Revisar
#	Errores - Descripción		Corregido	Fecha de corrección
1	El formato de la fecha era poco entendible para las personas		sí	15/06/2021

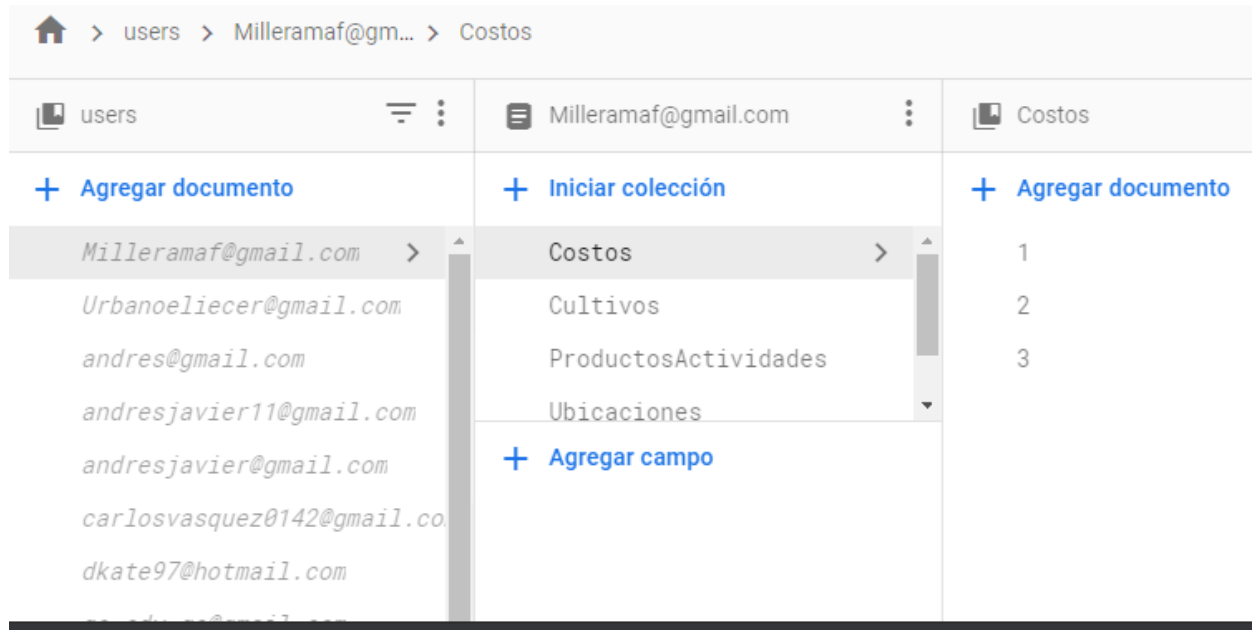
Tabla 18. Prueba de validación para consultar la última copia de seguridad

#	Usuario	Sitio web	Datos	Revisar
1		Presentar formulario para ingresar credenciales	-	-
2	Ingresar correo	-	Galdysfr64@hotmail.com	-
3	Ingresar contraseña	-	G123456F	-
4	Seleccionar botón 'consultar'	Realizar petición a Firebase para validar credenciales	-	Que el usuario esté registrado en la aplicación móvil.
5	-	Presentar pantalla con la fecha de la última copia de seguridad.	-	-
#	Errores -Descripción		Corregido	Fecha de la corrección
1	Al cambiar de pestaña la sesión finalizaba		sí	23/06/2021

2. Pruebas alfa

Esta prueba se encuentra descrita en el numeral 3.3.3, la Figura 81 es una captura de la plataforma de Firebase que demuestra algunos usuarios que participaron.

Figura 81. Algunos usuarios de la prueba alfa



3. Pruebas beta

La prueba beta se encuentra descrita en el numeral 3.3.4, el siguiente es el correo que se les envió a los usuarios que participaron en la prueba beta en especial a los estudiantes del IPRED:

Reciban un cordial saludo.

Agradecemos que haya podido participar en el plan de pruebas del proyecto AgroLibreta (aplicación móvil). A continuación, indicamos la dinámica del proceso a seguir:

1. Revisar los video tutoriales para conocer qué es AgroLibreta, por qué surge la necesidad de desarrollarla y cómo utilizarla, consulte los videos en los siguientes enlaces:
 - Introducción: Primer video
 - De donde surge: Segundo video
 - Paso a paso: Tercer video
2. Una vez revisados los tutoriales podrán descargar la aplicación, registrarse e iniciar sesión. Se puede descargar de la play store por el nombre AgroLibreta o utilizar el enlace que los lleva directamente:

https://play.google.com/store/apps/details?id=com.andresdeisy.agrolibreta_v2

3. Luego de iniciar sesión la idea es que registren:
 - Un cultivo.
 - Costos.
 - Un registro fotográfico (factura) y asocie un costo.
 - Un Modelo de Referencia.

4. Terminado el paso 3 podrán realizar las siguientes acciones:
 - Consultar costos.
 - Ver informe y descargar PDF.
 - Realizar copia de seguridad (sincronización de los datos).
 - Consultar fecha de la última copia de seguridad (en el sitio web solo después de realizar copia de seguridad).
 - * Enlace al sitio web: <https://agrolibretav1.web.app/index.html>

5. Navegar por la aplicación de manera libre y probando todas las demás funcionalidades que desee.

6. Comunicarse a cualquiera de los números de WhatsApp o correos si tiene alguna duda o para solicitar ingresar a un grupo de WhatsApp creado con el propósito de instruirlos. Agradecemos enviarnos pantallazos de cualquier error encontrado con el asunto 'error encontrado en AgroLibreta'.
 - 3193212377
 - 3172783700
 - andres.cuadros@correo.uis.edu.co
 - deisy.rangel@correo.uis.edu.co

7. Contestar la encuesta para conocer su experiencia como usuario, esta será enviada a su correo cuando se considere oportuno. Este punto es muy importante ya que los resultados son requeridos en el trabajo de grado.

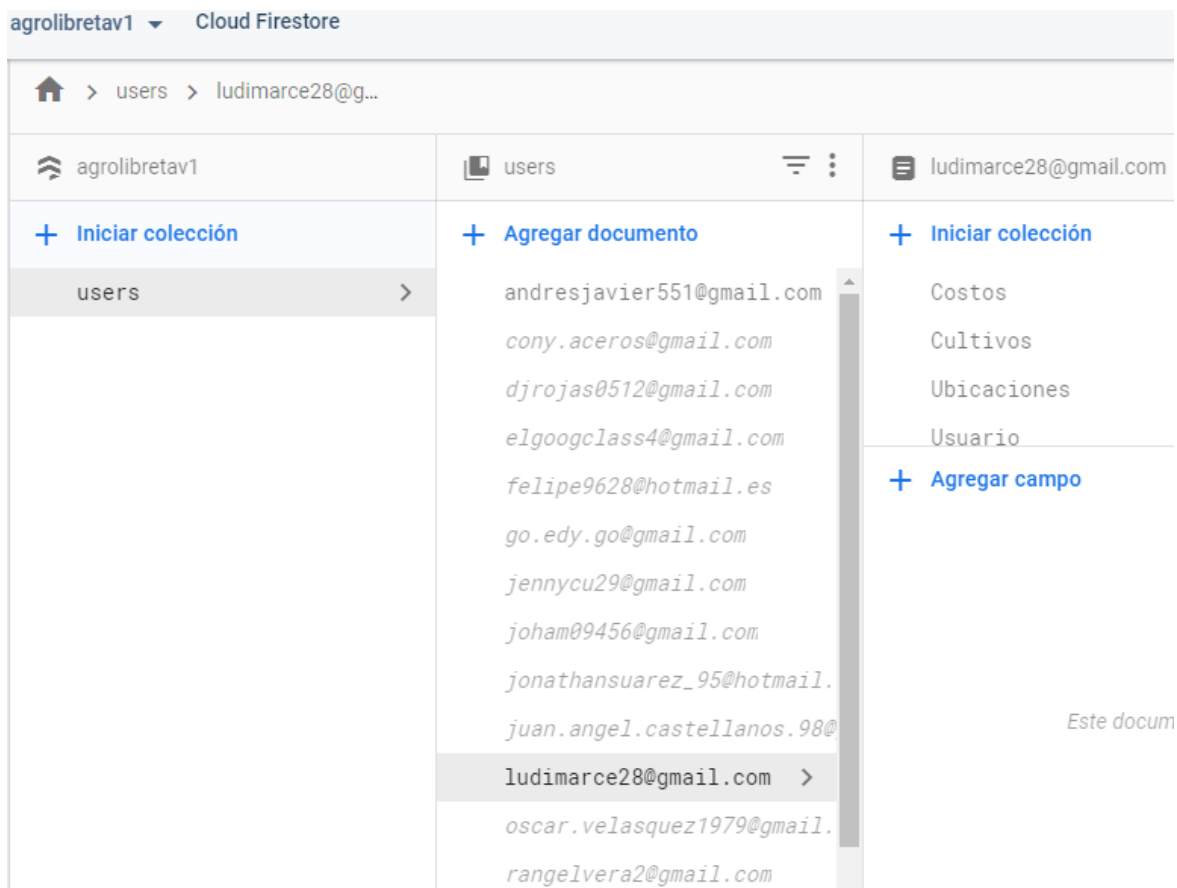
NOTA: No dude en comunicarse a cualquiera de los números de WhatsApp mencionados si tiene alguna duda o si desea ingresar a un grupo de WhatsApp creado con el propósito de instruirlos.

*Tenga en cuenta que el software está pensado para cultivos de arveja.

Atentamente,
Andres Javier Cuadros Sanabria - 2151144
Deisy Katherine Rangel Flórez - 2142669
Estudiantes de ingeniería de sistemas UIS

En la Figura 82 se pueden observar algunos usuarios que participaron en la prueba beta, algunos de ellos son productores de la región.

Figura 82. Algunos usuarios de la prueba beta



4. Pruebas de estrés

Las siguientes figuras representan la evidencia de la prueba de estrés realizada a la misma hora con cinco usuarios, en estas se puede observar una captura de la plataforma Firebase en la cual todos los usuarios tienen la misma fecha y hora.

Figura 83. Primer usuario para la prueba de estrés

```
apellidos: ██████████  
documento: ██████████  
email: "jennycu29@gmail.com"  
fechaNacimiento: "29-08-1989"  
fechaUltimaSincro: "21-08-2021 12:26"  
idUsuario: 1  
nombres: "Jenny"
```

La Figura 84 es una captura que realizó el usuario número dos para la prueba de estrés.

Figura 84. Segundo usuario para la prueba de estrés

```
apellidos: ██████████  
documento: ██████████  
email: "andresjavier551@gmail.com"  
fechaNacimiento: "12-03-1997"  
fechaUltimaSincro: "21-08-2021 12:26"  
idUsuario: 1  
nombres: "Andrés Javier"
```

La Figura 85 es una captura que realizó el usuario número tres para la prueba de estrés.

Figura 85. Tercer usuario para la prueba de estrés

```
apellidos: [REDACTED]
documento: [REDACTED]
email: "deisykatherinerangelflorez@gmail.com"
fechaNacimiento: "05-06-1997"
fechaUltimaSincro: "21-08-2021 12:26"
idUsuario: 1
nombres: "Deisy"
```

La Figura 86 es una captura que realizó el usuario número cuatro para la prueba de estrés.

Figura 86. Cuarto usuario para la prueba de estrés

```
apellidos: [REDACTED]
documento: [REDACTED]
email: "ludimarce28@gmail.com"
fechaNacimiento: "12-06-1992"
fechaUltimaSincro: "21-08-2021 12:26"
idUsuario: 1
nombres: "Ludis"
```

La Figura 87 es una captura que realizó el usuario número cinco para la prueba de estrés.

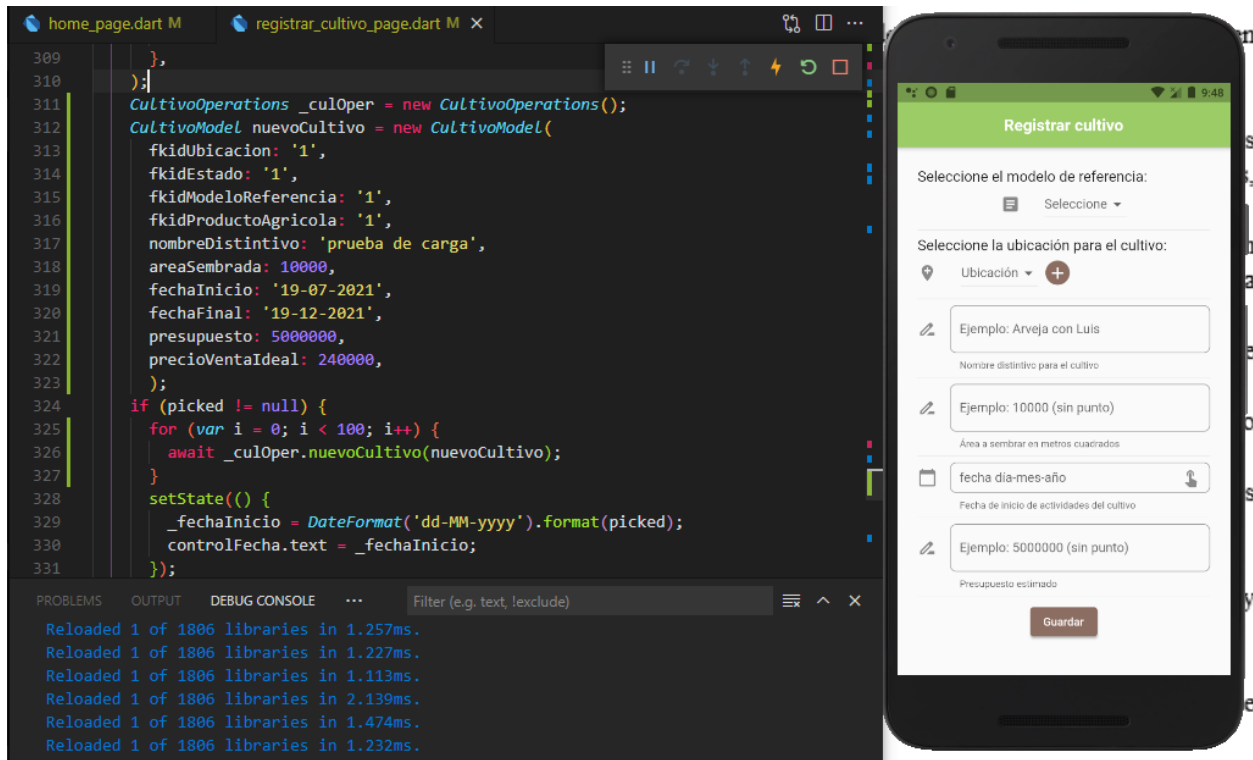
Figura 87. Quinto usuario para la prueba de estrés

```
apellidos: ██████████ (s  
documento: ██████████  
email: "zamoraeu123@gmail.com"  
fechaNacimiento: "16-07-1971"  
fechaUltimaSincro: "21-08-2021 12:26"  
idUsuario: 1  
nombres: "Eulalia"
```

5. Pruebas de carga

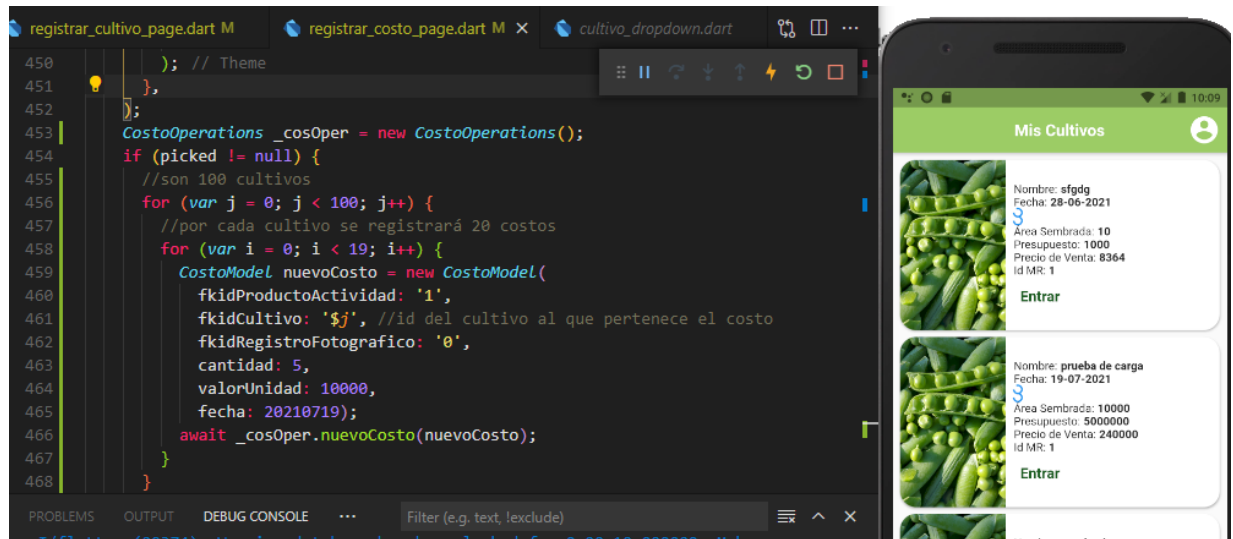
Las siguientes figuras son pantallazos de las pruebas de carga realizadas a la aplicación, insertando más de 2000 registros. En la Figura 88 por medio de un ciclo 'for' se insertaron de manera automática 101 registros de cultivos, se pudo observar que el comportamiento de la aplicación con esta cantidad de registros era normal.

Figura 88. Carga de 101 cultivos



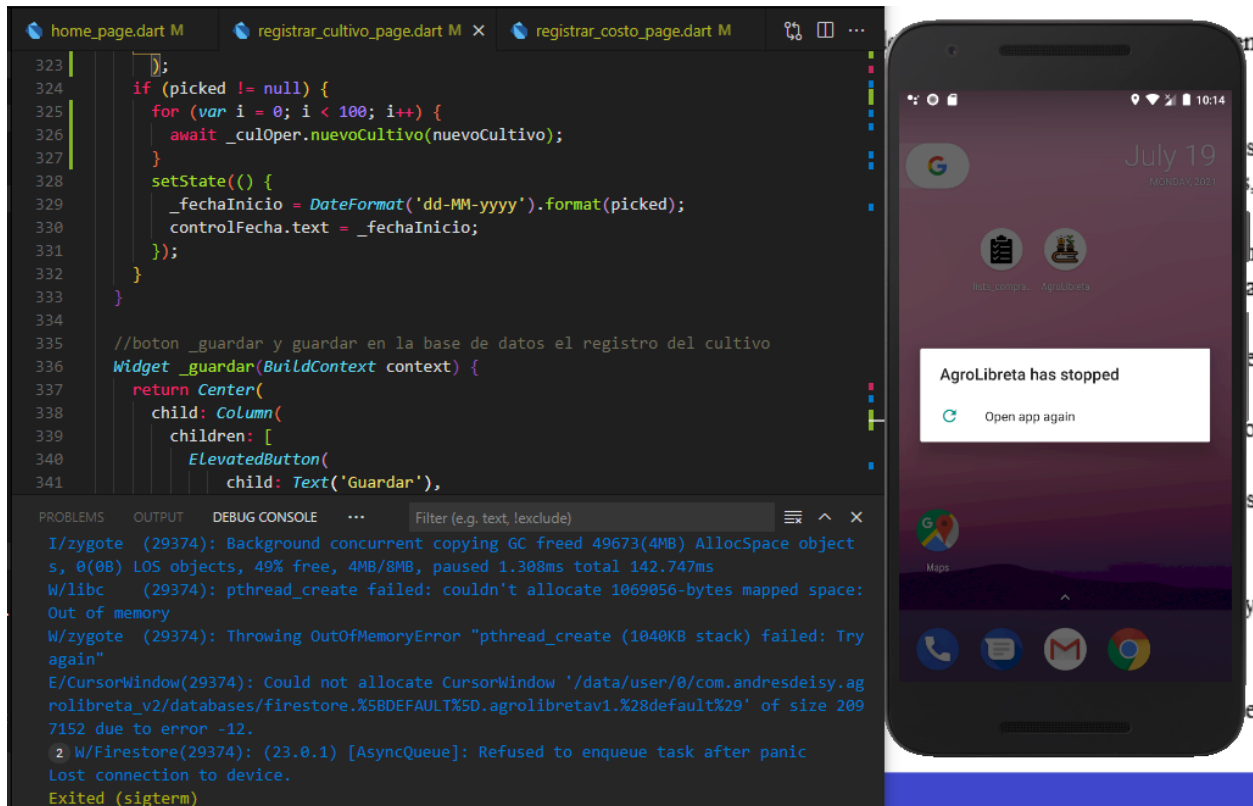
Con un ciclo 'for anidado en otro' se registraron 20 costos a cada uno de los 101 cultivos en la Figura 89 la aplicación se tornó algo lenta en las vistas home, informes y costos al momento de filtrar.

Figura 90. Aplicación al tener los 2100 registros vista home



En la Figura 91. Sincronizar más de 2100 registros se intentó sincronizar los 2100 registros mencionados en la figura anterior y en la cual se puede apreciar que la aplicación móvil se detuvo y no permitió realizar esa cantidad de transacciones al mismo tiempo.

Figura 91. Sincronizar más de 2100 registros



Apéndice H. Resultados de las encuestas.

En este apéndice se presentan los resultados obtenidos en las encuestas realizadas a los usuarios para las pruebas alfa y beta

Para los resultados de la encuesta a usuarios para la prueba alfa, se realizó una encuesta por medio de Google forms de 17 preguntas de las cuales tres fueron preguntas abiertas y fue contestada por 8 usuarios. La Figura 92 es la primera pregunta ‘¿Recomendaría usar la aplicación móvil AgroLibreta a otras personas?’, se puede observar que todas las respuestas fueron positivas.

Figura 92. Pregunta # 1 - Encuesta para la prueba alfa



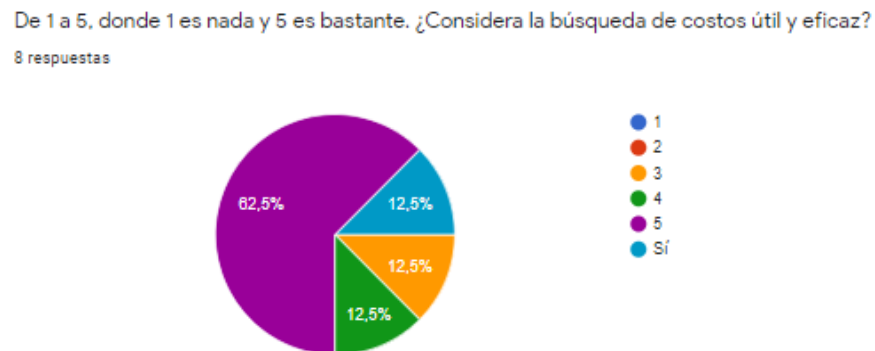
En la Figura 93 pregunta ‘¿Considera que la aplicación móvil AgroLibreta es una herramienta útil?’, se puede observar que todas las respuestas fueron afirmativas.

Figura 93. Pregunta # 2 - Encuesta para la prueba alfa



La Figura 94 representa la pregunta número tres ‘¿Considera la búsqueda de costos útil y eficaz?’, se observa que con un 75% las dos opciones, la número 5, donde 5 es bastante y 1 es nada y la opción ‘Sí, que a la gran mayoría de personas les pareció útil y eficaz.

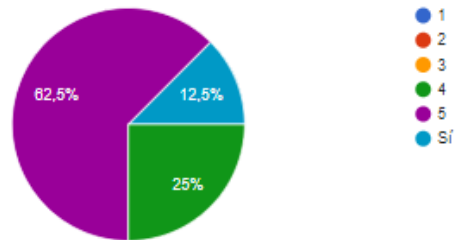
Figura 94. Pregunta # 3 - Encuesta para la prueba alfa



En la Figura 95 con un 75% se puede concluir que para la población de la prueba alfa la aplicación móvil AgroLibreta les pareció intuitiva y fácil de usar.

Figura 95. Pregunta # 4 - Encuesta para la prueba alfa

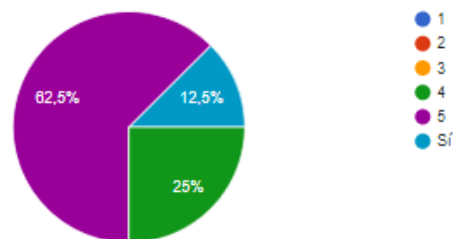
De 1 a 5, donde 1 es nada y 5 es bastante. ¿Considera que la app es intuitiva o fácil de usar?
8 respuestas



La Figura 96 representa la pregunta número cinco, en la cual se puede observar que la mayoría de las personas con un 75% comprenden la funcionalidad del modelo de referencia en un cultivo.

Figura 96. Pregunta # 5 - Encuesta para la prueba alfa

De 1 a 5, donde 1 es nada y 5 es bastante. ¿Comprende la funcionalidad del modelo de referencia en un cultivo?
8 respuestas



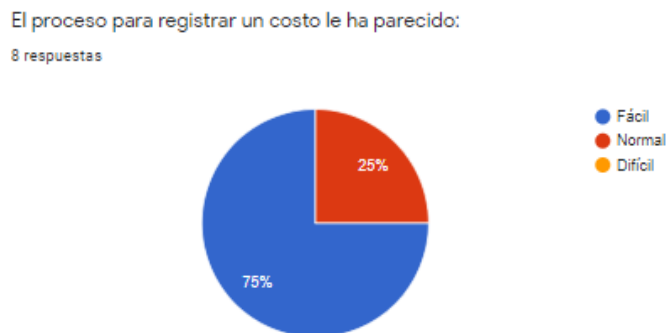
En la Figura 97 se puede observar que a la mayoría de las personas les pareció normal el registro de un cultivo.

Figura 97. Pregunta # 6 - Encuesta para la prueba alfa



En la Figura 98 se puede observar que a la mayoría de las personas que contestaron esta encuesta les pareció fácil el proceso para registrar un costo.

Figura 98. Pregunta # 7 - Encuesta para la prueba alfa



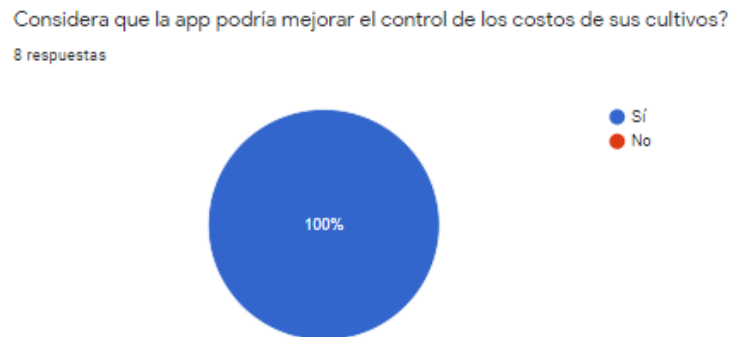
La Figura 99 representa la pregunta número ocho ‘¿La asignación de costos a un registro fotográfico le ha parecido?’, se puede observar que a la mayoría de las personas les pareció fácil.

Figura 99. Pregunta # 8 - Encuesta para la prueba alfa



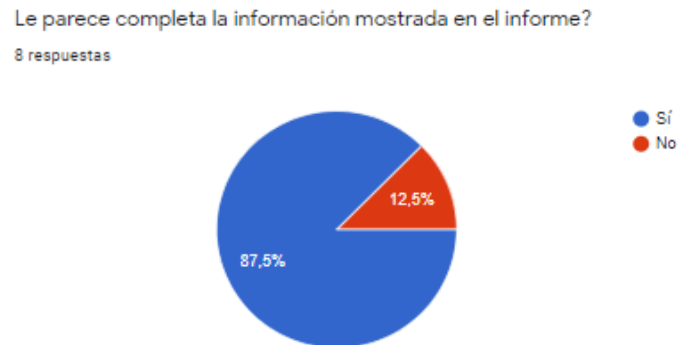
En la Figura 100 se observa que todas las personas que contestaron esta encuesta consideran que la aplicación móvil podría mejorar el control de los costos en los cultivos.

Figura 100. Pregunta # 9 - Encuesta para la prueba alfa



La Figura 101 representa la pregunta número diez '¿Le parece completa la información mostrada en el informe?', se puede observar que con un 87.5% la respuesta fue positiva.

Figura 101. Pregunta # 10 - Encuesta para la prueba alfa



En la Figura 102 se observa que la captura tomada a una factura o evento comercial para un registro fotográfico es útil con un 87.5% de afirmación.

Figura 102. Pregunta # 11- Encuesta para la prueba alfa



La Figura 103 contiene la pregunta '¿Cree usted que es útil descargar en su dispositivo el informe?', se puede observar que todas las personas contestaron afirmativamente.

Figura 103. Pregunta # 12 - Encuesta para la prueba alfa



En la Figura 104 se observa que todas las personas que contestaron la encuesta consideran que el tiempo que lleva obtener el costo total en un cultivo es menor con la aplicación móvil.

Figura 104. Pregunta #13 - Encuesta para la prueba alfa



La Figura 105 representa la primera pregunta abierta de la encuesta para la prueba alfa, en ella se pueden observar las respuestas de algunas personas para la pregunta ‘¿Qué es lo que más le gustó o resalta de la aplicación móvil?’

Figura 105. Pregunta #14 - Encuesta para la prueba alfa

Que es lo que mas le gusto o resalta de la app?

8 respuestas

Mas fácil para hacer las cuentas

Lo intuitiva que es.

Fácil uso

El registro fotografico y el informe final

En el informe final , la parte del gráfico muestra el valor máximo que se deben ir en costos como sugerencia para obtener ganancias netas y me permite llevar mejor seguimiento a los costos a la hora de cultivar

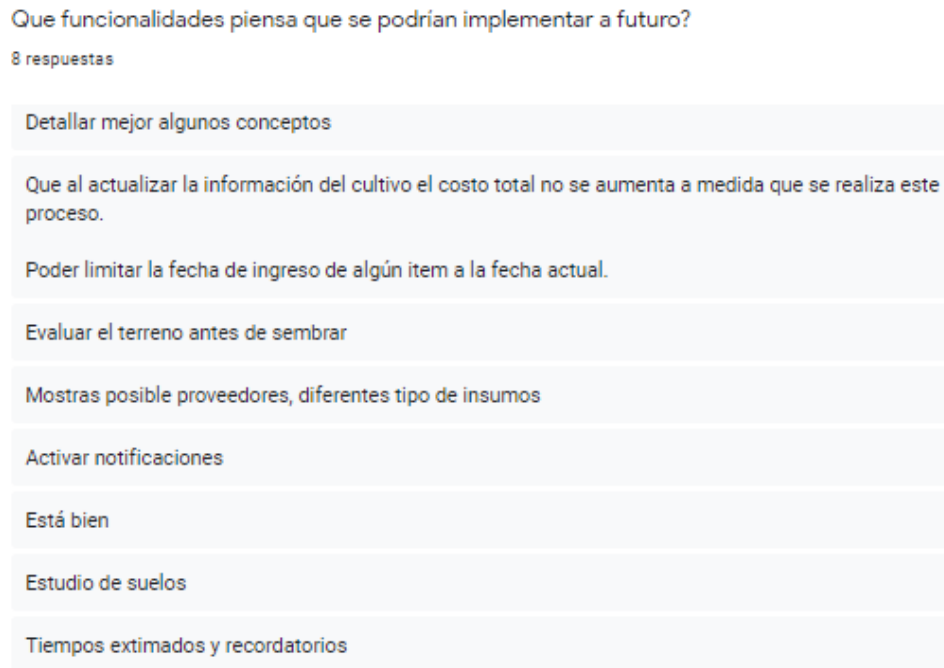
Muy eficaz

Aplicación de las TIC al sector rural.

Su diseño

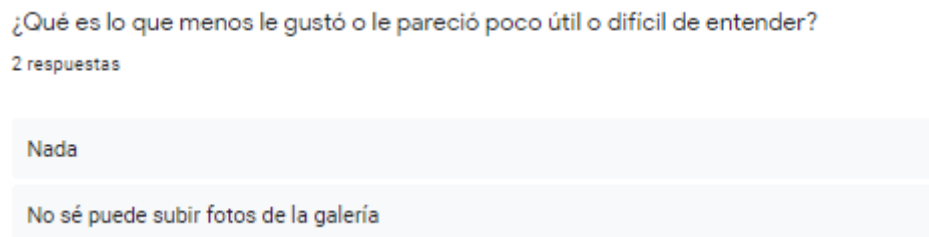
En la Figura 106 se observan las respuestas de algunas personas a la pregunta ‘¿Qué funcionalidades piensa que se podrían implementar a futuro?’.

Figura 106. Pregunta #15 - Encuesta para la prueba alfa



La Figura 107 contiene las respuestas a la pregunta abierta ‘¿Qué es lo que menos le gustó o le pareció útil o difícil de entender?’.

Figura 107. Pregunta #16 - Encuesta para la prueba alfa



Para los resultados de la encuesta a usuarios de la prueba beta, se realizó una encuesta a productores de la región y a estudiantes del IPRED por medio de Google forms de 19 preguntas de las cuales 4 son abiertas.

La Figura 108 representa la pregunta abierta para la sección correspondiente al sitio web de la encuesta realizada para la prueba beta y se observan las respuestas a la pregunta ‘¿En general como le ha parecido el sitio web?’

Figura 108. Pregunta # 1 - Encuesta para la prueba beta

AgroLibreta encuesta para usuarios: Sitio web

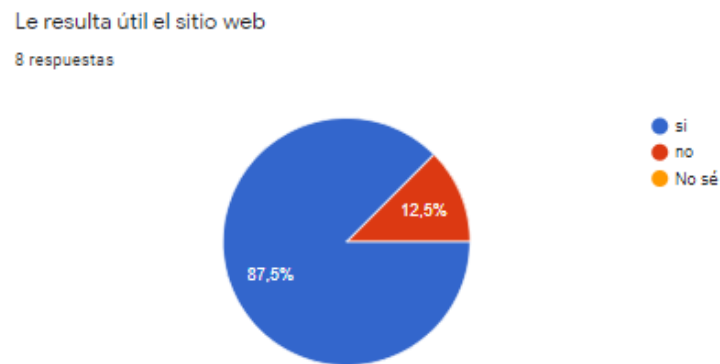
En general como le ha parecido el sitio web

8 respuestas

- Muy buena
- Genial
- muy bien
- Bueno
- No me parece muy util.
- util y buena
- Excelente propuesta
- Muy bien

La Figura 109 corresponde a la sección del sitio web se puede observar que con un 87.5% la mayoría de las personas respondieron afirmativamente a la pregunta ‘¿Le resulta útil el sitio web?’.

Figura 109. Pregunta # 2 - Encuesta para la prueba beta

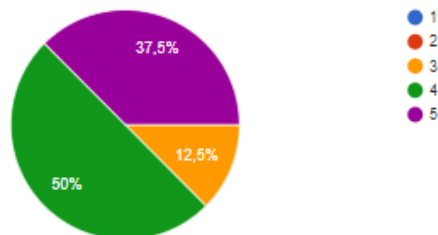


La Figura 110 corresponde a una afirmación de la sección del sitio web, ‘Resulta fácil o difícil consultar la fecha de la última copia de seguridad’, con un 50% la respuesta número cuatro fue la de mayor porcentaje.

Figura 110. Pregunta # 3 - Encuesta para la prueba beta

Seleccione de 1 a 5 si esta de acuerdo o no con la afirmación: Resulta fácil consultar la fecha de mi ultima copia de seguridad en el sitio web.

8 respuestas



La Figura 111 corresponde a una pregunta abierta sobre la aplicación móvil, en ella se pueden observar algunas respuestas sobre la pregunta ‘¿Qué es lo que menos le gusto o le pareció poco útil o difícil de entender?’

Figura 111. Pregunta # 4 - Encuesta para la prueba beta

¿Qué es lo que menos le gustó o le pareció poco útil o difícil de entender?

8 respuestas

Nada

nada

Difícil de entender la parte del precio de venta pero luego entendí y sirve bastante

La sensación de que los iconos de la interface inicial de los costos me va a enviar al resumen de ese rubro.

explicar el modelo de referencia

Un punto a mejorar sería el establecer montos límites un poco más elevados, ya que estos aumentan con mucha frecuencia.

La Figura 112 corresponde a una pregunta abierta sobre la aplicación móvil y contiene las respuestas a la pregunta ‘¿Qué es lo que más le gustó o resalta de la aplicación móvil?’.

Figura 112. Pregunta # 5 - Encuesta para la prueba beta

¿Qué es lo que mas le gusto o resalta de la app?

8 respuestas

Que me permite conocer los costos en un cultivo de una forma rápida, pero precisa, y que obtengo los informes detallados.

Lo especifico que es.

facil de usar

La visualización de la información

Es un buen inicio para del control de gastos de los cultivadores os cultivadores, aunque me gustaría saber si puede haber una opción de modo off line, para cuando el usuario no tenga cobertura de datos el la pueda usar y cuando recupere la señal el los datos se sincronicen.

es facil de usar

Es muy completa, cuenta con los elementos principales que se deben tener en cuenta en un proceso de producción.

La Figura 113 corresponde a la pregunta ‘¿Qué funcionalidades piensa que se podrían implementar a futuro?’, se observan algunas respuestas como ‘Que debería tener la opción de poner fechas para programar actividades a futuro’.

Figura 113. Pregunta # 6 - Encuesta para la prueba beta

¿Qué funcionalidades piensa que se podrían implementar a futuro?

8 respuestas

Que de la opción para otros cultivos, y que también permita sacar costos para explotaciones pecuarias

Guardar información de los cultivos para poder hacer cultivos mejorados y sin malgastar diinero

niguna

Consultar costos de Internet

Los iconos de la interface donde muestra los conceptos debería enviar al resumen de cada uno para saber cuanto he gastado en cada item, es necesario que los valores tengan separadores de miles, se puede mejorar el grafico se barras dejando los conceptos en Angulo de 45 para que los nombres no se sobrepongan,

mejoramiento del cultivo

Proveedores y clientes

Que debería tener la opción de poner fechas para programar actividades a futuro

En la Figura 114 se observa que todas las personas contestaron de manera afirmativa a la pregunta ‘¿Considera que el tiempo que lleva obtener el costo total de un cultivo es menor con la aplicación móvil AgroLibreta?’.

Figura 114. Pregunta # 7 - Encuesta para la prueba beta

Considera usted que el tiempo que lleva obtener el costo total de un cultivo es menor con la app?

8 respuestas



La Figura 115 contiene la pregunta '¿cree que es útil descargar en su dispositivo el informe general de un cultivo?', en ella se obtuvo una respuesta mayormente positiva.

Figura 115. Pregunta # 8 - Encuesta para la prueba beta

Cree usted que es útil descargar en su dispositivo el informe?

8 respuestas

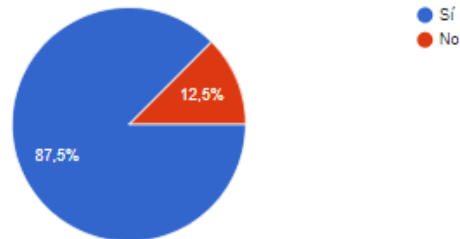


En la Figura 116 se observa que con un porcentaje de 87.5% las personas contestaron afirmativamente a la pregunta '¿Considera que la captura tomada a una factura o evento comercial para crear un registro fotográfico es útil?'.

Figura 116. Pregunta # 9 - Encuesta para la prueba beta

Considera que la captura tomada a una factura o evento comercial para crear un registro fotográfico es útil?

8 respuestas



La Figura 117 contiene las respuestas a la pregunta ‘¿Le parece completa la información mostrada en el informe general de un cultivo?’ en la cual se obtuvo una respuesta favorable.

Figura 117. Pregunta # 10 - Encuesta para la prueba beta

Le parece completa la información mostrada en el informe?

8 respuestas



La Figura 118 ‘¿Considera que la aplicación móvil AgroLibreta podría mejorar el control de los costos de sus cultivos?’, con un porcentaje de 62.5% se observa que la mayoría de las respuestas fueron positivas.

Figura 118. Pregunta # 11 - Encuesta para la prueba beta



La Figura 119 corresponde a la pregunta ‘¿Cómo le ha parecido la asignación de costos a un registro fotográfico?’, se observa un 50 a 50% en las respuestas normal y fácil.

Figura 119. Pregunta # 12 - Encuesta para la prueba beta



La Figura 120 representa la pregunta ‘¿Cómo le ha parecido el proceso para registrar un costo?’, en ella se puede observar que con un 62.5% la mayoría de las personas contestaron ‘normal’.

Figura 120. Pregunta # 13 - Encuesta para la prueba beta



En la Figura 121 se observan las respuestas a la pregunta ‘¿Cómo le ha parecido el registro de un cultivo?’, con un 87.5% a la mayoría de las personas les pareció normal

Figura 121. Pregunta #14 - Encuesta para la prueba beta

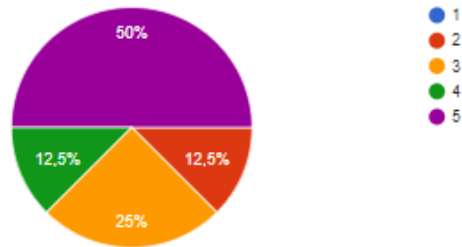


La Figura 122 contiene la respuesta a la afirmación ‘Sé que es el modelo de referencia y para qué sirve’, la respuesta favorable obtuvo un 50%.

Figura 122. Pregunta # 15 - Encuesta para la prueba beta

Seleccione de 1 a 5 si esta de acuerdo o no con la afirmación: Sé que es el modelo de referencia y para que sirve.

8 respuestas

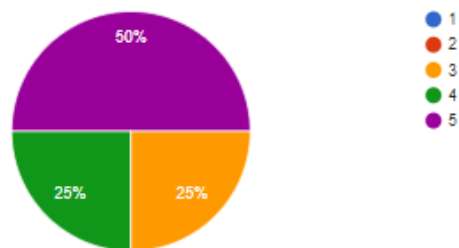


La Figura 123 contiene la respuesta a la afirmación 'La búsqueda de costos es útil y eficaz', la respuesta más favorable obtuvo un 50%.

Figura 123. Pregunta # 16 - Encuesta para la prueba beta

Seleccione de 1 a 5 si esta de acuerdo o no con la afirmación: La búsqueda de costos es útil y eficaz.

8 respuestas

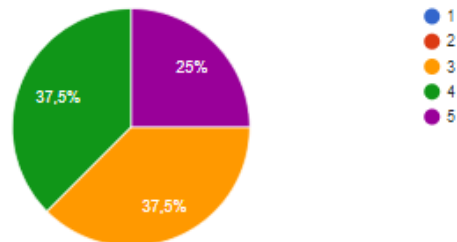


La Figura 124 representa la afirmación 'La aplicación es intuitiva' debieron contestar de 1 a 5 que tanto les parecía, la respuesta con mayor porcentaje de 37.5% fue la número cuatro.

Figura 124. Pregunta # 17 - Encuesta para la prueba beta

Seleccione de 1 a 5 si esta de acuerdo o no con la afirmación: La aplicación es intuitiva.

8 respuestas

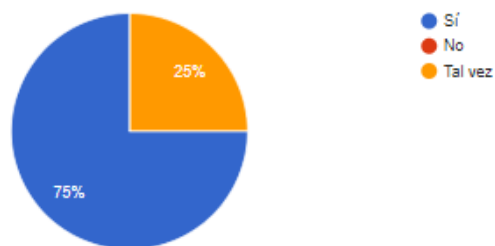


La Figura 125 representa la respuesta a la pregunta ‘¿Recomendaría usar la aplicación móvil AgroLibreta a otros productores agrícolas?’, se observa que un 75% contestaron ‘sí’ y un 25% contestaron ‘tal vez’

Figura 125. Pregunta # 18 - Encuesta para la prueba beta

¿Recomendaría usar la app a otros productores agrícolas?

8 respuestas



La Figura 126 contiene la pregunta ‘¿Considera que la aplicación móvil es una herramienta útil?’, en ella se puede observar que todas las personas contestaron de manera positiva.

Figura 126. Pregunta # 19 - Encuesta para la prueba beta

¿Considera que la app es una herramienta útil?

8 respuestas



● Sí
● No

Apéndice I. Visita a los agricultores

En este apéndice se evidencian algunas fotografías que se tomaron en la visita realizada a la vereda el Aventino municipio de Mutiscua Norte de Santander, se puede observar en la Figura 127, Figura 128 y Figura 129 a los productores que amablemente dieron su autorización para utilizar estas fotografías, estos mismos participaron en las pruebas de la aplicación móvil AgroLibreta.

Figura 127. Experiencia del primer productor



Figura 128. Experiencia de la segunda agricultora

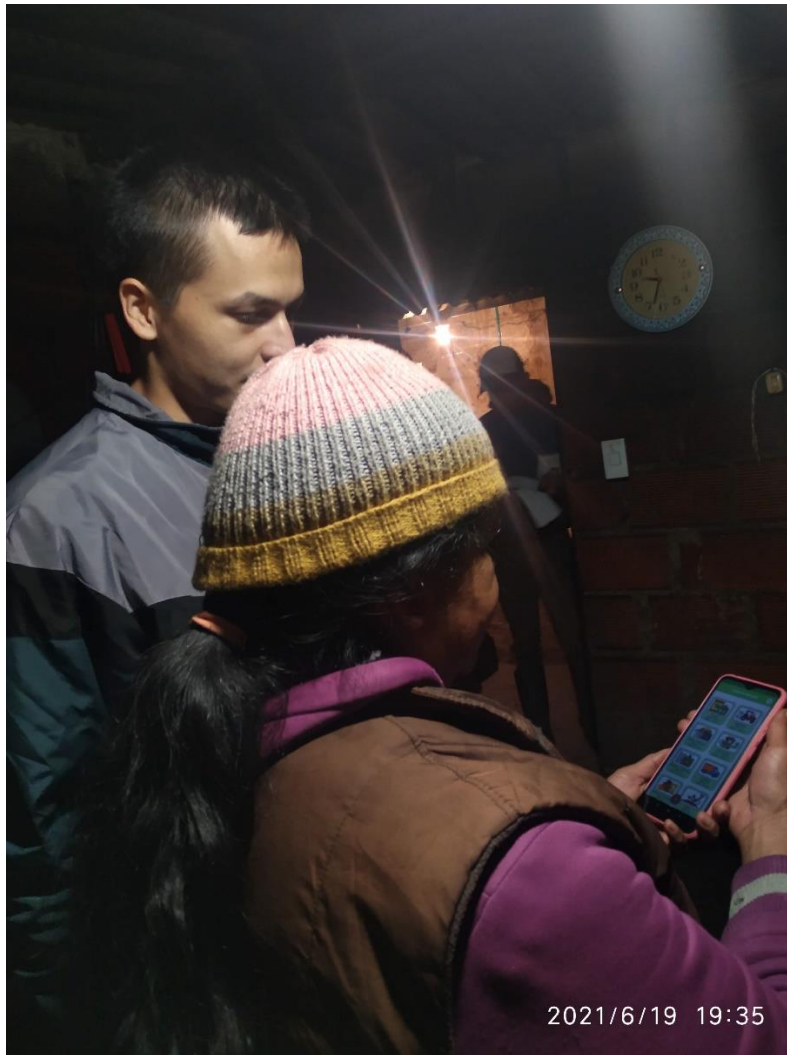


Figura 129. Experiencia del tercer agricultor



En la Figura 130 y Figura 131 se evidencia a los autores de este proyecto en un cultivo de arveja en la finca la Piñuela, vereda el Aventino, municipio de Mutiscua Norte de Santander.

Figura 130. Andres Cuadros Sanabria en un cultivo de arveja



Figura 131. Deisy Rangel Flórez en un cultivo de arveja

