



UNIVERSIDAD
NACIONAL
AUTÓNOMA DE
NICARAGUA,
MANAGUA
UNAN - MANAGUA

Facultad Regional Multidisciplinaria de Chontales

“Cornelio Silva Argüello”

Departamento de Ciencia, Tecnología y Salud

Proyecto de Graduación para optar al título de:

Ingeniero en Sistemas de Información

TÍTULO:

Desarrollo de una aplicación móvil y web utilizando inteligencia artificial, para el manejo integrado en la detección temprana de plagas y enfermedades, basados en los manuales técnicos de granos básicos del Instituto Nacional Tecnológico (INATEC) e Instituto Nacional de Tecnología Agropecuaria (INTA) en las principales siembras de Nicaragua, en la ciudad de Juigalpa-Chontales durante el segundo Semestre del año 2023.

CARRERA:

Ingeniería en Sistemas de Información

AUTORES:

Br. Jeyson Gabriel Jiménez Guevara

Br. José Ramón González Sevilla

Br. Ruddy Josué Mendoza

TUTORES:

M.Sc. Miriam Patricia Téllez Marín

M.Sc. Saira María Urbina Cienfuegos

Juigalpa, Chontales, diciembre 2023

¡A la Libertad por la Universidad!

Autores

Br. Jeyson Gabriel Jiménez Guevara

jeysongabriel.jimenez@gmail.com

Br. José Ramón González Sevilla

jramongs15@gmail.com

Br. Ruddy Josué Mendoza

ruddyjosuemendoza@gmail.com

Tutores

M.Sc. Miriam Patricia Téllez Marín

miriam.tellez@unan.edu.ni

M.Sc. Saira María Urbina Cienfuegos

surbina@unan.edu.ni

Valoración Docente

La tecnología es un recurso importante en la vida diaria de los seres humanos. Con el uso de aplicaciones web o móviles adecuadas es posible comunicarse con facilidad, realizar transacciones y gestionar servicios, esto agiliza procesos complejos y reduce costos significativos. Son incontables las aplicaciones que ofrecen soluciones innovadoras que incluyen herramientas relevantes con Inteligencia Artificial para dar respuesta a las necesidades de los usuarios.

Uno de los problemas actuales que existe en el sector agrícola de Nicaragua, es las plagas y enfermedades que afectan los cultivos de granos básicos, las instituciones pertinentes brindan acompañamiento a los agricultores para asesorarlos y evitar pérdidas cuantiosas en la producción, con base en este contexto se desarrolló una aplicación web y móvil utilizando Inteligencia Artificial para el manejo integrado en la detección temprana evitando la propagación con medidas preventivas y correctivas.

En la documentación presentada del proyecto, es posible observar la aplicación de metodologías y estándares en la búsqueda de un trabajo con un grado de calidad pertinente, además de, la implementación de tecnologías innovadoras en un proceso tradicional, estas evidencias se corresponden con los requisitos establecidos por la UNAN Managua para aceptar este proyecto como defensa de grado, que demuestra el cumplimiento de un aprendizaje integral por parte de los estudiantes.

Por lo antes expuesto, se considera que los estudiantes **Jeyson Gabriel Jiménez Guevara, José Ramón González Sevilla y Ruddy Josué Mendoza**, reúnen los requisitos para la evaluación del jurado examinador, en la defensa de trabajo de grado para optar al título de Ingeniero en Sistemas de Información.

M.Sc. Saira María Urbina Cienfuegos

M.Sc. Miriam Patricia Téllez Marín

Dedicatoria

Dedicamos este proyecto de graduación final a aquellos que han sido pilares fundamentales en nuestro camino. A Dios, fuente de fortaleza y guía, por brindarnos la inspiración y la sabiduría para perseguir nuestros sueños y superar los desafíos. A nuestros padres, cuyo amor incondicional, apoyo constante y sacrificios innumerables nos han permitido llegar hasta aquí. Su fe en nosotros nos ha impulsado a alcanzar nuestras metas. A nuestros amigos, que han sido nuestros cómplices, confidentes y fuentes inagotables de alegría. Han compartido risas y lágrimas, y su amistad ha enriquecido nuestras vidas. A nuestros maestros, quienes han compartido sus conocimientos y han sido faros de orientación en nuestro viaje educativo. Su paciencia y dedicación han sido invaluableles. A nuestros hermanos, cuyo apoyo inquebrantable y vínculo indestructible nos ha brindado fuerza y consuelo en momentos difíciles.

Br. Jeyson Gabriel Jiménez Guevara

Br. José Ramón González Sevilla

Br. Ruddy Josué Mendoza

Agradecimientos

En primer lugar, quiero agradecer a Dios por brindarme la fuerza, la sabiduría y la perseverancia para llevar a cabo este proyecto.

A mi madre, un pilar inquebrantable en mi vida, le debo una deuda eterna de gratitud. Su amor, apoyo incondicional y sacrificio han sido mi fuente de inspiración constante. Este logro es un testimonio de su amor y dedicación.

A mis maestros, a quienes considero guías en mi educación, les agradezco por su conocimiento, orientación y paciencia. A mis hermanos, por ser parte integral de mi vida y brindarme apoyo.

Br. Jeyson Gabriel Jiménez Guevara

Quiero extender mi gratitud a mis padres, cuyo incansable esfuerzo y honesto trabajo diario han sido el motor que me ha impulsado a alcanzar mis metas y convertirme en una persona de éxito. Su apoyo inquebrantable a lo largo de mi vida ha sido fundamental para que yo pueda perseguir mis sueños.

Además, deseo expresar mi agradecimiento a los dedicados docentes que nos han guiado a lo largo de nuestra carrera académica. Han invertido el tiempo necesario para compartir sus valiosos conocimientos y nos han brindado constante motivación. También agradezco profundamente a mi pareja, quien ha sido un apoyo incondicional en este viaje académico, brindándome amor y comprensión en cada etapa de mi formación académica.

Br. José Ramón González Sevilla

Agradezco a Dios por su amor y misericordia, cuidándome todos los días y permitiéndome llegar hasta aquí. A mi madre, quien me dio la vida, por siempre: protegerme, apoyarme, aconsejarme, celebrar cada uno de mis triunfos (por pequeños que estos sean) y ser un pañuelo donde derramar lágrimas cuando he llegado a necesitarlo.

A mis compañeros Jeyson Gabriel Jiménez Guevara y José Ramón González Sevilla, quienes han puesto todo su tiempo y esfuerzo en este trabajo para que alcance la mejor calidad que podamos brindar. Gracias por permitirme ser parte de este gran equipo.

Br. Ruddy Josué Mendoza

Resumen ejecutivo

El proyecto se enfoca en el desarrollo de software para la detección temprana de plagas y enfermedades en cultivos de granos básicos en Nicaragua, cruciales para la seguridad alimentaria y la economía del país, con un enfoque en el frijol, sorgo y maíz, cuya producción alcanzó 8.4 millones de quintales en el ciclo 2022 abasteciendo el consumo nacional con buenos inventarios en los mercados durante todo el año. La aplicación permite a los agricultores e instituciones monitorear y detectar las amenazas localizadas en los cultivos haciendo uso de inteligencia artificial, obtener información, acceder a información climática, hacer uso de mapas interactivos y comunicarse mediante un foro. Se identificaron diversas fuentes de ingresos para el proyecto, entre ellas, los anuncios publicitarios dentro de la aplicación mediante un modelo freemium, el financiamiento de instituciones relacionadas con la agricultura en el país y la venta del modelo de IA. Para el desarrollo del proyecto se ha estimado un presupuesto de C\$186.992,20 córdobas, que incluye contratación de servicios, adquisición de mobiliario y equipo. Se implementó la metodología ágil Scrum utilizando la herramienta web Taiga.io, organizando el proceso en 5 sprints. La aplicación móvil fue desarrollada con Android Studio, Android SDK y Java, junto con Material Design. La aplicación web se desarrolló con Visual Studio Code, JavaScript y Bootstrap. Como base de datos se utilizó Firebase, a través de Cloud Firestore y Realtime Database. Se realizaron pruebas de caja negra haciendo uso de Espresso y pruebas de caja blanca con AndroidJUnit4.

Palabras clave: inteligencia artificial, granos básicos, agricultura, detección, plagas.

Índice

I. Solución	1
II. Diseño o Modelo	2
2.1. Diseño de funcionamiento.....	2
2.2. Diseño de navegación.....	6
2.3. Diseño arquitectónico.....	8
2.3.1. Diagrama nivel 0 móvil	8
2.3.2. Diagrama nivel 1 móvil	9
2.3.3. Diagrama nivel 2 móvil	10
2.3.4. Diagrama nivel 0 web	11
III. Justificación de la tecnología	12
3.1. Selección del framework de desarrollo móvil.....	12
3.2. Selección del entorno de desarrollo web.....	15
3.3. Selección de herramienta de prototipado	18
3.4. Selección de herramienta de diseños vectoriales	19
3.5. Selección de herramienta para entrenamiento de modelo de aprendizaje automático	20
3.5.1. Proceso de funcionamiento.....	22
3.6. Selección de librerías de diseño	27
3.7. Selección del gestor de base de datos.....	28
IV. Justificación de la metodología	30
V. Oportunidad en el mercado	32
VII. Propuesta de valor	33
6.1. Roles.....	34
6.1.1. Agricultor.....	34
6.1.2. Usuario no registrado.....	35
6.1.3. Institución	36
6.1.4. Administrador.....	36
VII. Ciclo del proyecto	37
7.1. Historias de usuario.....	37
7.2. Product Backlog	47
7.3. Product Backlog organizado en Sprints	48
7.4. Sprint Backlog.....	49

7.4.1. Sprint 1	49
7.4.1.1. Entregables Sprint 1.....	52
7.4.1.2. Pruebas de software Sprint 1.....	54
7.4.2. Sprint 2	55
7.4.2.1. Entregables Sprint 2.....	58
7.4.2.2. Pruebas de software Sprint 2.....	60
7.4.3. Sprint 3	61
7.4.3.1. Entregables Sprint 3.....	63
7.4.3.2. Pruebas de software Sprint 3.....	65
7.4.4. Sprint 4	66
7.4.4.1. Entregables Sprint 4.....	68
7.4.4.2. Pruebas de software Sprint 4.....	70
7.4.5. Sprint 5	71
7.4.5.1. Entregables Sprint 5.....	73
7.4.5.2. Pruebas de software Sprint 5.....	74
VIII. Presupuesto	74
IX. Marco lógico.....	76
9.1. Árbol del problema.....	77
9.2. Árbol de objetivos	78
9.3. Árbol de solución V1	79
9.4. Árbol de Solución V2.....	80
9.5. Matriz de marco lógico	81
X. Cronograma de actividades.....	88
XI. Bibliografía.....	90
XII. Anexos.....	93
12.1. Modelo CANVAS.....	94
12.2. Manual técnico	95
12.3. Manual de usuario	114
12.4. Plan de pruebas de software.....	161
12.5. Participación en eventos nacionales.....	244

Índice de tablas

Tabla 1. Comparación de tecnologías de desarrollo móvil.....	12
Tabla 2. Comparación de tecnologías de desarrollo web.....	15
Tabla 3. Comparación de herramientas de prototipado	18
Tabla 4. Comparación de herramientas de diseño vectorial	19
Tabla 5. Comparación de herramienta para modelo de aprendizaje automático	20
Tabla 6. Comparación de librerías de diseño	27
Tabla 7. Comparación de gestores de base de datos	28
Tabla 8. Comparación de metodologías de desarrollo de software	30
Tabla 9. Historia de usuario 1	38
Tabla 10. Historia de usuario 2	38
Tabla 11. Historia de usuario 3	38
Tabla 12. Historia de usuario 4	39
Tabla 13. Historia de usuario 5	39
Tabla 14. Historia de usuario 6	40
Tabla 15. Historia de usuario 7	40
Tabla 16. Historia de usuario 8	41
Tabla 17. Historia de usuario 9	41
Tabla 18. Historia de usuario 10	42
Tabla 19. Historia de usuario 11	42
Tabla 20. Historia de usuario 12	42
Tabla 21. Historia de usuario 13	43
Tabla 22. Historia de usuario 14	43
Tabla 23. Historia de usuario 15	44
Tabla 24. Historia de usuario 16	44
Tabla 25. Historia de usuario 17	45
Tabla 26. Historia de usuario 18	45
Tabla 27. Historia de usuario 19	46
Tabla 28. Historia de usuario 20	46
Tabla 29. Product Backlog.....	47
Tabla 30. Product Backlog organizado en Sprints.....	48
Tabla 31. Sprint 1	49
Tabla 32. Sprint 2.....	55
Tabla 33. Sprint 3	61
Tabla 34. Sprint 4.....	66
Tabla 35. Sprint 5.....	71
Tabla 36. Presupuesto para desarrollo del proyecto	75
Tabla 37. Matriz de marco lógico	81
Tabla 38. Cronograma de actividades.....	88

Índice de figuras

Figura 1. Diseño o modelo móvil usuario agricultor	3
Figura 2. Diseño o modelo móvil de usuario no registrado	4
Figura 3. Diseño o modelo web	5
Figura 4. Diseño de navegación móvil del usuario agricultor	6
Figura 5. Diseño de navegación web	7
Figura 6. Diagrama nivel 0 móvil	8
Figura 7. Diagrama nivel 1 móvil	9
Figura 8. Diagrama nivel 2 móvil	10
Figura 9. Diagrama nivel 0 web	11
Figura 10. Carga de datos en Teachable Machine	22
Figura 11. Entrenamiento en Teachable Machine	23
Figura 12. Precisión por clase en Teachable Machine	24
Figura 13. Predicción con imagen que no se usó en el entrenamiento	25
Figura 14. Predicción con imagen que se usó en el entrenamiento	25
Figura 15. Exportación del modelo en Teachable Machine	26
Figura 16. Burndown chart Sprint 1	51
Figura 17. H00 - Detección de plagas y enfermedades (Entrenamiento de modelo)	52
Figura 18. H00 - Detección de plagas y enfermedades (Resultados de detección)	52
Figura 19. H00 - Detección de plagas y enfermedades (Interfaz de detección)	52
Figura 20. H01 - Información sobre plagas y enfermedades (Lista)	53
Figura 21. H01 - Información sobre plagas y enfermedades (Selección de cultivo)	53
Figura 22. H01 - Información sobre plagas y enfermedades (Información)	53
Figura 23. H01 - Información sobre plagas y enfermedades (Tratamiento)	53
Figura 24. H15 - Plagas y enfermedades detectadas	54
Figura 25. Pruebas de software Sprint 1	54
Figura 26. Burndown chart Sprint 2	58
Figura 27. H03 - Pronóstico del tiempo	58
Figura 28. H14 - Marcadores en el mapa de plagas y enfermedades	59
Figura 29. H16 – Inicio de sesión	59
Figura 30. H08 – Foro agrícola	59
Figura 31. H04 – Registro de plagas y enfermedades	59
Figura 32. Pruebas de software Sprint 2	60
Figura 33. Burndown chart Sprint 3	63
Figura 34. H07 - Mapa de plagas y enfermedades	63
Figura 35. H17 - Dashboard de plagas y enfermedades detectadas	64
Figura 36. H02 - Clima actual	64
Figura 37. H11 - Registro de agricultor	64
Figura 38. Pruebas de software Sprint 3	65
Figura 39. Burndown chart Sprint 4	68
Figura 40. H19 - Manual de introducción	68
Figura 41. H09 - Perfil del agricultor	69

Figura 42. H06 - Mapa de zonas de cultivos de granos básicos	69
Figura 43. H05 - Historial de plagas y enfermedades.....	69
Figura 44. Pruebas de software Sprint 4	70
Figura 45. Burndown chart Sprint 5	72
Figura 46. H12 - Visualización de calificaciones de la aplicación	73
Figura 47. H13 - Cantidad de usuarios registrados.....	73
Figura 48. H18 - Splash Screen	73
Figura 49. H10 - Calificación de la aplicación	73
Figura 50. Pruebas de software Sprint 5	74
Figura 51. Árbol del problema.....	77
Figura 52. Árbol de objetivos	78
Figura 53. Árbol de solución V1	79
Figura 54. Árbol de solución V2	80
Figura 55. Modelo CANVAS	94

I. Solución

El desarrollo de Placuix, una aplicación móvil y web para la detección de plagas y enfermedades en los cultivos de granos básicos en Nicaragua es una iniciativa que revoluciona la agricultura en la región. Según el Ministerio Agropecuario (MAG, 2023), “El cultivo de frijol rojo ha mostrado un crecimiento sostenido durante los últimos años, garantizando el consumo nacional e impulsando el crecimiento de las exportaciones, las cuales en el año 2022 cerraron con 115.8 millones de dólares”. El MAG (2023) afirma que, “La producción nacional del maíz alcanzó 8.4 millones de quintales, representando crecimiento de 1.5%, en comparación a la producción del ciclo anterior, volúmenes producidos por más de 180 mil pequeños productores que cultivan en las diferentes épocas del ciclo”. Flores (2019) plantea que el sorgo es una alternativa y sustituto del maíz, además de representar gran importancia para la alimentación de animales; por ejemplo, concentrados de cerdo y concentrado para la alimentación de ganado. Debido a la relevancia de estos cultivos se hace sumamente necesario aprovechar las últimas tendencias tecnológicas, como la Inteligencia Artificial (IA), para proporcionar a los agricultores una herramienta poderosa para el manejo de sus cultivos de manera más eficiente.

Un aspecto fundamental de esta solución es la detección y gestión de plagas y enfermedades. La Inteligencia Artificial (IA) puede desempeñar un papel crucial al permitir que la aplicación imite la inteligencia humana al procesar imágenes de los cultivos. Esta tecnología avanzada brinda la capacidad de realizar una detección precisa y oportuna, lo que, a su vez, posibilita tomar medidas preventivas y correctivas de manera eficaz.

Para garantizar el éxito de esta iniciativa, se requiere una colaboración cercana con expertos agrícolas y el acceso a datos locales confiable. Adicionalmente, hay riesgos que deben ser abordados cuidadosamente, como la precisión de la detección y la posible resistencia de los agricultores a adoptar nuevas tecnologías; para ello, es esencial brindar capacitación y soporte adecuados, así como fomentar una comunidad en línea donde los agricultores puedan compartir experiencias y obtener ayuda de expertos, funcionalidad que también se incluye en la aplicación.

En cuanto a la demografía y la cobertura, el enfoque inicial sería en las regiones agrícolas clave de Nicaragua, pero con la intención de expandirse gradualmente a nivel nacional. Esto permitiría llegar a una amplia gama de agricultores y promover prácticas más sostenibles en toda la industria agrícola del país.

II. Diseño o Modelo

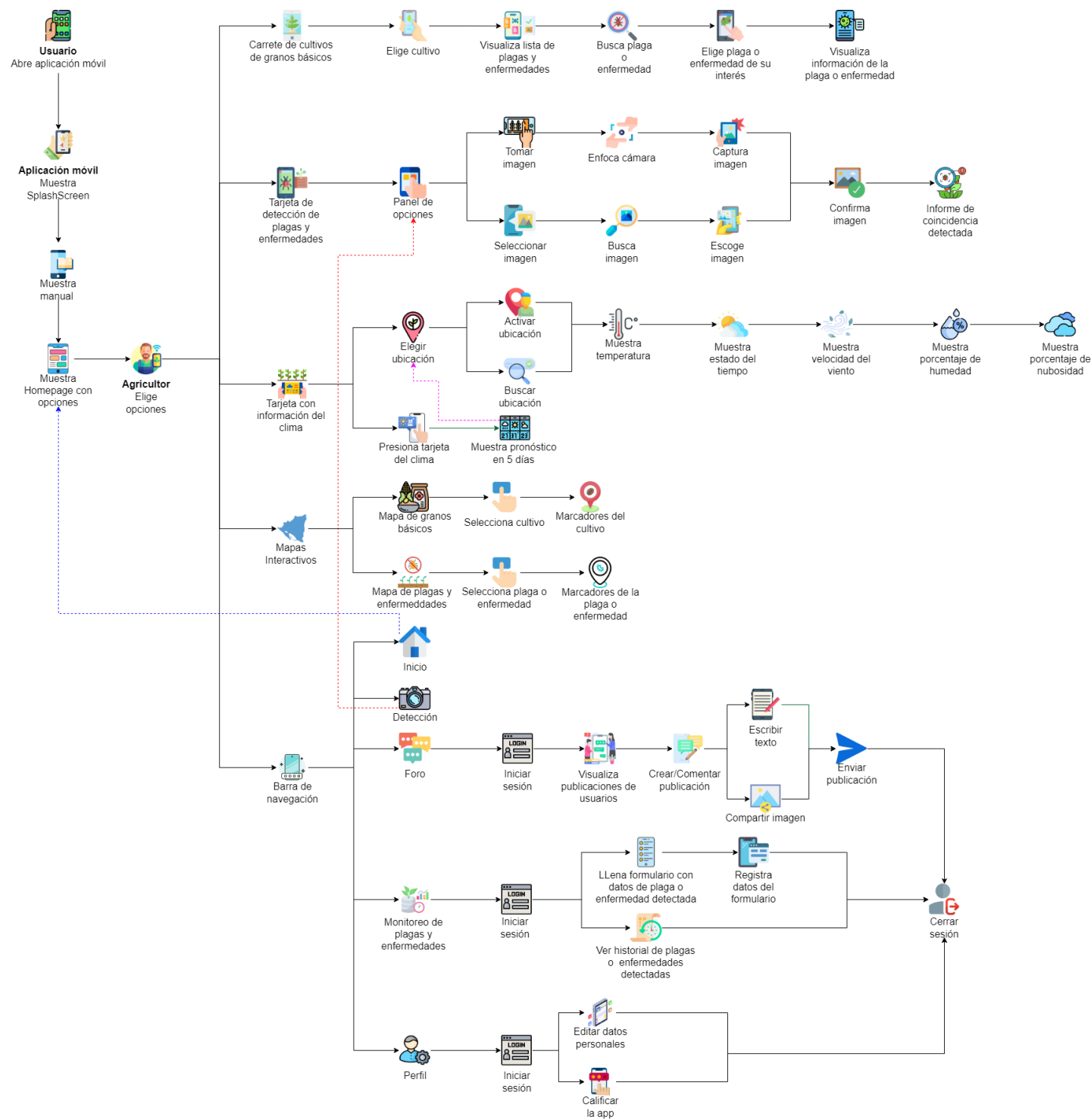
A continuación, se muestran representaciones gráficas del funcionamiento que tendrá tanto la aplicación móvil como la aplicación web desarrollad. Placuix maneja diferentes tipos de usuarios de los cuales se derivan una serie de opciones que representarán las funcionalidades principales de la aplicación. Los cuatro tipos de usuarios están dados por el agricultor, el usuario no registrado, el administrador y las instituciones.

2.1. Diseño de funcionamiento

A continuación, se muestra el funcionamiento de la aplicación móvil y web por medio de diferentes figuras que representan la interacción entre el software y los distintos usuarios.

Figura 1

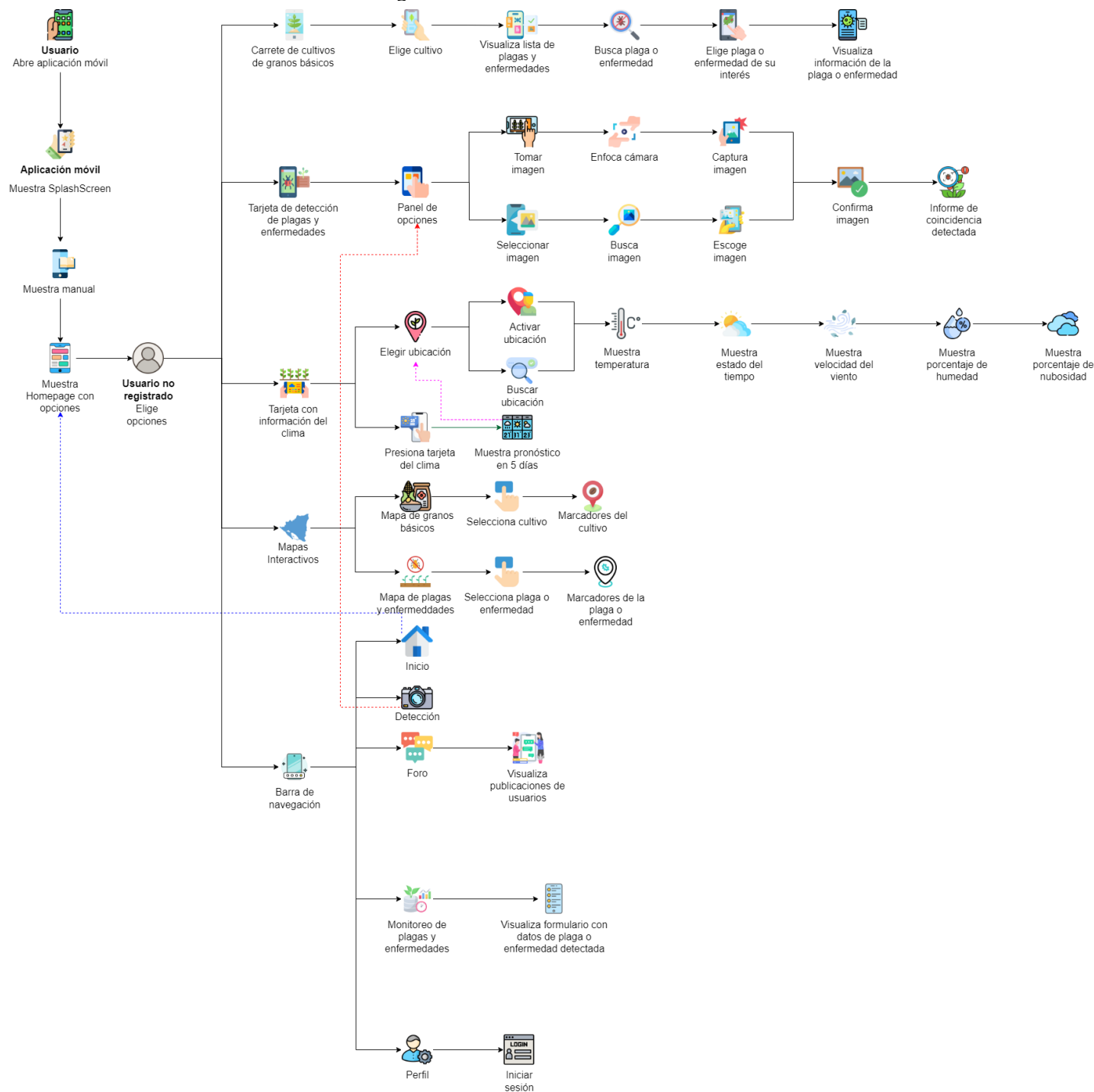
Diseño o modelo móvil de usuario agricultor



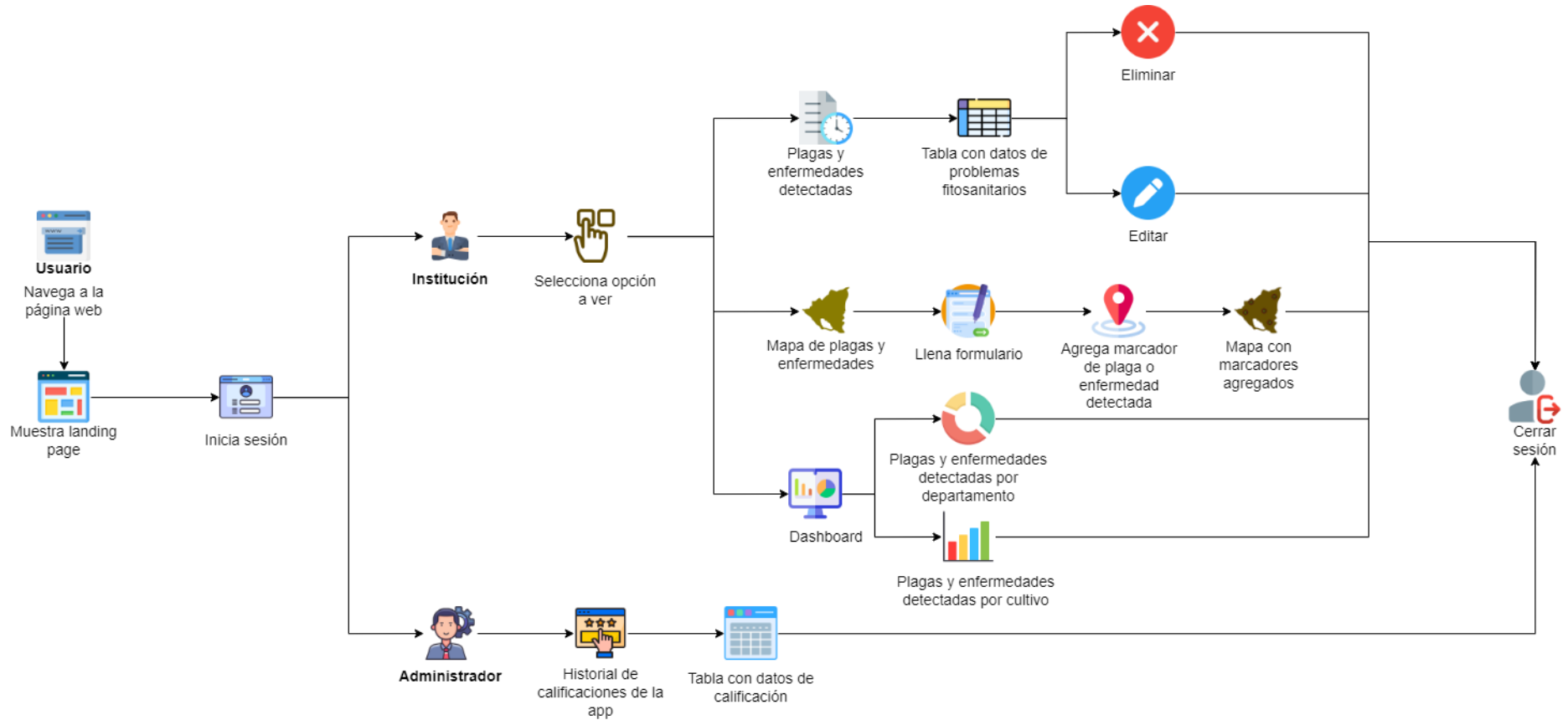
Nota. Funcionamiento de la aplicación móvil con el usuario agricultor.

Figura 2

Diseño o modelo móvil de usuario no registrado



Nota. Funcionamiento de la aplicación móvil con el usuario no registrado.

Figura 3*Diseño o modelo web*

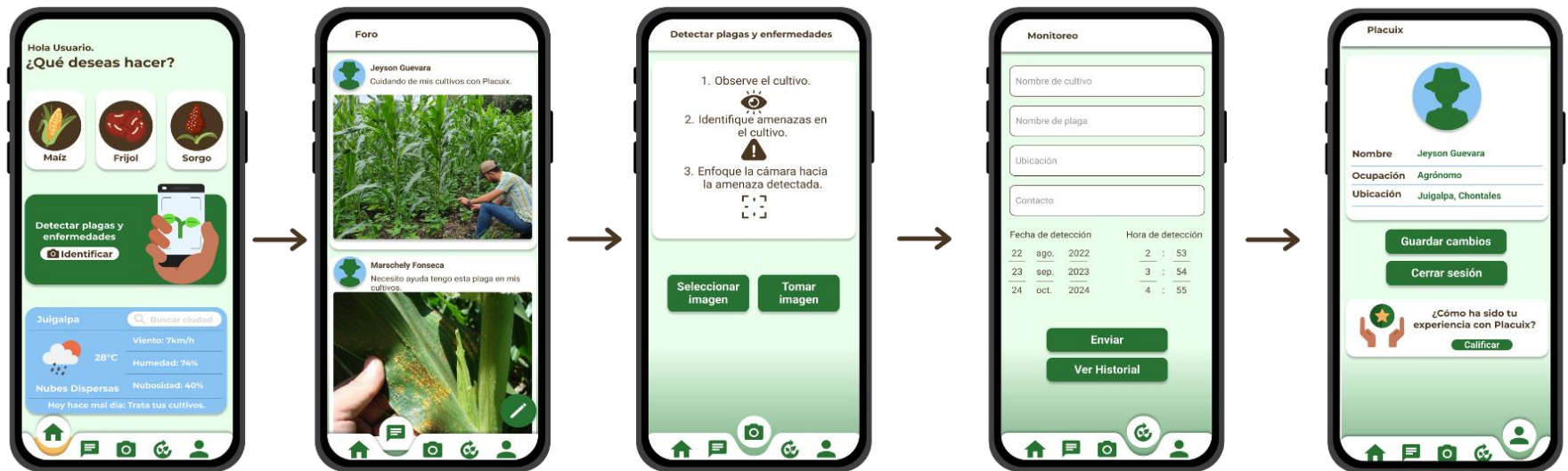
Nota. Funcionamiento de la aplicación web con los usuarios institución y administrador.

2.2. Diseño de navegación

La aplicación móvil PlacuiX presenta una interfaz de usuario simple y bien organizada. En la pantalla principal, los usuarios ven una pantalla con diferentes tarjetas de tamaño considerable para su selección. Pueden acceder a otras secciones de la aplicación, como Foro, Detección, Monitoreo y Perfil desde un menú de navegación en la parte inferior. La navegación es intuitiva y los usuarios pueden moverse fácilmente entre las secciones sin problemas.

Figura 4

Diseño de navegación móvil del usuario agricultor

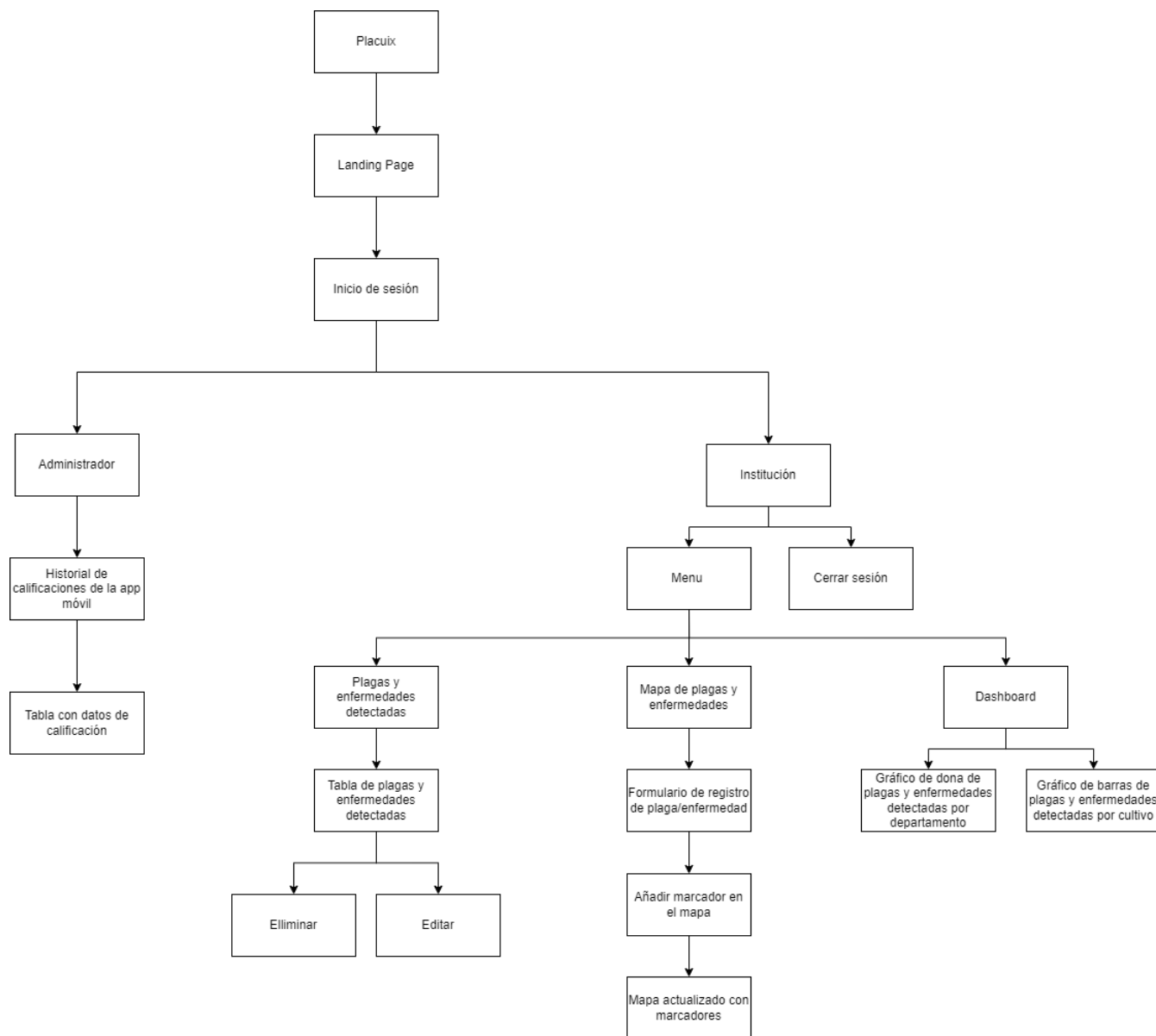


Nota. La aplicación móvil tiene una navegación fácil e intuitiva entre sus diferentes pantallas por medio de una barra en la parte inferior como se muestra en la figura 4, con el rol de usuario agricultor.

La aplicación web también presenta navegación simple y bien organizada para brindar una mejor experiencia a los usuarios.

Figura 5

Diseño de navegación web



Nota. La aplicación web cuenta con una barra de navegación por medio de la cual pueden acceder los usuarios administrador e institución.

2.3. Diseño arquitectónico

A continuación, se define la estructura y las responsabilidades de los componentes que conforman la arquitectura del proyecto de software.

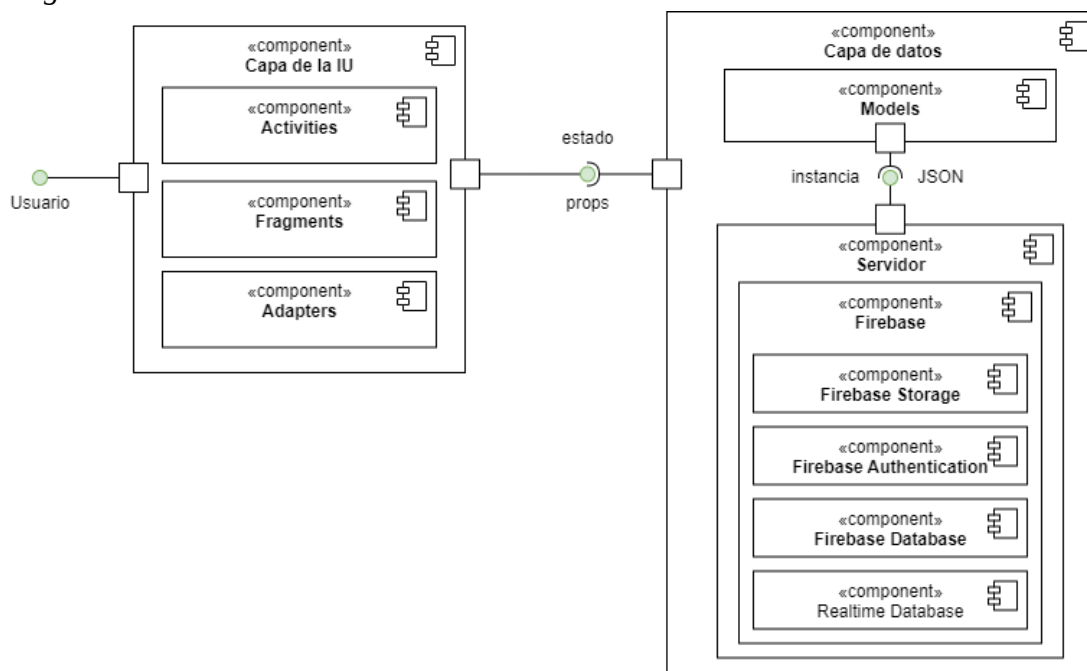
Una de las arquitecturas recomendadas en Android es el patrón de arquitectura en capas y es bajo la cual se desarrolla la aplicación móvil. Esta comparte similitudes con el patrón arquitectónico Modelo-Vista-Controlador (MVC), ya que ambos se basan en la separación de responsabilidades en diferentes componentes para lograr un código más organizado, pero está pensada en un flujo unidireccional de datos (UDF) en todas las capas de la app. Según Google Developers (2023), los datos viajan desde la fuente de datos hacia la IU, los eventos del usuario fluyen desde la IU hasta la fuente de información donde los datos se modifican y se exponen en un tipo inmutable.

2.3.1. Diagrama nivel 0 móvil

La figura 6 representa los módulos principales en los que está dividida la aplicación móvil, los cuales son: la capa de presentación que contiene activities, fragments y adapters,

Figura 6

Diagrama nivel 0 móvil



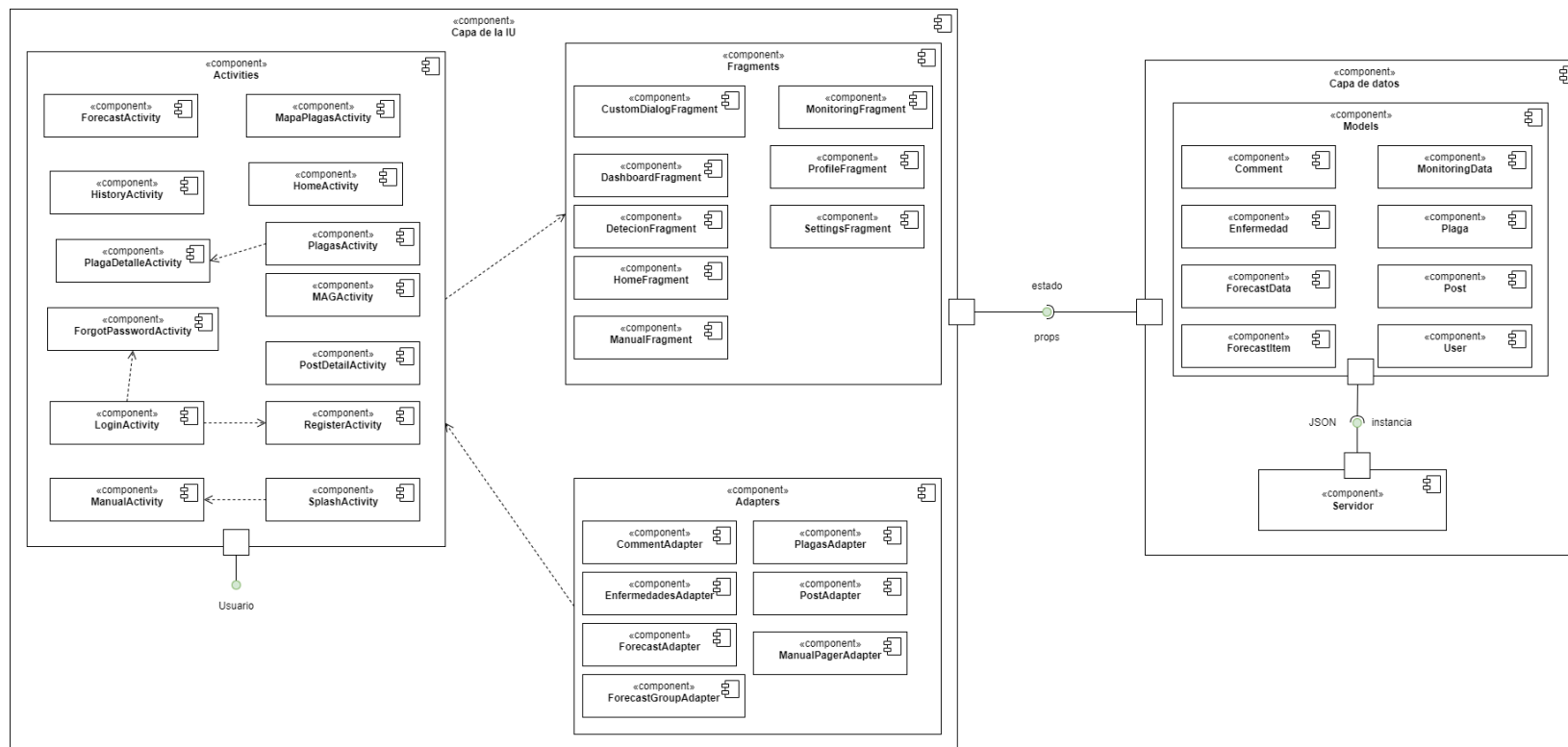
Nota. Los componentes envían y reciben props de la capa de datos que contiene models que a su vez obtiene información del servidor de Firebase.

2.3.2. Diagrama nivel 1 móvil

La figura 7 refleja los componentes y las relaciones de cada módulo de la aplicación móvil, definiendo los subcomponentes de Activities, Fragments, Adapters y Models.

Figura 7

Diagrama nivel 1 móvil



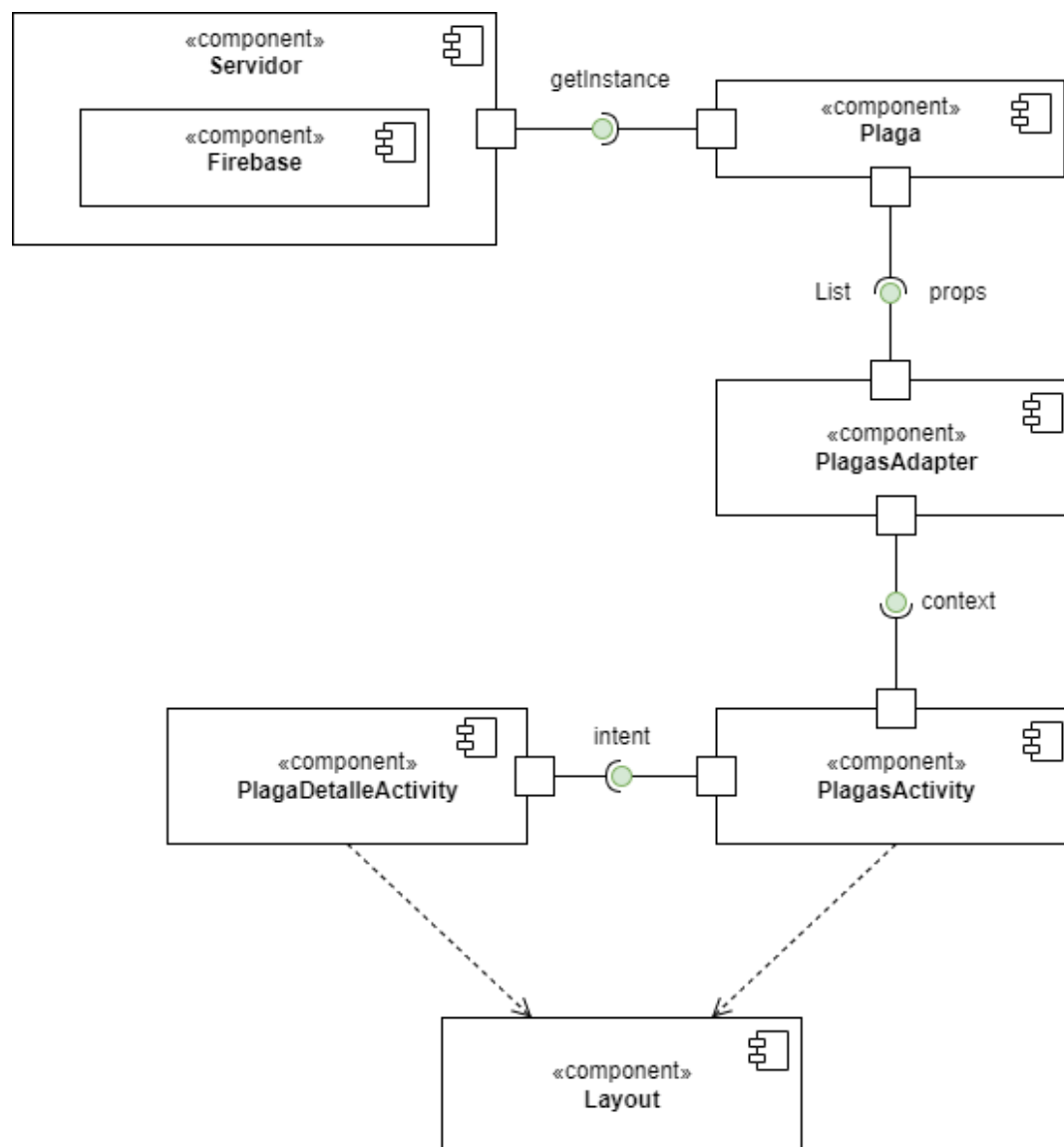
Nota. Las flechas indican la relación que tiene cada módulo de la aplicación entre sus componentes.

2.3.3. Diagrama nivel 2 móvil

Se detalla el flujo de datos que es unidireccional y cómo se implementan los llamados Layout, que se encargan de mostrar los Activities, desde la obtención de datos del servidor de Firebase pasando por los Model y Adapters.

Figura 8

Diagrama nivel 2 móvil



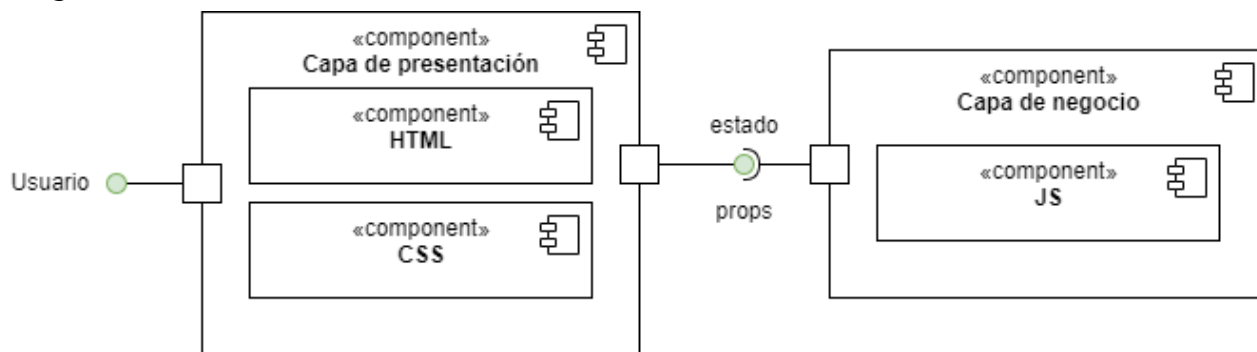
Nota. La figura representa la interacción de la aplicación móvil y su flujo interno, según la arquitectura en capas.

2.3.4. Diagrama nivel 0 web

La figura 9 refleja cómo está dividida la estructura de la aplicación web, en donde se muestra la capa de presentación que se encarga de la interfaz de usuario, dividiendo los archivos HTML y CSS. Por otro lado, está la capa de la lógica de la aplicación, en donde se coloca la lógica de negocio, organizando los archivos JavaScript para informar a la capa de presentación cómo se comportará mediante propiedades y datos. Es importante destacar que, aunque las arquitecturas en capas y la arquitectura cliente-servidor son conceptos diferentes, se combinan en la aplicación. En el presente proyecto, la aplicación web tiene una arquitectura en capas donde la capa de presentación (cliente) se comunica con la capa de lógica de aplicación (servidor) a través de solicitudes HTTP.

Figura 9

Diagrama nivel 0 web



Nota. La figura representa cómo la capa de presentación actúa por medio del usuario para comunicarse con el servidor, obteniendo datos y servicios.

III. Justificación de la tecnología

A continuación, se presenta la justificación de las tecnologías elegidas para el proyecto de software, tras una evaluación exhaustiva que incluye comparaciones entre diferentes opciones. Esta elección se basa en la alineación con las necesidades específicas de la aplicación móvil y web, la experiencia del equipo y la eficacia demostrada de estas tecnologías en proyectos similares.

3.1. Selección del framework de desarrollo móvil

Tabla 1

Comparación de tecnologías de desarrollo móvil

Cuadro comparativo de tecnologías para el desarrollo de aplicaciones móviles								
Nombre	Licencia	Conocimientos técnicos requeridos			Ponderación		Descripción	Ventajas
		Lenguaje	Experiencia	Curva de aprendizaje	Experiencia	Curva de aprendizaje		
Ionic	Licencia MIT	JavaScript	80%	20%	45%	30%	Es un framework que proporciona conjunto de herramientas de interfaz de usuario de código abierto para crear aplicaciones móviles de alto rendimiento y calidad utilizando tecnologías web (HTML, CSS y JavaScript) con	• Componentes intuitivos de interfaz de usuario • Plazos de desarrollo de proyectos más rápidos • Uniformidad de la interfaz de usuario • Optimización del rendimiento • Desarrollo multiplataforma
		TypeScript	10%	40%				

							integraciones para marcos populares. ^a
React Native	Licencia MIT	JavaScript	80%	20%	61.66%	30%	Es un framework que combina las mejores partes del desarrollo nativo con React, la mejor biblioteca JavaScript de su clase para crear interfaces de usuario. Se puede usar en proyectos de Android e iOS o crear una aplicación completamente nueva desde cero. ^b • Recarga en vivo para ver los últimos cambios en el código • Comunidad de apoyo • Mejor rendimiento de la aplicación • Minimiza el tiempo de desarrollo • Soporta plugins de terceros • Interfaz de usuario sencilla • Biblioteca fácil de usar • Framework para Android y iOS
		JSX	30%	50%			
		CSS	75%	20%			
Flutter	Código abierto	Dart	0%	95%	0%	95%	Es un framework de código abierto de Google para crear aplicaciones hermosas, compiladas de forma nativa para móviles, web y escritorio a partir de una sola base de código. ^c • Aplicaciones de alto rendimiento • Capacidad para previsualizar los cambios • Fácil de aprender • Diseños que consumen menos recursos • Económico • Comunidad y documentación

					• Mejora la productividad	
Android SDK	Licencia Apache, versión 2.0 freeware	Java	90%	10%	Android SDK es el entorno de desarrollo integrado (IDE) oficial que se usa en el desarrollo de apps para Android. Basado en el potente editor de código y las herramientas para desarrolladores de IntelliJ IDEA. ^d	• Un sistema de compilación flexible basado en Gradle • Un emulador rápido y cargado de funciones • Un entorno unificado • Ediciones en vivo para actualizar en tiempo real • Integración con GitHub y plantillas de código • Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones.
				65%		
		Kotlin	40%	20%		

Nota. ^a Ionic (2023). ^b React Native (2023). ^c Flutter (2023). ^d Android Developers (2023)

Habiendo evaluado las tecnologías, se ha seleccionado Android SDK para el desarrollo móvil de la aplicación, ya que el equipo ha adquirido experiencia con el desarrollo de proyectos anteriores, haciendo que la curva de aprendizaje sea relativamente corta y el proceso de desarrollo sea más eficiente.

3.2. Selección del entorno de desarrollo web

Tabla 2

Comparación de tecnologías de desarrollo web

Cuadro comparativo de tecnologías para el desarrollo de aplicaciones web								
Nombre	Licencia	Conocimientos técnicos requeridos			Ponderación		Descripción	Ventajas
		Lenguaje	Experiencia	Curva de aprendizaje	Experiencia	Curva de aprendizaje		
Visual Studio Code	Licencia MIT	JavaScript	80%	20%	55%	27.5%	Visual Studio Code es un editor de código fuente liviano pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para diferentes frameworks y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución. ^a	• Es gratuito y de código abierto. • Es altamente personalizable. • Es compatible con varios lenguajes de programación. • Tiene una gran comunidad. • Es compatible con varias plataformas. • Contiene IntelliSense.
		Java	90%	10%				
		Python	0%	50%				
		PHP	50%	30%				

Visual Studio	Licencia Propietaria	C#	75%	60%	53.75%	42.5%	Es una herramienta de desarrollo eficaz que permite completar todo el ciclo de desarrollo en un solo lugar. Es un entorno de desarrollo integrado (IDE) completo que puede usar para escribir, editar, depurar y compilar el código y, luego, implementar la aplicación. ^b • Crea aplicaciones para cualquier plataforma. • Diseñadores, editores, depuradores y generadores de perfiles en una sola herramienta. • Ofrece miles de extensiones. • Permite una gran cantidad de lenguajes de programación.
		Java	90%	10%			
		C++	50%	50%			
		Python	0%	50%			

NetBeans	Licencia Apache. Versión 2.0	Java	90%	10%	30%	46.66%	Apache NetBeans es mucho más que un editor de texto. Destaca el código fuente sintáctica y semánticamente, le permite refactorizar el código fácilmente, con una variedad de herramientas útiles y potentes. ^c • Código abierto • Multiplataforma • Manejo automático de la memoria. • Multilenguaje.
		JSP	0%	60%			
		Haskell	0%	70%			

Nota. ^a Visual Studio Code Team (2023). ^b Visual Studio Colaboradores (2023). ^c Neil et al. (2023)

Habiendo evaluado las tecnologías, se ha seleccionado Visual Studio Code como editor de texto para el desarrollo de la aplicación web, ya que el equipo ha adquirido experiencia con el desarrollo de proyectos anteriores, haciendo que la curva de aprendizaje sea relativamente corta y el proceso de desarrollo sea más eficiente.

3.3. Selección de herramienta de prototipado

Tabla 3

Comparación de herramientas de prototipado

	Figma	Sketch	Adobe XD
Plataforma	Aplicación basada en navegador.	Aplicación de escritorio y navegador.	Aplicación de escritorio y móvil.
Sistema operativo	MacOS, Windows, Linux.	MacOS.	MacOS, Windows, iOS, Android.
Colaboración	Colaboración en tiempo real.	Colaboración en tiempo real para suscriptores Sketch de MacOS.	Colaboración en tiempo real en proyectos sincronizados con la nube.
Cómo comenzar	Lecciones y ejercicios de diseño.	Documentación.	Tutoriales en video y guías paso a paso.
Complementos	Biblioteca creciente disponible en la aplicación.	Gran biblioteca, descargada externamente.	Biblioteca creciente disponible en la aplicación.
Manipulación de vectores	Redes vectoriales.	Ruta de vectores.	Ruta de vectores.
Precio	Versión inicial gratuita o \$12 USD / mes por editor.	Prueba gratuita de 30 días, luego \$10 USD / mes por editor.	Prueba gratuita de 7 días, luego \$52.99 USD / mes.
Curva de aprendizaje	15%	25%	10%

Nota. Información tomada de Coursera (2023).

Como herramientas de prototipado fueron seleccionadas Adobe XD y Figma debido a su disponibilidad en múltiples plataformas, facilidad de colaboración en la nube, recursos de aprendizaje sólidos y curvas de aprendizaje más suaves en comparación con Sketch. Además, sus precios y la creciente biblioteca de complementos en las aplicaciones hacen que sean opciones atractivas para diseñar prototipos de aplicaciones móviles y web.

3.4. Selección de herramienta de diseños vectoriales

Tabla 4

Comparación de herramientas de diseño vectorial

Tipo de trabajo	Adobe Illustrator	Adobe Photoshop	Adobe InDesign
Editar fotos		✓	
Diseño web		✓	
Creación de arte digital	✓	✓	
Wireframing	✓		
Creación de logos	✓		✓
Creación de tarjetas de presentación o volantes	✓		✓
Documentos multipágina			✓
Precio	Prueba gratuita de 7 días, luego \$52.99 USD / mes.		
Curva de aprendizaje	15%	40%	60%

Nota. Información tomada de Souza (2022).

Fue seleccionado Adobe Illustrator por destacarse en la creación de arte digital, diseño de logos, tarjetas de presentación, volantes y algunas tareas de diseño web y móvil, además de tener una curva de aprendizaje baja, lo que lo hace más fácil de aprender en comparación con Adobe Photoshop y Adobe InDesign.

3.5. Selección de herramienta para entrenamiento de modelo de aprendizaje automático

Tabla 5

Comparación de herramienta para modelo de aprendizaje automático

Características	Teachable Machine	TensorFlow	Azure Machine Learning
Código abierto	No	Sí	No
Interfaz visual	Sí	No	Sí
Experiencia basada en código	No	Sí	Sí
Bibliotecas de ciencia de datos	No	Sí (Tensorflow, scikit-learning)	Sí (Python)
Entrenamiento de modelos	Sí	Sí	Sí
Implementación de modelos	Sí	Sí	Sí
Curva de aprendizaje	10%	80%	60%

Nota. Información tomada de Google Developers (2021) y Google (2019).

Fue seleccionado Teachable Machine por su enfoque en la facilidad de uso a través de una interfaz visual y una curva de aprendizaje relativamente baja, lo que lo hace adecuado para usuarios sin experiencia en creación de modelos de aprendizaje automático. TensorFlow y Azure Machine Learning son opciones más robustas que requieren alta experiencia en IA y ofrecen más flexibilidad, especialmente en entornos empresariales o de investigación, por lo que no son tan adecuadas según los requerimientos del presente proyecto.

Google Teachable Machine es una herramienta de aprendizaje automático que permite a los usuarios entrenar a un ordenador para que reconozca imágenes, sonidos y posturas (Google, 2019). Las tecnologías que utiliza son las siguientes:

Redes Neuronales: Según Ramos (2023), son un tipo de aprendizaje automático que intenta imitar el funcionamiento del cerebro humano. Las redes neuronales consisten en capas de neuronas, donde cada neurona produce un solo valor basado en sus entradas, y estos valores se pasan a través de la red para producir una salida.

Teachable Machine funciona específicamente con **Redes Neuronales Convolucionales (CNN)**, que desde el punto de vista de IBM (2023), proporcionan un enfoque más escalable para las tareas de clasificación de imágenes y reconocimiento de objetos, aprovechando principios del álgebra lineal, específicamente la multiplicación de matrices, para identificar patrones dentro de una imagen. Sin embargo, pueden ser muy exigentes desde el punto de vista informático, ya que requieren unidades de procesamiento gráfico (GPU) para entrenar los modelos.

Aprendizaje Supervisado: Teachable Machine utiliza aprendizaje supervisado, que implica entrenar a un modelo utilizando datos etiquetados que incluyen entradas junto con las respuestas correctas. Ferrer (2023), afirma que “Su principal objetivo es que el modelo aprenda a mapear las entradas a las salidas correctas y pueda hacer predicciones precisas sobre datos no etiquetados”. En este caso, los datos etiquetados son las imágenes o sonidos que se suben y agrupan en diferentes clases.

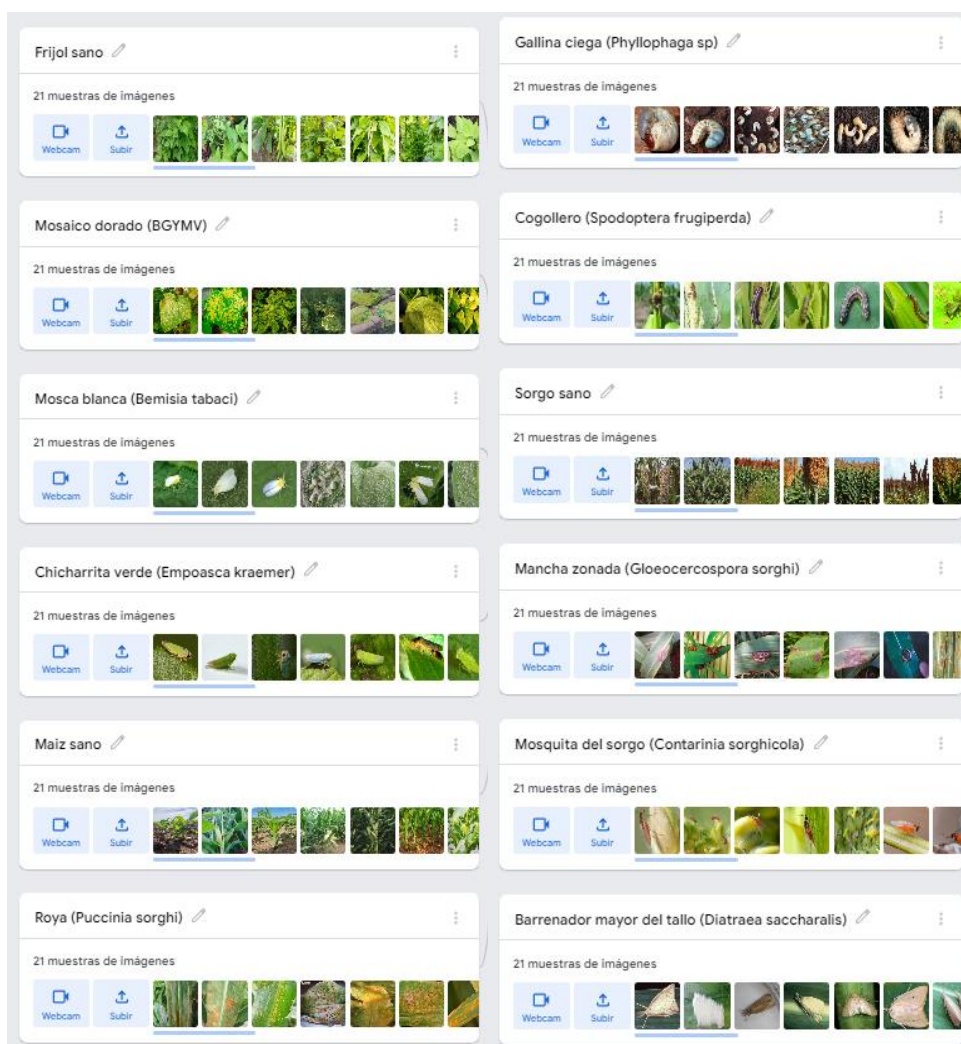
Backpropagation: Este es un método utilizado para entrenar redes neuronales en el aprendizaje supervisado. La Universidad en Internet (UNIR, 2023), afirma que después de cada paso hacia adelante a través de una red, backpropagation realiza un paso hacia atrás mientras ajusta los parámetros del modelo (pesos y sesgos), para minimizar los errores y mejorar en el proceso de aprendizaje y precisión de las predicciones.

3.5.1. Proceso de funcionamiento

Carga de datos: Se suben las imágenes a Teachable Machine, agrupándolas en diferentes clases según lo que se quiere que la máquina aprenda, en este caso, plagas y enfermedades de los cultivos del maíz, frijol y sorgo. Se crearon 12 clases, a las cuales se agregaron 21 muestras de imágenes. Es importante que todas las clases tengan la misma cantidad de imágenes, ya que en caso contrario el porcentaje de precisión se vería afectado mostrando resultados incorrectos. Al cargar las imágenes, la herramienta les hace un pre procesamiento dejándolas cuadradas con un tamaño de 224x224px.

Figura 10

Carga de datos en Teachable Machine



Nota. Figura de clases generadas por teachablemachine.withgoogle.com

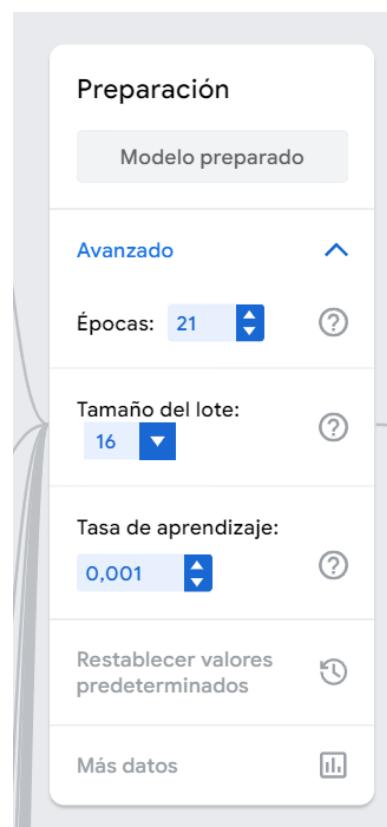
Entrenamiento: Una vez que se han subido los datos, comienza el entrenamiento. Durante este proceso, la red neuronal de Teachable Machine intenta aprender a distinguir entre las diferentes clases de datos que se han subido. Esto se hace ajustando los pesos y sesgos de la red para minimizar el error entre las predicciones de la red y las etiquetas reales de los datos.

Se tiene la opción de acceder a configuración avanzada y modificar parámetros como épocas, tamaño del lote y tasa de aprendizaje, que según Google (2019) son lo siguiente:

- **Épocas:** Cuando el modelo de preparación ha procesado todas las muestras del conjunto de datos de preparación una vez, se dice que se ha completado una época. En el modelo entrenado, se establecen 21 épocas, por lo que procesará 21 veces todo el conjunto de datos de preparación. Por lo general, cuanto mayor es el número, mejor aprenderá el modelo a predecir los datos.
- **Tamaño del lote:** Un lote es un conjunto de muestras que se usan en una iteración de preparación. En el modelo actual se tienen 252 imágenes y un tamaño de lote de 16. Esto significa que las 252 imágenes se dividirán en lotes de 16, lo que da como resultado 15 lotes. Cuando el modelo complete los 5 lotes, completará una época.
- Brownlee (2020) señala que, la **tasa de aprendizaje** es un hiperparámetro que controla cuánto cambiar el modelo en respuesta al error estimado cada vez que se actualizan las ponderaciones del modelo. Elegir la tasa de aprendizaje es un desafío, ya que un valor demasiado pequeño puede resultar en un largo proceso de entrenamiento que podría atascarse, mientras que un valor demasiado grande puede resultar en el aprendizaje de un conjunto subóptimo de pesos demasiado rápido o un proceso de entrenamiento inestable.

Figura 11

Entrenamiento en Teachable Machine



Nota. Figura tomada de teachablemachine.withgoogle.com

Validación: Durante el entrenamiento, una parte de los datos se reserva para la validación. Estos datos de validación se utilizan para evaluar qué tan bien está aprendiendo la red y para ajustar los parámetros de la red durante el entrenamiento. Google (2019) afirma que utiliza un 85% de los ejemplos que se suministra para el entrenamiento, esto para clasificar correctamente nuevos datos en las clases creadas. El otro 15% se utilizan para las muestras de prueba, las cuales nunca son usadas para entrenar el modelo y una vez que el modelo fue entrenado estas se utilizan para verificar como se desempeña el mismo.

Figura 12

Precisión por clase en Teachable Machine

Precisión por clase ?		
CLASS	ACCURACY	# SAMPLES
Frijol sano	1.00	4
Mosaico dorado (BG...	1.00	4
Mosca blanca (Bemi...	1.00	4
Chicharrita verde ...	1.00	4
Maiz sano	1.00	4
Roya (Puccinia sor...	1.00	4
Gallina ciega (Phy...	1.00	4
Cogollero (Spodopt...	1.00	4
Sorgo sano	1.00	4
Mancha zonada (Glo...	1.00	4
Mosquita del sorgo...	1.00	4
Barrenador mayor d...	1.00	4

Nota. Figura generada por teachablemachine.withgoogle.com

En la figura 12 se puede observar que como resultado de los parámetros establecidos en la etapa de entrenamiento se obtuvo que la precisión o exactitud por clase fue muy buena para las doce clases, todas con 1.00 de precisión.

Prueba: Una vez que la red ha sido entrenada, se puede probarla utilizando nuevos datos que no fueron utilizados durante el entrenamiento. Esto permite ver qué tan bien la red ha aprendido a generalizar a partir de los datos de entrenamiento.

Figura 13

Predicción con imagen que no se usó en el entrenamiento

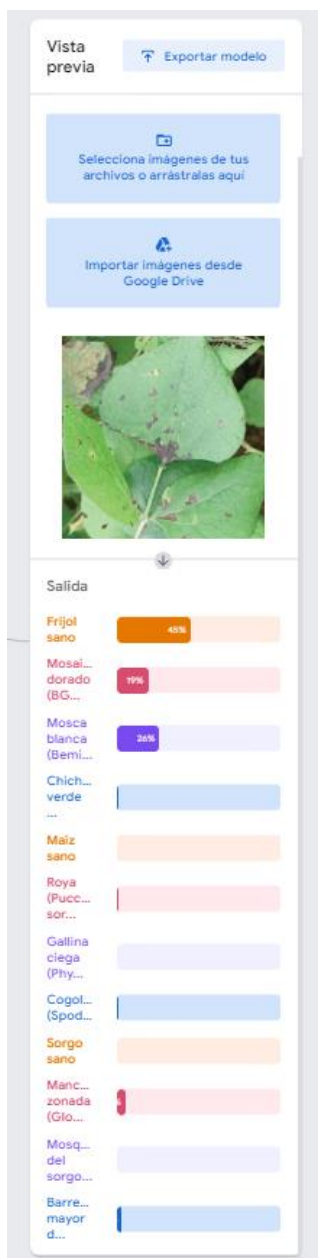


Figura 14

Predicción con imagen que se usó en el entrenamiento



Nota. Figuras generadas por teachablemachine.withgoogle.com

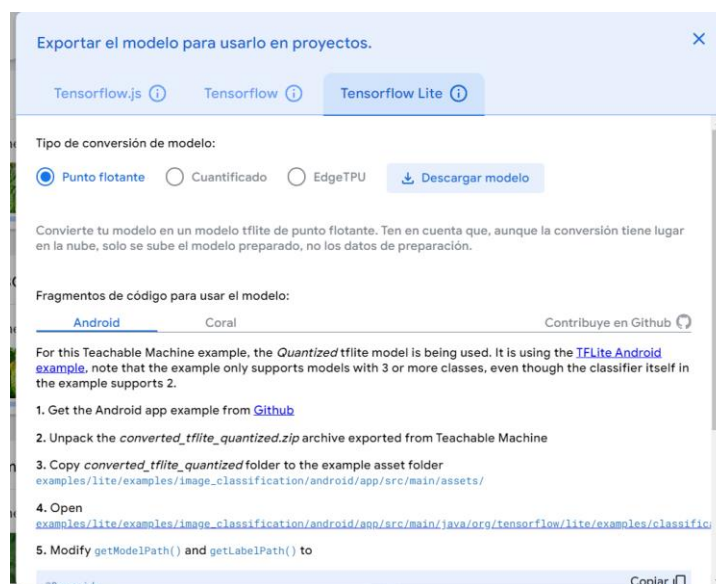
Una vez entrenado el modelo y se conocen los resultados del entrenamiento, se sabe que al momento de hacer una predicción con una imagen que no se utilizó en el entrenamiento, el porcentaje de precisión del modelo no será muy alto y no sabrá a qué clase pertenece con exactitud. En la Figura 13 se puede ver como el modelo no está seguro de a qué clase pertenece la imagen que se ingresa como prueba, ya que para la clase Frijol sano da una precisión del 45%, para la clase Mosaico dorado una precisión del 19% y para la clase Mosca blanca de un 26%.

Sin embargo, en la Figura 14 se puede ver como el modelo arroja para la imagen de prueba un valor de 100% de precisión en la clase Roya, debido a que sí fue entrenado con imágenes relacionadas a dicha clase.

Exportación: Finalmente, una vez que se está satisfecho con el rendimiento del modelo, se exporta para usarlo en el desarrollo del proyecto. Teachable Machine permite exportar el modelo en varios formatos, incluyendo TensorFlow Lite, lo que permite utilizar el modelo en una aplicación móvil.

Figura 15

Exportación del modelo en Teachable Machine



Nota. Figura generada por teachablemachine.withgoogle.com

Google Developers (2021) afirma que, “TensorFlow Lite es un conjunto de herramientas que ayuda a los desarrolladores a ejecutar sus modelos en dispositivos incorporados, móviles o de IoT, y les permite implementar el aprendizaje automático integrado en el dispositivo.”

3.6. Selección de librerías de diseño

Tabla 6

Comparación de librerías de diseño

	Bootstrap	Material Design	Tailwind
Descripción	Bootstrap es la biblioteca HTML, CSS y JS más popular del mundo. Es una herramienta potente, extensible y repleta de características para el desarrollo frontend. Permite construir sitios rápidos y receptivos, personalizar con Sass, utilizar un sistema de cuadrícula preconstruido y componentes, y dar vida a los proyectos con potentes plugins de JavaScript. ^a	Material Design es un sistema adaptable de directrices, componentes y herramientas que respaldan las mejores prácticas de diseño de interfaz de usuario. Respaldado por código abierto, Material Design agiliza la colaboración entre diseñadores y desarrolladores, y ayuda a los equipos a crear rápidamente productos hermosos. ^b	Tailwind CSS es un marco de CSS de utilidad primero para construir rápidamente sitios web modernos sin salir de tu HTML. Está repleto de clases como flex, pt-4, text-center y rotate-90 que se pueden componer para construir cualquier diseño, directamente en tu marcado. ^c
Ventajas	• Fácil de usar • Diseño adaptable • Integración con librerías JavaScript • Uso de Less • Documentación completa	• Innovación y clasicismo • Experiencia unificada • Basado en la realidad táctil • Componentes predefinidos • Influenciado por el diseño de impresión • Renueva la interfaz gráfica de usuario Android	• Flujo de trabajo rápido • Fácil de entender y utilizar • Muy personalizable y escalable • Bien optimizado • Buena documentación oficial
Desventajas	• Integración a medio proyecto • Cambiar de versión • No es ligero • Adaptar diseño a un grid de 12 columnas • Sitios web se ven muy similares	• Complejidad • Limitaciones del contenido • Crecimiento de sombras y elevación	• Marcado puede parecer desorganizado • No es fácil de aprender sin conocimientos de CSS • Construir todo desde cero • No hay componentes predefinidos
Curva de aprendizaje	20%	10%	70%

Nota. ^a Bootstrap (2023). ^b Google (2023). ^c Tailwind CSS (2023).

Se seleccionó Bootstrap para el desarrollo de la aplicación web debido a su facilidad de uso y versatilidad, mientras que Material Design se eligió para la aplicación móvil para garantizar una experiencia de usuario coherente y moderna en dispositivos Android. Esta elección se basó en las fortalezas y ventajas específicas de cada tecnología en función de las necesidades y plataformas de desarrollo.

3.7. Selección del gestor de base de datos

Tabla 7

Comparación de gestores de base de datos

Base de datos	Tipo de datos	Licencia	Soporte	Experiencia	Curva de aprendizaje	Alojamiento
Cloud Firestore	Orientada a documentos	Licencia Atribución 4.0 de Creative Commons	Android, iOS, Web	90%	30%	En la nube
Realtime Database	Orientada a documentos	Licencia Atribución 4.0 de Creative Commons	Android, iOS, Web	90%	20%	En la nube
MySQL	Relacional	Licencia Pública General de GNU/Licencia comercial	Apps móviles, web, Windows, Linux, MacOS	80%	20%	Local y en la nube
SQLite	Relacional	Dominio público	Apps móviles y web	0%	25%	Local y en la nube

Nota. Información generada por solid IT (2023).

Los gestores de Base de Datos elegidos fueron Cloud Firestore y Realtime Database, ya que estos son los más accesibles tomando en cuenta que sus alojamientos son en la nube y que brindan un plan de uso totalmente gratuito. Cabe señalar que estos servicios están integrados dentro de la plataforma de Backend as a Service (BaaS) desarrollada por Google llamada Firebase, la cual ofrece además almacenamiento y hosting gratuito para aplicaciones web y móviles.

Tamplin (2016) afirma que, firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2012 y adquirida por Google en 2014.

Tras lanzar el servicio chat, Tampin y Lee observaron que dicho chat estaba siendo utilizado en gran escala por los desarrolladores para pasar paquetes de información de sus aplicaciones, como el estado de las partidas en el caso de los juegos. Fue entonces cuando decidieron separar ambas funcionalidades, el sistema de chat y el sistema de arquitectura en

tiempo real que lo propulsaba, dando como resultado la fundación de Firebase en abril de 2012. (Aguiar, 2018)

Firebase ofrece una serie de ventajas que son especialmente relevantes para este tipo de aplicación:

- **Escalabilidad:** Es altamente escalable y puede manejar grandes volúmenes de datos y usuarios, lo que es crucial para una aplicación que podría necesitar procesar una gran cantidad de imágenes y datos de detección de plagas y enfermedades.
- **Tiempo real:** Ofrece una base de datos en tiempo real que permite a los usuarios obtener actualizaciones en tiempo real sobre la detección de plagas y enfermedades. Esto es especialmente útil para mantener a los agricultores y otros usuarios informados sobre el estado de sus cultivos.
- **Almacenamiento de archivos:** Proporciona un servicio de almacenamiento de archivos que permite a los usuarios cargar y almacenar imágenes y otros archivos relacionados con la detección de plagas y enfermedades. Esto es fundamental para una aplicación que depende en gran medida de datos visuales.
- **Autenticación y autorización:** Ofrece capacidades de autenticación y autorización, lo que significa que puedes controlar quién tiene acceso a la aplicación y qué funcionalidades pueden utilizar. Esto es importante para garantizar la privacidad y la seguridad de los datos de detección de plagas y enfermedades.
- **Integración con otras herramientas de Google:** Se integra con otras herramientas de Google, como Google Cloud Platform y Google AI Platform, lo que facilita la implementación de la inteligencia artificial para la detección de plagas y enfermedades.
- **Costos razonables:** Ofrece un modelo de precios flexible que permite a los startups y a las aplicaciones en crecimiento escalar sus costos de acuerdo con su uso, lo que puede resultar rentable para proyectos de detección de plagas y enfermedades que estén empezando.
- **Fácil de usar:** Es conocido por su facilidad de uso y su capacidad para permitir un desarrollo rápido de aplicaciones. Esto es especialmente útil cuando se está desarrollando una aplicación móvil y web compleja que incluye componentes de inteligencia artificial.

IV. Justificación de la metodología

Tabla 8

Comparación de metodologías de desarrollo de software

	SCRUM	KANBAN	Waterfall (cascada)
Descripción	Scrum es un marco de gestión de proyectos de metodología ágil que ayuda a los equipos a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. Los equipos aprenden a través de las experiencias, se autoorganizan mientras abordan un problema y reflexionan sobre sus victorias y derrotas para mejorar continuamente. ^a	Kanban se basa en una filosofía centrada en la mejora continua, donde las tareas se “extraen” de una lista de acciones pendientes en un flujo de trabajo constante. ^b	El modelo Waterfall es una metodología para la gestión de proyectos que se divide en distintas fases secuenciales. Cada fase comienza recién cuando ha terminado la anterior. Este enfoque para la gestión de proyectos surgió a partir de los sectores de fabricación y construcción, en los que cada hito debe estar finalizado para poder avanzar con el proceso de producción. ^c
Proceso de desarrollo	Se basa en sprints, ciclos de tiempo fijo donde se completa una cantidad de trabajo establecida.	Se basa en el desarrollo incremental del ciclo de vida de un proyecto y en aspectos visuales.	Se basa en una secuencia lineal de etapas de desarrollo, donde cada etapa depende de la anterior.
Roles	• Product owner • Scrum master • Equipo de desarrollo	No hay roles obligatorios, pero se han identificado dos funciones: • Service Request Manager • Service Delivery Manager	• Gerente de proyecto • Analista de negocios • Equipo de desarrollo • Equipo de pruebas
Herramientas	• Producto backlog • Sprint backlog • Incremento	• Tablero kanban	• Microsoft Project • GanttPRO • Zoho Projects
Curva de aprendizaje	Curva de aprendizaje rápida debido a los conceptos de iteraciones más pequeñas y reuniones diarias.	Depende del grado de éxito obtenido durante el aprendizaje en el transcurso del tiempo.	Curva de aprendizaje alta debido a la necesidad de planificación detallada y documentación extensa.

Nota. ^a Drumond (2023). ^b Martins (2022). ^c Laoyan (2022)

Tras haber analizado las metodologías más populares, se ha seleccionado la metodología Scrum para el desarrollo de la aplicación móvil y web de detección de plagas y enfermedades en cultivos, justificándose además por las siguientes razones:

- **Enfoque iterativo e incremental:** Scrum es conocido por su enfoque iterativo e incremental, lo que significa que el proyecto se divide en entregas incrementales y funcionales. En el caso de una aplicación móvil de detección de plagas y enfermedades en cultivos, esto permite que el equipo pueda desarrollar y entregar funcionalidades clave de manera temprana y rápida, lo que puede ser extremadamente valioso para los usuarios y para el negocio.
- **Adaptabilidad y respuesta al cambio:** En el ámbito agrícola, las plagas y enfermedades pueden surgir repentinamente o evolucionar rápidamente. Scrum proporciona flexibilidad y capacidad de respuesta al cambio, lo que permite al equipo adaptarse rápidamente a nuevas necesidades o requerimientos que puedan surgir durante el desarrollo. Esto es esencial para asegurar que la aplicación pueda hacer frente a las demandas cambiantes del entorno agrícola y brindar una solución actualizada y efectiva.
- **Colaboración multidisciplinaria:** Scrum fomenta la formación de equipos multidisciplinarios que incluyen desarrolladores, diseñadores, expertos en agricultura y otros profesionales relevantes. En el contexto de una aplicación móvil de detección de plagas y enfermedades en cultivos, contar con la colaboración de expertos agrícolas es fundamental para comprender los desafíos y requisitos específicos del sector. La participación activa de estos expertos durante todo el proceso de desarrollo garantiza que la solución final sea precisa, confiable y valiosa para los agricultores.
- **Retroalimentación temprana y continua:** Scrum incorpora revisiones y retrospectivas al final de cada sprint, lo que permite obtener comentarios tempranos y continuos sobre el producto en desarrollo. En el caso de una aplicación móvil de detección de plagas y enfermedades en cultivos, la retroalimentación de los agricultores y otros usuarios involucrados es esencial para garantizar que la aplicación cumpla con sus necesidades y expectativas. La capacidad de ajustar y mejorar la aplicación en función de esta retroalimentación mejora su calidad y utilidad en el campo.

V. Oportunidad en el mercado

El Gobierno de Reconciliación y Unidad Nacional (2021), asegura que, en Nicaragua, la agricultura es de gran importancia para la economía y seguridad alimentaria, donde su actividad genera la producción suficiente para satisfacer en su totalidad la demanda de consumo de granos básicos como el maíz y el frijol, incluso llegando a sobrepasar esa cantidad requerida. Junto con el sorgo, estos son cultivos esenciales en la dieta local y representan una parte significativa de la producción agrícola del país; convirtiéndolo en un sector que requiere especial atención y protección de factores que puedan ponerlo en riesgo.

Siendo una de las regiones más afectadas por los efectos del cambio climático, Nicaragua experimenta patrones climáticos impredecibles, sequías más intensas, y lluvias erráticas que crean un ambiente perfecto para el surgimiento de plagas y enfermedades (Ribalaygua et al., 2011). Además, la falta de recursos y acceso a información adecuada hace que sea difícil para los agricultores identificar y manejar eficazmente las amenazas, las cuales generan un riesgo de experimentar escasez de alimentos y aumentos en los precios de productos básicos, lo que impacta a toda la población.

Según la Agencia de Cooperación Internacional del Japón (2010), “El manejo integrado de plagas y enfermedades es una práctica agrícola sostenible que busca minimizar el uso de pesticidas químicos y promover en su lugar métodos más ecológicos y eficientes”. Esta tendencia global se está adoptando cada vez más en el país, y una aplicación móvil que brinde orientación y recursos específicos sobre el manejo integrado sería muy relevante.

Nicaragua ha experimentado un aumento en la integración de teléfonos móviles en los últimos años, y una aplicación móvil de detección de plagas y enfermedades en cultivos de granos básicos abordaría una necesidad crítica en el sector agrícola, proporcionaría soluciones tecnológicas accesibles, y tendría un impacto significativo en la seguridad alimentaria y la economía del país. Por lo tanto, esta iniciativa representa una oportunidad de mercado sólida y beneficiosa tanto para los agricultores como para la sociedad en general, aprovechando la oportunidad de colaborar con el gobierno y organizaciones no gubernamentales (ONG) que trabajan en el sector para ampliar su alcance y promover prácticas de manejo integrado en la agricultura.

VII. Propuesta de valor

La propuesta de valor de Placuix radica en ofrecer una solución integral de agricultura de precisión respaldada por inteligencia artificial. Lo que hace única a esta aplicación en comparación con otras en el mercado se puede destacar de la siguiente manera:

- **Detección precisa de plagas y enfermedades sin conexión a internet**

Placuix utiliza inteligencia artificial para una identificación precisa de plagas y enfermedades en los cultivos sin necesidad de tener conexión a internet, tomando en cuenta que en la mayoría de las áreas de siembras no se dispone de buena señal. Esta capacidad permite a los agricultores tomar medidas inmediatas y específicas, evitando pérdidas significativas y optimizando la producción.

- **Recomendaciones de tratamiento basadas en manejo integrado**

La aplicación no solo se limita a diagnosticar las amenazas, sino que también ofrece recomendaciones de tratamiento basadas en prácticas de manejo integrado. Esto significa que las acciones sugeridas son amigables con el medio ambiente y consideran diversas estrategias para abordar las amenazas de manera sostenible.

- **Datos climáticos locales y futuros**

Proporciona información detallada sobre las condiciones climáticas locales y futuras. Esto permite a los agricultores tomar decisiones informadas sobre el momento adecuado para tratar los cultivos, maximizando la eficacia de las medidas tomadas.

- **Comunidad agrícola fortalecida**

La inclusión de un foro dentro de la aplicación crea una comunidad agrícola interactiva. Los agricultores pueden compartir experiencias, conocimientos y mejores prácticas, fomentando la colaboración y el aprendizaje mutuo.

- **Registro histórico y monitoreo institucional**

Permite mantener un registro histórico de plagas y enfermedades. Este historial facilita el monitoreo institucional a nivel nacional, lo que contribuye a la gestión eficiente y a la toma de decisiones basada en datos a nivel macro.

En conjunto, Placuix optimiza la gestión agrícola al proporcionar herramientas avanzadas para el manejo de plagas y enfermedades. Esto no solo beneficia a nivel individual, sino que también contribuye a prácticas agrícolas más sostenibles y al fortalecimiento de la colaboración entre agricultores.

6.1. Roles

A continuación, se describe detalladamente cómo Placuix satisface las necesidades de cada uno de los cuatro roles de usuarios (agricultor, usuario no registrado, institución, administrador).

6.1.1. Agricultor

- **Detección de plagas y enfermedades:** El agricultor puede detectar plagas y enfermedades en sus cultivos mediante imágenes. Puede tomar fotos directamente desde su dispositivo o subirlas desde la galería de fotos de su teléfono. La aplicación utiliza tecnología de procesamiento de imágenes para identificar plagas y enfermedades de manera precisa.
- **Información climática en tiempo real:** La aplicación proporciona información climática actual basada en la ubicación del agricultor. Muestra datos como temperatura, velocidad del viento, porcentaje de humedad, porcentaje de nubosidad y el estado del día.
- **Pronóstico del clima a 5 días:** El agricultor también tiene acceso a un pronóstico del clima para los próximos cinco días, lo que le permite planificar sus actividades agrícolas de manera efectiva.
- **Información detallada sobre plagas y enfermedades:** La aplicación proporciona información detallada sobre plagas y enfermedades específicas según el tipo de cultivo seleccionado. Esta información incluye descripciones, síntomas de la plaga o enfermedad y recomendaciones de tratamiento utilizando prácticas de manejo integrado.
- **Foro de comunidad agrícola:** Los agricultores pueden participar en un foro donde pueden realizar publicaciones relacionadas con sus cultivos, hacer preguntas y compartir sus experiencias. También pueden comentar las publicaciones de otros usuarios, lo que fomenta el intercambio de conocimientos y la colaboración entre agricultores.
- **Historial de detección de plagas y enfermedades:** Los agricultores pueden registrar las plagas y enfermedades que detecten en sus cultivos a través de un formulario en la

aplicación. Esta información se almacena en un historial que el agricultor puede consultar en cualquier momento para realizar un seguimiento de los problemas pasados.

- **Mapas interactivos:** Los agricultores tienen acceso a mapas interactivos que muestran las zonas de cultivo de granos básicos en Nicaragua. También pueden ver un mapa de plagas y enfermedades detectadas en la zona, lo que les permite estar al tanto de las amenazas cercanas a sus cultivos y tomar medidas preventivas.

6.1.2. Usuario no registrado

- **Detección de plagas y enfermedades:** Puede detectar plagas y enfermedades en sus cultivos mediante imágenes. Puede tomar fotos directamente desde su dispositivo o subirlas desde la galería de fotos de su teléfono.
- **Información climática en tiempo real:** La aplicación proporciona información climática actual basada en la ubicación del usuario. Muestra datos como temperatura, velocidad del viento, porcentaje de humedad, porcentaje de nubosidad y el estado del día.
- **Pronóstico del clima a 5 días:** Tiene acceso a un pronóstico del clima para los próximos cinco días, lo que le permite planificar sus actividades agrícolas de manera efectiva.
- **Información detallada sobre plagas y enfermedades:** La aplicación proporciona información detallada sobre plagas y enfermedades específicas según el tipo de cultivo seleccionado. Esta información incluye descripciones, síntomas de la plaga o enfermedad y recomendaciones de tratamiento utilizando prácticas de manejo integrado.
- **Foro de comunidad agrícola:** Los usuarios no registrados pueden observar las publicaciones relacionadas por otros usuarios, pero no pueden comentarlas, al igual que no pueden realizar sus propias publicaciones.

La propuesta se enriquece aún más al incluir las funcionalidades específicas para el usuario institucional. Este usuario desempeña un papel crucial en la gestión y supervisión de la salud de los cultivos. A continuación, se detallan las características exclusivas que están disponibles para el usuario institucional a través de la aplicación web:

6.1.3. Institución

- **Registro de plagas y enfermedades en el mapa interactivo:** Los usuarios institucionales tienen la capacidad de acceder a un mapa interactivo desde la aplicación web. En este mapa, pueden agregar marcadores que representan plagas y enfermedades detectadas en ubicaciones específicas. Los marcadores incluyen información detallada sobre la amenaza detectada, como el nombre de la plaga o enfermedad, la fecha de detección y una descripción breve. Esto permite una visualización efectiva de las áreas afectadas.
- **Tabla de seguimiento de plagas y enfermedades:** En la aplicación web, los usuarios institucionales pueden acceder a una tabla que muestra todas las plagas y enfermedades detectadas y registradas en la aplicación móvil por parte de los agricultores. Esta tabla proporciona una visión general de las amenazas actuales y pasadas, permitiendo un seguimiento eficaz de la evolución de los problemas en los cultivos.
- **Dashboard de plagas y enfermedades:** Se le presenta una gráfica de dona con las plagas y enfermedades detectadas por departamento en el país, así como una gráfica de barra de la cantidad de amenazas detectadas por cultivo (frijol, maíz, sorgo).

El usuario administrador desempeña un papel fundamental en la gestión y el análisis de datos, lo que contribuye a mejorar continuamente la calidad del servicio. A continuación, se describen las características específicas disponibles para el administrador en la aplicación móvil:

6.1.4. Administrador

- **Gestión de usuarios registrados:** El administrador tiene acceso a una funcionalidad que le permite ver la cantidad de usuarios registrados.
- **Evaluación de utilidad mediante gráfico de pastel:** El administrador tiene la capacidad de recopilar y analizar la retroalimentación de los usuarios sobre la utilidad de la aplicación. Esto se logra a través de la opción de calificación, donde los usuarios pueden expresar si consideran que la aplicación es útil o no útil. El administrador recopila estos datos y presenta una representación visual en forma de gráfico de pastel para evaluar la satisfacción general de los usuarios.

VII. Ciclo del proyecto

Placuix utilizó la metodología Scrum para la gestión del desarrollo de software en este proyecto. Siguiendo los principios ágiles, se recorre un ciclo de desarrollo dinámico y colaborativo, con el objetivo de crear una aplicación móvil y una plataforma web que transformen la forma en que agricultores, observadores y usuarios institucionales interactúan en el ámbito agrícola de Nicaragua.

7.1. Historias de usuario

A continuación, se presentan los requisitos clave de la aplicación, organizados según los roles de usuario definidos en la metodología Scrum, siendo los siguientes:

- **Agricultor:** Utilizará la aplicación para detectar plagas y enfermedades, acceder a información sobre cultivos, ver el clima, registrar detecciones, ver mapas interactivos, acceder a un foro agrícola (publicar y comentar).
- **Administrador:** Utilizará la aplicación para ver la cantidad de usuarios registrados, ver calificaciones de la aplicación.
- **Institución:** Utilizará la aplicación para agregar marcadores en el mapa de plagas y enfermedades, ver una tabla de plagas y enfermedades detectadas, y colaborar en la gestión de amenazas agrícolas.
- **Usuario no registrado:** Utilizará la aplicación para detectar plagas y enfermedades, acceder a información sobre cultivos, ver el clima, ver mapas interactivos, acceder a un foro agrícola (visualizar).

Cada historia de usuario define los elementos esenciales, como la necesidad a satisfacer, la prioridad, el impacto, la validación y la estimación, para guiar el desarrollo de funcionalidades específicas y mejorar la aplicación.

Tabla 9*Historia de usuario 1*

H00- Detección de plagas y enfermedades mediante imágenes			
Como	Agricultor / Usuario no registrado.		
Quiero	Utilizar la cámara de mi dispositivo o cargar imágenes desde la galería de fotos.		
Para	Detectar plagas y enfermedades en mis cultivos de manera eficiente y precisa.		
Validación	- Analizar imágenes de cultivos. - Proporcionar resultados claros que identifiquen la presencia de plagas o enfermedades.	Impacto	10
	Incluir el nombre específico sobre la plaga o enfermedad detectada.	Prioridad	5
	Detectar una variedad de plagas y enfermedades comunes en los cultivos de granos básicos.	Estimación	120 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 1.

Tabla 10*Historia de usuario 2*

H01- Información sobre plagas y enfermedades			
Como	Agricultor / Usuario no registrado.		
Quiero	Acceder a una lista de plagas y enfermedades que afectan a los cultivos de maíz, frijol y sorgo.		
Para	Obtener información detallada sobre las amenazas potenciales para mis cultivos y aprender sobre las recomendaciones de tratamiento utilizando prácticas de manejo integrado.		
Validación	Proporcionar una lista de plagas y enfermedades específicas que afectan a los cultivos de maíz, frijol y sorgo.	Impacto	10
	Incluir descripciones detalladas de cada plaga o enfermedad, incluyendo su descripción física y los daños que causan a los cultivos.	Prioridad	5
	Proporcionar recomendaciones claras y prácticas de manejo integrado para controlar o prevenir estas plagas y enfermedades.	Estimación	60 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 2.

Tabla 11*Historia de usuario 3*

H02- Clima actual			
Como	Agricultor / Usuario no registrado.		
Quiero	Ver el clima actual para mi ubicación actual o para una ubicación específica.		
Para	Tomar decisiones informadas sobre las actividades agrícolas diarias, como siembra, cosecha o riego, teniendo en cuenta las condiciones climáticas actuales.		
Validación	Obtener automáticamente la ubicación del agricultor o permitirle buscar una ubicación específica.	Impacto	7

Mostrar la temperatura actual, la humedad, la velocidad del viento, el nivel de nubosidad y el estado general del día (soleado, nublado, lluvioso, etc.).	Prioridad	4
Proporcionar información precisa y actualizada para reflejar las condiciones climáticas en tiempo real.	Estimación	24 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 3.

Tabla 12

Historia de usuario 4

H03- Pronóstico del tiempo		
Como	Agricultor / Usuario no registrado.	
Quiero	Ver el pronóstico del tiempo futuro para mi ubicación actual o para una ubicación específica.	
Para	Planificar actividades agrícolas a corto y largo plazo, como siembra, cosecha, aplicación de pesticidas y riego, teniendo en cuenta las condiciones climáticas previstas.	
Validación	Obtener automáticamente la ubicación del agricultor o permitirle buscar una ubicación específica.	Impacto 8
	Proporcionar un pronóstico a corto plazo (para las próximas 24 horas) y uno a largo plazo (para los próximos 5 días).	Prioridad 4
	Incluir detalles sobre la temperatura, la humedad, la velocidad del viento, el nivel de nubosidad y el estado general del día.	Estimación 12 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 4.

Tabla 13

Historia de usuario 5

H04- Registro de plagas y enfermedades		
Como	Agricultor.	
Quiero	Tener la capacidad de registrar las plagas y enfermedades que detecto en mis cultivos a través de un formulario.	
Para	Documentar y compartir información importante sobre la presencia de plagas o enfermedades en mis cultivos y colaborar en la prevención y el manejo adecuado.	
Validación	Incluir campos para ingresar el nombre del cultivo afectado, el nombre de la plaga o enfermedad detectada, la fecha y hora de detección, la ubicación y un número de contacto.	Impacto 8
		Prioridad 4
		Estimación 4 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 5.

Tabla 14*Historia de usuario 6*

H05- Historial de plagas y enfermedades			
Como	Agricultor.		
Quiero	Ver un historial de plagas y enfermedades que se han registrado previamente en un formato de tabla.		
Para	Revisar y analizar la información de plagas y enfermedades anteriores en mis cultivos.		
Validación	• Proporcionar una tabla organizada que muestre todos los registros previos de plagas y enfermedades.	Impacto	7
	• Incluir detalles como el nombre del cultivo afectado, el nombre de la plaga o enfermedad, la fecha y hora de detección, la ubicación y el número de contacto en cada entrada en la tabla.	Prioridad	3
	• Permitir ordenar y filtrar la tabla para buscar registros específicos o realizar análisis.	Estimación	6 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 6.

Tabla 15*Historia de usuario 7*

H06- Mapa de zonas de cultivos de granos básicos			
Como	Agricultor / Usuario no registrado.		
Quiero	Acceder a un mapa interactivo que muestre las zonas en las que se cultivan granos básicos en Nicaragua.		
Para	Obtener una visión general de las áreas de cultivo de granos básicos en mi región y en todo el país, lo que me ayudará en la toma de decisiones relacionadas con la planificación de cultivos y la identificación de posibles colaboraciones con otros agricultores.		
Validación	• Mostrar en el mapa interactivo de manera clara y precisa las áreas donde se cultivan granos básicos, como maíz, frijol y sorgo, en Nicaragua.	Impacto	6
	• Permitir hacer zoom y explorar el mapa para obtener detalles específicos de ubicación.	Prioridad	3
	• Incluir herramientas de búsqueda y filtro para encontrar información sobre cultivos en áreas específicas.	Estimación	1 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 7.

Tabla 16*Historia de usuario 8*

H07- Mapa de plagas y enfermedades detectadas			
Como	Agricultor / Usuario no registrado.		
Quiero	Acceder a un mapa interactivo que muestre las zonas en las que han sido detectadas plagas y enfermedades en los cultivos.		
Para	Obtener información sobre las áreas afectadas por plagas y enfermedades en mi región y en todo el país, lo que me ayudará a tomar medidas preventivas y a estar alerta ante posibles amenazas para mis cultivos.		
Validación	Mostrar en el mapa interactivo de manera clara y precisa las áreas donde se han detectado plagas y enfermedades en los cultivos.	Impacto	7
	Permitir hacer zoom y explorar el mapa para obtener detalles específicos de ubicación.	Prioridad	4
	Incluir herramientas de búsqueda y filtro para encontrar información sobre plagas y enfermedades en áreas específicas.	Estimación	1 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 8.

Tabla 17*Historia de usuario 9*

H08- Foro agrícola			
Como	Agricultor.		
Quiero	Participar en un foro agrícola en la aplicación móvil.		
Para	Plantear preguntas, compartir información y experiencias, y aprender de otros agricultores, además de poder proporcionar respuestas y consejos a otros miembros de la comunidad agrícola.		
Validación	Permitir la creación de publicaciones en el foro, ingresando el nombre de la duda o pregunta, una descripción detallada y adjuntando imágenes relevantes.	Impacto	8
	Permitir ver las publicaciones realizadas por otros agricultores.	Prioridad	4
	Permitir realizar comentarios en las publicaciones de otros usuarios, utilizando mensajes de texto para proporcionar respuestas, consejos o compartir experiencias.	Estimación	48 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 9.

Tabla 18*Historia de usuario 10*

H09- Perfil de agricultor			
Como	Agricultor.		
Quiero	Visualizar y actualizar la información de mi perfil.		
Para	Mantener mi información actualizada y personalizar mi perfil de acuerdo a mis preferencias y cambios en mi situación.		
Validación	Permitir el acceso a la sección de edición de perfil desde la configuración de la aplicación.	Impacto	6
	Permitir cambiar mi imagen de perfil, cargando una nueva imagen desde mi galería	Prioridad	3
	Permitir la edición del nombre de usuario, profesión y ubicación.		
	Mostrar un mensaje de confirmación cuando los cambios han sido guardados correctamente.	Estimación	16 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 10.

Tabla 19*Historia de usuario 11*

H10- Calificación de la aplicación			
Como	Agricultor.		
Quiero	Calificar la aplicación y proporcionar retroalimentación sobre su utilidad.		
Para	Expresar mi opinión sobre la aplicación y ayudar a los desarrolladores a tomar decisiones informadas basadas en mi experiencia.		
Validación	Mostrar una ventana emergente que invite al Agricultor a calificar la aplicación, con opciones para calificarla como "útil" o "no útil".	Impacto	5
	Permitir al agricultor elegir una de las opciones de calificación y enviar su elección.	Prioridad	3
	Registrar la calificación seleccionada en la base de datos para futuras referencias y análisis.	Estimación	8 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 11.

Tabla 20*Historia de usuario 12*

H11- Registro de agricultor			
Como	Agricultor.		
Quiero	Registrarme en la aplicación para obtener una cuenta personalizada.		
Para	Acceder a ciertas funcionalidades de la aplicación, como la capacidad de realizar publicaciones en el foro y llevar un registro de plagas y enfermedades.		
Validación	Permitir el acceso a la opción de registro desde la pantalla de inicio de sesión de la aplicación.	Impacto	7

	• Validar el correcto llenado del formulario de registro. • Mostrar un mensaje de registro exitoso una vez completado el proceso. • Permitir el inicio de sesión en la cuenta después de registrarse.	Prioridad	4
		Estimación	8 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 12.

Tabla 21

Historia de usuario 13

H12- Visualización de calificaciones de la aplicación			
Como	Administrador.		
Quiero	Ver las calificaciones de la aplicación proporcionadas por otros usuarios, incluyendo si la calificaron como "útil" o "no útil".		
Para	Obtener una comprensión de la percepción general de la aplicación por parte de la comunidad de agricultores y tomar decisiones informadas sobre su posible utilidad.		
Validación	• Mostrar las calificaciones proporcionadas por los usuarios mediante un gráfico de pastel. • Incluir un indicador en cada calificación que muestre si el usuario la consideró "útil" o "no útil".	Impacto	6
		Prioridad	3
		Estimación	4 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 13.

Tabla 22

Historia de usuario 14

H13- Cantidad de usuarios registrados			
Como	Administrador.		
Quiero	Ver la cantidad total de usuarios registrados en la aplicación.		
Para	Obtener una idea de la comunidad de usuarios y la adopción de la aplicación en la comunidad agrícola, lo que puede proporcionar una perspectiva sobre su popularidad y utilidad.		
Validación	• Mostrar claramente la cantidad total de usuarios registrados mediante un texto.	Impacto	5
		Prioridad	3
		Estimación	1 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 14.

Tabla 23*Historia de usuario 15*

H14- Marcadores en el mapa de plagas y enfermedades			
Como	Institución.		
Quiero	Agregar marcadores en el mapa interactivo de plagas y enfermedades, junto con el nombre de la plaga o enfermedad detectada.		
Para	Documentar y marcar de manera efectiva las ubicaciones de plagas y enfermedades detectadas en colaboración con otros agricultores y autoridades agrícolas, y facilitar su seguimiento y control.		
Validación	Permitir el acceso a la función de agregar marcadores en el mapa desde la aplicación web.	Impacto	8
	Permitir la opción de proporcionar el nombre de la plaga o enfermedad detectada al agregar un marcador.	Prioridad	4
		Estimación	48 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 15.

Tabla 24*Historia de usuario 16*

H15- Plagas y enfermedades detectadas			
Como	Institución.		
Quiero	Ver una tabla que muestre todas las plagas y enfermedades detectadas por los usuarios en la aplicación.		
Para	Tener una visión completa y organizada de las amenazas agrícolas registradas, lo que facilitará la toma de decisiones y la gestión de medidas de control y prevención.		
Validación	Permitir el acceso a la tabla de plagas y enfermedades detectadas desde la aplicación web de usuario Institucional.	Impacto	9
	Incluir información detallada sobre cada plaga o enfermedad, como el nombre, la fecha de detección, la ubicación, el cultivo afectado y el nombre del agricultor que realizó la detección.	Prioridad	5
	Permitir la opción de ordenar y filtrar la tabla para buscar información específica.	Estimación	48 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 16.

Tabla 25

Historia de usuario 17

H16- Inicio de sesión			
Como	Agricultor / Administrador / Institución.		
Quiero	Iniciar sesión en la aplicación.		
Para	Acceder a mi cuenta personalizada y aprovechar las funcionalidades.		
Validación	Mostrar formulario que permita realizar la autenticación.	Impacto	8
	Validar campos del formulario.	Prioridad	4
	Obtener el usuario y mostrar opciones correspondientes.	Estimación	24 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 17.

Tabla 26

Historia de usuario 18

H17- Dashboard de plagas y enfermedades detectadas			
Como	Institución.		
Quiero	Tener la capacidad de visualizar datos sobre plagas y enfermedades detectadas por cultivo y por departamento, de una manera clara y efectiva, a través de gráficas.		
Para	Tomar decisiones informadas y estratégicas en la gestión de plagas y enfermedades, identificar patrones y tendencias, y asignar recursos de manera eficiente.		
Validación	Mostrar los gráficos con un desglose claro y legible de la cantidad de plagas y enfermedades detectadas en cada departamento y cultivo.	Impacto	9
	Actualizar automáticamente la información de los gráficos cuando se agreguen nuevos registros o se modifiquen datos.	Prioridad	3
	Permitir hacer clic en una porción del gráfico para obtener detalles adicionales.	Estimación	12 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 18.

Tabla 27*Historia de usuario 19*

H18- Splash Screen			
Como	Agricultor/Usuario no registrado.		
Quiero	Ver una pantalla de presentación al abrir la aplicación		
Para	Tener una experiencia visual atractiva y confirmar que la aplicación se está cargando.		
Validación	Mostrar automáticamente el splash screen al abrir la aplicación, sin necesidad de interacción del usuario.	Impacto	5
	Mostrar el splash screen durante 3 segundos exactos.	Prioridad	2
	Mostrar el logotipo o la identidad visual de la aplicación en el splash screen, proporcionando una breve introducción a la marca.	Estimación	3 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 19.

Tabla 28*Historia de usuario 20*

H19- Manual de introducción			
Como	Agricultor/Usuario no registrado.		
Quiero	Ver un manual introductorio que conste de 5 pantallas que se muestran solo la primera vez que abro la aplicación móvil.		
Para	Aprender rápidamente cómo utilizar la aplicación y aprovechar al máximo sus características.		
Validación	Proporcionar información clara y concisa sobre una función específica de la aplicación en cada pantalla del manual.	Impacto	7
	Mostrar las pantallas del manual con un diseño atractivo y visualmente agradable que faciliten su lectura y comprensión.	Prioridad	2
	Permitir la navegación entre las pantallas del manual de manera intuitiva.	Estimación	24 hrs

Nota. La tabla contiene las funciones del software según la historia de usuario N° 20.

7.2. Product Backlog

A continuación, se presenta una tabla que detalla las historias de usuario, asignando puntuaciones de prioridad en una escala de 1 a 5 y puntuaciones de impacto en una escala de 1 a 10.

Tabla 29
Product Backlog

ID	Historia de usuario	Prioridad	Impacto	Importancia	Tiempo estimado
H00	Detección de plagas y enfermedades mediante imágenes	5	10	50	21 días
H01	Información sobre plagas y enfermedades	5	10	50	8 días
H02	Clima actual	4	7	28	5 días
H03	Pronóstico del tiempo	4	8	32	5 días
H04	Registro de plagas y enfermedades	4	8	32	2 días
H05	Historial de plagas y enfermedades	3	7	21	3 días
H06	Mapa de zonas de cultivos de granos básicos	3	6	18	1 día
H07	Mapa de plagas y enfermedades detectadas	4	7	28	6 días
H08	Foro agrícola	4	8	32	8 días
H09	Perfil del agricultor	3	6	18	10 días
H10	Calificación de la aplicación	3	5	15	3 días
H11	Registro de agricultor	4	7	28	2 días
H12	Visualización de calificaciones de la aplicación	3	6	18	2 días
H13	Cantidad de usuarios registrados	3	5	15	2 días
H14	Marcadores en el mapa de plagas y enfermedades	4	8	32	3 días
H15	Plagas y enfermedades detectadas	5	9	45	4 días
H16	Inicio de sesión	4	8	32	1 día
H17	Dashboard de plagas y enfermedades detectadas	3	9	27	1 día
H18	Splash Screen	2	5	10	1 día
H19	Manual de introducción	4	6	24	1 día

Nota. Las puntuaciones en la tabla ayudan a calcular la importancia de cada historia y estimar el tiempo necesario para su implementación.

7.3. Product Backlog organizado en Sprints

Se presentan las historias de usuario organizadas en orden descendente de importancia, lo que facilita su agrupación en Sprints para la planificación y desarrollo del proyecto.

Tabla 30

Product Backlog organizado en Sprints

ID	Historia de usuario	Importancia	Sprint programado	Fecha de inicio	Fecha de fin
H00	Detección de plagas y enfermedades mediante imágenes	50	1	14/08/23	03/09/23
H01	Información sobre plagas y enfermedades	50	1	14/08/23	21/08/23
H15	Plagas y enfermedades detectadas	45	1	22/08/23	25/08/23
H03	Pronóstico del tiempo	32	2	04/09/23	08/09/23
H04	Registro de plagas y enfermedades	32	2	11/09/23	12/09/23
H08	Foro agrícola	32	2	13/09/23	20/09/23
H14	Marcadores en el mapa de plagas y enfermedades	32	2	21/09/23	23/09/23
H16	Inicio de sesión	32	2	24/09/23	24/09/23
H02	Clima actual	28	3	25/09/23	29/09/23
H07	Mapa de plagas y enfermedades detectadas	28	3	01/10/23	05/10/23
H11	Registro de agricultor	28	3	06/10/23	07/10/23
H17	Dashboard de plagas y enfermedades detectadas	27	3	08/10/23	08/10/23
H19	Manual de introducción	24	4	09/10/23	09/10/23
H05	Historial de plagas y enfermedades	21	4	10/10/23	11/10/23
H06	Mapa de zonas de cultivos de granos básicos	18	4	12/10/23	12/10/23
H09	Perfil del agricultor	18	4	13/10/23	22/10/23
H12	Visualización de calificaciones de la aplicación	18	5	23/10/23	24/10/23
H10	Calificación de la aplicación	15	5	25/10/23	27/10/23
H13	Cantidad de usuarios registrados	15	5	28/10/23	28/10/23
H18	Splash Screen	10	5	29/10/23	29/10/23

7.4. Sprint Backlog

Es la lista de tareas que se eligen del Product Backlog en la planificación del sprint actual. Estas tareas son esenciales para llevar a cabo el desarrollo de un incremento del producto.

7.4.1. Sprint 1

A continuación, se describe el inicio del primer sprint, que tiene una duración de 3 semanas y abarca 3 historias de usuario junto con un total de 28 tareas, que incluyen las pruebas de software necesarias.

Tabla 31
Sprint 1

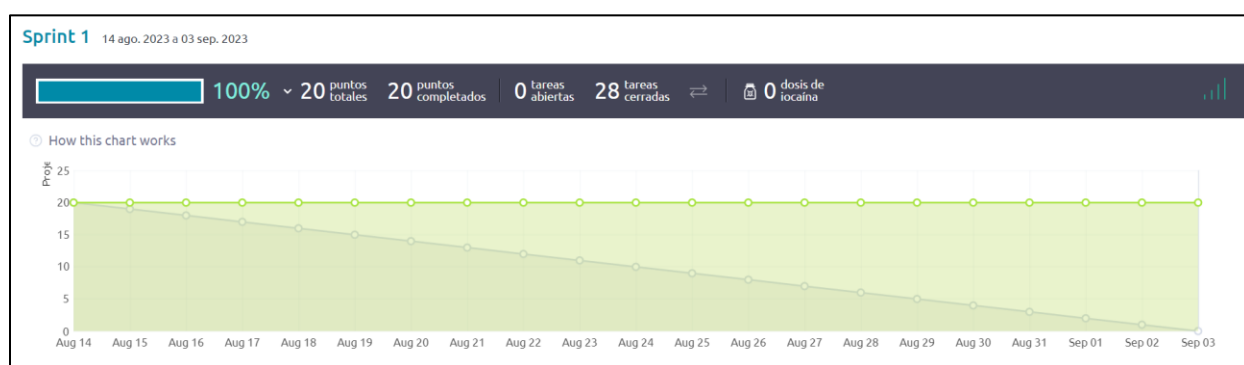
ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H00	Detección de plagas y enfermedades	Diseñar la interfaz de usuario de la función de detección de plagas y enfermedades	3 hrs	Ramón González
		Crear prototipos interactivos para evaluar la usabilidad y la experiencia del usuario	3 hrs	Ramón González
		Diseñar elementos visuales claros para representar los resultados de detección	2 hrs	Ramón González
		Colaborar con el investigador para recopilar y organizar datos sobre plagas y enfermedades en la agricultura para incorporarlos en el diseño	2 hrs	Ramón González
		Implementar la funcionalidad de acceso a la cámara del dispositivo y carga de imágenes desde la galería.	2 hrs	Jeyson Guevara
		Desarrollar el algoritmo de detección de plagas y enfermedades	2 hrs	Jeyson Guevara
		Entrenar e integrar el modelo de aprendizaje de plagas y enfermedades agrícolas para referencias de detección.	12 hrs	Jeyson Guevara
		Programar la lógica de presentación de resultados detallados	1 hrs	Jeyson Guevara
		Recopilar datos e información sobre plagas y enfermedades agrícolas comunes en los cultivos de granos básicos	10 hrs	Ruddy Mendoza
		Colaborar con el diseñador para asegurarse de que la interfaz de usuario incluya la información necesaria para la detección.	2 hrs	Ruddy Mendoza
H01	Información sobre plagas y enfermedades	Diseñar una interfaz de usuario clara y fácil de navegar para acceder a la lista de plagas y enfermedades en los cultivos de maíz, frijol y sorgo.	4 hrs	Ramón González

H15	Plagas y enfermedades detectadas	Crear un diseño de pantalla que muestre las descripciones detalladas de las plagas y enfermedades.	2 hrs	Ramón González
		Diseñar y crear la base de datos	4 hrs	Jeyson Guevara
		Implementar la funcionalidad que permita a los usuarios acceder a la lista de plagas y enfermedades en los cultivos específicos.	3 hrs	Jeyson Guevara
		Desarrollar la lógica para mostrar descripciones detalladas de cada plaga y enfermedad en la interfaz de usuario.	4 hrs	Jeyson Guevara
		Programar la funcionalidad que presente recomendaciones claras y prácticas de manejo integrado para el control y la prevención de plagas y enfermedades.	2 hrs	Jeyson Guevara
		Recopilar datos sobre las plagas y enfermedades que afectan a los cultivos de maíz, frijol y sorgo	4 hrs	Ruddy Mendoza
		Proporcionar recomendaciones basadas en prácticas de manejo integrado para el control y la prevención de cada plaga o enfermedad.	3 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario intuitiva y eficiente que permita a las instituciones acceder a la tabla de plagas y enfermedades detectadas.	3 hrs	Ramón González
		Crear un diseño de pantalla que muestre de manera organizada la información detallada sobre cada plaga o enfermedad	2 hrs	Ramón González
		Diseñar elementos de interfaz que permitan a los usuarios ordenar y filtrar la tabla para buscar información específica.	1 hrs	Ramon González
		Creación y diseño de la base de datos	1 hrs	Jeyson Guevara
		Implementar la funcionalidad que permita a las instituciones acceder a la tabla de plagas y enfermedades detectadas a través de la aplicación web.	3 hrs	Jeyson Guevara
		Programar la lógica que recopile y muestre información detallada sobre cada plaga o enfermedad	3 hrs	Jeyson Guevara
		Desarrollar la funcionalidad de ordenar y filtrar la tabla para que los usuarios puedan buscar información específica de manera eficiente.	1 hrs	Jeyson Guevara

Colaborar con el diseñador y el desarrollador para definir los campos de información detallada que deben incluirse en la tabla.	1 hrs	Ruddy Mendoza
Proporcionar retroalimentación y guía sobre la mejor manera de presentar los datos en la tabla.	1 hrs	Ruddy Mendoza
Pruebas de software	3 hrs	Jeyson Guevara

Figura 16

Burndown chart Sprint 1

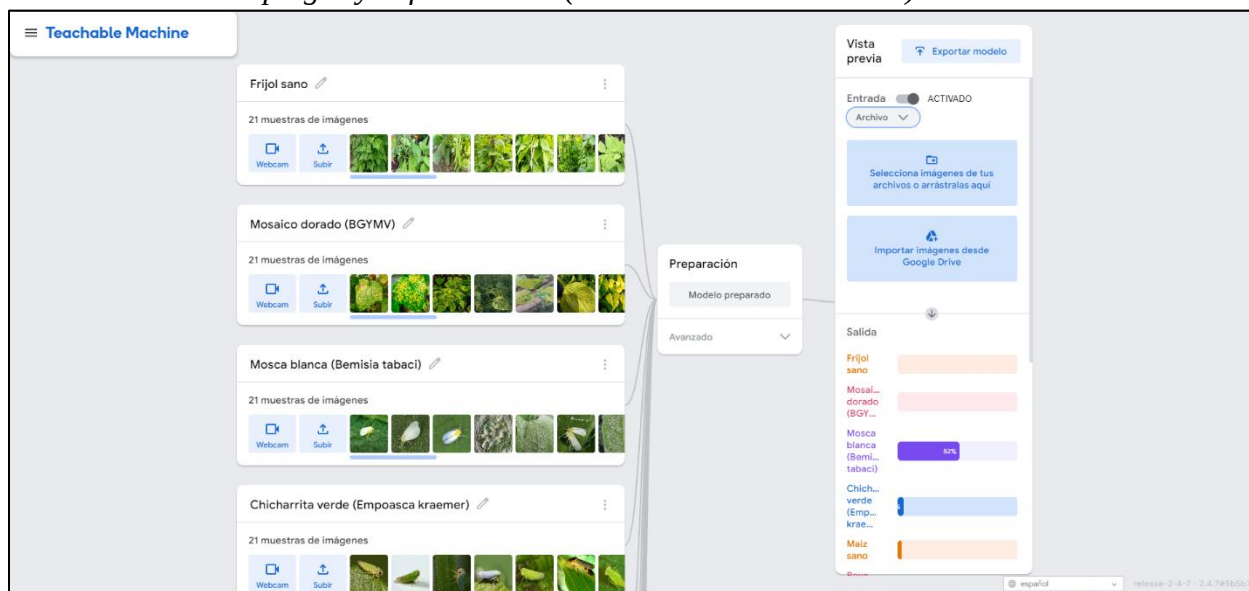


Nota. Muestra el progreso aproximado o proyectado del Sprint 1 en función de las tareas cerradas estimadas de la historia de usuario. Las historias de usuarios abiertas y terminadas se observar en la barra superior de color negro. Figura generada por taiga.io.

7.4.1.1. Entregables Sprint 1.

Figura 17

H00 - Detección de plagas y enfermedades (Entrenamiento de modelo)



Nota. Figura tomada desde teachablemachine.withgoogle.com.

Figura 19

H00 - Detección de plagas y enfermedades (Interfaz de detección)

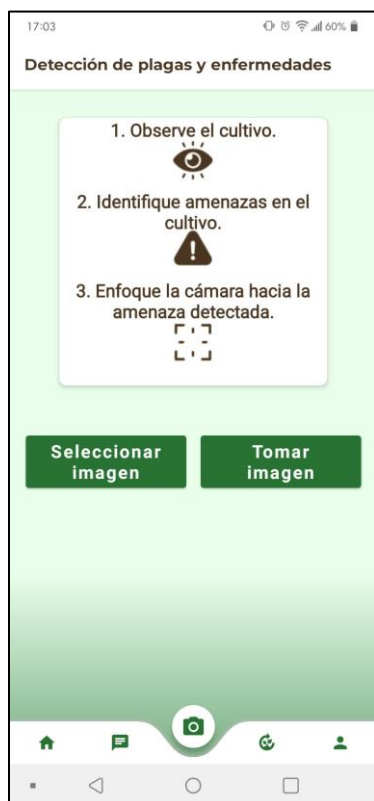


Figura 18

H00 - Detección de plagas y enfermedades (Resultados de detección)



Figura 21
H01 - Información sobre plagas y enfermedades (Selección de cultivo)



Figura 22
H01 - Información sobre plagas y enfermedades (Información)



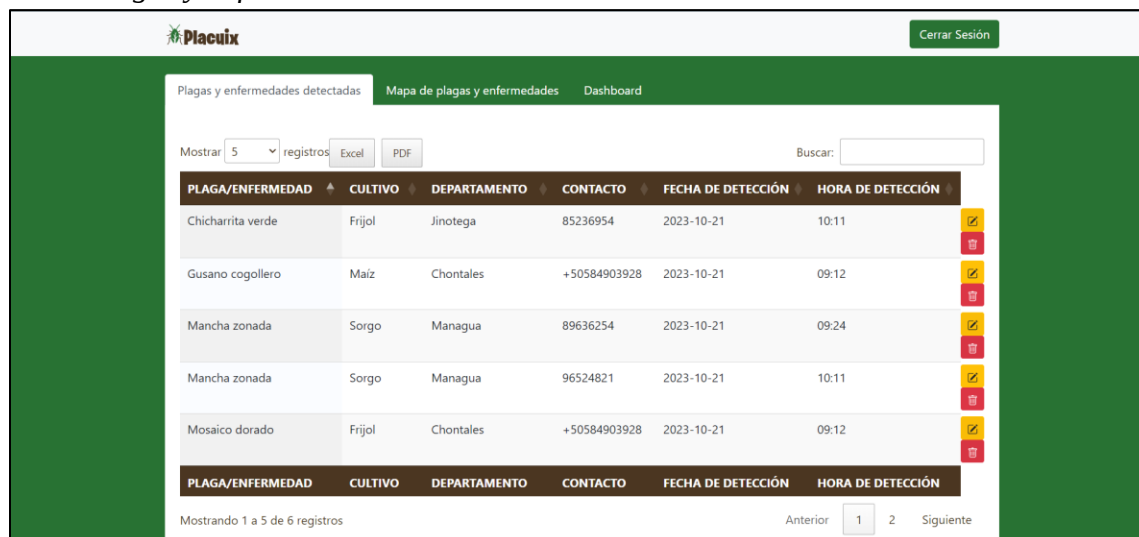
Figura 20
H01 - Información sobre plagas y enfermedades (Lista)



Figura 23
H01 - Información sobre plagas y enfermedades (Tratamiento)



Figura 24
H15 - Plagas y enfermedades detectadas



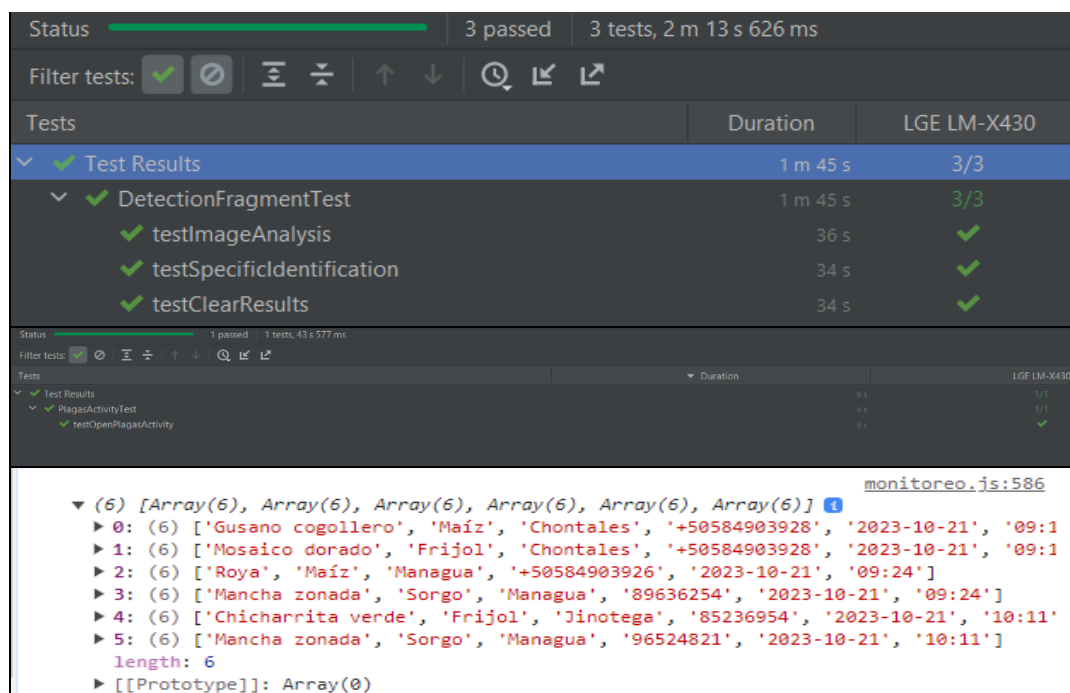
The screenshot shows the Placuix web application interface. At the top, there is a navigation bar with the Placuix logo and a 'Cerrar Sesión' button. Below the navigation bar, there are three tabs: 'Plagas y enfermedades detectadas' (selected), 'Mapa de plagas y enfermedades', and 'Dashboard'. The main content area displays a table of detected pests and diseases. The table has columns for 'PLAGA/ENFERMEDAD', 'CULTIVO', 'DEPARTAMENTO', 'CONTACTO', 'FECHA DE DETECCIÓN', and 'HORA DE DETECCIÓN'. There are also buttons for 'Mostrar' (5 registros), 'Excel', and 'PDF'. A search bar is located at the top right of the table. The table contains five rows of data. At the bottom, there is a pagination bar showing 'Mostrando 1 a 5 de 6 registros' and navigation buttons for 'Anterior', '1', '2', and 'Siguiente'.

PLAGA/ENFERMEDAD	CULTIVO	DEPARTAMENTO	CONTACTO	FECHA DE DETECCIÓN	HORA DE DETECCIÓN
Chicharrita verde	Frijol	Jinotega	85236954	2023-10-21	10:11
Gusano cogollero	Maíz	Chontales	+50584903928	2023-10-21	09:12
Mancha zonada	Sorgo	Managua	89636254	2023-10-21	09:24
Mancha zonada	Sorgo	Managua	96524821	2023-10-21	10:11
Mosaico dorado	Frijol	Chontales	+50584903928	2023-10-21	09:12

7.4.1.2. Pruebas de software Sprint 1.

A continuación, se muestra el resultado de la ejecución de pruebas de software.

Figura 25
Pruebas de software Sprint 1



The screenshot shows the Android Studio test results interface. At the top, there is a status bar indicating '3 passed' and '3 tests, 2 m 13 s 626 ms'. Below the status bar, there are filter buttons for 'Filter tests:'. The main content area displays a table of test results. The table has columns for 'Tests', 'Duration', and 'LGE LM-X430'. There are two sections of test results. The first section shows 'Test Results' with a duration of '1 m 45 s' and a status of '3/3'. It includes sub-tests: 'DetectionFragmentTest' (1 m 45 s, 3/3), 'testImageAnalysis' (36 s, ✓), 'testSpecificIdentification' (34 s, ✓), and 'testClearResults' (34 s, ✓). The second section shows 'Test Results' with a duration of '1 passed' and '1 tests, 43 s 577 ms'. It includes sub-tests: 'PlagasActivityTest' (0 s, 1/1) and 'testOpenPlagasActivity' (0 s, ✓). At the bottom, there is a console log showing the output of the 'monitoreo.js:586' script, which displays an array of 6 objects, each containing pest/disease information.

Tests	Duration	LGE LM-X430
Test Results	1 m 45 s	3/3
DetectionFragmentTest	1 m 45 s	3/3
testImageAnalysis	36 s	✓
testSpecificIdentification	34 s	✓
testClearResults	34 s	✓

Tests	Duration	LGE LM-X430
Test Results	1 passed	1 tests, 43 s 577 ms
PlagasActivityTest	0 s	1/1
testOpenPlagasActivity	0 s	✓


```

(6) [Array(6), Array(6), Array(6), Array(6), Array(6), Array(6)] monitoreo.js:586
  0: (6) ['Gusano cogollero', 'Maíz', 'Chontales', '+50584903928', '2023-10-21', '09:12']
  1: (6) ['Mosaico dorado', 'Frijol', 'Chontales', '+50584903928', '2023-10-21', '09:12']
  2: (6) ['Roya', 'Maíz', 'Managua', '+50584903926', '2023-10-21', '09:24']
  3: (6) ['Mancha zonada', 'Sorgo', 'Managua', '89636254', '2023-10-21', '09:24']
  4: (6) ['Chicharrita verde', 'Frijol', 'Jinotega', '85236954', '2023-10-21', '10:11']
  5: (6) ['Mancha zonada', 'Sorgo', 'Managua', '96524821', '2023-10-21', '10:11']
  length: 6
  [[Prototype]]: Array(0)
  
```

Nota. Se puede observar que hubo una cobertura de código del 100% en los componentes DetectionFragment, DashboardFragment, Monitoreo. Figura generada por Android Studio y consola de Google Chrome.

7.4.2. Sprint 2

El segundo sprint tiene una duración de 3 semanas y consta de 5 historias de usuario y 41 tareas en total, incluyendo las pruebas de software correspondientes.

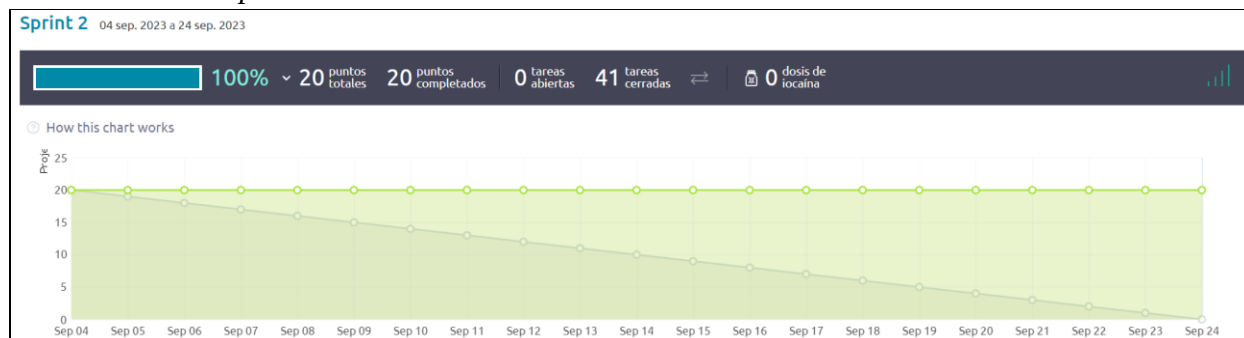
Tabla 32
Sprint 2

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H03	Pronóstico del tiempo	Diseñar una interfaz de usuario amigable que permita a los usuarios ver el pronóstico del tiempo.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre claramente el pronóstico a corto plazo y a largo plazo para la ubicación actual o una ubicación específica.	1 hrs	Ramón González
		Diseñar elementos visuales que representen los datos de temperatura, humedad, velocidad del viento, nubosidad y estado del día de manera comprensible.	1 hrs	Ramón González
		Implementar la funcionalidad que obtenga automáticamente la ubicación actual del agricultor o permita la búsqueda de ubicaciones específicas.	2 hrs	Jeyson Guevara
		Programar la obtención de datos meteorológicos a partir de la API.	1 hrs	Jeyson Guevara
		Desarrollar la lógica para presentar el pronóstico del tiempo a corto plazo (próximas 24 horas) y a largo plazo (próximos 5 días).	3 hrs	Jeyson Guevara
		Integrar la funcionalidad de presentación de detalles sobre temperatura, humedad, velocidad del viento, nubosidad y estado del día.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y detalles clave a mostrar en la interfaz de usuario.	1 hrs	Ruddy Mendoza
		Proporcionar orientación sobre la presentación efectiva de datos meteorológicos para que sean útiles para la planificación agrícola.	1 hrs	Ruddy Mendoza
H04	Registro de plagas y enfermedades	Diseñar una interfaz de usuario intuitiva y fácil de usar para el formulario de registro de plagas y enfermedades.	1 hrs	Ramón González

H08 Foro agrícola	Crear un diseño de formulario que incluya campos solicitados	1 hrs	Ramón González
	Diseñar elementos visuales que guíen al agricultor a través del proceso de registro de manera efectiva.	1 hrs	Ramón González
	Implementar la funcionalidad del formulario de registro, incluyendo la creación de campos y la lógica de validación.	2 hrs	Jeyson Guevara
	Programar la lógica para registrar la información ingresada por el agricultor	1 hrs	Jeyson Guevara
	Integrar un sistema de almacenamiento de datos para guardar la información registrada de manera segura.	2 hrs	Jeyson Guevara
	Desarrollar la funcionalidad de envío de la información a la base de datos de plagas y enfermedades detectadas.	1 hrs	Jeyson Guevara
	Colaborar con el diseñador y el desarrollador para definir los campos clave que deben incluirse en el formulario de registro.	1 hrs	Ruddy Mendoza
	Diseñar una interfaz de usuario atractiva y fácil de usar para el foro agrícola en la aplicación móvil.	2 hrs	Ramón González
	Crear un diseño de pantalla que permita a los agricultores crear y comentar publicaciones	2 hrs	Ramón González
	Diseñar elementos visuales que faciliten la navegación por las publicaciones y los comentarios de otros agricultores.	1 hrs	Ramón González
	Implementar la funcionalidad del foro agrícola	3 hrs	Jeyson Guevara
	Programar la lógica que permita a los agricultores ver las publicaciones realizadas por otros usuarios	2 hrs	Jeyson Guevara
	Desarrollar la funcionalidad que permita a los agricultores comentar las publicaciones de otros usuarios utilizando mensajes de texto	2 hrs	Jeyson Guevara
	Diseñar y crear la base de datos	1 hrs	Jeyson Guevara
	Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la creación de publicaciones y comentarios.	1 hrs	Ruddy Mendoza

H14	Marcadores en el mapa de plagas y enfermedades	Diseñar una interfaz de usuario en la aplicación web que permita agregar marcadores en el mapa de plagas y enfermedades de manera intuitiva.	1 hrs	Ramón González
		Crear un diseño de pantalla que incluya un campo para ingresar el nombre de la plaga o enfermedad detectada al agregar un marcador.	1 hrs	Ramón González
		Diseñar elementos visuales que indiquen claramente cómo colocar el marcador de manera precisa en el mapa interactivo.	1 hrs	Ramón González
		Implementar la funcionalidad que permita a las instituciones agregar marcadores en el mapa interactivo desde la aplicación web	2 hrs	Jeyson Guevara
		Programar la lógica que permita a los usuarios agregar el nombre de la plaga o enfermedad detectada al colocar un marcador en el mapa.	2 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que garantice la colocación precisa del marcador en la ubicación deseada en el mapa interactivo.	1 hrs	Jeyson Guevara
		Integrar una base de datos que almacene la información de ubicaciones de marcadores junto con los nombres de plagas o enfermedades detectadas	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y detalles clave al agregar marcadores en el mapa.	1 hrs	Ruddy Mendoza
H16	Inicio de sesión	Diseñar una interfaz de inicio de sesión que sea clara y fácil de usar.	1 hrs	Ramón González
		Crear un diseño de formulario de inicio de sesión	1 hrs	Ramón González
		Diseñar elementos visuales que guíen a los usuarios a través del proceso de autenticación de manera efectiva.	1 hrs	Ramón González
		Implementar la funcionalidad del formulario de inicio de sesión, incluyendo la validación de campos y la autenticación de usuarios.	1 hrs	Jeyson Guevara
		Programar la lógica que permita verificar la autenticidad de las credenciales ingresadas por los usuarios.	1 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que muestre las opciones correspondientes una vez que un usuario ha iniciado sesión con éxito.	1 hrs	Jeyson Guevara

Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en el formulario de inicio de sesión.	1 hrs	Ruddy Mendoza
Pruebas de software	4 hrs	Jeyson Guevara

Figura 26*Burndown chart Sprint 2*

Nota. Muestra el progreso aproximado o proyectado del Sprint 2 en función de las tareas cerradas estimadas de la historia de usuario. Las historias de usuarios abiertas y terminadas se observar en la barra superior de color negro. Figura generada por taiga.io.

7.4.2.1. Entregables Sprint 2.

Figura 27*H03 - Pronóstico del tiempo*

Figura 31
H04 – Registro de plagas y enfermedades

Fecha de Detección			Hora de Detección	
01	oct.	2022	4	48
02	nov.	2023	5	: 49
03	dic.	2024	6	50

Figura 30
H08 – Foro agrícola

Figura 29
H16 – Inicio de sesión

Figura 28
H14 - Marcadores en el mapa de plagas y enfermedades

7.4.2.2. Pruebas de software Sprint 2.

A continuación, se muestra el resultado de la ejecución de pruebas donde se puede observar que hubo una cobertura de código del 100% en los componentes.

Figura 32
Pruebas de software Sprint 2

Status	3 passed	3 tests, 41 s 601 ms
Filter tests:	☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
Tests	Duration	LGE LM-X430
✓ Test Results	13 s	3/3
✓ LocationSearchTest ✓ verifyLocationSearch ✓ verifyDeviceLocationsDisplayed ✓ verifyEnteredLocationsDisplayed	13 s 6 s 3 s 3 s	3/3 ✓ ✓ ✓
Filter tests:	☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
Tests	Duration	LGE LM-X430
✓ Test Results	2 s	1/1
✓ HomeFragmentTest ✓ testFabButtonVisibility	2 s 2 s	1/1 ✓
Status	1 passed	1 tests, 30 s 625 ms
Filter tests:	☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
Tests	Duration	LGE LM-X430
✓ Test Results	9 s	1/1
✓ MonitoringFragmentTest ✓ testMonitoringFormSubmission	9 s 9 s	1/1 ✓
Status	1 passed	1 tests, 37 s 890 ms
Filter tests:	☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
Tests	Duration	LGE LM-X430
✓ Test Results	2 s	1/1
✓ HomeFragmentTest ✓ testLoginMessageVisibility	2 s 2 s	1/1 ✓

Elements	Console	Sources	Network	Performance	>>	9	⚙️	⋮	×
☐ Hide network ☐ Preserve log ☐ Selected context only ☒ Group similar messages in console ☒ Show CORS errors in console	☐ Log XMLHttpRequests ☒ Eager evaluation ☒ Autocomplete from history ☒ Treat code evaluation as user action	Marcador guardado con ID: DB1Ibss5G7XNNrMFQ7Vz con el nombre de: monitoreo.js:706 Chicharrita verde							

Nota. Los componentes probados son DashboardFrament, MonitoringFragment, HomeFragment, Monitoreo. Figura generada por Android Studio y consola de Google Chrome.

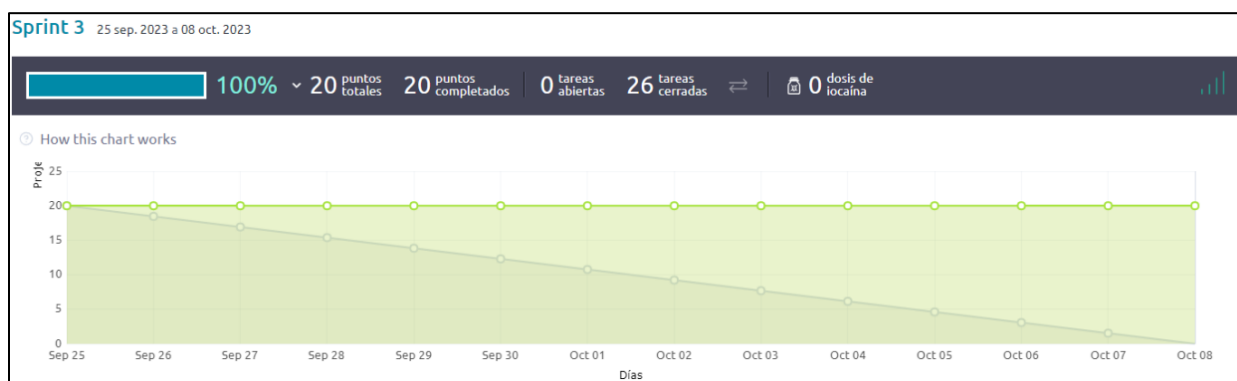
7.4.3. Sprint 3

El tercer sprint tiene una duración de 3 semanas y consta de 4 historias de usuario y 26 tareas en total, incluyendo las pruebas de software correspondientes.

Tabla 33
Sprint 3

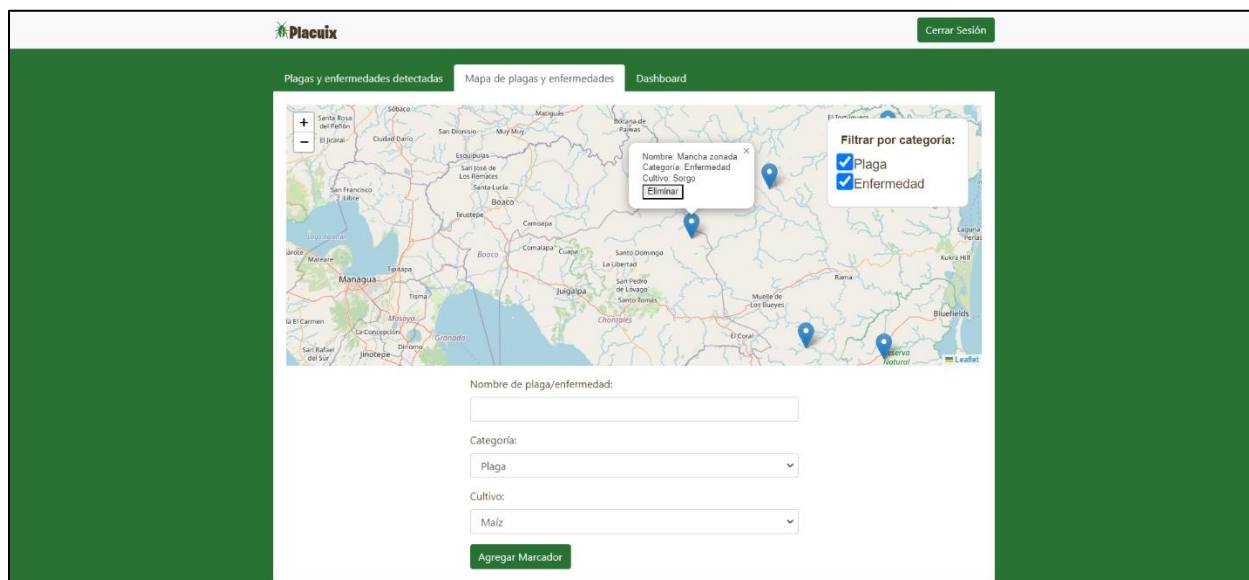
ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H02	Clima actual	Diseñar una interfaz de usuario que permita ver el clima actual de manera clara y fácil de entender.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre de manera efectiva los datos climáticos	1 hrs	Ramón González
		Diseñar elementos visuales que reflejen el estado del día de forma intuitiva (soleado, nublado, lluvioso, etc.).	1 hrs	Ramón González
		Implementar la funcionalidad que permita obtener automáticamente la ubicación actual del agricultor o permitirle buscar una ubicación específica.	2 hrs	Jeyson Guevara
		Programar la lógica para obtener datos precisos y actualizados regularmente sobre el clima	2 hrs	Jeyson Guevara
		Desarrollar la funcionalidad para mostrar esta información en tiempo real en la interfaz de usuario.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave relacionados con la visualización de datos climáticos.	1 hrs	Ruddy Mendoza
H07	Mapa de plagas y enfermedades	Diseñar una interfaz de usuario atractiva y fácil de usar para el mapa interactivo de plagas y enfermedades.	2 hrs	Ramón González
		Crear un diseño de pantalla que muestre de manera efectiva las áreas donde se han detectado plagas y enfermedades en los cultivos.	2 hrs	Ramón González
		Diseñar elementos visuales que permitan a los usuarios hacer zoom y explorar el mapa para obtener detalles específicos de ubicación.	1 hrs	Ramón González
		Implementar la funcionalidad del mapa interactivo	3 hrs	Jeyson Guevara
		Programar la lógica para permitir a los usuarios hacer zoom y explorar el mapa	1 hrs	Jeyson Guevara

H11	Registro de agricultor	para obtener detalles de ubicación		
		Desarrollar las herramientas de búsqueda y filtro que permitan a los usuarios encontrar información sobre plagas y enfermedades en áreas específicas.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización del mapa de plagas y enfermedades.	1 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario clara y atractiva que permita acceder a la opción de registro desde la pantalla de inicio de sesión de la aplicación.	1 hrs	Ramón González
		Crear un diseño de formulario de registro que incluya campos para recopilar la información necesaria.	1 hrs	Ramón González
		Diseñar elementos visuales que guíen a los usuarios a través del proceso de registro de manera efectiva.	1 hrs	Ramón González
		Implementar la funcionalidad que permita acceder a la opción de registro desde la pantalla de inicio de sesión de la aplicación.	2 hrs	Jeyson Guevara
		Programar la lógica para validar y procesar la información ingresada por los Agricultores en el formulario de registro.	1 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que envíe una confirmación de registro exitoso una vez que el proceso de registro esté completo.	1 hrs	Jeyson Guevara
		Integrar un sistema de autenticación que permita a los Agricultores iniciar sesión en sus cuentas después de registrarse para acceder a las funcionalidades de la aplicación.	1 hrs	Jeyson Guevara
H17	Dashboard de plagas y enfermedades detectadas	Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en el formulario de registro.	1 hrs	Ruddy Mendoza
		Diseño de gráficos por cultivo y departamento	4 hrs	Ramón González
		Investigación de librerías de implementación de gráficos	2 hrs	Ruddy Mendoza
		Desarrollo de la funcionalidad del dashboard	6 hrs	Jeyson Guevara
		Pruebas de software	3 hrs	Jeyson Guevara

Figura 33*Burndown chart Sprint 3*

Nota. Muestra el progreso aproximado o proyectado del Sprint 3 en función de las tareas cerradas estimadas de la historia de usuario. Las historias de usuarios abiertas y terminadas se observar en la barra superior de color negro. Figura generada por taiga.io.

7.4.3.1. Entregables Sprint 3.

Figura 34*H07 - Mapa de plagas y enfermedades*

7.4.3.2. Pruebas de software Sprint 3.

En la figura 38 se muestra el resultado de la ejecución de pruebas del sprint 3 donde se puede observar que hubo una cobertura de código del 100% en los componentes.

Figura 38

Pruebas de software Sprint 3

The screenshot shows the Chrome DevTools Console with the following content:

- Console Log:** A list of markers with names and file locations:
 - Nombre del marcador: Gusano cogollero [monitoreo.js:665](#)
 - Nombre del marcador: Chicharrita verde [monitoreo.js:665](#)
 - Nombre del marcador: Gusano cogollero [monitoreo.js:665](#)
 - Nombre del marcador: Gallina ciega [monitoreo.js:665](#)
 - Nombre del marcador: Gusano cogollero [monitoreo.js:665](#)
 - Nombre del marcador: Mosquita del sorgo [monitoreo.js:665](#)
 - Nombre del marcador: Mosaico dorado [monitoreo.js:665](#)
 - Nombre del marcador: Mosaico dorado [monitoreo.js:665](#)
 - Nombre del marcador: Roya del maíz [monitoreo.js:665](#)
 - Nombre del marcador: Mancha zonada [monitoreo.js:665](#)
 - Nombre del marcador: Gusano cogollero [monitoreo.js:665](#)
 - Nombre del marcador: Mosaico dorado [monitoreo.js:665](#)
 - showPlaga: false [monitoreo.js:795](#)
 - showEnfermedad: false [monitoreo.js:796](#)
 - 6 Marker category: Plaga [monitoreo.js:802](#)
 - 4 Marker category: Enfermedad [monitoreo.js:802](#)
- Test Results Section 1:**
 - Status: 1 passed, 1 tests
 - Filter tests: [Icons]
 - Tests Table:

Tests	Duration	LGE LM-X430
✓ Test Results	4 s	1/1
✓ LoginActivityPerformanceTest	4 s	1/1
✓ testLoginPerformance	4 s	✓
- Test Results Section 2:**
 - Status: 2 passed, 2 tests, 34 s 593 ms
 - Filter tests: [Icons]
 - Tests Table:

Tests	Duration	LGE LM-X430
✓ Test Results	14 s	2/2
✓ RegisterActivityTest	14 s	2/2
✓ testRegistrationSuccess	8 s	✓
✓ testRegistrationFailure	6 s	✓

Nota. Los componentes probados son Monitoreo, RegisterActivity, LoginActivity, Figura generada por Android Studio y consola de Google Chrome.

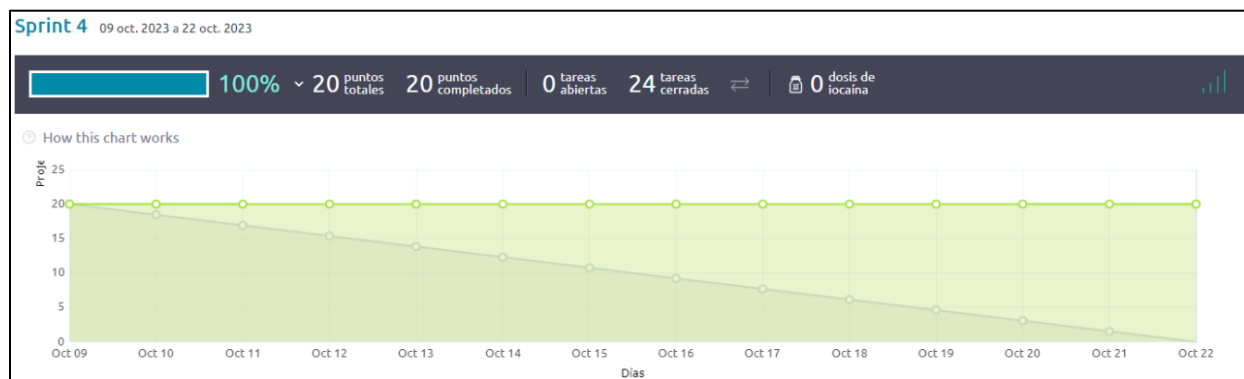
7.4.4. Sprint 4

El cuarto sprint tiene una duración de 2 semanas y consta de 4 historias de usuario y 24 tareas en total, incluyendo las pruebas de software correspondientes.

Tabla 34
Sprint 4

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H19	Manual de introducción	Diseño de las pantallas del manual	8 hrs	Ramón González
		Describir las funcionalidades principales	2 hrs	Ruddy Mendoza
		Desarrollo de la lógica del manual	10 hrs	Jeyson Guevara
		Funcionalidad de deslizamiento e intuitividad	4 hrs	Jeyson Guevara
H05	Historial de plagas y enfermedades	Diseñar una interfaz de usuario que permita ver el historial de plagas y enfermedades en un formato de tabla de manera clara y organizada.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre la tabla de registros de plagas y enfermedades de forma eficiente.	1 hrs	Ramón González
		Diseñar elementos visuales que faciliten la navegación y la interacción con la tabla, incluyendo opciones de ordenamiento y filtrado.	2 hrs	Ramón González
		Implementar la funcionalidad que permita cargar y mostrar la tabla de registros de plagas y enfermedades en la interfaz de usuario.	1 hrs	Jeyson Guevara
		Programar la lógica para obtener y presentar la información detallada de cada entrada en la tabla	2 hrs	Jeyson Guevara
		Desarrollar las opciones de ordenamiento y filtrado	2 hrs	Jeyson Guevara
		Integrar base de datos que mantenga actualizado el historial de plagas y enfermedades	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización de la tabla de historial.	1 hrs	Ruddy Mendoza
H06	Mapa de zonas de	Diseñar una interfaz de usuario que permita acceder al mapa interactivo de	1 hrs	Ramón González

H09	cultivos de granos básicos	zonas de cultivos de granos básicos en Nicaragua de manera clara y atractiva.		
		Crear un diseño de pantalla que muestre el mapa de forma eficiente.	1 hrs	Ramón González
		Implementar la funcionalidad que permita cargar y mostrar el mapa interactivo de zonas de cultivos de granos básicos en Nicaragua en la interfaz de usuario.	1 hrs	Jeyson Guevara
	Perfil del agricultor	Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización del mapa de zonas de cultivos de granos básicos.	1 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario que permita visualizar y actualizar su información de perfil de manera clara y atractiva.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre la sección de edición de perfil en la configuración de la aplicación.	1 hrs	Ramón González
		Diseñar elementos visuales que permita cambiar imagen de perfil, editar nombre de usuario, profesión y ubicación.	1 hrs	Ramón González
		Implementar la funcionalidad que permita acceder a la sección de edición de perfil desde la configuración de la aplicación.	2 hrs	Jeyson Guevara
		Programar la lógica para cargar y cambiar la imagen de perfil, así como para editar el nombre de usuario, profesión y ubicación.	1 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que guarde los cambios realizados en el perfil y proporcione una confirmación de que han sido guardados correctamente.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización y edición del perfil.	1 hrs	Ruddy Mendoza
		Pruebas de software	3 hrs	Jeyson Guevara

Figura 39**Burndown chart Sprint 4**

Nota. Muestra el progreso aproximado o proyectado del Sprint 4 en función de las tareas cerradas estimadas de la historia de usuario. Las historias de usuarios abiertas y terminadas se observar en la barra superior de color negro. Figura generada por taiga.io.

7.4.4.1. Entregables Sprint 4.

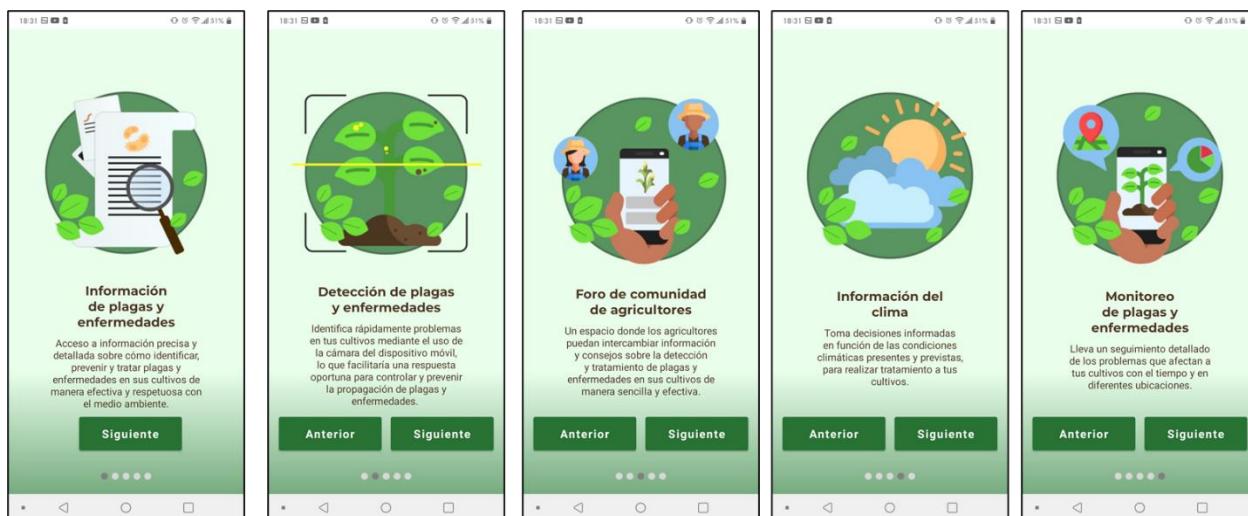
Figura 40**H19 - Manual de introducción**

Figura 43
H05 - Historial de plagas y enfermedades

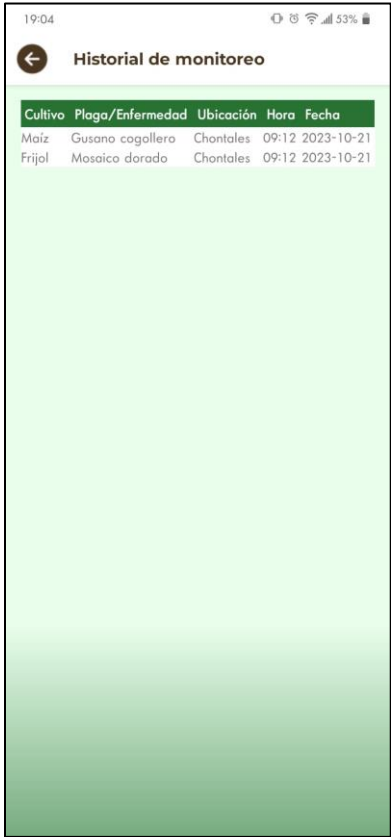


Figura 42
H06 - Mapa de zonas de cultivos de granos básicos

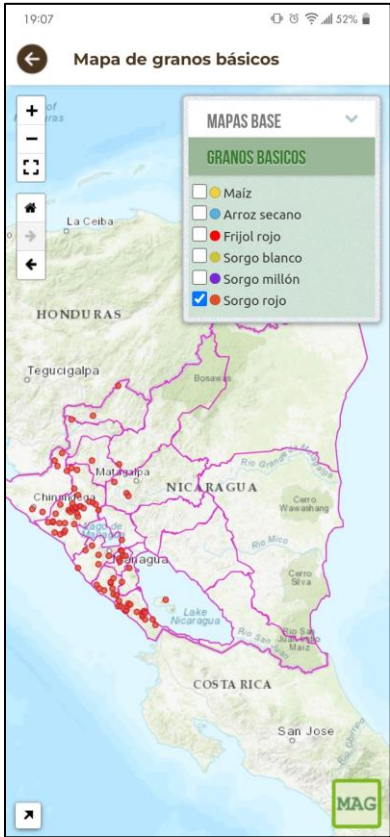


Figura 41
H09 - Perfil del agricultor



7.4.4.2. Pruebas de software Sprint 4.

En la figura 44 se muestra el resultado de la ejecución de pruebas del sprint 4 donde se puede observar que hubo una cobertura de código del 100% en los componentes.

Figura 44

Pruebas de software Sprint 4

Status	1 passed	1 tests, 23 s 832 ms
Filter tests:         		
Tests	Duration	LGE LM-X430
✓ Test Results	1 s	1/1
✓ HistoryActivityTest ✓ testHistoryTableContents	1 s 1 s	1/1 ✓
Status	1 passed	1 tests, 36 s 777 ms
Filter tests:         		
Tests	Duration	LGE LM-X430
✓ Test Results	2 s	1/1
✓ MAGActivityTest ✓ testMapLoading	2 s 2 s	1/1 ✓
Status	2 passed	2 tests, 59 s 57 ms
Filter tests:         		
Tests	Duration	LGE LM-X430
✓ Test Results	25 s	2/2
✓ ProfileFragmentTest ✓ testProfileFragmentElements ✓ testProfileFragmentSaveChanges	25 s 2 s 22 s	2/2 ✓ ✓

Nota. Los componentes probados son HistoryActivity, MAGActivity, ProfileFragment. Figura generada por Android Studio.

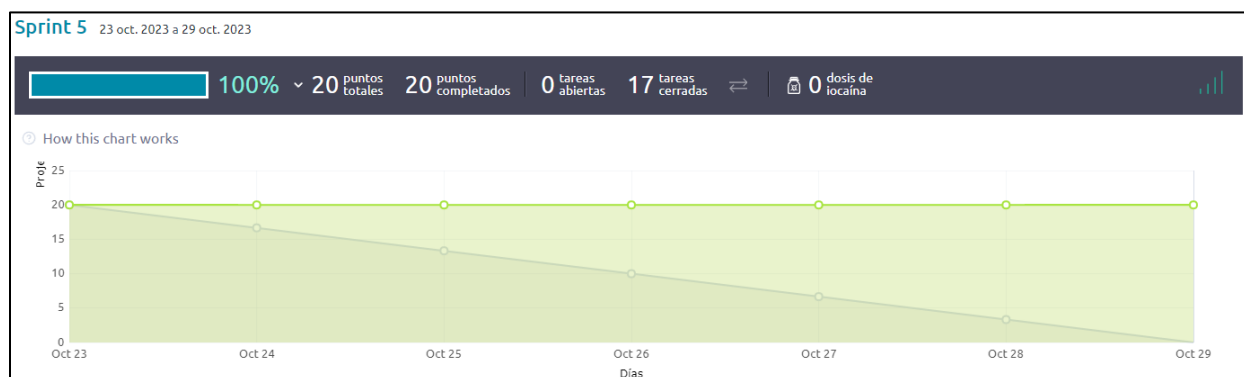
7.4.5. Sprint 5

El quinto sprint tiene una duración de 1 semana y consta de 4 historias de usuario y 17 tareas en total, incluyendo las pruebas de software correspondientes.

Tabla 35
Sprint 5

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H12	Visualización de calificaciones de la aplicación	Diseñar una interfaz de usuario que permita ver las calificaciones de la aplicación de manera clara y efectiva.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre las calificaciones en forma de gráfico de pastel.	1 hrs	Ramón González
		Diseñar elementos visuales que indiquen claramente si una calificación fue considerada "útil" o "no útil"	1 hrs	Ramón González
		Implementar la funcionalidad que permita acceder y visualizar las calificaciones de la aplicación.	2 hrs	Jeyson Guevara
		Programar la lógica para mostrar las calificaciones en un gráfico de pastel, con indicadores claros de "útil" o "no útil".	2 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir la presentación adecuada de las calificaciones de la aplicación.	1 hrs	Ruddy Mendoza
H10	Calificación de la aplicación	Diseñar una ventana emergente que invite a calificar la aplicación de manera atractiva y no intrusiva.	1 hrs	Ramón González
		Crear elementos visuales claros para las opciones de calificación, como "útil" o "no útil".	1 hrs	Ramón González
		Implementar la funcionalidad de la ventana emergente que aparecerá para invitar a calificar la aplicación.	1 hrs	Jeyson Guevara
		Programar la lógica para registrar la calificación seleccionada en la base de datos de la aplicación.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir la apariencia adecuada de la ventana emergente de calificación.	1 hrs	Ruddy Mendoza
H13		Diseñar elementos visuales que permitan mostrar de manera clara y efectiva el	1 hrs	Ramón González

		número de usuarios registrados en la interfaz.		
Cantidad de usuarios registrados		Programar la lógica para obtener el número total de usuarios registrados y mostrarlo claramente en un elemento de texto en la interfaz.	1 hrs	Jeyson Guevara
		Proporcionar orientación sobre cómo presentar la información de manera efectiva para que pueda ser interpretada fácilmente.	1 hrs	Ruddy Mendoza
H18 Splash Screen		Diseño del splash screen	1 hrs	Ramón González
		Desarrollo de la lógica del splash screen	2 hrs	Jeyson Guevara
		Pruebas del software	3 hrs	Jeyson Guevara

Figura 45*Burndown chart Sprint 5*

Nota. Muestra el progreso aproximado o proyectado del Sprint 5 en función de las tareas cerradas estimadas de la historia de usuario. Las historias de usuarios abiertas y terminadas se observan en la barra superior de color negro. Figura generada por taiga.io.

7.4.5.1. Entregables Sprint 5.

Figura 46

H12 - Visualización de calificaciones de la aplicación

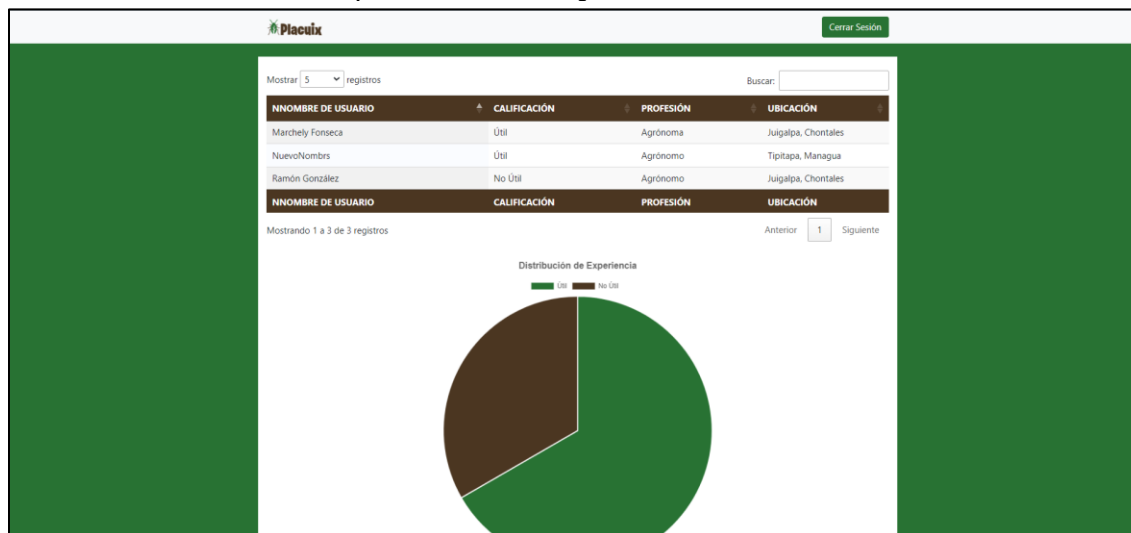


Figura 47

H13 - Cantidad de usuarios registrados



Figura 49

H10 - Calificación de la aplicación



Figura 48

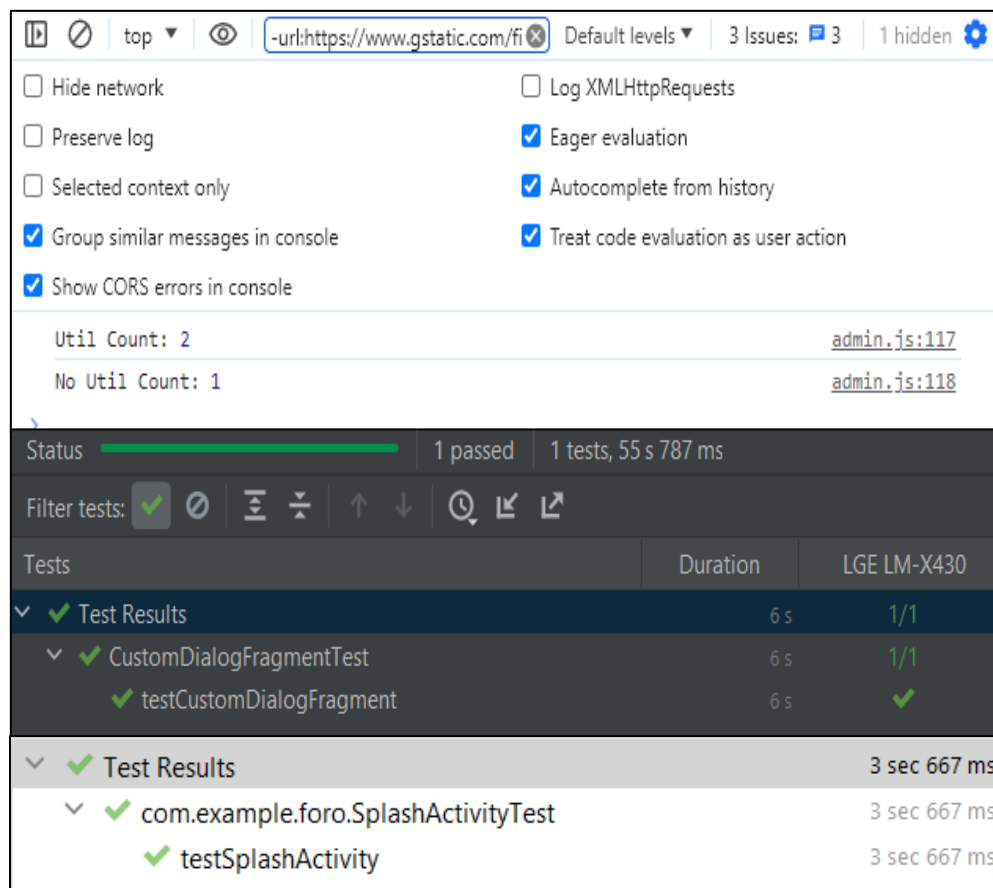
H18 - Splash Screen



7.4.5.2. Pruebas de software Sprint 5.

En la figura 50 se muestra el resultado de la ejecución de pruebas del sprint 5 donde se puede observar que hubo una cobertura de código del 100% en los componentes.

Figura 50
Pruebas de software Sprint 5



Nota. Los componentes probados son Admin, ProfileFragment, SplashAcitivity. Figura generada por Android Studio y consola de Google Chrome.

VIII. Presupuesto

A continuación, se proporciona el desglose financiero que incluye los servicios a ser contratados, los recursos humanos necesarios, así como la adquisición de software y equipos esenciales para la ejecución del proyecto.

Tabla 36*Presupuesto para desarrollo del proyecto*

	Tasa/Mes	Meses	Total
Recursos Humanos			
Desarrollador	C\$7.712,00	4	C\$30.848,00
Diseñador gráfico	C\$6.932,00	4	C\$27.728,00
Investigador	C\$5.397,00	4	C\$21.588,00
Servicios			
Energía Eléctrica	C\$1.560,00	4	C\$6.240,00
Internet	C\$1.850,00	4	C\$7.400,00
Transporte	C\$1.380,00	4	C\$5.520,00
Alimentación	C\$12.000,00	4	C\$48.000,00
Agua	C\$730,00	4	C\$2.920,00
Cuenta de desarrollador de Google Play	C\$914,00	-	C\$914,00
Gestor de base de datos	C\$5.666,80	4	C\$22.667,20
Hosting (Mensual)	C\$840,00	4	C\$3.360,00
Dominio (Anual)	C\$383,00	-	C\$383,00
Software			
Android Studio	-	4	C\$0,00
Teachable Machine	-	4	C\$0,00
Visual Studio Code	-	4	C\$0,00
Figma	-	4	C\$0,00
Adobe Creative Cloud			
• Adobe Photoshop	C\$1940,00	1	C\$1940,00
• Adobe Illustrator			
• Adobe XD			
	Valor unitario por depreciación	Meses	Total
Equipo			
Papelería	1000	-	C\$1.000,00
Computadora 1	450	4	C\$1.800,00
Computadora 2	313	4	C\$1.252,00
Computadora 3	233	4	C\$932,00
Celular 1	300	4	C\$1.200,00
Celular 2	200	4	C\$800,00
Celular 3	125	4	C\$500,00
Monto total			C\$186.992,20

IX. Marco lógico

Esta sección se enfoca en la conceptualización, diseño, implementación y evaluación de un proyecto utilizando la metodología del marco lógico. La herramienta principal que resulta de esta metodología analítica es la Matriz de marco lógico, que resume los objetivos del proyecto, cómo se planea lograrlos, los supuestos claves involucrados y cómo se llevará a cabo el seguimiento y la evaluación de los recursos y resultados del proyecto.

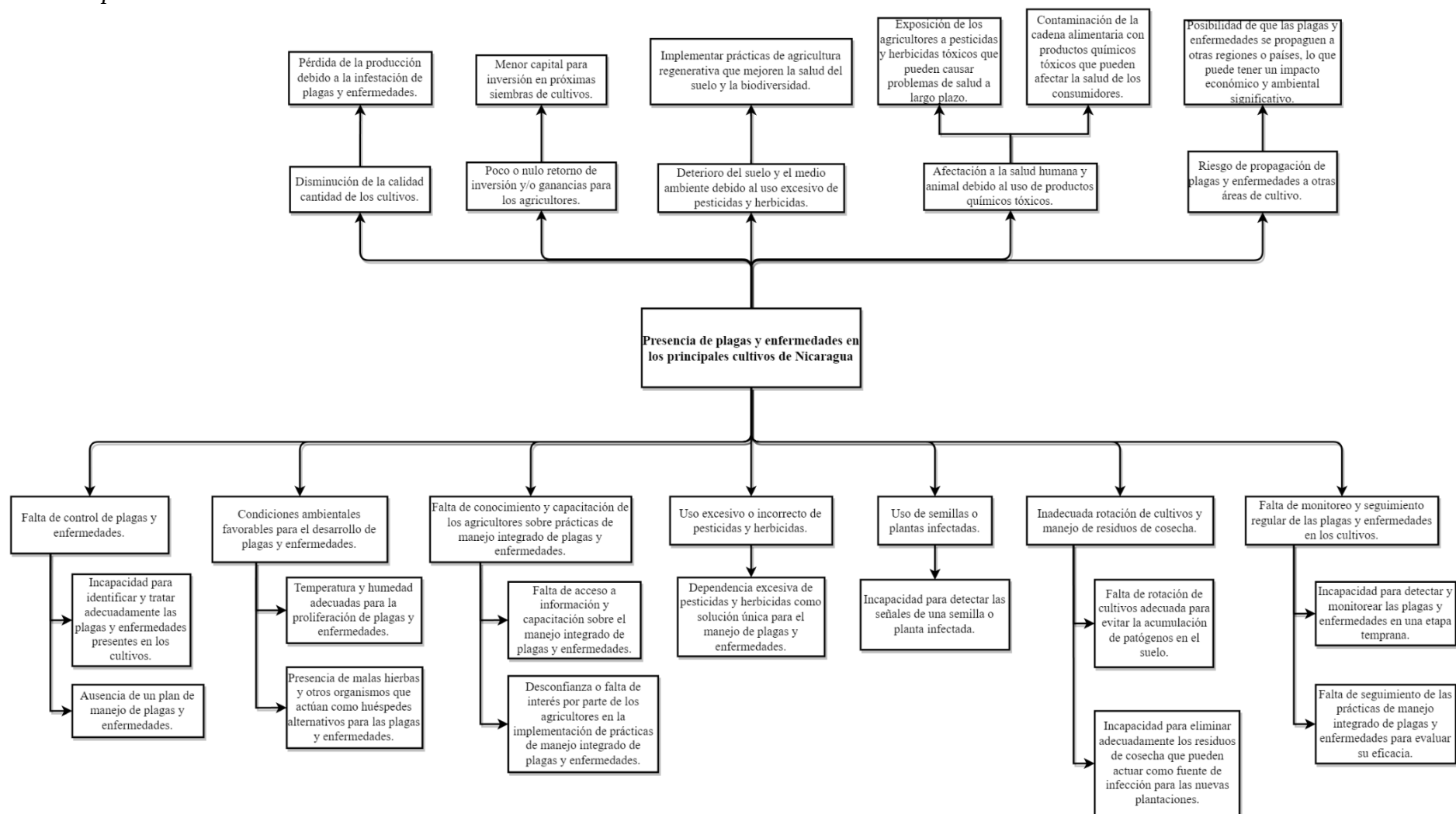
Según la CEPAL, en su SERIE manuales N° 42 (2005), este enfoque se divide en dos etapas que se desarrollan paso a paso durante las fases de identificación y diseño en el ciclo de vida del proyecto:

- **Identificación del problema y alternativas de solución:** En esta fase, se analiza la situación actual para crear una visión de la situación deseada y seleccionar las estrategias necesarias para alcanzarla. Se llevan a cabo cuatro tipos de análisis: el análisis de las partes interesadas, el análisis de los problemas, el análisis de los objetivos y el análisis de las estrategias. Durante esta etapa, se desarrollan herramientas como el árbol del problema, el árbol de objetivos y el árbol de la solución.
- **Etapas de planificación:** En esta etapa, la idea del proyecto se traduce en un plan operativo concreto para su ejecución. La Matriz de marco lógico se elabora en esta fase, en la que se definen las actividades, los recursos necesarios y se establece una línea de tiempo para su implementación.

9.1. Árbol del problema

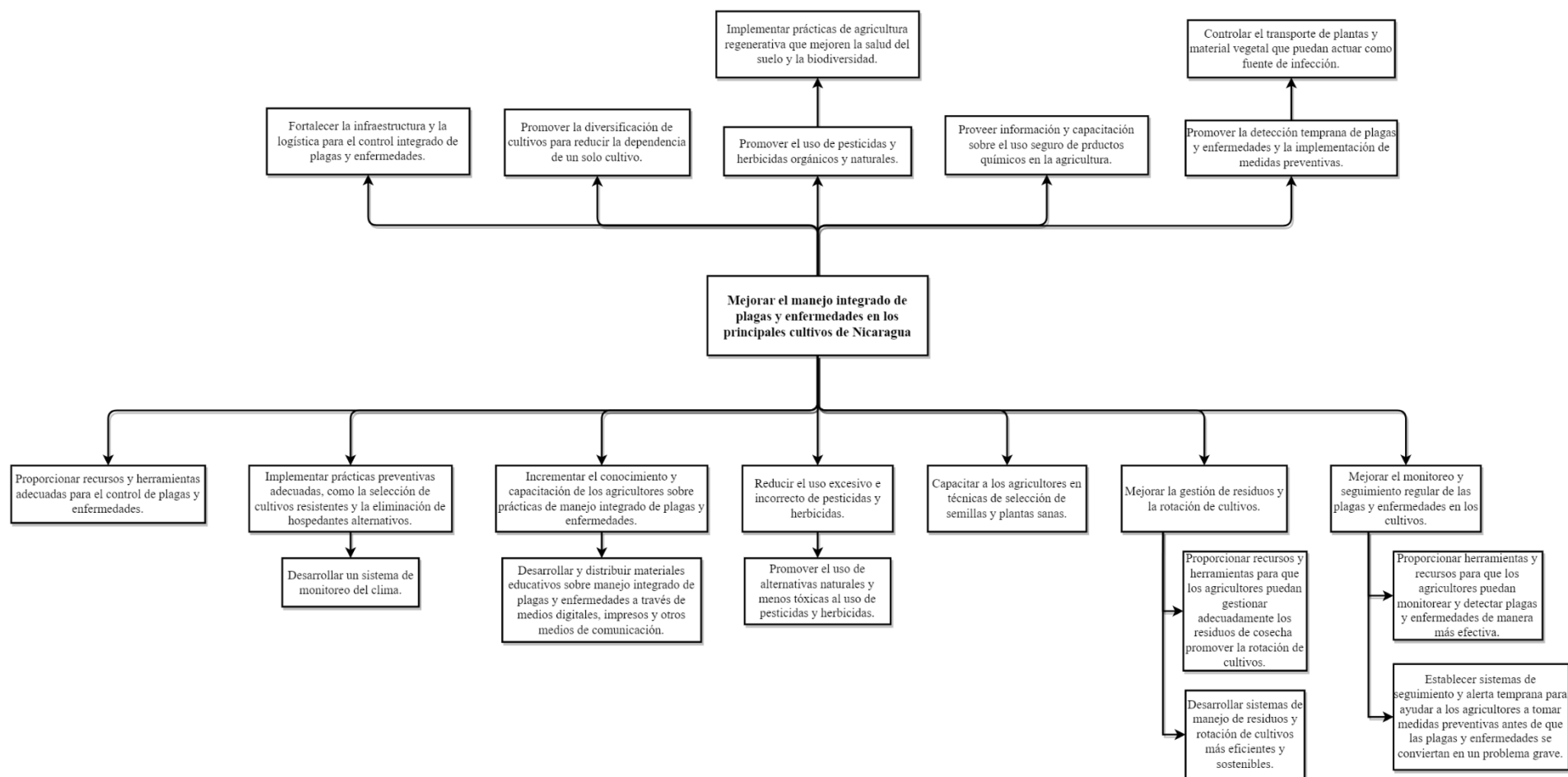
Figura 51

Árbol del problema



9.2. Árbol de objetivos

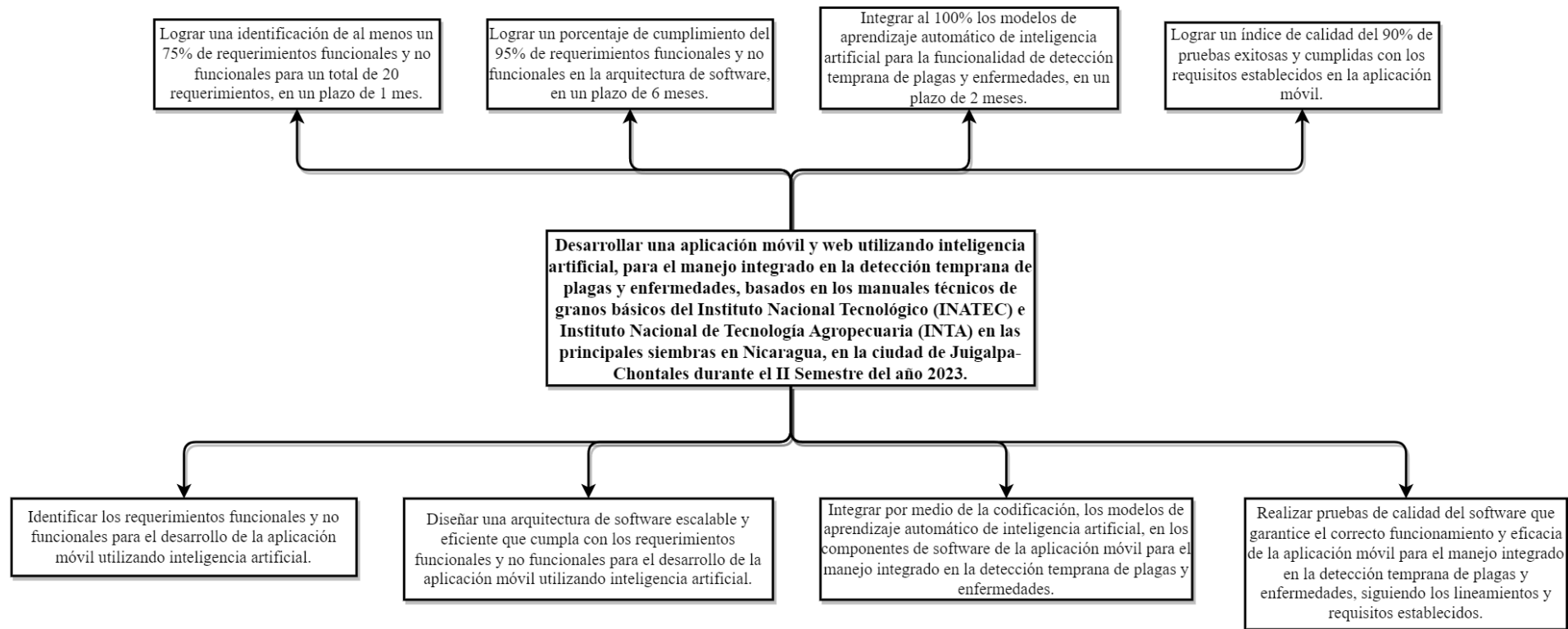
Figura 52
Árbol de objetivos



9.3. Árbol de solución V1

Figura 53

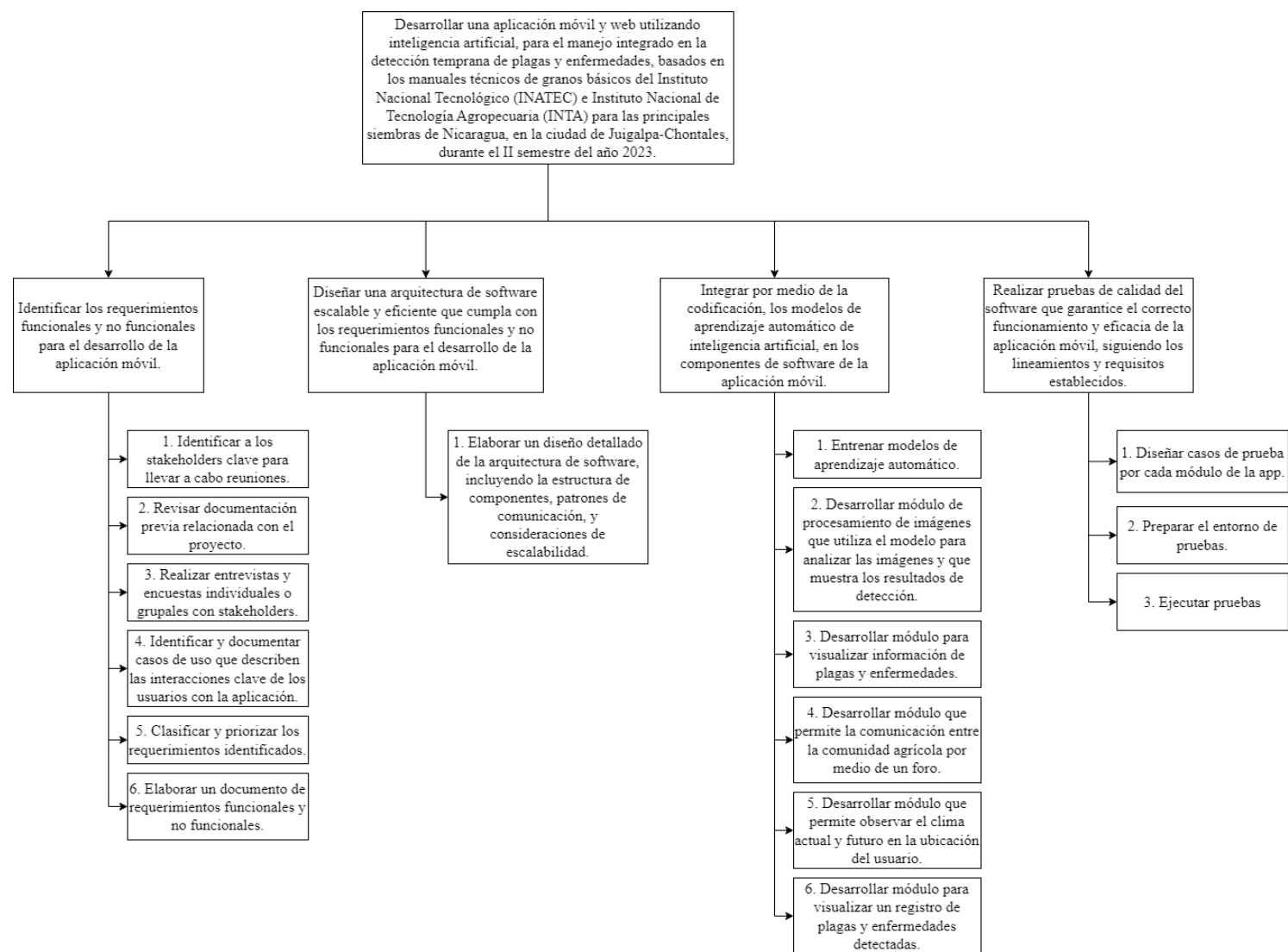
Árbol de solución V1



9.4. Árbol de Solución V2

Figura 54

Árbol de solución V2



9.5. Matriz de marco lógico

Tabla 37

Matriz de marco lógico

Resumen narrativo de objetivos	Descripción	Indicador	Medios de verificación	Supuestos
Fin	Mejorar el manejo integrado de plagas y enfermedades en los principales cultivos de Nicaragua.	● Reducción del uso de pesticidas químicos en un 30% en los principales cultivos de Nicaragua en un plazo de 24 meses. ● Aumento del 15% en el uso de prácticas de manejo integrado en cultivos en un plazo de 24 meses.	● Registros de ventas de pesticidas químicos en el Ministerio Agropecuario de Nicaragua. ● Encuestas a agricultores para recopilar datos sobre el uso del MIPE. ● Informe con datos en el Plan Nacional de la lucha contra la pobreza y para el desarrollo humano.	● Disponibilidad de alternativas efectivas y sostenibles para el control de plagas y enfermedades. ● Colaboración activa de los agricultores en la adopción de prácticas de manejo integrado. ● Acceso a financiamiento continuo para la implementación del proyecto.

Propósito	Desarrollar una aplicación móvil y web utilizando inteligencia artificial, para el manejo integrado en la detección temprana de plagas y enfermedades, basados en los manuales técnicos de granos básicos del Instituto Nacional Tecnológico (INATEC) e Instituto Nacional de Tecnología Agropecuaria (INTA) en las principales siembras de Nicaragua, en la ciudad de Juigalpa-Chontales durante el segundo Semestre del año 2023.	● 100% de cumplimiento del cronograma de actividades. ● Desarrollo de la aplicación móvil en 90% durante el II semestre del 2023.	● Cantidad de módulos y características completadas durante el II semestre del año 2023.	● Acceso generalizado a dispositivos móviles y conectividad en las zonas agrícolas de Nicaragua. ● Capacitación efectiva para los agricultores. ● Interés y participación activa de los usuarios en el uso de la aplicación móvil.
Componentes	1. Identificar los requerimientos funcionales y no funcionales para el desarrollo de la aplicación.	● Lograr una identificación de al menos un 75% de requerimientos funcionales y no funcionales para un total de 20 requerimientos, en un plazo de 1 mes.	● Documento formal que enumera y describe todos los requerimientos funcionales y no funcionales identificados.	● Acceso a los stakeholders clave para obtener información relevante sobre los requerimientos.
	2. Diseñar una arquitectura de software escalable y eficiente que cumpla con los requerimientos funcionales y no funcionales para el desarrollo de la aplicación.	● Lograr un porcentaje de cumplimiento del 95% de requerimientos funcionales y no funcionales en la	● Documento técnico que describe la arquitectura de software diseñada, incluyendo los componentes, las	● Acceso a las tecnologías para implementar la arquitectura diseñada de manera eficiente y efectiva.

	arquitectura de software, en un plazo de 6 meses.	relaciones entre ellos y cómo se cumplen los requerimientos.	
3. Integrar por medio de la codificación, los modelos de aprendizaje automático de inteligencia artificial, en los componentes de software de la aplicación.	Integrar al 100% los modelos de aprendizaje automático de inteligencia artificial en los módulos de la aplicación en un plazo de 2 meses.	Registros de código y documentación que demuestran la integración de los modelos de IA con los componentes de la aplicación móvil.	Suficiente potencia de cómputo y recursos de hardware para ejecutar los modelos de IA y su relación con los componentes de manera eficiente en dispositivos móviles
4. Realizar pruebas de calidad del software que garantice el correcto funcionamiento y eficacia de la aplicación, siguiendo los lineamientos y requisitos establecidos.	Lograr un índice de calidad del 90% de pruebas exitosas y cumplidas con los requisitos establecidos en la aplicación móvil.	Plan detallado que describe los tipos de pruebas a realizar, los criterios de aceptación y los recursos asignados con sus resultados.	Disposición de entornos de prueba adecuados para llevar a cabo pruebas exhaustivas sin afectar la producción.
Actividades Componente 1 Identificar los requerimientos funcionales y no funcionales para el desarrollo de la aplicación.	Requerimientos funcionales y no funcionales identificados al 100% durante el II semestre del 2023.	Documento formal que enumera y describe todos los requerimientos funcionales y no funcionales identificados.	Colaboración de manera efectiva entre el equipo de desarrollo y los stakeholders para recopilar y validar los requerimientos funcionales y no funcionales.

1. Identificar a los stakeholders clave para llevar a cabo reuniones.	• 10 stakeholders identificados	• Registro de participantes en las reuniones.	• Los stakeholders clave están dispuestos a participar en reuniones iniciales.
2. Revisar documentación previa relacionada con el proyecto.	• Porcentaje de información relevante recopilada de la documentación existente.	• Documento de resumen de la información recopilada.	• La documentación previa es accesible y completa
3. Realizar entrevistas y encuestas individuales o grupales con stakeholders.	• Porcentaje de respuestas recibidas.	• Datos recopilados de encuestas y entrevistas.	• Las respuestas proporcionadas son honestas y representan las necesidades reales.
4. Identificar y documentar casos de uso que describen las interacciones clave de los usuarios con la aplicación.	• Porcentaje de requerimientos priorizados.	• Documentación de casos de uso.	• Los stakeholders pueden proporcionar retroalimentación efectiva sobre los casos de uso.
5. Clasificar y priorizar los requerimientos identificados.	• Porcentaje de requerimientos priorizados.	• Documento de priorización de requerimientos.	• Los stakeholders pueden participar en la clasificación y priorización.
6. Elaborar un documento de requerimientos funcionales y no funcionales.	• Documento de requerimientos funcionales y no funcionales completado al 100%.	• Documento de requerimientos.	• Los stakeholders pueden revisar y aprobar la documentación.
Componente 2 Diseñar una arquitectura de software escalable y eficiente	• Estructura y componentes de la arquitectura de la	• Documento de diseño detallado de la arquitectura.	• El diseño detallado cumple con la arquitectura de

que cumpla con los requerimientos funcionales y no funcionales para el desarrollo de la aplicación.	aplicación definidos al 100% durante el II semestre del 2023.		referencia y los requerimientos.
1. Elaborar un diseño detallado de la arquitectura de software, incluyendo la estructura de componentes, patrones de comunicación, y consideraciones de escalabilidad.	• Diseño detallado de la arquitectura completado al 100%.	• Gráfico de diseño detallado de la arquitectura mediante UML.	• Surgimiento de limitaciones técnicas.
Componente 3 Integrar por medio de la codificación, los modelos de aprendizaje automático de inteligencia artificial, en los componentes de software de la aplicación.	• 100% de modelos de aprendizaje automático integrados con éxito en componentes de la aplicación móvil durante el II semestre del 2023.	• Registro de implementación de modelos de aprendizaje automático en los componentes de software de la aplicación móvil.	• Los recursos de hardware y software necesarios para la ejecución de los modelos están disponibles y funcionales.
1. Entrenar modelos de aprendizaje automático.	• 100% de modelos entrenados al finalizar el Sprint 1.	• Modelo de aprendizaje automático en Google Teachable Machine.	• Surgimiento de fallas en el software de entrenamiento de modelos de aprendizaje automático.
2. Desarrollar módulo de procesamiento de imágenes que utiliza el modelo para analizar las imágenes y que muestra los resultados de detección.	• Módulo operando al 100% al finalizar el Sprint 1.	• Interfaz que muestra los resultados de detección de plagas y enfermedades en imágenes analizadas.	• Cambios en los recursos de la tecnología usada para el procesamiento de modelos.

3. Desarrollar módulo para visualizar información de plagas y enfermedades.	• Módulo operando al 100% al finalizar el Sprint 2.	• Interfaz que muestra lista de plagas y enfermedades con información detallada.	• Inactividad de la base de datos por parte del proveedor.
4. Desarrollar módulo que permite la comunicación entre la comunidad agrícola por medio de un foro.	• Módulo operando al 100% al finalizar el Sprint 4.	• Interfaz que muestra publicaciones de usuarios y permite comentarlas, • Opción que permite crear publicaciones a través de preguntas con imágenes.	• Inactividad de la base de datos por parte del proveedor.
5. Desarrollar módulo que permite observar el clima actual y futuro en la ubicación del usuario.	• Módulo operando al 100% al finalizar el Sprint 4.	• Interfaz que muestra el clima de la ubicación del usuario con datos de Temperatura, Humedad, Viento y Nubosidad.	• Caída del servicio de API de clima.
6. Desarrollar módulo para visualizar un registro de plagas y enfermedades detectadas.	• Módulo operando al 100% al finalizar el Sprint 5.	• Interfaz que muestra lista de plagas y enfermedades detectadas por el usuario. • Interfaz que muestra mapa de Nicaragua con plagas y enfermedades detectadas en la zona.	• Fallas en el servicio de API del mapa. • Fallas en la base de datos por parte del proveedor.
Realizar pruebas de calidad del software que garantice el correcto funcionamiento y	• 90% de pruebas de software realizadas en los módulos de la	• Documentación del plan de pruebas de software.	• Acceso a dispositivos móviles para realizar pruebas en diferentes entornos.

eficacia de la aplicación, siguiendo los lineamientos y requisitos establecidos.		aplicación durante el II semestre del 2023.	
1. Diseñar casos de prueba por cada módulo de la app.	• 100% de casos de prueba diseñados.	• Documentación de casos de prueba.	• Los requisitos y lineamientos de prueba proporcionados son claros y completos.
2. Preparar el entorno de pruebas.	• Entorno de pruebas preparado al 100%.	• Registro de configuración del entorno de pruebas.	• Disposición del hardware y software necesario.
3. Ejecutar pruebas	• 90% de casos de pruebas ejecutados durante el II semestre del 2023.	• Informe de seguimiento de defectos.	• Aplicación móvil lista para ser sometida a pruebas y no presenta errores críticos.

X. Cronograma de actividades

La gestión del proyecto para el desarrollo de la aplicación móvil y web Placuix, se ha previsto para realizarse en un total de 11 semanas.

Tabla 38

Cronograma de actividades

EDT	Nombre de la tarea	Fecha de inicio	Fecha final	Duración
PL1	Desarrollo de una aplicación móvil y web para la detección de plagas y enfermedades utilizando inteligencia artificial	14/08/23	29/10/23	77d
PL1.1	Análisis	16/08/23	01/09/23	17d
PL1.1.1	Marco lógico	16/08/23	29/08/23	14d
PL1.1.1.1	Árbol del problema	16/08/23	17/08/23	2d
PL1.1.1.2	Árbol de objetivos	18/08/23	19/08/23	2d
PL1.1.1.3	Árbol de solución	20/08/23	21/08/23	2d
PL1.1.1.4	Matriz de marco lógico	23/08/23	29/08/23	7d
PL1.1.1.4.1	Descripción	23/08/23	24/08/23	2d
PL1.1.1.4.2	Indicador	25/08/23	26/08/23	2d
PL1.1.1.4.3	Medios de verificación	27/08/23	27/08/23	1d
PL1.1.1.4.4	Supuestos	28/08/23	29/08/23	2d
PL1.1.2	Solución	31/08/23	01/09/23	2d
PL1.2	Diseño	01/09/23	15/09/23	15d
PL1.2.1	Diseño / Modelo	02/09/23	02/09/23	1d
PL1.2.2	Diseño de interfaces de la aplicación	01/09/23	05/09/23	5d
PL1.2.2.1	Diseño de interfaces para la aplicación móvil	01/09/23	05/09/23	5d
PL1.2.2.2	Diseño de interfaces para la aplicación web	01/09/23	05/09/23	5d
PL1.2.3	Diseño de navegación	03/09/23	04/09/23	2d
PL1.2.4	Diseño arquitectónico	13/09/23	15/09/23	3d
PL1.3	Justificación de la tecnología	16/09/23	19/09/23	4d
PL1.4	Justificación de la metodología	20/09/23	21/09/23	2d
PL1.5	Elaboración de oportunidad de mercado	22/09/23	22/09/23	1d
PL1.6	Elaboración del modelo CANVAS	23/09/23	26/09/23	4d
PL1.7	Elaboración de propuesta de valor	28/09/23	30/09/23	3d
PL1.8	Definición del ciclo del proyecto según metodología	02/10/23	04/10/23	3d
PL1.8.1	Creación de historias de usuario	02/10/23	02/10/23	1d

PL1.8.2	Definición de tareas	03/10/23	03/10/23	1d
PL1.8.3	Definición de sprints	04/10/23	04/10/23	1d
PL1.9	Desarrollo	14/08/23	29/10/23	77d
PL1.9.1	Sprint 1	14/08/23	03/09/23	21d
PL1.9.1.1	Detección de plagas y enfermedades mediante imágenes	14/08/23	03/09/23	21d
PL1.9.1.2	Información sobre plagas y enfermedades	14/08/23	21/08/23	8d
PL1.9.1.3	Plagas y enfermedades detectadas	22/08/23	25/08/23	4d
PL1.9.2	Sprint 2	04/09/23	24/09/23	21d
PL1.9.2.1	Pronóstico del tiempo	04/09/23	08/09/23	5d
PL1.9.2.2	Registro de plagas y enfermedades	11/09/23	12/09/23	2d
PL1.9.2.3	Foro agrícola	13/09/23	20/09/23	8d
PL1.9.2.4	Marcadores en el mapa de plagas y enfermedades	21/09/23	23/09/23	3d
PL1.9.2.5	Inicio de sesión	24/09/23	24/09/23	1d
PL1.9.3	Sprint 3	25/09/23	08/10/23	14d
PL1.9.3.1	Clima actual	25/09/23	29/09/23	5d
PL1.9.3.2	Mapa de plagas y enfermedades detectadas	01/10/23	05/10/23	5d
PL1.9.3.3	Registro de agricultor	06/10/23	07/10/23	2d
PL1.9.3.4	Dashboard de plagas y enfermedades detectadas	08/10/23	08/10/23	1d
PL1.9.4	Sprint 4	09/10/23	22/10/23	14d
PL1.9.4.1	Manual de introducción	09/10/23	09/10/23	1d
PL1.9.4.2	Historial de plagas y enfermedades	10/10/23	12/10/23	3d
PL1.9.4.3	Mapa de zonas de cultivos de granos básicos	12/10/23	12/10/23	1d
PL1.9.4.4	Perfil del agricultor	13/10/23	22/10/23	10d
PL1.9.5	Sprint 5	23/10/23	29/10/23	7d
PL1.9.5.1	Visualización de calificaciones de la aplicación	23/10/23	24/10/23	2d
PL1.9.5.2	Calificación de la aplicación	25/10/23	27/10/23	3d
PL1.9.5.3	Cantidad de usuarios registrados	28/10/23	28/10/23	1d
PL1.9.5.4	Splash Screen	29/10/23	29/10/23	1d
PL1.10	Pruebas de software	14/08/23	29/10/23	77d
PL1.11	Manual de usuario	23/10/23	29/10/23	7d
PL1.12	Manual técnico	23/10/23	29/10/23	7d

XI. Bibliografía

- Agafonkin, V. (2023). *Leaflet Quick Start Guide*. Recuperado el 2 de Noviembre de 2023, de Leaflet: <https://bit.ly/47ecYjM>
- Agencia de Cooperación Internacional del Japón. (2010). *Guía del Manejo Integrado de Plagas (MIP) para técnicos y productores. Versión 1*. Recuperado el 10 de Octubre de 2023, de JICA: <https://bit.ly/46XhYcM>
- Aguiar, E. U. (2018). *Firestore en el desarrollo de aplicaciones móviles*. Mazatlán. Recuperado el 2 de Noviembre de 2023, de <https://bit.ly/45WwCQe>
- Android Developers. (15 de Agosto de 2023). *Android Studio Electric Eel*. Recuperado el 2 de Nov de 2023, de Developers: <https://bit.ly/49nlwXj>
- Android Developers. (7 de Agosto de 2023). *Introducción a Android Studio*. Recuperado el 4 de Octubre de 2023, de Android Developers: <https://bit.ly/48zi2Rn>
- Android Developers. (21 de Febrero de 2023). *Notas de la versión de la Plataforma del SDK*. Recuperado el 2 de Noviembre de 2023, de Developers: <https://bit.ly/3QwUkNr>
- Bootstrap. (2023). *The most popular HTML, CSS, and JS library in the world*. Recuperado el 4 de Octubre de 2023, de Bootstrap: <https://bit.ly/46iuTWg>
- Brownlee, J. (12 de Septiembre de 2020). *Understand the Impact of Learning Rate on Neural Network Performance*. Recuperado el 3 de Diciembre de 2023, de Machine Learning Mastery: <https://bit.ly/3Rsu73W>
- CEPAL. (2005). *Metodología del marco lógico para la planificación, el seguimiento y la evaluación de proyectos y programas*. Santiago de Chile: Naciones Unidas. Recuperado el 23 de Octubre de 2023, de <https://bit.ly/3T7pptI>
- Coursera. (21 de Julio de 2023). *Figma vs. Sketch vs. Adobe XD: ¿Qué herramienta de diseño es mejor?* Recuperado el 4 de Octubre de 2023, de Coursera: <https://bit.ly/3ZGcyjq>
- Drumond, C. (2023). *Qué es scrum y cómo empezar*. Recuperado el 2 de Noviembre de 2023, de ATLASSIAN: <https://bit.ly/3SpssNN>
- Ferrer, S. (25 de Agosto de 2023). *Tipos de Aprendizaje en Machine Learning: Supervisado y No Supervisado*. Recuperado el 3 de Diciembre de 2023, de IEBS: <https://bit.ly/3RoJZoj>
- Flores, A. (6 de Abril de 2022). *INTA presenta nuevas variedades de sorgo rojo de alta calidad industrial*. Recuperado el 3 de Octubre de 2023, de El 19 Digital: <https://bit.ly/3rB7iB5>
- Flutter. (2023). *Build apps for any screen*. Recuperado el 4 de Octubre de 2023, de Flutter: <https://bit.ly/45ig7NZ>
- Gobierno de Reconciliación y Unidad Nacional. (2021). *Plan Nacional de lucha contra la pobreza y para el desarrollo humano 2022-2026*. Recuperado el 10 de Octubre de 2023, de <http://bit.ly/3tWOLA9>

- Google. (7 de Noviembre de 2019). *Teachable Machine*. Recuperado el 3 de Diciembre de 2023, de Google Creative Lab: <https://bit.ly/3MAi1my>
- Google. (2023). *Acerca de*. Recuperado el 1 de Noviembre de 2023, de Teachable Machine: <https://bit.ly/3MAi1my>
- Google. (2023). *Get Started*. Recuperado el 1 de Noviembre de 2023, de Material Design: <https://bit.ly/3SpCT3Q>
- Google. (2023). *Material Design*. Recuperado el 4 de Octubre de 2023, de Material Design: <https://bit.ly/48CGjpA>
- Google Developers. (1 de Septiembre de 2021). *Tensorflow Lite*. Recuperado el 21 de Octubre de 2023, de TensorFlow: <https://bit.ly/47nQz3C>
- Google Developers. (27 de Julio de 2023). *Guía de arquitectura de apps*. Recuperado el 2 de Diciembre de 2023, de Android Developers: <https://bit.ly/47EKHn7>
- Google Developers. (23 de Septiembre de 2023). *Información sobre los proyectos de Firebase*. Recuperado el 2 de Noviembre de 2023, de Firebase: <https://bit.ly/463mzJc>
- HTML. (6 de Noviembre de 2023). *Living Standard*. Recuperado el 7 de Noviembre de 2023, de HTML: <https://bit.ly/40w8Ppm>
- IBM. (2023). *¿Qué es Java?* Recuperado el 3 de Noviembre de 2023, de IBM: <https://bit.ly/3QwUTH3>
- IBM. (2023). *¿Qué son las redes neuronales convolucionales?* Recuperado el 3 de Diciembre de 2023, de IBM: <https://bit.ly/3T9hGvw>
- Ionic. (2023). *Ionic Documentation*. Recuperado el 3 de Octubre de 2023, de Ionic Framework: <https://bit.ly/3RJQrGQ>
- Laoyan, S. (29 de Septiembre de 2022). *Qué es la metodología waterfall y cuándo utilizarla*. Recuperado el 2 de Noviembre de 2023, de asana: <https://bit.ly/3QHvIld>
- Martins, J. (10 de Octubre de 2022). *¿Qué es la metodología Kanban y cómo funciona?* Recuperado el 2 de Noviembre de 2023, de asana: <https://bit.ly/3QHLqOg>
- MDN contributors. (13 de Marzo de 2023). *CSS*. Recuperado el 3 de Noviembre de 2023, de mdn web docs: <https://bit.ly/3SwJoSh>
- Ministerio Agropecuario. (Febrero de 2023). *Producción de frijol rojo nicaragüense*. Recuperado el 3 de Octubre de 2023, de Ministerio Agropecuario: <https://bit.ly/3LKso73>
- Ministerio Agropecuario. (Febrero de 2023). *Producción de maíz nicaragüense*. Recuperado el 3 de Octubre de 2023, de Ministerio Agropecuario: <https://bit.ly/3PlaxyJ>
- Murray, J., Dieiev, A., starball, & Dent, A. (Octubre de 2023). *Updates*. Recuperado el 2 de Noviembre de 2023, de Visual Studio Code: <https://bit.ly/3tWOuWX>

- Neil, E. M. (1 de Septiembre de 2023). *Welcome to Apache NetBeans*. Recuperado el 2 de Noviembre de 2023, de Apache NetBeans: <https://bit.ly/3u4EIc5>
- Ramos, E. (15 de Agosto de 2023). *¿Qué es una red neuronal y cómo funciona?* Recuperado el 3 de Diciembre de 2023, de EBAC: <https://bit.ly/3uFNTA9>
- React Native. (2023). *Learn once, write anywhere*. Recuperado el 4 de Octubre de 2023, de React Native: <https://bit.ly/48Clr1F>
- Ribalaygua, J., De Loma-Ossorio, E., Córdoba, M., Torres, L., Lahoz, C., & Arias, A. y. (2011). *Enfoques innovadores en la simulación del cambio climático y su impacto en la seguridad alimentaria. La experiencia en Nicaragua*. Recuperado el 10 de Octubre de 2023, de <https://bit.ly/40izdml>
- Robledano, A. (12 de Agosto de 2019). *Qué es Javascript*. Recuperado el 1 de Noviembre de 2023, de OpenWebinars: <https://bit.ly/3QPn1Xe>
- solid IT. (2023). *System Properties Comparison Firebase Realtime Database vs. MySQL vs. SQLite*. Recuperado el 29 de Octubre de 2023, de DB-ENGINES: <https://bit.ly/3RakkOT>
- Souza, B. (12 de Abril de 2022). *Differences between Photoshop, Illustrator and InDesign*. Recuperado el 4 de Octubre de 2023, de pun:chat: <https://bit.ly/3ZENI9f>
- Tailwind CSS. (2023). *Rapidly build modern websites without ever leaving your HTML*. Recuperado el 4 de Octubre de 2023, de tailwindcss: <https://bit.ly/3RLMAct>
- Tamplin, J. (18 de Mayo de 2016). *Firebase expands to become a unified app platform*. Recuperado el 2 de Noviembre de 2023, de The firebase blog: <https://bit.ly/3Sqt1H1>
- UNIR. (17 de Febrero de 2023). *¿Qué es el algoritmo backpropagation para el entrenamiento de redes neuronales?* Recuperado el 3 de Diciembre de 2023, de UNIR La Universidad en Internet: <https://bit.ly/3RrMhDd>
- UVDB community. (2023). *OpenWeather: Weather forecasts, nowcasts and history in a fast and elegant way*. Recuperado el 3 de Noviembre de 2023, de OpenWeather: <https://bit.ly/3u7jH0H>
- Visual Studio Code Team. (1 de Octubre de 2023). *Documentation for Visual Studio Code*. Recuperado el 2 de Noviembre de 2023, de Visual Studio Code: <https://bit.ly/47kopGN>
- Visual Studio Colaboradores. (31 de Octubre de 2023). *¿Qué es Visual Studio? | Microsoft Learn*. Recuperado el 2 de Noviembre de 2023, de Visual Studio: <https://bit.ly/478eMLi>

XII. Anexos

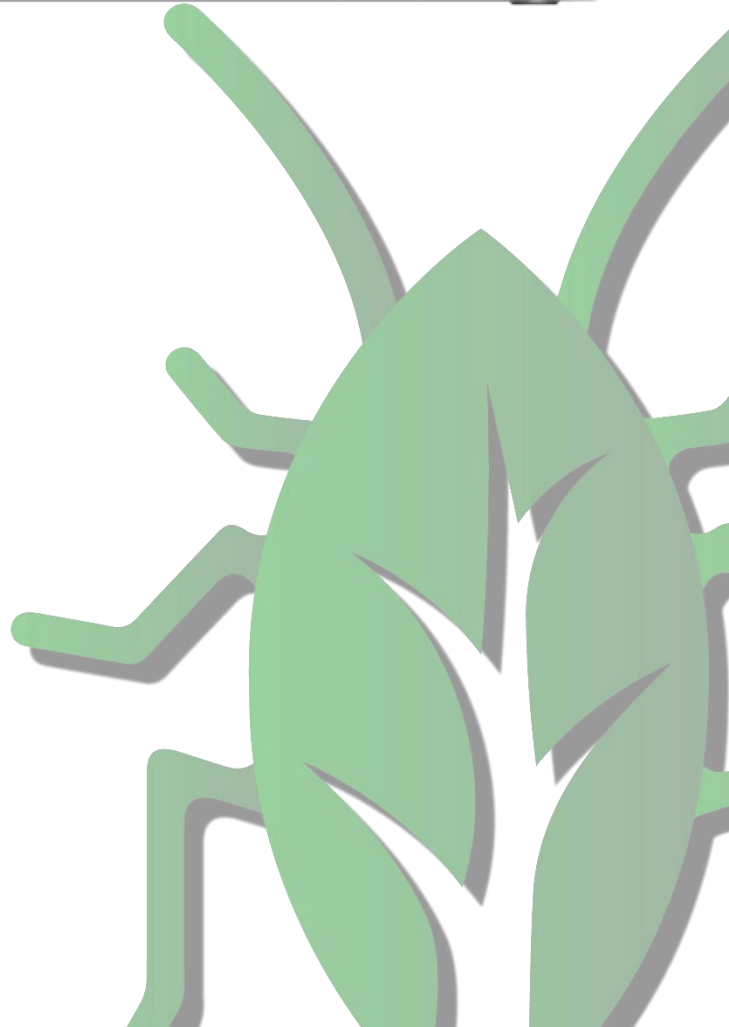
12.1. Modelo CANVAS

Figura 55

Modelo CANVAS



Manual técnico



Índice

I.	Introducción.....	1
II.	Requerimientos técnicos	1
2.1.	Requisitos de instalación del sistema	1
III.	Frameworks y estándares utilizados	2
IV.	Estructura de la aplicación	5
4.1.	Aplicación móvil	5
4.2.	Aplicación web.....	14
V.	Modelo de datos.....	16

Índice de figuras

Figura 1.	Carpeta raíz móvil	5
Figura 2.	Carpeta app.....	6
Figura 3.	Carpeta src	7
Figura 4.	Archivo AndroidManifest	8
Figura 5.	Carpeta values	9
Figura 6.	Archivo strings.xml.....	9
Figura 7.	Archivo activity_login.xml	10
Figura 8.	Carpeta layout.....	10
Figura 9.	Panel de diseño	10
Figura 10.	Carpeta drawable.....	11
Figura 11.	Carpeta Java	11
Figura 12.	LoginActivity.java.....	12
Figura 13.	Componentes Activities	12
Figura 14.	Componentes Adapters.....	12
Figura 15.	CommentAdapter.java.....	12
Figura 16.	HomeFragment.java	13
Figura 17.	Componentes Fragments	13
Figura 18.	User.java.....	13
Figura 19.	Componentes Models	13
Figura 20.	Carpeta raíz web	14
Figura 21.	Carpeta JS (Capa Negocio)	15
Figura 22.	Carpeta Capa Presentación	15
Figura 23.	Modelo de datos en cloud firestore	16
Figura 24.	Modelo de datos en realtime database.....	17

I. Introducción

Placuix es una aplicación móvil y web dirigida a la detección temprana de plagas y enfermedades en los cultivos de granos básicos, la cual tiene acceso a tres tipos de usuarios: agricultores, instituciones y el administrador.

Para alcanzar estos propósitos se ha hecho uso del conjunto de herramientas de desarrollo de Android Studio llamado Android SDK, el cual se adecua a las necesidades actuales de las aplicaciones móviles con dicho sistema operativo. Por otro lado, para llevar a cabo la aplicación web fue utilizado el editor de texto de Visual Studio Code, ya que es la herramienta de desarrollo web más popular entre la comunidad de desarrolladores de software.

Este manual ha sido diseñado teniendo en cuenta la inclusión de todos los detalles técnicos esenciales que son requeridos para operar y supervisar la aplicación de manera efectiva.

II. Requerimientos técnicos

Para el correcto funcionamiento de la aplicación móvil es necesario tener cubiertos una serie de requisitos, tanto de hardware como software.

2.1. *Requisitos de instalación del sistema*

Requisitos de software

- Sistema operativo: Android 10 o superior.

Requisitos de hardware

- Procesador: MediaTek Helio P22 2GHz o superior.
- Memoria RAM: 2GB
- Tamaño de la pantalla: 6.1"
- Cámara: Dual, 13MP+5MP o superior
- Almacenamiento: 1GB

Asimismo, se deben establecer los requerimientos mínimos de hardware y software necesarios para el correcto funcionamiento de la aplicación web.

Requisitos de software

- Sistema operativo: Windows 7 o superior.
- Navegador: Google Chrome, Mozilla Firefox o Microsoft Edge.

Requisitos de hardware

- Procesador: 1.5 Ghz
- Memoria RAM: 2GB
- Almacenamiento en disco duro: 500GB
- Tamaño de la pantalla: 15,6"

III. Frameworks y estándares utilizados

Firestore (v12.8.0)



Firestore es una plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google. Ofrece una variedad de servicios y herramientas que ayudan a los desarrolladores a crear aplicaciones de alta calidad, escalables y seguras. (Google Developers, 2023)

Android SDK (nivel de api 29)



El Android SDK (Software Development Kit) es un conjunto de herramientas de desarrollo de software proporcionadas por Google para permitir a los programadores crear aplicaciones y juegos para el sistema operativo Android. (Android Developers, 2023)

Android Studio (Electric Eel 2022.1.1)



Android Studio es el entorno de desarrollo integrado (IDE) oficial de Google para el desarrollo de aplicaciones móviles en el sistema operativo Android. Es una herramienta poderosa y altamente especializada diseñada específicamente para facilitar la creación, prueba y depuración de aplicaciones Android. (Android Developers, 2023)

Java (SE 20)



Java es un lenguaje de programación de alto nivel, orientado a objetos y de propósito general. Fue desarrollado por Sun Microsystems (que luego fue adquirida por Oracle Corporation) en la década de 1990 y se ha convertido en uno de los lenguajes de programación más populares y ampliamente utilizados en el mundo. (IBM, 2023)

Material Design (Material 3)



Material Design es un sistema de diseño desarrollado por Google que se utiliza para crear interfaces de usuario coherentes, atractivas y funcionales en aplicaciones móviles y web. Fue introducido por Google en 2014 y ha influido significativamente en el diseño de aplicaciones en la plataforma Android y en aplicaciones web. (Google, 2023)

Visual Studio Code (v1.84)



Visual Studio Code, a menudo abreviado como VS Code, es un editor de código fuente de código abierto desarrollado por Microsoft. Se centra en la edición de código y proporciona una experiencia de desarrollo más ligera y altamente personalizable. (Murray et al., 2023)

JavaScript (ECMAScript 2020)



JavaScript es un lenguaje de programación de alto nivel que se utiliza principalmente en el desarrollo web para agregar interactividad y funcionalidad a las páginas web. Fue creado originalmente por Netscape Communications y se convirtió en un estándar en la industria, siendo ampliamente adoptado en todos los navegadores web modernos. (Robledano, 2019)

HTML (HTML 5)



HTML, o Hypertext Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje estándar utilizado para crear páginas web y estructurar el contenido en la World Wide Web. Fue creado en la década de 1990 y ha evolucionado a lo largo de los años, siendo supervisado por el World Wide Web Consortium (W3C), una organización que establece estándares para la web. (HTML, 2023)

CSS (CSS3)



CSS, o Cascading Style Sheets (Hojas de Estilo en Cascada), es un lenguaje de marcado utilizado en el desarrollo web para definir el aspecto y el diseño visual de las páginas web y documentos HTML. (MDN contributors, 2023)

Bootstrap (v4.6)



Bootstrap es un popular marco de diseño front-end de código abierto desarrollado por Twitter, y ahora mantenido por la comunidad de código abierto. Este marco proporciona un conjunto de herramientas y recursos preestablecidos para facilitar el diseño y la creación de sitios web y aplicaciones web modernas, responsivas y atractivas. (Bootstrap, 2023)

Teachable Machine (Teachable Machine 2.0)



Google Teachable Machine es una herramienta en línea desarrollada por Google que permite a las personas, incluso aquellas sin experiencia en programación o aprendizaje automático, crear modelos de aprendizaje automático personalizados y entrenarlos utilizando datos de entrada de imágenes, sonido o gestos. La herramienta utiliza una técnica llamada "aprendizaje profundo" o "deep learning" para crear modelos de aprendizaje automático. (Google, 2023)

Tensorflow Lite (v2.13)



TensorFlow Lite es una versión ligera y eficiente de TensorFlow, que es un marco de aprendizaje automático de código abierto desarrollado por Google. TensorFlow Lite está diseñado específicamente para llevar a cabo inferencia de modelos de aprendizaje automático en dispositivos con recursos limitados, como teléfonos móviles, dispositivos IoT (Internet de las cosas) y sistemas embebidos. La inferencia es el proceso de utilizar un modelo de aprendizaje automático previamente entrenado para realizar predicciones o clasificaciones en nuevos datos de entrada. (Google Developers, 2021)

OpenWeatherMap (v2.0)



OpenWeatherMap (OWM) es un servicio de datos meteorológicos y pronóstico del clima en línea que proporciona información sobre las condiciones climáticas en todo el mundo. (UVDB community, 2023)

Leaflet (v1.9.4)



Leaflet es una biblioteca de código abierto en JavaScript utilizada para crear mapas interactivos y personalizables en sitios web y aplicaciones web. (Agafonkin, 2023)

IV. Estructura de la aplicación

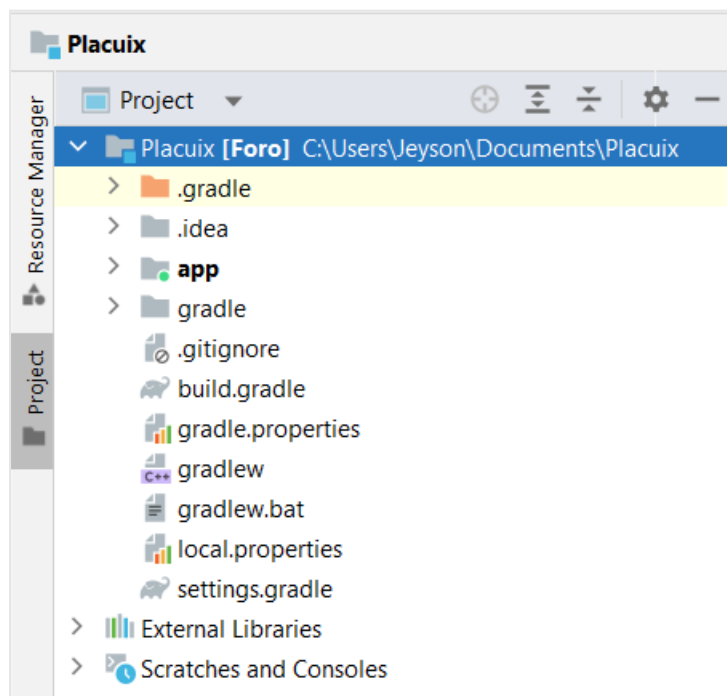
A continuación, se muestra cómo están estructurados los proyectos de la aplicación móvil y web.

4.1. Aplicación móvil

Se ejecuta **Android Studio** y se selecciona el proyecto “**Placuix**”. Se navega al panel de la parte izquierda llamado “**Project**”, el cual representa la estructura del proyecto. Se muestran las carpetas: idea, app, build, gradle y otros archivos de configuración.

Figura 1

Carpeta raíz móvil

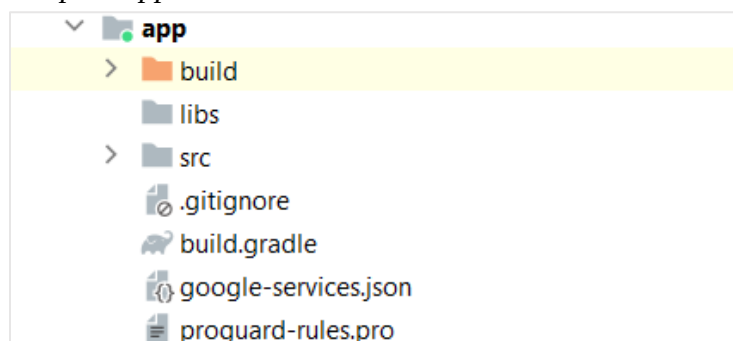


Nota. Las figuras que se muestran para la estructura de la aplicación móvil son tomadas desde el IDE de Android Studio.

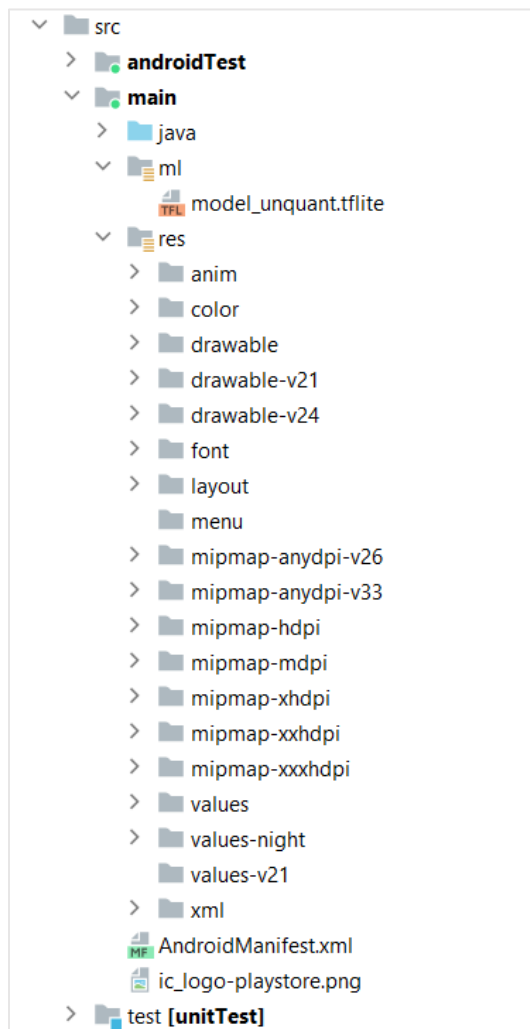
La carpeta “**app**” es la que contiene todo lo relacionado con el proyecto, en esta se incluyen los archivos necesarios para que la aplicación sea empaquetada.

Figura 2

Carpeta app



- La carpeta “**build**” contiene información y archivos temporales que son necesarios para compilar y empaquetar la aplicación, pero generalmente no es necesario interactuar directamente con su contenido.
- La carpeta “**libs**” es un lugar común para almacenar archivos binarios, como bibliotecas de terceros en formato .jar (Java Archive) o archivos .so (Shared Object).
- La carpeta “**src**” es una parte fundamental que almacena el código fuente de la aplicación. Esta carpeta se organiza en varios subdirectorios que representan las distintas fuentes de código y recursos del proyecto.
- El archivo “**build.gradle**” es donde se declaran las dependencias de terceros que la aplicación necesita para funcionar. Estas dependencias se descargan automáticamente por Gradle cuando se construye el proyecto.
- El archivo “**google-services.json**” es esencial para habilitar y configurar servicios de Google, como Firebase.
- El archivo “**proguard-rules.pro**” es utilizado para especificar reglas de ofuscación y minimización de código cuando se aplica ProGuard.

Figura 3*Carpeta src*

Normalmente la mayor parte del tiempo y fuerzas se usa la carpeta **src (Source)**.

- Dentro de ella se ubica la carpeta **main**, la cual contiene todos los **archivos fuente Java** para la aplicación.
- La carpeta **ml (machine learning)** contiene los archivos o modelos de aprendizaje automático que la aplicación utiliza para realizar tareas específicas, como reconocimiento de imágenes, procesamiento de lenguaje natural o cualquier otra tarea relacionada con el ML.
- La carpeta **res (Resources)** que contiene los recursos del proyecto (iconos, sonido, diseños, etc.) y el archivo **AndroidManifest.xml**.

¿Qué es el Android Manifest?

Es un **archivo XML** que contiene nodos descriptivos sobre las características de una aplicación Android. Características como los **building blocks** existentes, la **versión de SDK** usada, los permisos necesarios para ejecutar algunos servicios y muchas más. En pocas palabras el Android Manifest es un panorama de toda la aplicación.

Figura 4
Archivo AndroidManifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.foro">

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_logo"
        android:label="@string/placuix"
        android:networkSecurityConfig="@xml/network_security_config"
        android:roundIcon="@mipmap/ic_logo_round"
        android:supportRtl="true"
        android:theme="@style/Theme.For2"
        tools:targetApi="31">

        <activity...>
        <activity...>
        <activity...>
        <activity...>
        <activity...>
        <activity android:name=".Activities.RegisterActivity" />
        <activity android:name=".Activities.PostDetailActivity" />
        <activity android:name=".Activities.LoginActivity" />
        <activity android:name=".Activities.ForecastActivity" />
        <activity android:name=".Activities.PlagaDetalleActivity" />
        <activity android:name=".Activities.HistoryActivity" />
        <activity android:name=".Activities.ManualActivity" />
        <activity android:name=".Activities.MAGActivity" />
        <activity android:name=".Activities.Home" />
        <meta-data...>
        <meta-data...>
```

Se observa que manifest posee dos atributos, xmlns:android y package. El primero no se cambiar nunca, ya que es el **namespace** del archivo.

- El atributo package indica el nombre del **paquete Java** que soporta a la aplicación.
- La etiqueta <application> representa como estará construida la aplicación. Dentro de ella se definen nodos referentes a las **actividades** que contiene, las **librerías incluidas**, los Intents, Providers, y demás componentes.
- <activity> tiene dos atributos: name, el cual se refiere a la **clase Java** que hace referencia a esta

actividad (Comienza con un punto «.»). Y el atributo **label** que hace referencia al texto que se mostrará en la cabecera de la actividad.

¿Qué utilidad tiene el archivo strings.xml?

Dentro de la carpeta “res” encontraremos todos aquellos recursos tercerizados para la aplicación. Esta práctica de excluir los atributos de la aplicación a través de archivos externos, permite reducir la complejidad de diseño en las interfaces.

Uno de los recursos más relevantes es el archivo **strings.xml** que se encuentra dentro de la subcarpeta values. Este fichero almacena todas las cadenas que se muestran en los widgets (controles, formas, botones, vistas, etc.) de las actividades.

Figura 5
Carpeta values

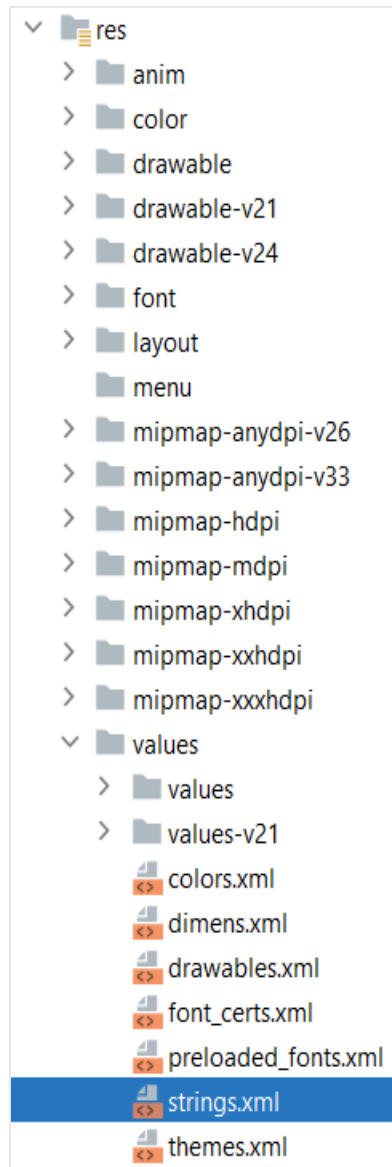


Figura 6
Archivo strings.xml

```
<resources>
    <string name="app_name">PlacuixFinal</string>
    <string name="title_activity_home">Home</string>
    <string name="navigation_drawer_open">Open navigation drawer</string>
    <string name="navigation_drawer_close">Close navigation drawer</string>
    <string name="nav_header_title">Android Studio</string>
    <string name="nav_header_subtitle">android.studio@android.com</string>
    <string name="nav_header_desc">Navigation header</string>
    <string name="action_settings">Settings</string>

    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>
    <string name="title_activity_main">MainActivity</string>
    <string name="title_home">Home</string>
    <string name="title_dashboard">Dashboard</string>
    <string name="title_notifications">Notifications</string>

    <string name="trigo_description">Imagen de trigo</string>
    <string name="frijoles_description">Imagen de frijoles</string>
    <string name="maiz_description">Imagen de maiz</string>
    <string name="temperature"> %d°C</string>
    <string name="wind_speed">Viento: %d km/h</string>
    <string name="humidity">Humedad: %d%%</string>
    <string name="cloudiness">Nubosidad: %d%%</string>

    <string-array name="city_options">...</string-array>
</resources>
```

- El archivo strings.xml es muy útil para los desarrolladores. Una de sus grandes utilidades es facilitar el uso **de múltiples idiomas** en la aplicación. Esto se debe a que permite externalizar las cadenas del código java y seleccionar la **versión** del archivo strings.xml con el lenguaje necesitado.

¿Qué hay dentro de la carpeta layout?

En la carpeta layout se encuentran los archivos de diseño de todas las actividades, fragmentos y adaptadores. Estos archivos representan el diseño de la interfaz de los componentes antes mencionados. En ellos se establecen todos los widgets que vaya a agregar a la actividad.

Figura 8
Carpeta layout

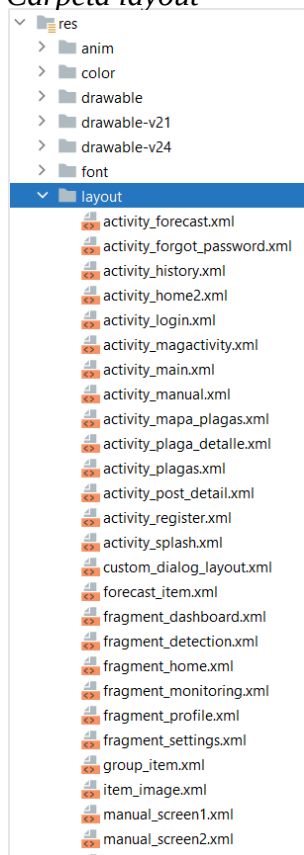
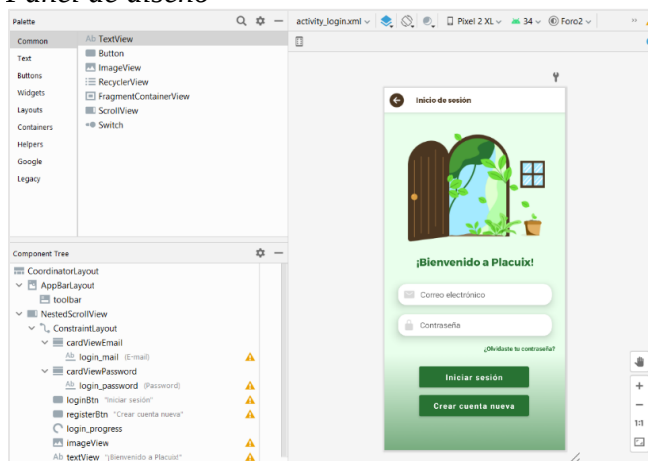


Figura 7
Archivo activity_login.xml



- Construir la interfaz a través de nodos XML es mucho más sencillo que la creación a través de código Java.

Figura 9
Panel de diseño



- Adicionalmente **Android Studio** proporciona un panel de diseño estilo **Drag and Drop**, lo cual facilita demasiado la creación de una interfaz de usuario. Con este diseñador solo se ubican los componentes donde se plazca y automáticamente son generados como nodos en el archivo.

La carpeta drawable

La carpeta "**drawable**" es una carpeta importante en el contexto de desarrollo de aplicaciones Android. Esta carpeta se utiliza para almacenar recursos gráficos, como imágenes y archivos de iconos, que se utilizan en la interfaz de usuario de la aplicación.

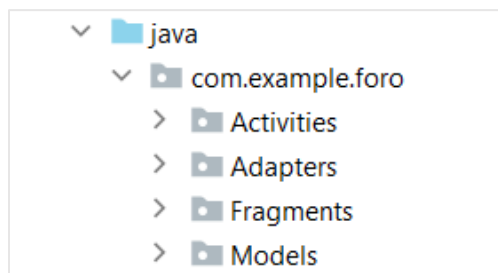
Figura 10
Carpeta drawable



La carpeta Java

La carpeta "**java**" es una parte fundamental de la estructura de directorios en proyectos de desarrollo de aplicaciones Android. En esta carpeta, generalmente ubicada dentro del módulo de la aplicación en el proyecto, se almacena el código fuente escrito en el lenguaje de programación Java (o Kotlin, que es otro lenguaje comúnmente utilizado para el desarrollo de aplicaciones Android).

Figura 11
Carpeta Java



- Las "**activities**" son componentes esenciales en una aplicación Android. Representan las diferentes pantallas o interfaces de usuario de la aplicación y controlan la lógica y la interacción de usuario específicas de cada pantalla.

Figura 29
Componentes Activities

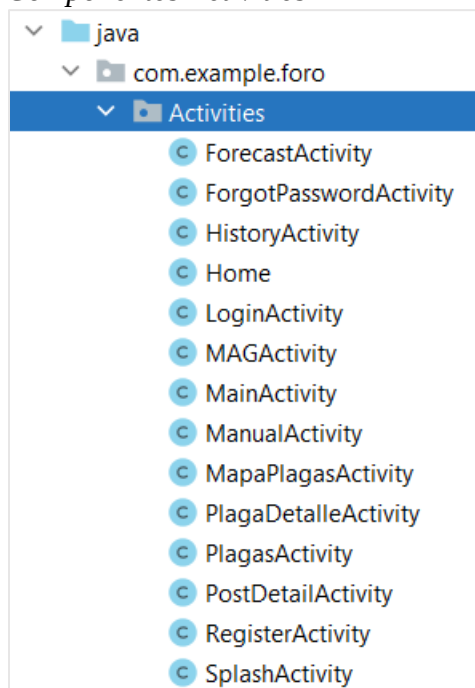


Figura 20
LoginActivity.java



- Los "**adapters**" en Android son componentes que se utilizan para conectar conjuntos de datos a vistas, como las vistas de un RecyclerView, un ListView o un Spinner.

Figura 47
Componentes Adapters

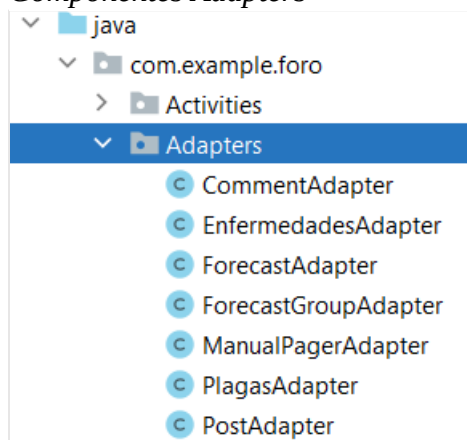
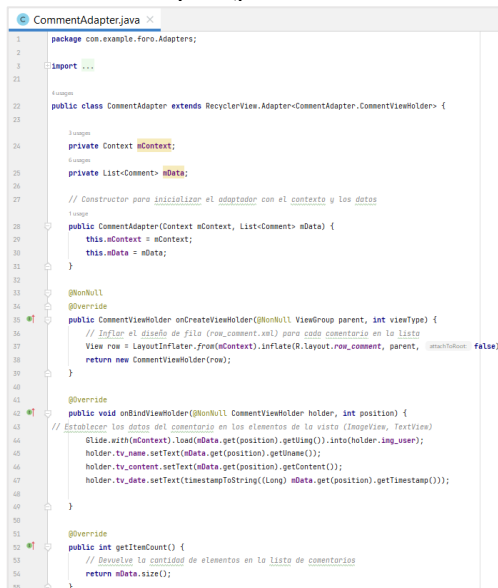


Figura 38
CommentAdapter.java



- Los “**fragments**” son componentes modulares de la interfaz de usuario que pueden combinarse en actividades para crear interfaces de usuario más flexibles y reutilizables.

Figura 65
Componentes Fragments

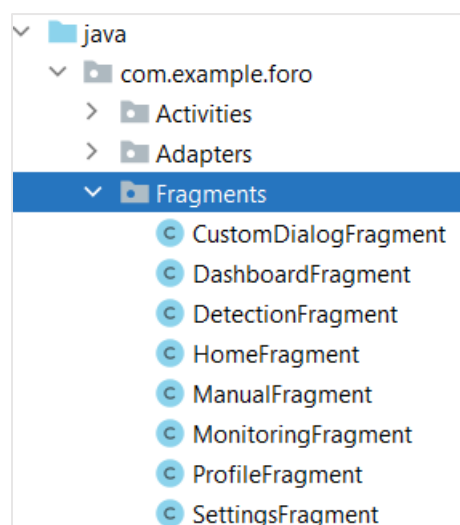


Figura 56
HomeFragment.java

```

1 package com.example.foro.Fragments;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
51
52 public class HomeFragment extends Fragment {
53
54     private static final int PReqCode = 2;
55     private static final int REQUEST_CODE = 2;
56     RecyclerView postRecyclerView;
57     PostAdapter postAdapter;
58     FirebaseDatabase firebaseDatabase;
59     DatabaseReference databaseReference;
60     FirebaseAuth mAuth;
61     FirebaseUser currentUser;
62     List<Post> postList;
63     Dialog popupAddPost;
64     ImageView popupUserImage, popupPostImage, popupAddBtn;
65     TextView popupUserName, popupTitle, popupDescription;
66     ProgressBar popupClickProgress;
67     private Uri pickedImgUri = null;
68
69     public HomeFragment() {
70         // Constructor público necesario para un Fragment
71     }
72

```

- Los “**models**” suelen ser clases que definen la estructura y el comportamiento de los datos utilizados en la aplicación, como objetos de usuario, elementos de una lista, datos de API, configuraciones, y otros tipos de información.

Figura 82
Componentes Models

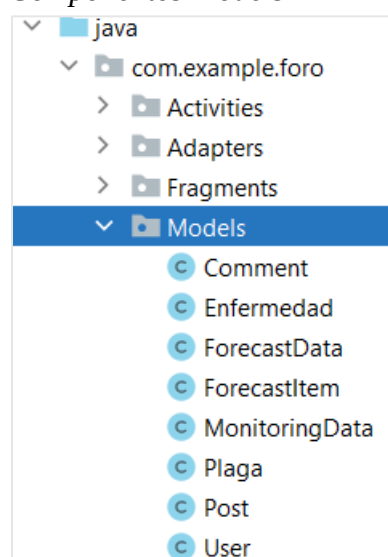


Figura 74
User.java

```

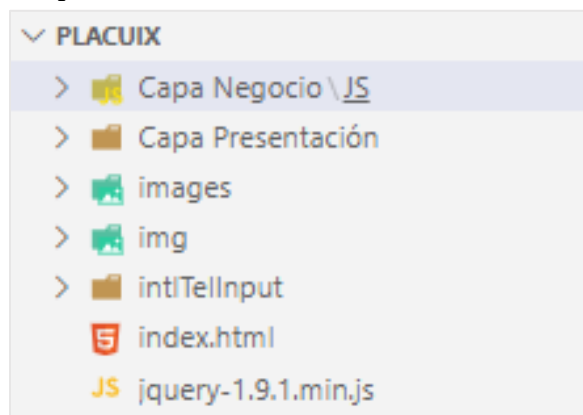
1 // Importamos la clase necesaria para este paquete
2 package com.example.foro.Models;
3
4 // Definimos una clase llamada User que represente información de usuario
5 public class User {
6
7     private String username; // Nombre de usuario
8     private String profession; // Profesión del usuario
9     private String city; // Ciudad del usuario
10
11     // Constructor vacío requerido para Firebase.
12     public User() {
13         // Constructor vacío requerido para Firebase.
14     }
15
16     // Constructor de la clase que recibe los datos necesarios para crear un objeto User
17     public User(String username, String profession, String city) {
18         this.username = username;
19         this.profession = profession;
20         this.city = city;
21     }
22
23     // Métodos "getter" para acceder a los atributos privados desde fuera de la clase
24
25     // Método para obtener el nombre de usuario
26     public String getUsername() { return username; }
27
28     // Método para establecer el nombre de usuario
29     public void setUsername(String username) { this.username = username; }
30
31     // Método para obtener la profesión del usuario
32     public String getProfession() { return profession; }
33
34     // Método para establecer la profesión del usuario
35     public void setProfession(String profession) { this.profession = profession; }
36
37     // Método para obtener la ciudad del usuario
38     public String getCity() { return city; }
39
40     // Método para establecer la ciudad del usuario
41     public void setCity(String city) { this.city = city; }
42
43 }

```

4.2. Aplicación web

Figura 91

Carpeta raíz web



Nota. Las figuras que se muestran para la estructura de la aplicación web son tomadas desde el editor de código de Visual Studio Code.

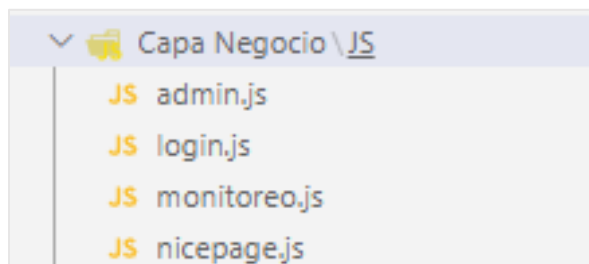
- **Capa Negocio:** Esta capa se encarga de la lógica empresarial de la aplicación. Aquí se implementan las reglas de negocio, el procesamiento de datos y la manipulación de la información. La capa de negocio no se preocupa por la presentación o la interacción con el usuario, sino por el procesamiento de datos y la toma de decisiones basadas en reglas.
- **Capa Presentación:** Esta capa se encarga de la interfaz de usuario y la interacción con el usuario final. Incluye componentes como formularios, vistas, páginas web y otros elementos visuales que permiten a los usuarios interactuar con la aplicación. La capa de presentación no contiene lógica empresarial, sino que se centra en la representación y la interacción con los datos.
- **Images:** En esta carpeta se almacenan todas las imágenes a usarse en la aplicación web.
- El archivo "**jquery-1.9.1.min.js**" es una versión de la biblioteca JavaScript jQuery. jQuery es una de las bibliotecas JavaScript más populares y ampliamente utilizadas en el desarrollo web. Facilita la manipulación del DOM (Document Object Model), la gestión de eventos, las animaciones, las solicitudes AJAX y muchas otras tareas comunes en el desarrollo web.
- El archivo "**index.html**" es el componente central en el desarrollo web y actúa como la página principal. Define la estructura, el contenido y otros elementos esenciales que los navegadores web utilizan para representar visualmente el sitio a los visitantes.

Componente Capa Negocio

La carpeta "JS" dentro de la "capa de negocio" es una convención de nomenclatura que a menudo se utiliza en proyectos web para organizar archivos JavaScript relacionados con la lógica empresarial de la aplicación.

Figura 95

Carpeta JS (Capa Negocio)

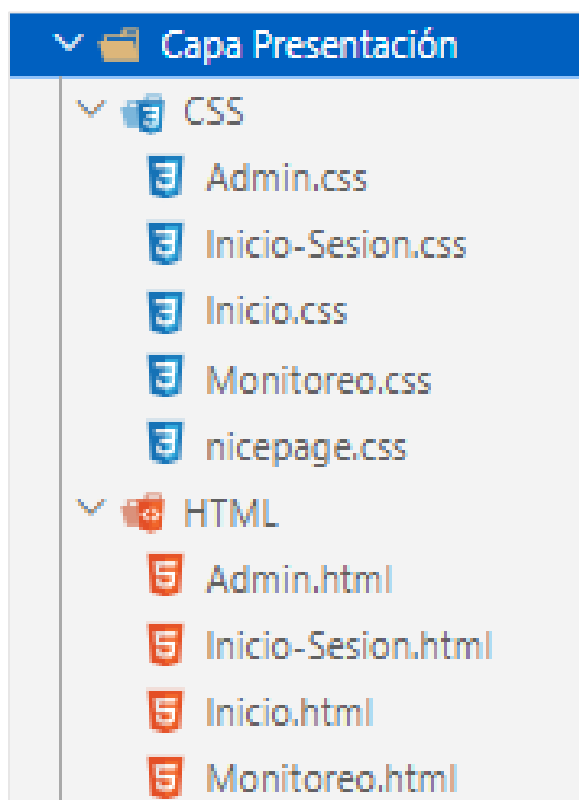


- El archivo **“admin.js”** se encarga de manejar la lógica para el usuario administrador.
- El archivo **“login.js”** maneja la lógica para la pantalla de inicio de sesión.
- El archivo **“monitoreo.js”** maneja la lógica para el usuario institucional.

Componente Capa Presentación

Figura 104

Carpeta Capa Presentación



- La carpeta **"HTML"** contiene archivos html que definen la estructura de la página web de la aplicación. Cada archivo html representa una página o una sección de la interfaz de usuario, conteniendo encabezados, párrafos, enlaces, formularios, imágenes y otros elementos. Esta contiene los archivos: admin, inicio-sesion, inicio, monitoreo.
- La carpeta **"CSS"** contiene archivos CSS (Cascading Style Sheets) que definen los estilos y la presentación visual de la página web de la aplicación para controlar aspectos como colores, fuentes, márgenes, espaciado, alineación y otros estilos visuales de la interfaz de usuario. Esta contiene los archivos de estilo correspondientes a cada archivo html.

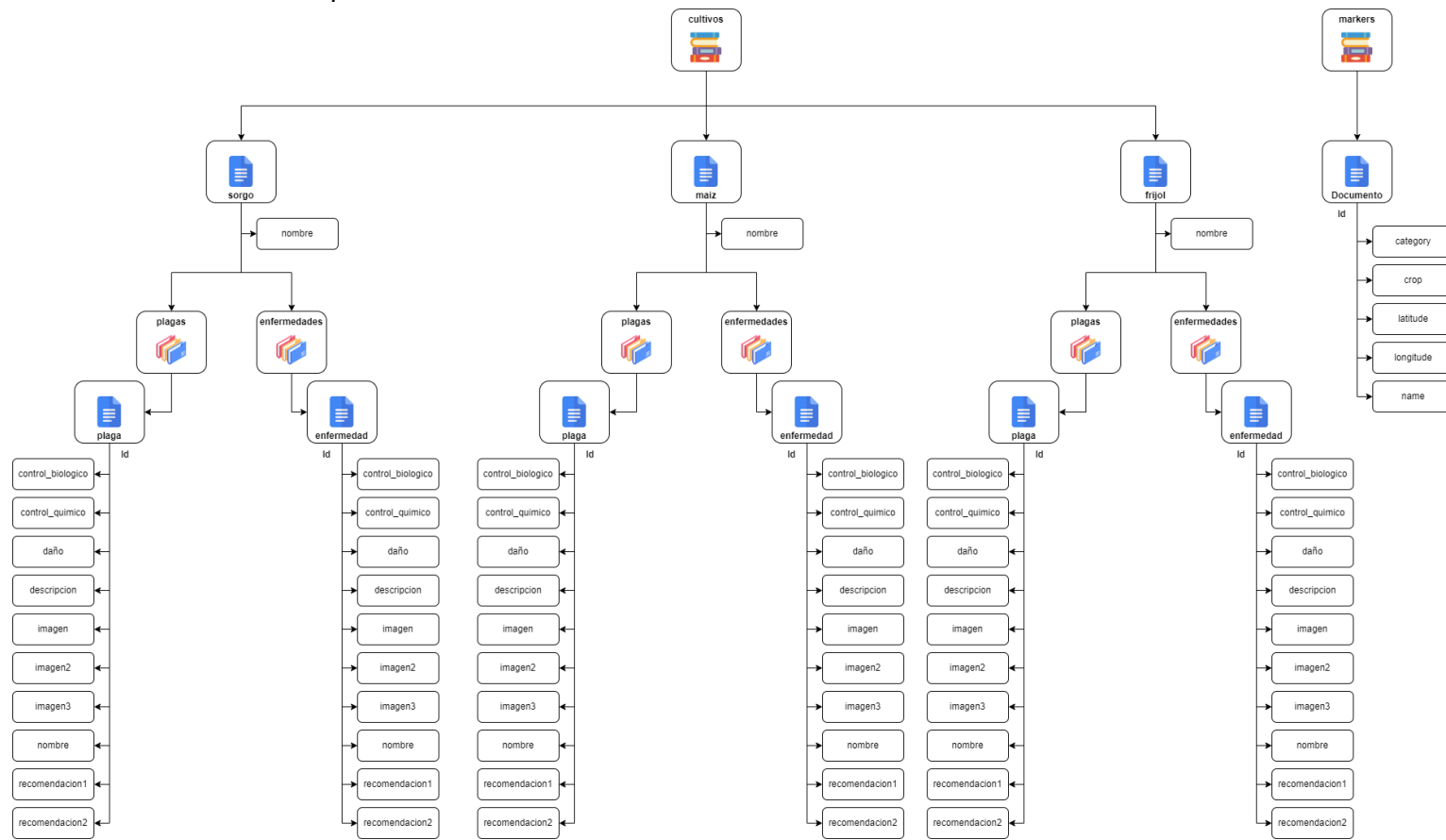
V. Modelo de datos

Cloud Firestore

Cloud Firestore es una base de datos NoSQL orientada a los documentos. A diferencia de una base de datos SQL, no hay tablas ni filas. en su lugar, se almacenan los datos en *documentos*, que se organizan en *colecciones*.

Figura 113

Modelo de datos en cloud firestore

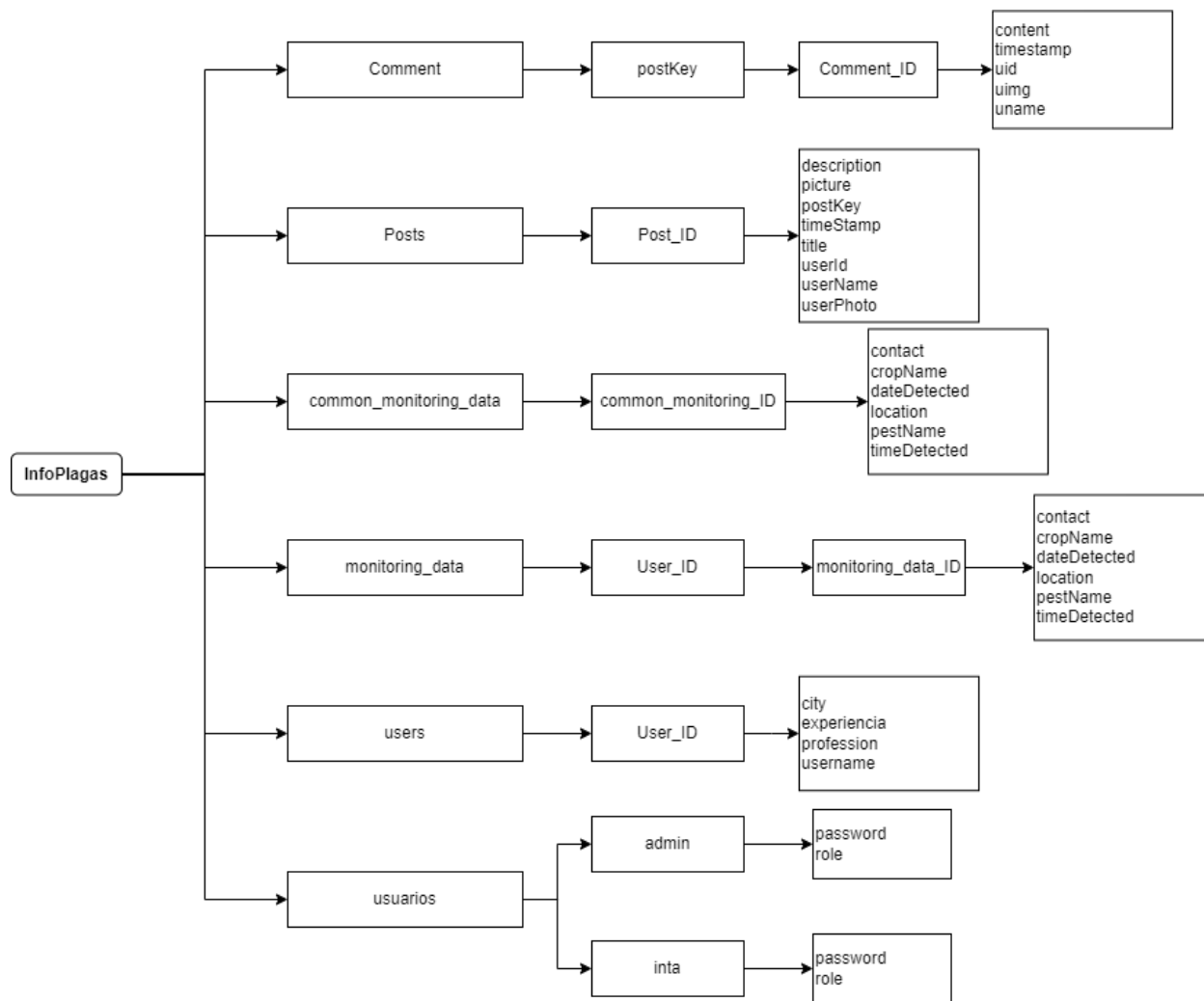


Realtime Database

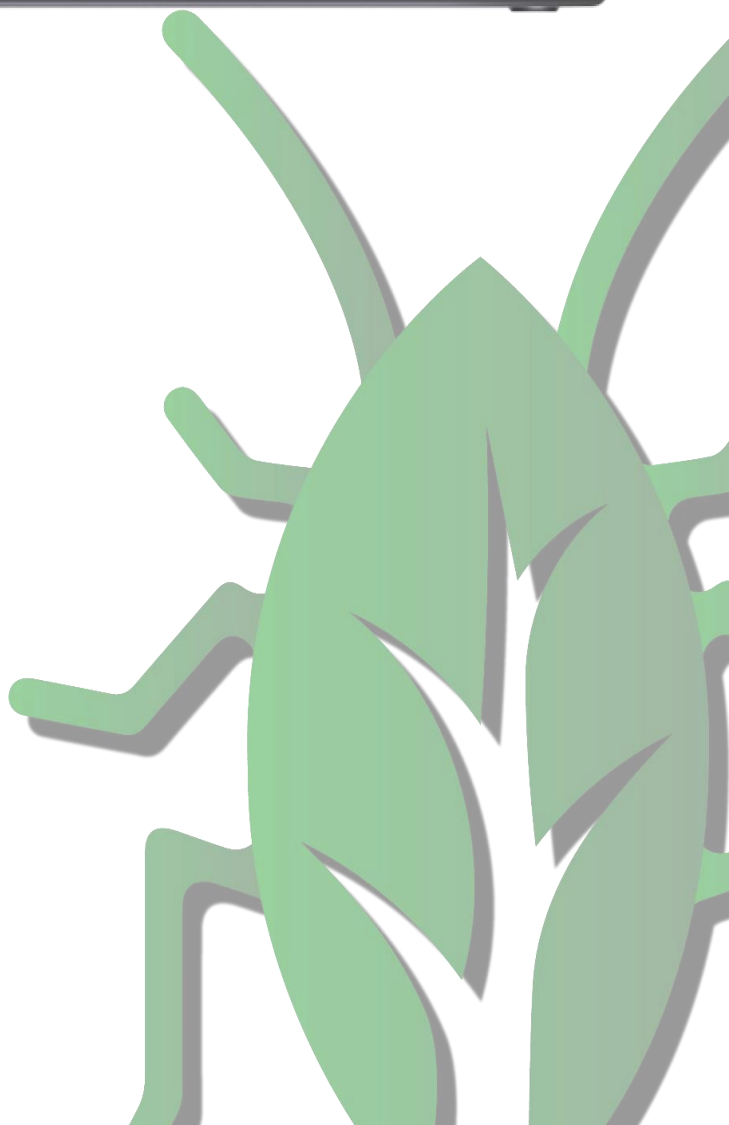
Todos los datos de Firebase Realtime Database se almacenan como objetos JSON. La base de datos puede conceptualizarse como un árbol JSON alojado en la nube. A diferencia de una base de datos de SQL, no hay tablas ni registros. Cuando se agregan datos al árbol JSON, estos se convierten en un nodo de la estructura JSON existente con una clave asociada.

Figura 121

Modelo de datos en realtime database



Manual de usuario



Índice

1.	Sobre la aplicación Placuix.....	1
2.	Requerimientos del sistema	2
3.	Usuario agricultor aplicación móvil	3
3.1.	Instalación	3
3.2.	Splash Screen	4
3.3.	Manual de la aplicación.....	4
3.4.	Página principal (Home)	5
3.4.1.	Información de plagas y enfermedades en los cultivos	6
3.4.2.	Detección de plagas y enfermedades en los cultivos.....	8
3.4.3.	Información del clima.....	11
3.4.4.	Mapas interactivos.....	12
3.4.5.	Barra de navegación	14
4.	Mensajes de error y resolución de problemas aplicación móvil.....	22
4.1.	Error de detección	22
4.2.	Error de campo vacío al iniciar sesión	23
4.3.	Error en el campo contraseña al registrarse	24
5.	Usuario no registrado aplicación móvil.....	25
5.1.	Foro de la comunidad.....	25
5.2.	Monitoreo de plagas y enfermedades	26
5.3.	Calificar la aplicación.....	27
6.	Usuario agricultor aplicación web	28
7.	Usuario admin aplicación web	30
7.1.	Iniciar sesión	30
7.2.	Panel de control de los usuarios agricultores.	31
8.	Usuario institucional aplicación web.....	32
8.1.	Iniciar sesión	32
8.2.	Historial de plagas y enfermedades detectadas	33
8.3.	Editando un registro de plaga o enfermedad detectada.....	34
8.4.	Eliminando un registro de plaga o enfermedad detectada.....	35
8.5.	Mapa de plagas y enfermedades.....	36

8.5.1. Agregando un nuevo registro de plaga o enfermedad detectada	36
8.6. Panel (Dashboard)	38
9. Mensajes de error y resolución de problemas aplicación web	39
9.1. Error de campo vacío al iniciar sesión	39
9.2. Error al agregar un marcado al mapa de plagas y enfermedades	40
10. Glosario.....	41

Índice de tablas

Tabla 39. Requerimientos para instalar la aplicación móvil.....	2
Tabla 40. Requerimientos para navegar en la aplicación web	2

Índice de figuras

Figura 1. Pantalla del teléfono que muestra el ícono de la aplicación Placuix.....	3
Figura 2. Splash Screen	4
Figura 3. Manual de la aplicación.....	4
Figura 4. Pantalla principal 2.....	5
Figura 5. Pantalla principal 1	5
Figura 6. Carrete de cultivos.....	6
Figura 7. Listado de información de plagas y enfermedades	6
Figura 8. Información de plaga o enfermedad.....	7
Figura 9. Tarjeta detección de plaga o enfermedad	8
Figura 10. Pantalla de detección de plagas y enfermedades.....	8
Figura 11. Galería del dispositivo móvil	9
Figura 12. Interfaz de la cámara del dispositivo móvil.....	9
Figura 13. Resultado de la detección	10
Figura 14. Tarjeta de información del clima.....	11
Figura 15. Pronóstico del clima	11
Figura 16. Tarjetas de mapas interactivos	12
Figura 17. Mapa de granos básicos del MAG	12
Figura 18. Popup de mapa de granos	12
Figura 19. Mapa de plagas y enfermedades.....	13
Figura 20. Popup de mapa de plagas	13
Figura 21. Barra de navegación seleccionada en el Inicio	14
Figura 22. Pantalla de perfil sin cuenta.....	14
Figura 23. Barra de navegación seleccionada en perfil	14
Figura 24. Pantalla para registrarse.....	15
Figura 25. Pantalla de inicio de sesión	16
Figura 26. Pantalla para restablecer contraseña.....	16
Figura 27. Pantalla de perfil con cuenta	17
Figura 28. Pantalla de valoración de la aplicación	17
Figura 29. Barra de navegación seleccionada en el foro	18
Figura 30. Pantalla del foro de la comunidad	18
Figura 31. Pantalla para realizar una publicación.....	19
Figura 32. Pantalla de publicación seleccionada	19
Figura 33. Barra de navegación seleccionada en monitoreo	20
Figura 34. Pantalla de monitoreo	20
Figura 35. Pantalla de historial de monitoreo	21
Figura 36. Barra de navegación seleccionada en la detección.....	21
Figura 37. Pantalla de error de detección	22
Figura 38. Instructivo para hacer una detección exitosa.....	22
Figura 39. Pantalla de error de iniciar sesión.....	23
Figura 40. Pantalla de error en registrarse	24
Figura 41. Foro de la comunidad usuario no registrado	25

Figura 42. Publicación de la comunidad usuario no registrado	25
Figura 43. Monitoreo de plagas y enfermedades usuario no registrado	26
Figura 44. Calificar la aplicación usuario no registrado	27
Figura 45. Inicio de aplicación web 1	28
Figura 46. Inicio de aplicación web 2	28
Figura 47. Inicio de aplicación web 3	29
Figura 48. Inicio de aplicación web 4	29
Figura 49. Pantalla de inicio de sesión usuario admin	30
Figura 50. Redes sociales usuario admin	30
Figura 51. Pantalla de aplicación web de panel de control de usuarios	31
Figura 52. Pantalla de inicio de sesión de usuario institucional	32
Figura 53. Redes sociales usuario institucional	32
Figura 54. Tabla de registros	33
Figura 55. Opciones de registros	33
Figura 56. Edición de información de un registro	34
Figura 57. Confirmación para eliminar un registro	35
Figura 58. Agregar marcador en el mapa	36
Figura 59. Vista de marcadores en el mapa	37
Figura 60. Información de un marcador	37
Figura 61. Vista del panel(dashboard)	38
Figura 62. Error de inicio de sesión aplicación web	39
Figura 63. Error de marcador de mapa	40

1. Sobre la aplicación Placuix

Placuix es una aplicación web y móvil para la detección temprana de plagas y enfermedades en cultivos de granos básicos. La aplicación móvil cuenta con la función de detectar plagas y enfermedades de forma offline haciendo uso de Inteligencia Artificial mediante la clasificación de imágenes seleccionando la imagen desde galería o haciendo uso de la cámara del dispositivo móvil, proporcionando el nombre de la amenaza detectada con el nivel de precisión. Cuenta con información detallada sobre plagas y enfermedades proporcionando recomendaciones de tratamiento basadas en prácticas de manejo integrado. Tiene un foro para compartir experiencias y conocimientos. Un historial de monitoreo en el que los agricultores pueden ir registrando los problemas encontrados para llevar un control de estos, cuyos datos se reflejan en la aplicación web diseñada para las instituciones encargadas de la gestión agrícola del país, de modo que puedan proporcionar ayuda y mantener alerta a la población por medio de un mapa interactivo en el que agregan las zonas en las que se han registrado apariciones de plagas y enfermedades en los cultivos.

Es muy sencillo aprender a usar este software ya que está hecho para que sea intuitivo, pero se asume que los usuarios estarán ya familiarizados con los términos, conceptos y métodos que se llevan a cabo en la aplicación. Ante esto, se presenta el manual de usuario de Placuix. Se debe estudiar detenidamente este documento antes de empezar a usar el software.

En esta guía para el usuario se ofrece una visión detallada de las características de la aplicación web y móvil indicando las instrucciones que deben seguirse paso a paso para realizar diversas tareas, además cuenta con posibles errores que el usuario pueda cometer y cómo solucionarlos y un glosario de palabras claves.

2. Requerimientos del sistema

Para el correcto funcionamiento de la aplicación móvil es necesario que el usuario tenga cubiertos una serie de requisitos, tanto de hardware como software, mencionados a continuación:

Tabla 1

Requerimientos para instalar la aplicación móvil

Requerimientos para la instalación de la aplicación móvil Placuix	
Requerimientos de software	
Versión de sistema operativo	Android 10 o superior
Requisitos de hardware	
Procesador	MediaTek Helio P22 Exynos 850
Memoria RAM	2GB
Tamaño de pantalla	6.1", 720 x 1520 pixeles
Cámara	13 megapíxeles

De igual manera para el correcto funcionamiento de la aplicación web es necesario que el usuario tenga cubiertos los siguientes requisitos, tanto hardware como software, mencionados a continuación:

Tabla 2

Requerimientos para navegar en la aplicación web

Requerimientos para la instalación de la aplicación web Placuix	
Requerimientos de software	
Sistema operativo	Windows 7 o superior
Navegador	Google Chrome Mozilla Firefox Microsoft Edge
Versiones	Versión 119.0.6045.105 Versión 119.0 Versión 118.0.2088.76 o superiores
Requisitos de hardware	
Procesador	1.5 Ghz
Memoria RAM	2GB
Almacenamiento en disco duro	500GB
Tamaño de la pantalla	800x600

3. Usuario agricultor aplicación móvil

3.1. Instalación

Una vez que se descarga la aplicación en el teléfono, la instalación de la misma se realiza de manera automática. Una vez que la aplicación se ha instalado correctamente es preciso ubicar el ícono ejecutable de la aplicación descargada, para asegurarse que se encuentra instalada correctamente (como se muestra en la Ilustración 1). Para hacer uso de la aplicación Placuix, sólo es necesario tocar el icono de la aplicación para abrirla y empezar a trabajar con ella.

Figura 1

Pantalla del teléfono que muestra el ícono de la aplicación Placuix



3.2. Splash Screen

Al abrir la aplicación el usuario debe esperar 5 segundos en la pantalla de bienvenida para que inicie correctamente.

Figura 2
Splash Screen

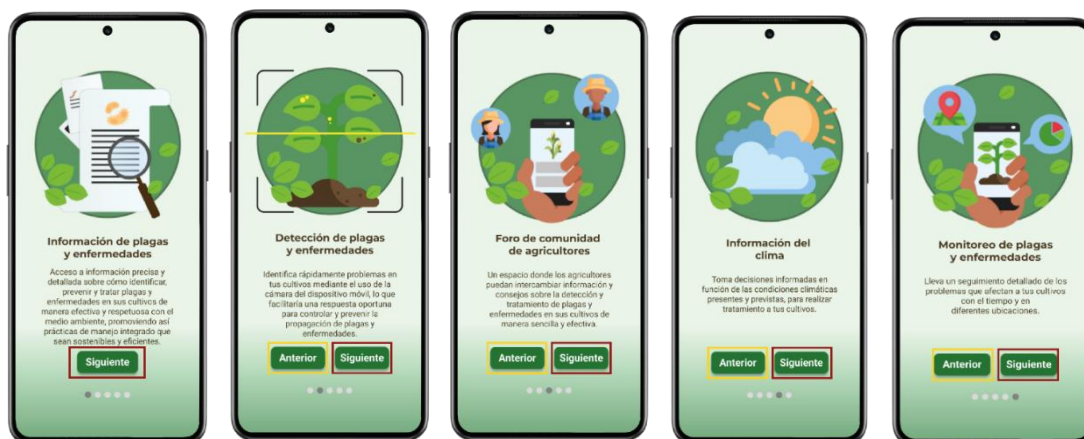


3.3. Manual de la aplicación

Al usuario se le presentará un pequeño instructivo de todas las funciones que tiene la aplicación donde tendrá que tocar el botón **Siguiente** para avanzar o **Anterior** para retroceder.

Importante: El instructivo de la aplicación solo se le presentará la primera vez que el usuario inicie la aplicación.

Figura 3
Manual de la aplicación



3.4. Página principal (Home)

La pantalla de inicio de la aplicación donde se encuentran todas las funciones de Placuix cuenta con distintos apartados divididos por secciones.

- 1 Información de plagas y enfermedades de los cultivos:** Un pequeño carrito donde se muestran los cultivos registrados en la aplicación.
- 2 Detección de plagas y enfermedades en los cultivos:** Una tarjeta la cual permite acceder al apartado de detección de plagas y enfermedades.
- 3 Información del clima:** Una tarjeta con información del clima de la ubicación donde el usuario se encuentre, donde refleja los grados celsius, viento, humedad, nubosidad y consejo del día para tratar los cultivos.
- 4 Mapas interactivos:** Dos tarjetas con dos mapas interactivos distintos, uno para ver que cultivo se siembra más en los departamentos de Nicaragua y otro para ver plagas y enfermedades registradas en los departamentos de Nicaragua.
- 5 Barra de navegación:** Una barra donde están las siguientes funciones de la aplicación: Foro de la comunidad agrícola, Detección de plagas y enfermedades, Monitoreo, Perfil del agricultor.

Figura 5

Pantalla principal 1



Figura 4

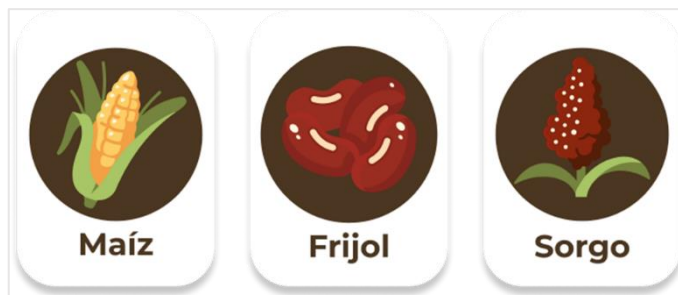
Pantalla principal 2



1

3.4.1. Información de plagas y enfermedades en los cultivos

Figura 6
Carrete de cultivos



Al tocar algún cultivo del carrete el usuario accederá a información precisa y detallada sobre cómo identificar, prevenir y tratar plagas y enfermedades en sus cultivos de manera efectiva y respetuosa con el medio ambiente, promoviendo así prácticas de manejo integrado que sean sostenibles y eficientes.

En pantalla se mostrará un listado de tarjetas de plagas y otro de enfermedades registradas en la aplicación, donde el usuario elegirá de cual desea información.

Figura 7

Listado de información de plagas y enfermedades

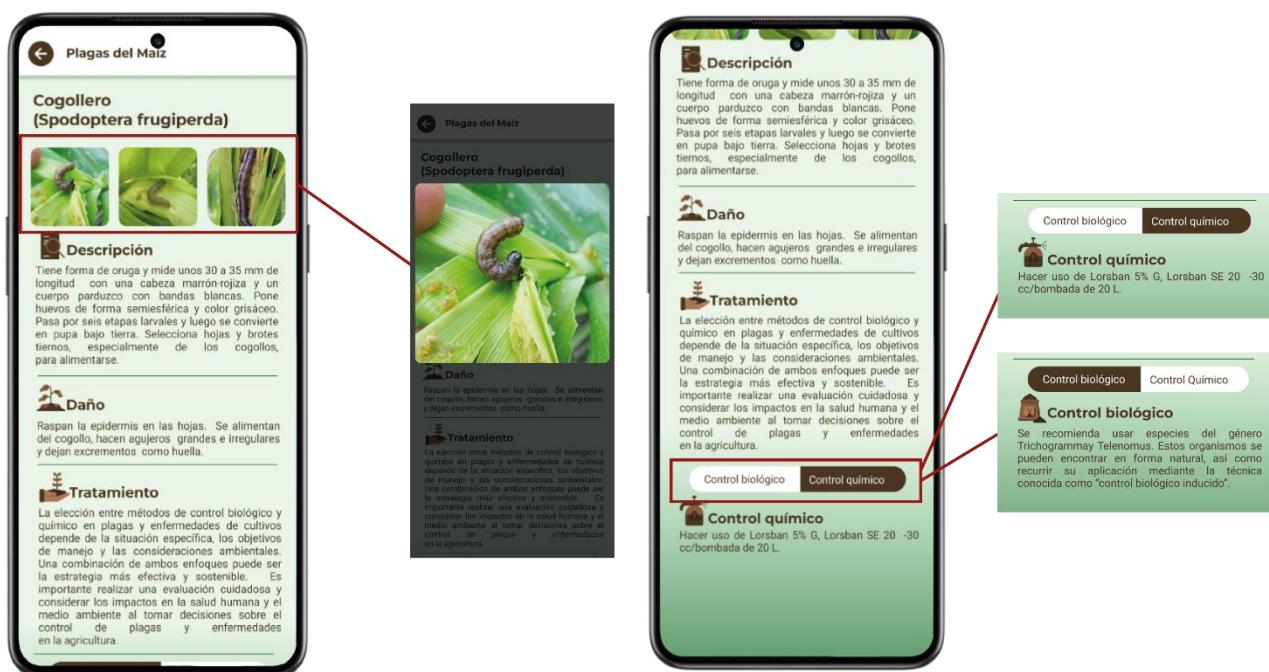


Al tocar una tarjeta ya sea de plagas o enfermedad la aplicación cargará una pantalla con el nombre común y el nombre científico de la plaga o enfermedad seleccionada, imágenes, descripción, el daño que causa, el tratamiento y también el control biológico o químico de esta.

El usuario tendrá la posibilidad de interactuar con las **imágenes mostradas en el carrete**, que al tocarlas podrá verlas más amplias y detalladamente; también podrá alternar entre ver el **control biológico o el control químico**.

Figura 8

Información de plaga o enfermedad



2

3.4.2. Detección de plagas y enfermedades en los cultivos

Figura 9*Tarjeta detección de plaga o enfermedad*

Al tocar la tarjeta de detectar plagas y enfermedades el usuario podrá identificar rápidamente problemas en tus cultivos mediante el uso de la cámara del dispositivo móvil, lo que facilitará una respuesta oportuna para controlar y prevenir la propagación de plagas y enfermedades.

Al usuario se le presentará la siguiente pantalla donde contará con un instructivo para hacer la detección de la mejor manera posible, seguido de este tendrá dos botones, **Seleccionar imagen** y **Tomar imagen**, aquí dependerá si el usuario quiere subir una imagen de su galería o tomar una imagen con la cámara de su dispositivo móvil.

Figura 10*Pantalla de detección de plagas y enfermedades*

Al seleccionar la opción de **Seleccionar imagen** se mostrará la interfaz de la galería de imágenes del dispositivo móvil.

Importante: La apariencia de esta pantalla dependerá del dispositivo que use el usuario ya que la galería de imágenes es distinta para cada marca de dispositivo móvil.

Figura 11
Galería del dispositivo móvil



Al seleccionar la opción de **Tomar imagen** se mostrará la interfaz cámara del dispositivo móvil.

Importante: La apariencia de esta pantalla dependerá del dispositivo que use el usuario ya que la interfaz de la cámara es distinta para cada marca de dispositivo móvil.

Figura 12
Interfaz de la cámara del dispositivo móvil



Luego de haber tomado la imagen o seleccionado una imagen de la galería, se mostrarán los resultados de la detección, donde se mostrará el nombre de la plaga o enfermedad detectada y la precisión de detección, en caso de que la planta este sana se mostrará el mensaje “Planta sana”, si el caso es que la foto fue mal tomada se presentará el mensaje “No se encontraron coincidencias”.

Si el usuario desea hacer una detección nueva, tendrá dos botones **Seleccionar imagen** para seleccionar la imagen desde la galería del dispositivo o **Tomar imagen** con la cámara del dispositivo móvil.

Figura 13

Resultado de la detección



3

3.4.3. Información del clima

Figura 14
Tarjeta de información del clima



El usuario podrá tomar decisiones informadas en función de las condiciones climáticas presentes y previstas, para realizar tratamiento a sus cultivos, podrá observar la temperatura, viento, humedad y nubosidad de la zona donde se encuentre, contará con un buscador para ver el clima de una zona en específico, y por último se le presentará un consejo en dependencia de cómo se encuentre clima, “Hoy hace buen día: Trata tus cultivos”. O “Hoy hace mal día: No tratar tus cultivos”

Al tocar la tarjeta se le mostrará un pronóstico del clima de los próximos cinco días en la zona que se encuentre o la zona que busco anteriormente.

Figura 15
Pronóstico del clima



4

3.4.4. Mapas interactivos

Figura 16

Tarjetas de mapas interactivos

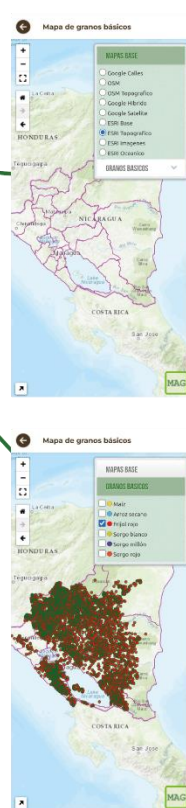


El usuario puede monitorear las plagas y enfermedades a través de un mapa interactivo y monitorear en que departamentos y municipios se siembra más de un cultivo en Nicaragua.

Al tocar la tarjeta del **mapa de granos básicos del MAG** el usuario elige entre tipos de **mapa base** para tener una mejor visión y perspectiva del mismo, también puede escoger que **granos básicos** desea observar en el mapa.

Figura 17

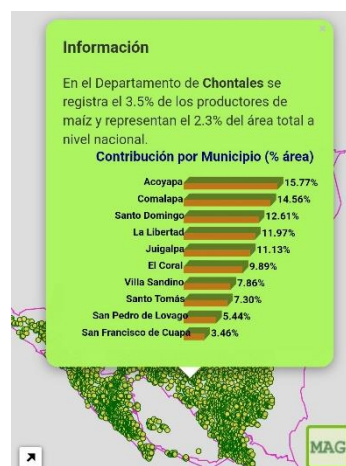
Mapa de granos básicos del MAG



- Si el usuario toca un marcador aparecerá una ventana emergente con información del lugar.

Figura 18

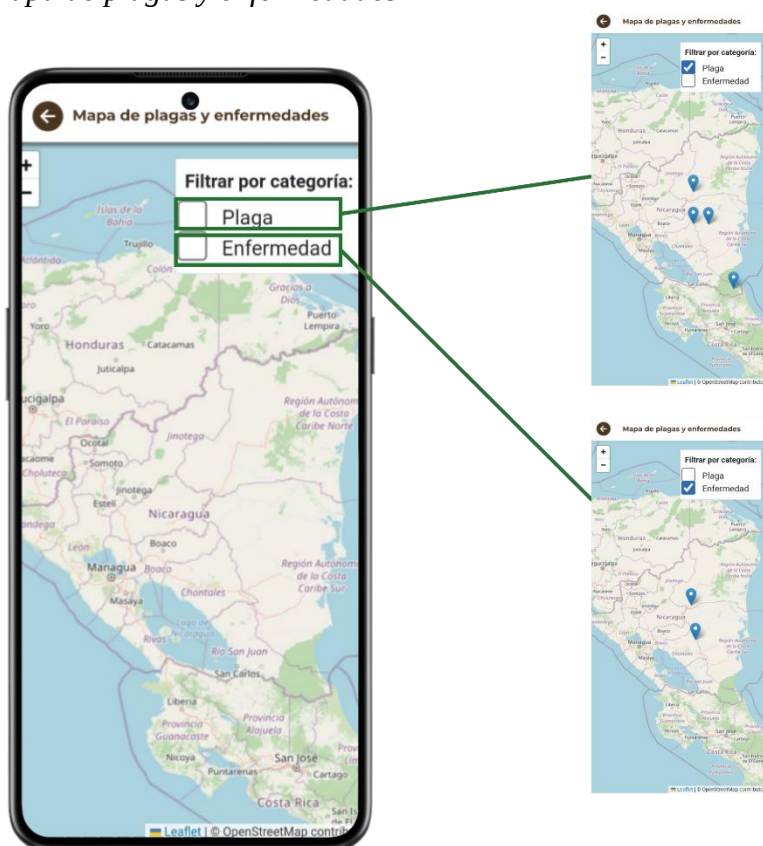
Popup de mapa de granos



Al tocar la tarjeta del **mapa de plagas y enfermedades detectadas** el usuario elige ver que **plaga** o **enfermedad** fue detectada por las instituciones gubernamentales en los departamentos de Nicaragua.

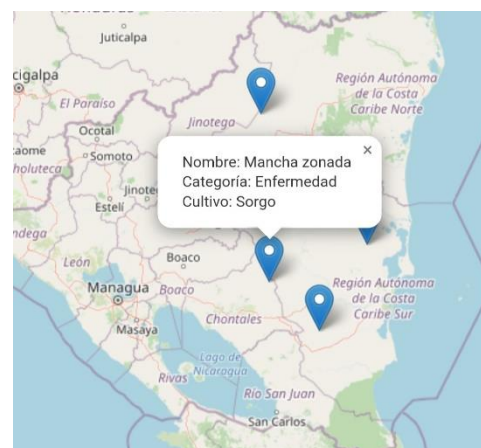
Importante: Esta función de la aplicación móvil va conectada con la aplicación web dedicada a las instituciones gubernamentales.

Figura 19
Mapa de plagas y enfermedades



- Si el usuario toca un marcador aparecerá una ventana emergente con información del lugar.

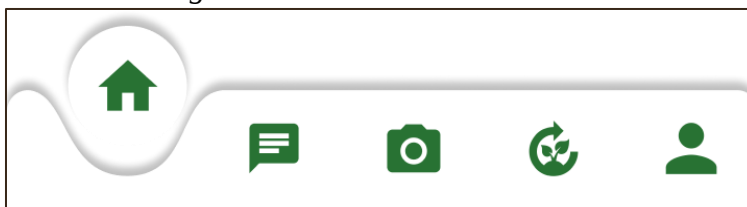
Figura 20
Popup de mapa de plagas



3.4.5. Barra de navegación

Figura 21

Barra de navegación seleccionada en el Inicio



El usuario puede acceder a otras secciones de la aplicación que solo se encuentran en la barra de navegación, como foro de la comunidad de agricultores, la detección de plagas y enfermedades (esta se encuentra tanto en la pantalla de inicio como en la barra de navegación por ser la principal función de la aplicación), Monitoreo de plagas y enfermedades y el perfil de usuario esto se encuentra en un menú de navegación en la parte inferior. La navegación es intuitiva y los usuarios pueden moverse fácilmente entre las secciones sin problemas.

El usuario al tocar el ícono de una persona accederá al apartado de perfil, donde se le mostrará una pantalla con dos botones, **Iniciar sesión** si el usuario ya tiene una cuenta y si no puede **Registrarse** para crear una cuenta nueva

Figura 23

Barra de navegación seleccionada en perfil

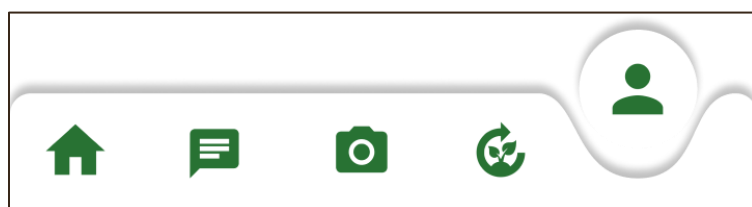
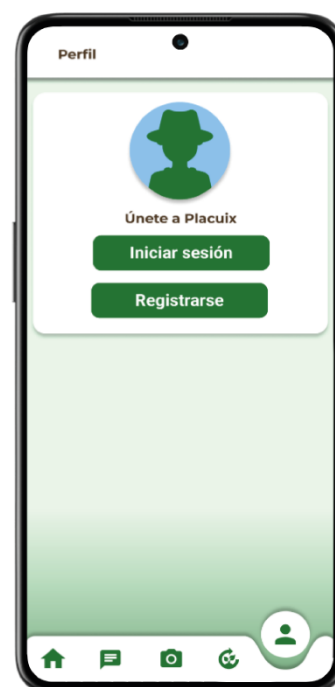


Figura 22

Pantalla de perfil sin cuenta



Al tocar el botón de **registrarse** la aplicación cargará una pantalla con un formulario para ingresar los datos con los que crearas tu cuenta: Imagen de perfil, Nombre de usuario, Correo electrónico, Contraseña y confirmar la contraseña, luego de haber llenado con sus datos se debe tocar en el botón **Crear cuenta**; en caso de que el usuario ya tenga una cuenta en la parte inferior podrá tocar **Accede** y se redirigirá a **iniciar sesión**.

Figura 24

Pantalla para registrarse



Al tocar el botón de **iniciar sesión** la aplicación cargará una pantalla para ingresar con una cuenta ya creada donde se pide el correo y contraseña, luego de ingresar los datos se debe tocar el botón **Iniciar sesión**, si el usuario no tiene una cuenta puede crear una nueva al tocar el botón **Crear cuenta nueva**.

Figura 25

Pantalla de inicio de sesión



En caso de que el usuario haya olvidado su contraseña podrá restablecerla tocando **¿Olvidaste tu contraseña?** y se cargará una pantalla donde se pide el correo electrónico con el que creó la cuenta.

Figura 26

Pantalla para restablecer contraseña



Luego de haber accedido con una cuenta a la aplicación aparecerán los datos que se ingresaron para crearla, donde el usuario tendrá la opción de cambiar su Imagen de perfil, Nombre, Ocupación y su ubicación, en caso de hacer algún cambio debe tocar el botón **Guardar cambios**, también cuenta con un botón para **Cerrar sesión** si deja cambiar de cuenta o crear una nueva cuenta.

Figura 27
Pantalla de perfil con cuenta



En caso de que el usuario quiera valorar la aplicación, podrá **calificar** si Placuix le ha parecido **Útil** o **No útil**.

Figura 28
Pantalla de valoración de la aplicación



Importante: Para poder utilizar las funciones de Monitoreo de plagas y enfermedades o Foro de la comunidad, se necesita tener una cuenta en la aplicación.

Si el usuario entra al foro de la comunidad y no tiene una cuenta le aparecerá un botón en el que lo redirigirá a la pantalla de inicio de sesión.

Inicia sesión para publicar contenido

El **foro de la comunidad** es un espacio donde los agricultores puedan intercambiar información y consejos sobre la detección y tratamiento de plagas y enfermedades en sus cultivos de manera sencilla y efectiva.

Figura 29

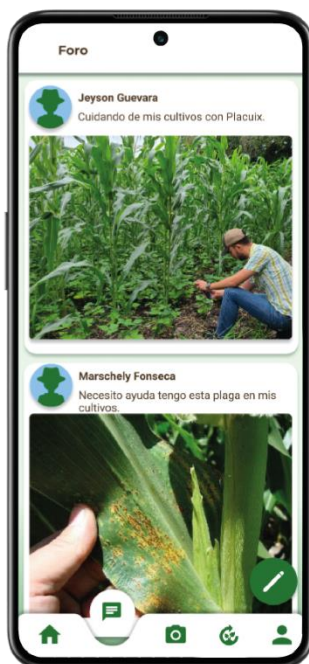
Barra de navegación seleccionada en el foro



En la pantalla del foro se pueden ver las publicaciones realizadas por la comunidad de agricultores

Figura 30

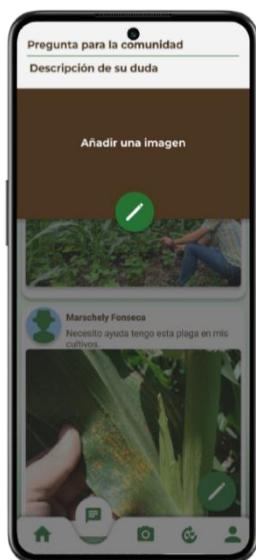
Pantalla del foro de la comunidad



Para realizar una publicación el usuario debe tocar el ícono de un lápiz y se le abrirá una pantalla emergente donde debe escribir la pregunta o el título, la descripción o duda y si desea añadir una imagen.

Figura 31

Pantalla para realizar una publicación



Si se desea ver los comentarios de los demás usuarios, solo se debe tocar la publicación y cargará una pantalla con la imagen, el título, la descripción y los comentarios de otros usuarios. Si desea escribir un comentario tendrá un cuadro de texto para escribir lo que piensa de la publicación.

Figura 32

Pantalla de publicación seleccionada



Figura 33

Barra de navegación seleccionada en monitoreo



El **monitoreo de plagas y enfermedades** sirve para llevar un seguimiento detallado de los problemas que afectan a tus cultivos con el tiempo y en diferentes ubicaciones.

Al tocar el ícono de monitoreo cargará una pantalla donde habrá un formulario donde el usuario debe escribir el nombre del cultivo, nombre de la plaga detectada, el contacto para que las instituciones puedan contactarse con él, el departamento donde se encontró la plaga y la fecha y hora. Luego de llenar el formulario se debe tocar el botón **Enviar**.

Figura 34

Pantalla de monitoreo

A screenshot of a mobile app interface titled 'Monitoreo'. It features a form with five input fields: 'Nombre de cultivo', 'Nombre de plaga', 'Contacto', and 'Departamento'. Below these is a table for recording detection dates and times. The table has two columns: 'Fecha de detección' and 'Hora de detección'. It contains three rows of data. At the bottom of the form are two green buttons: 'Enviar' and 'Ver Historial'. The bottom navigation bar is visible, with the 'Monitoreo' icon selected.

Si el usuario desea ver todos los monitoreos de plagas y enfermedades detectadas puede tocar el botón **Ver historial** y se le presentará una tabla con el historial de monitoreos realizados.

Figura 35

Pantalla de historial de monitoreo

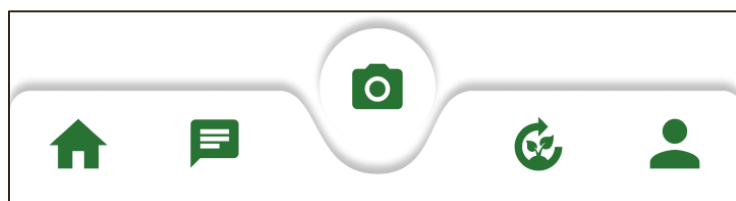


Cultivo	Plaga/Enfermedad	Ubicación	Hora	Fecha
Maiz	Gusano cogollero	Juigalpa	09:22	2023-10-11
			16:46	2023-10-11
			10:10	2023-10-2
hhh	bbbb	Atlántico Norte	04:35	2023-10-11
Maiz	Cogollero	Juigalpa, Chontales	16:04	2023-11
Sorgo	Masaico del sorgo	Tipitapa, Managua	16:04	2023-11
Frijol	Masaico dorado	Sébaco, Matagalpa	16:04	2023-11
Maiz	Cogollero	Juigalpa, Chontales	16:04	2023-11
Sorgo	Masaico del sorgo	Tipitapa, Managua	16:04	2023-11
Frijol	Masaico dorado	Sébaco, Matagalpa	16:04	2023-11

Importante: La función de monitoreo va de la mano con la aplicación web, ya que sirve como medio de contacto para que las instituciones ayuden.

Figura 36

Barra de navegación seleccionada en la detección



En la barra de navegación el ícono de la cámara sirve como un acceso directo para redirigir al usuario a la pantalla de detección de plagas y enfermedades debido a que es la función principal de la aplicación Placuix.

4. Mensajes de error y resolución de problemas aplicación móvil

El usuario agricultor puede tener problemas al utilizar la aplicación móvil de Placuix, es por ello que explicamos paso a paso cuales son los errores que puede cometer y como solucionarlos.

4.1. Error de detección

Si el usuario selecciona o toma una imagen movida o de una perspectiva errónea la aplicación le mandará el mensaje “**No se encontraron coincidencias**”

Figura 37

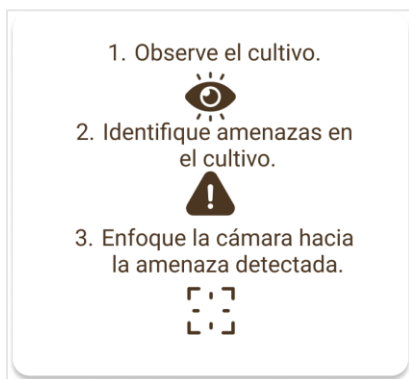
Pantalla de error de detección



Para solucionar esto el usuario debe seguir estas recomendaciones que se presentan en la pantalla de detección de plagas y seleccionar o tomar una foto clara.

Figura 38

Instructivo para hacer una detección exitosa



4.2. Error de campo vacío al iniciar sesión

Si el usuario al momento de ingresar en la aplicación móvil no llena un campo correctamente le mandará un mensaje **“Por favor, verifique todos los campos”**.

Figura 39

Pantalla de error de iniciar sesión



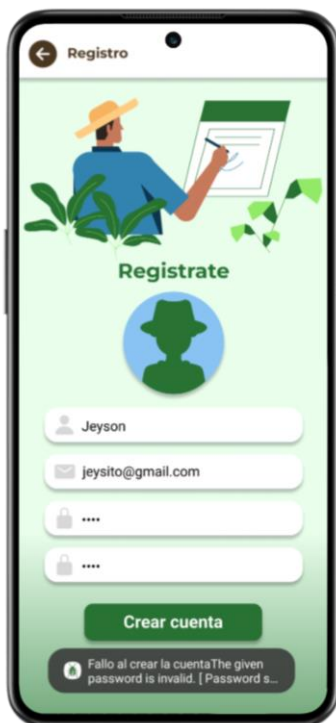
Para solucionar esto el usuario debe ingresar correctamente todos los campos requeridos para iniciar sesión en la aplicación.

4.3. Error en el campo contraseña al registrarse

Si el usuario al momento de crear una cuenta nueva no cumple con los caracteres totales establecidos para el campo contraseña le aparecerá un mensaje “Fallo al crear cuenta”.

Figura 40

Pantalla de error en registrarse



Para solucionar esto el usuario debe crear una contraseña con 6 caracteres como mínimo para crear su cuenta con éxito.

5. Usuario no registrado aplicación móvil

El usuario que no está registrado en la aplicación móvil tiene acceso completo a las funciones de detección de plagas y enfermedades, Información del clima y mapas interactivos, pero se le restringe el acceso completo a las siguientes funciones.

5.1. Foro de la comunidad

El usuario no registrado al acceder al foro de la comunidad no podrá hacer ninguna publicación, sin embargo, tendrá un botón que lo mandará a iniciar sesión para luego poder hacer sus preguntas a la comunidad.

Cabe destacar que el usuario no registrado podrá entrar a ver los detalles de las publicaciones de otros usuarios, pero tampoco podrá hacer comentarios al respecto.

Figura 41

*Foro de la comunidad
usuario no registrado*

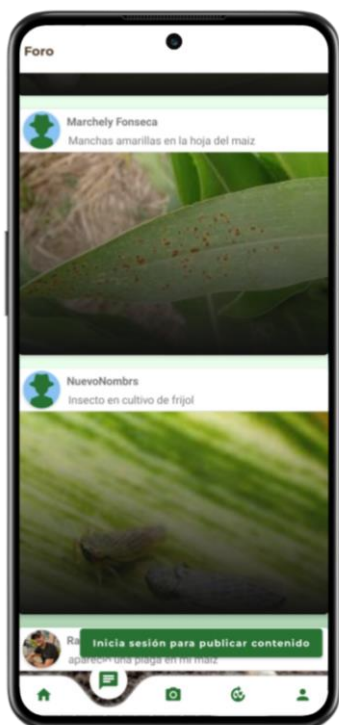


Figura 42

*Publicación de la comunidad
usuario no registrado*



5.2. Monitoreo de plagas y enfermedades

El usuario no registrado no podrá realizar ningún registro de monitoreo ni le dejará ver el historial ya que se necesita de tener una sesión iniciada en la aplicación para acceder a estas funciones.

Figura 43

*Monitoreo de plagas y enfermedades
usuario no registrado*

Monitoreo

Nombre de cultivo

Nombre de plaga

Contacto

Departamento

Fecha de Detección			Hora de Detección		
02	nov	2022	11	33	
03	dic	2023	12	34	
04	ene	2024	1	35	

Enviar

Ver Historial

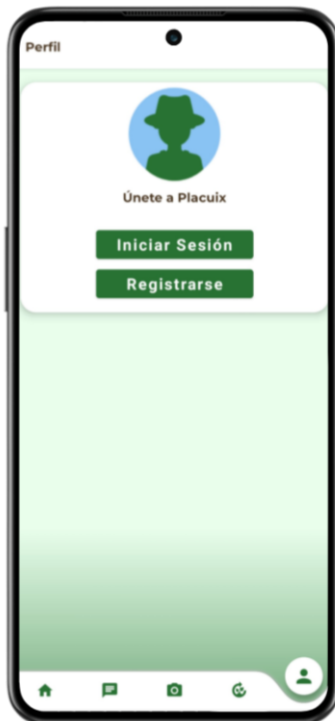
Inicia sesión para enviar los datos

5.3. Calificar la aplicación

El usuario no registrado no podrá acceder a calificar la aplicación ni tendrá un perfil propio.

Figura 44

*Calificar la aplicación
usuario no registrado*



6. Usuario agricultor aplicación web

El usuario agricultor puede entrar a la aplicación web únicamente para observar el **Lading Page** donde encontrará información sobre la aplicación, testimonios y para contactarse con el grupo de Placuix.

Figura 45

Inicio de aplicación web 1



Figura 46

Inicio de aplicación web 2



Figura 47
Inicio de aplicación web 3

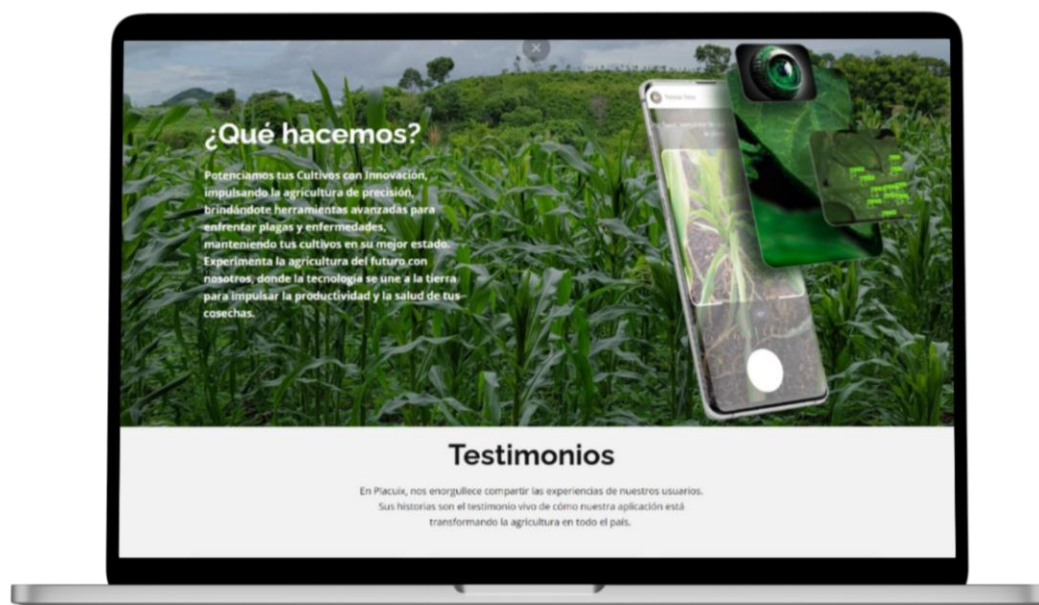
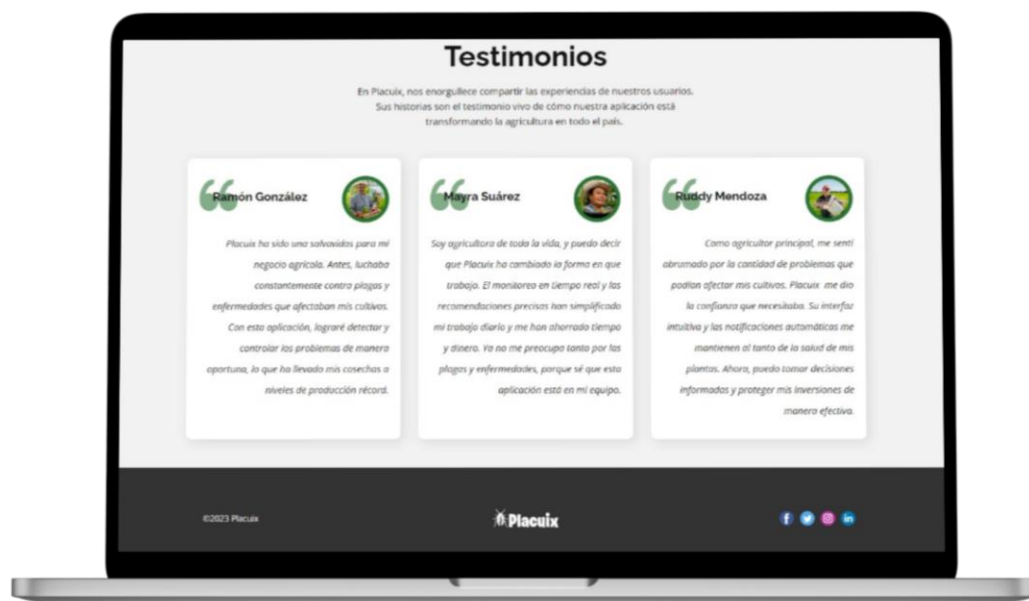


Figura 48
Inicio de aplicación web 4



En caso de que los usuarios quieran visitar las redes sociales de PlacuiX solo deberán tocar los íconos de la parte inferior

Importante: Cualquier usuario puede acceder al inicio de la aplicación web de PlacuiX.

7. Usuario admin aplicación web

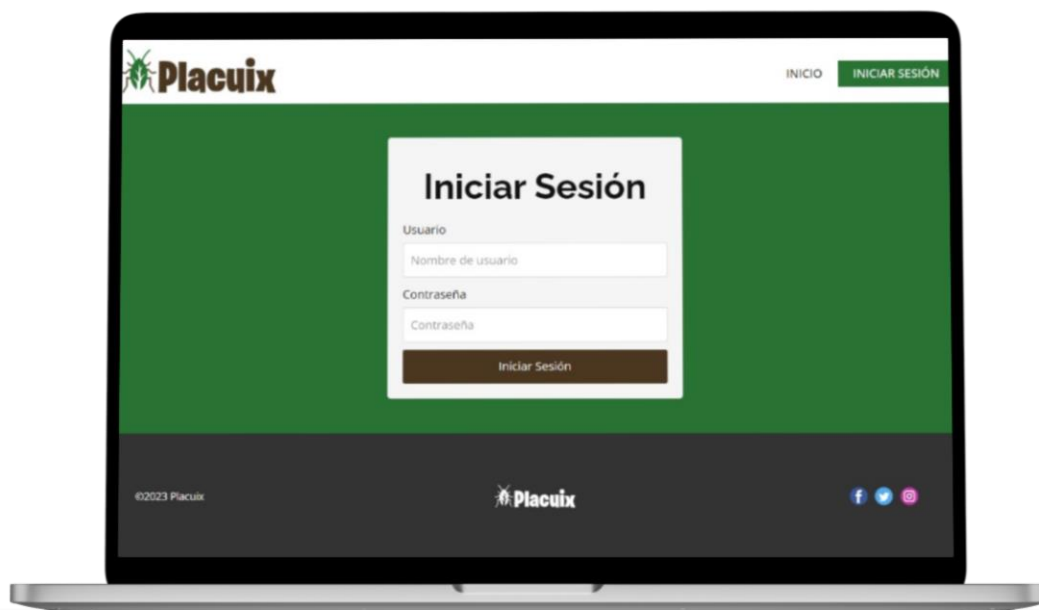
La función del usuario admin en Placuix es únicamente acceder a un panel de control donde puede observar a los usuarios agricultores que ingresan a la aplicación móvil y ver sus opiniones de que si Placuix para ellos ha sido útil o no.

7.1. Iniciar sesión

El usuario admin debe acceder a la aplicación web de Placuix donde deberá ingresar su usuario y contraseña para poder iniciar sesión.

Figura 49

Pantalla de inicio de sesión usuario admin



Si el usuario admin desea visitar las redes sociales de Placuix, solo debe tocar los íconos en la parte inferior.

Figura 50

Redes sociales usuario admin

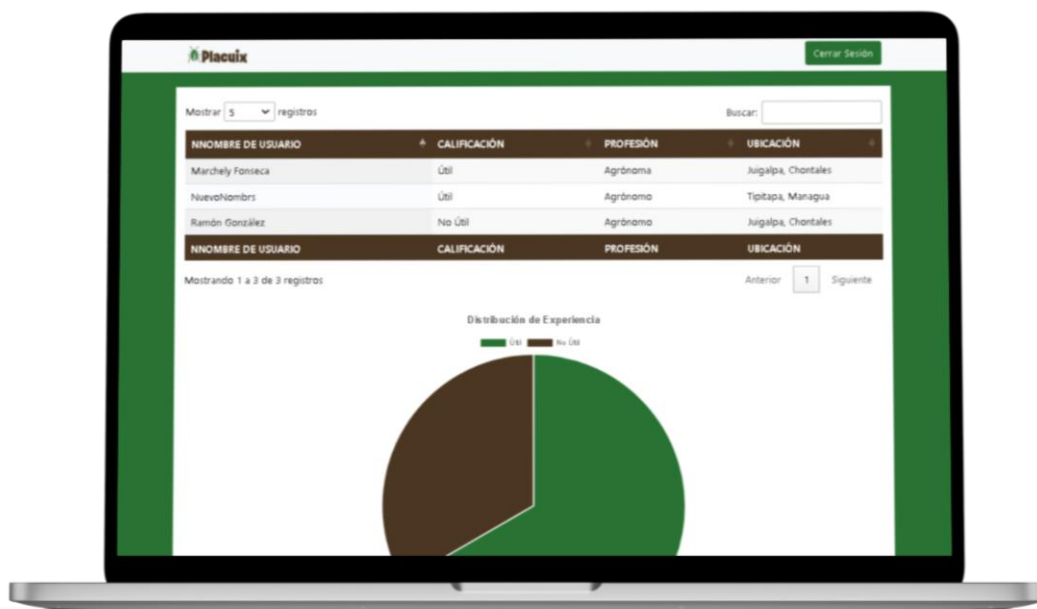


7.2. Panel de control de los usuarios agricultores

El usuario admin puede observar que usuarios y opiniones tiene la aplicación móvil Placuix.

Figura 51

Pantalla de aplicación web de panel de control de usuarios



Importante: Las opiniones se agregan por los agricultores en la aplicación móvil.

8. Usuario institucional aplicación web

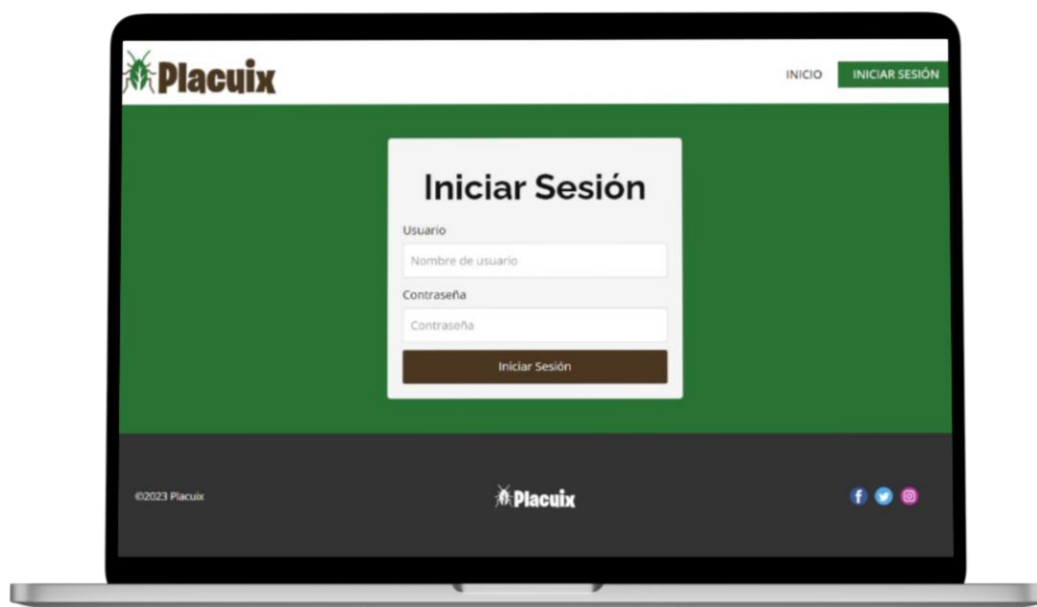
El usuario institucional tiene como funciones en la aplicación web un historial de las plagas y enfermedades detectadas, mapa de plagas y enfermedad y un panel para ver estadísticas de las plagas y enfermedades por departamentos.

8.1 Iniciar sesión

El usuario institucional debe acceder a la aplicación web de Placuix donde deberá ingresar su usuario y contraseña para poder iniciar sesión.

Figura 52

Pantalla de inicio de sesión de usuario institucional



Si el usuario institucional desea visitar las redes sociales de Placuix solo debe tocar los íconos de la parte inferior.

Figura 53

Redes sociales usuario institucional



8.2. Historial de plagas y enfermedades detectadas

Al tocar iniciar sesión se redirigirá al usuario al historial de plagas y enfermedades detectadas en los departamentos de Nicaragua.

Donde podrá observar una tabla con los registros de plagas y enfermedades detectadas con los campos Plaga/Enfermedad, Cultivo, Departamento, Contacto, Fecha de detección y Hora de detección.

Figura 54
Tabla de registros

PLAGA/ENFERMEDAD	CULTIVO	DEPARTAMENTO	CONTACTO	FECHA DE DETECCIÓN	HORA DE DETECCIÓN
Chicharrita verde	Frijol	Jinotega	85236954	2023-10-21	10:11
Gusano cogollero	Maíz	Chontales	+50584903928	2023-10-21	09:12
Mancha zonada	Sorgo	Managua	89636254	2023-10-21	09:24
Mancha zonada	Sorgo	Managua	96524821	2023-10-21	10:11
Mosaico dorado	Frijol	Chontales	+50584903928	2023-10-21	09:12

Si el usuario institucional presiona el ícono  se le permitirá editar los datos del registro de plaga o enfermedad seleccionado en un modal. Mientras que el botón con el ícono  cargará un modal que le pedirá confirmar la eliminación del registro de la tabla seleccionado.

Cuenta con las funciones de cuántos registros desea **mostrar** en pantalla, **imprimir** los registros mostrados en formato de Excel o PDF y con una **barra de búsqueda** para ver un registro en específico.

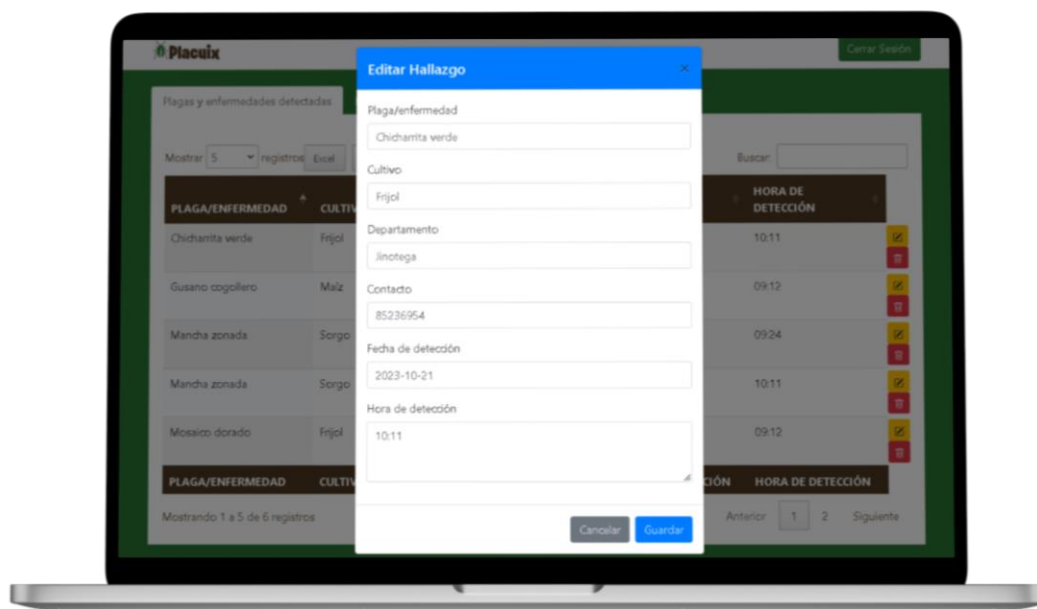
Figura 55
Opciones de registros

8.3. Editando un registro de plaga o enfermedad detectada

Una vez el usuario presiona el botón con el ícono  se le permitirá editar el registro a través de un modal.

Figura 56

Edición de información de un registro



Si el usuario desea cambiar la información del registro (Plaga/enfermedad, cultivo, Departamento, Contacto, Fecha de detección y hora de detección) deberá editar cualquiera de los campos que posee el modal.

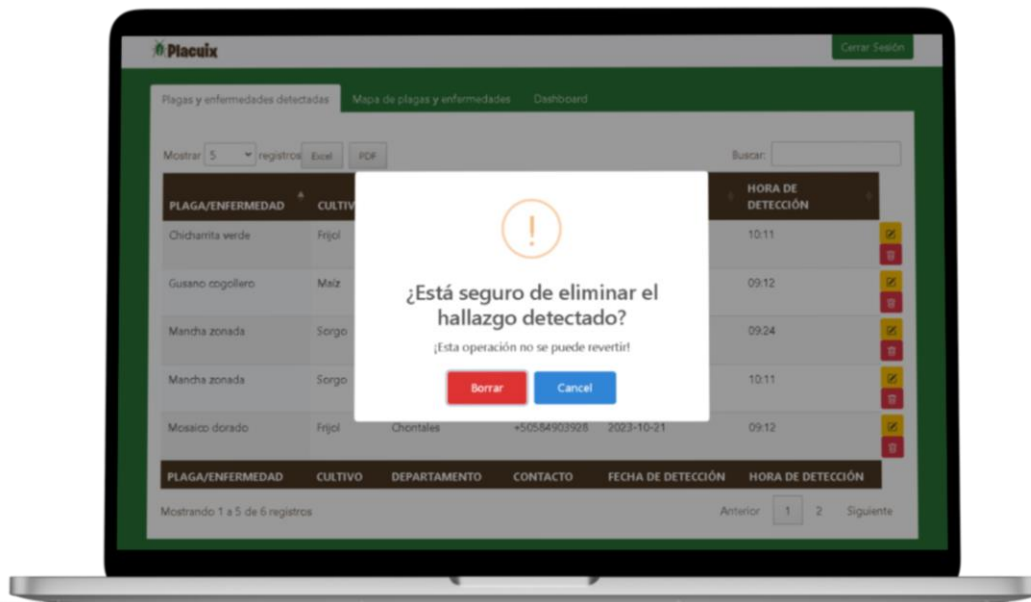
Una vez se han editado los campos se debe de presionar el botón “**Guardar**” y se guardaran los campos afectados.

8.4. Eliminando un registro de plaga o enfermedad detectada

Una vez el usuario presiona el botón con el ícono  se le pedirá que confirme la acción a través de un modal.

Figura 57

Confirmación para eliminar un registro



Si el usuario institucional está seguro de borrar el registro se debe presionar el botón **“Borrar”** y se borrará el registro seleccionado.

Importante: Los registros que se muestran en esta tabla son agregados por los agricultores desde la aplicación móvil.

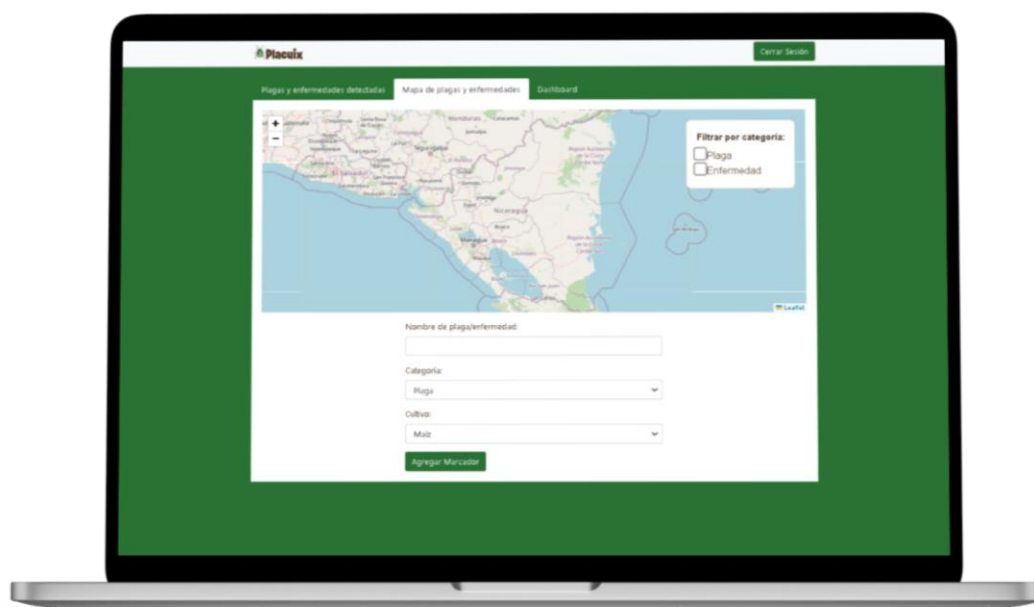
8.5. Mapa de plagas y enfermedades

8.5.1. Agregando un nuevo registro de plaga o enfermedad detectada

Al tocar la pestaña de Mapa de plagas y enfermedades la aplicación web cargará una pantalla donde se mostrará un mapa donde el usuario institucional deberá seleccionar con el puntero donde se detectó una plaga. Luego de localizar el lugar de la plaga o enfermedad detectada debe Escribir el **nombre de la plaga o enfermedad**, seleccionar la **categoría** si es una plaga o enfermedad y de que **cultivo** es la detección.

Figura 58

Agregar marcador en el mapa



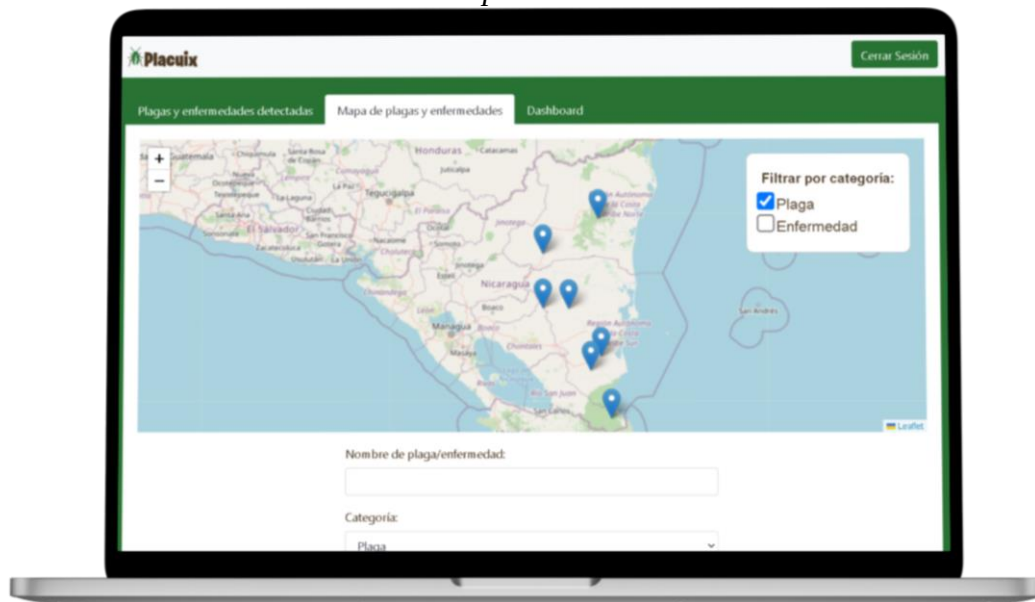
Una vez llenado todos los campos el usuario debe presionar el botón “Agregar Marcador” y a continuación se agregará el nuevo registro de detección.

Importante: Los marcadores que se agreguen aparecerán en la aplicación móvil para que todos los agricultores se informen.

Si el usuario desea ver en el mapa que registros de plagas o enfermedades están agregados puede filtrar por categoría **Plaga** o **Enfermedad**.

Figura 59

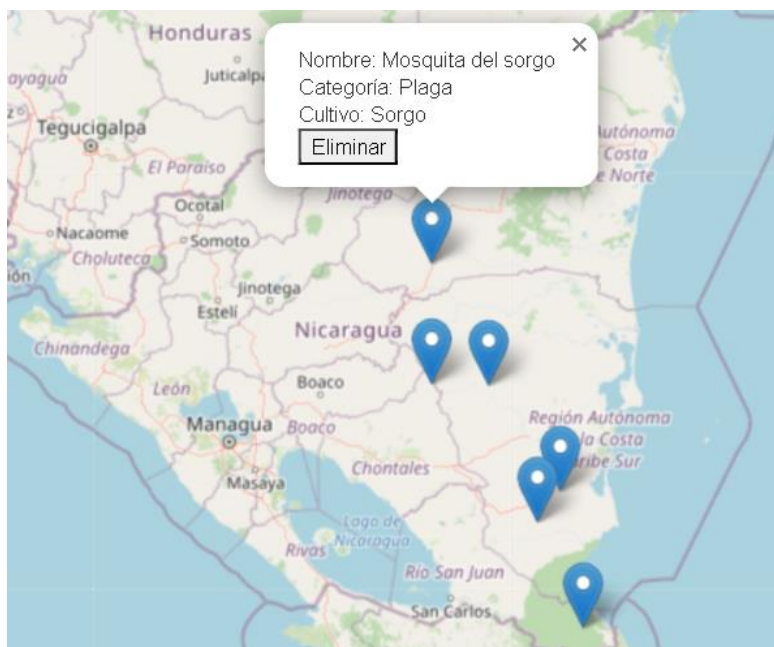
Vista de marcadores en el mapa



Además, tiene la opción de ver la información que tiene cada marcador, para mostrar la información del marcador solo debe presionarlo y le aparecerán los datos, también le permite **Eliminar** un marcador si lo colocó mal.

Figura 60

Información de un marcador

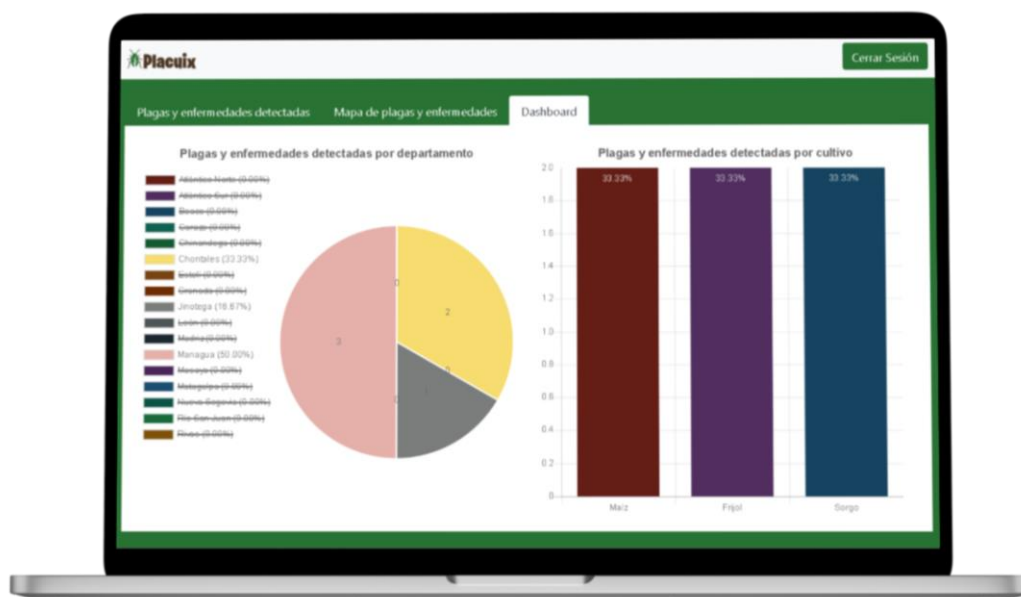


8.6. Panel (Dashboard)

El usuario institucional puede ver en un panel gráfico de las plagas y enfermedades detectadas por departamentos y por cultivos.

Figura 61

Vista del panel(dashboard)



9. Mensajes de error y resolución de problemas aplicación web

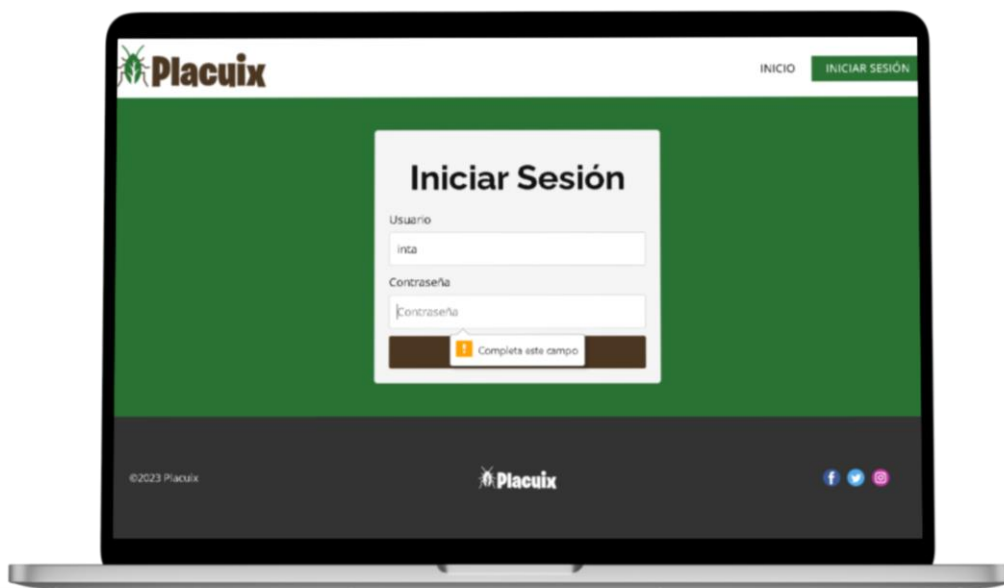
El usuario admin o institucional pueden tener problemas al utilizar la aplicación web de Placuix, es por ello que explicamos paso a paso cuales son los errores que puede cometer y como solucionarlos.

9.1. Error de campo vacío al iniciar sesión

Si el usuario al momento de ingresar en la aplicación web no llena un campo correctamente le mandará un mensaje “**Completa este campo**”.

Figura 62

Error de inicio de sesión aplicación web



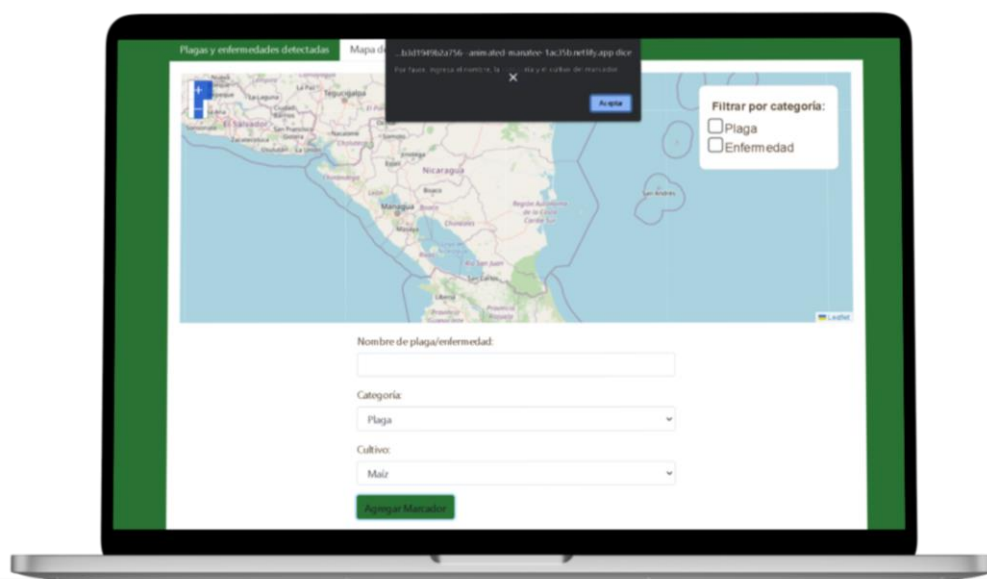
Para solucionar esto el usuario admin o institucional debe ingresar correctamente todos los campos requeridos para iniciar sesión en la aplicación.

9.2. Error al agregar un marcador al mapa de plagas y enfermedades

Si el usuario presiona el botón Agregar Marcador en el apartado de mapa de plagas y enfermedades y aún no ha llenado un campo le aparecerá un mensaje “Por favor, ingrese el nombre, la categoría y el cultivo del marcador”.

Figura 63

Error de marcador de mapa



Para solucionar esto el usuario institucional debe llenar todos los campos requeridos y presionar el botón **Agregar Marcador** para agregar un nuevo marcador al mapa.

10. Glosario

Placuix: es una abreviación de plagas y cultivos.

Plagas: Son organismos vivos como insectos, ácaros, nematodos o microorganismos que causan daño a los cultivos y afectan negativamente su desarrollo y producción.

Enfermedades: Son provocadas por patógenos como hongos, bacterias, virus o viroides que infectan a las plantas y causan síntomas y daños en su crecimiento y producción.

Cultivos: Son plantas que se cultivan en la agricultura con el fin de obtener alimentos, recursos naturales o materias primas.

Granos básicos: Son cultivos alimentarios de gran importancia económica y nutricional, como el maíz, el frijol y el sorgo, que constituyen la base de la alimentación en muchas regiones.

Detección: Es el proceso de identificar la presencia o la ocurrencia de algo, como la detección de plagas o enfermedades en los cultivos.

Maíz: Es un cultivo ampliamente cultivado en el mundo, con múltiples variedades y usos, utilizado tanto para alimentación humana como animal.

Frijol: Es una leguminosa que se cultiva por sus semillas comestibles y es una importante fuente de proteínas y nutrientes en la dieta humana.

Sorgo: Es un cultivo de cereal resistente a la sequía, utilizado principalmente para la producción de biocombustibles y como alimento para el ganado.

Clima: Es el conjunto de condiciones atmosféricas en un área determinada, incluyendo la temperatura, la humedad, la precipitación y el viento, que influyen en el crecimiento y desarrollo de los cultivos.

Foro: Es un espacio de discusión y intercambio de información en línea donde los agricultores, investigadores y expertos pueden compartir sus experiencias, conocimientos y preguntas relacionadas con la agricultura.

Monitoreo: Es el seguimiento y la observación continua de los cultivos, las plagas, las enfermedades y los factores ambientales para evaluar su estado y tomar medidas de control o manejo adecuadas.

Inteligencia artificial: Es un campo de la informática que desarrolla programas y sistemas capaces de realizar tareas que normalmente requieren de inteligencia humana, como el reconocimiento de patrones, la toma de decisiones o la resolución de problemas.

Manejo integrado: Es una estrategia de manejo de plagas y enfermedades que combina diferentes métodos y enfoques, incluyendo el uso de medidas preventivas, biológicas, culturales y químicas de manera coordinada y sostenible.

Agricultura: Es la actividad humana que se dedica al cultivo de plantas y la cría de animales para la obtención de alimentos, fibras, productos químicos y otros recursos naturales.

Agricultura de presión: Es un término que se refiere al uso intensivo y extensivo de prácticas agrícolas y tecnologías que ejercen presión sobre los ecosistemas y los recursos naturales, como la deforestación, el uso excesivo de fertilizantes o pesticidas, entre otros.

Splash Screen: Es la primera pantalla visible para el usuario cuando se inicia la aplicación. La pantalla de bienvenida es una de las pantallas más vitales de la aplicación, ya que es la primera experiencia del usuario con la aplicación.



**PLAN DE PRUEBAS
APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y
ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS**

“Creando soluciones, empoderando innovación”

Juigalpa, Chontales., diciembre de 2023

BITÁCORA DE CAMBIOS

Tabla 1

Bitácora de cambios del plan de pruebas

Fecha	Modificado por	Versión	Descripción
25/08/23	Br. Jeyson Guevara Br. Ramón González Br. Ruddy Mendoza	1.0	Definición del plan de pruebas
05/10/23	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	1.1	Características a ser probadas y a no ser probadas
29/10/23	Br. Jeyson Guevara	1.2	Definición de casos de pruebas, ejecución y resultados de pruebas
31/10/23	Br. Jeyson Guevara	1.3	Definición de criterios de aprobación y fallo, criterios de suspensión y reanudación, tareas de las pruebas, necesidades ambientales.

FORMATO PRELIMINAR AL DOCUMENTO

Tabla 2
Formato preliminar

Título:	PLAN DE PRUEBAS APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS		
ID:	PLAN_001_PLACUIX		
Fecha elaboración:	31/10/23		
Sumario	El propósito de este documento es examinar las funciones y procesos de los diferentes módulos y componentes de la aplicación Placuix, detectar y solucionar cualquier fallo o error que pueda surgir durante el periodo de pruebas. También busca confirmar si el sistema satisface los requisitos que garantizan su correcto funcionamiento.		
Palabras Claves:	Pruebas, fallos, componentes, aplicación, requerimientos, funcionalidad.		
Formato:	.pdf	Lenguaje:	Español
Dependencia:	Nicarasoft		
Versión:	1.3	Estado:	Aprobado
Autor (es):	Br. Jeyson Gabriel Jiménez Guevara Br. José Ramón González Sevilla Br. Ruddy Josué Mendoza		
Revisó:	Lic. Carmen Anielka Reyes Téllez M.Sc. Saira María Urbina Cienfuegos M.Sc. Miriam Patricia Téllez Marín		
Aprobó:	Lic. Carmen Anielka Reyes Téllez M.Sc. Saira María Urbina Cienfuegos M.Sc. Miriam Patricia Téllez Marín		
Información Adicional:	Se establecerán reuniones semanales de seguimiento para informar sobre el progreso de las pruebas y discutir cualquier problema o desafío que surja durante el proceso.		

TABLA DE CONTENIDO

BITÁCORA DE CAMBIOS.....	1
FORMATO PRELIMINAR AL DOCUMENTO.....	2
TABLA DE CONTENIDO.....	3
INDICE DE TABLAS	5
INDICE DE FIGURAS.....	6
1. INTRODUCCIÓN.....	7
1.1. OBJETIVO GENERAL	7
1.2. ESTRATEGIA DE PRUEBAS	7
1.3. ALCANCE	7
1.4. PROPOSITO	7
2. ENTREGABLES.....	8
2.1. DOCUMENTACIÓN A ENTREGAR.....	8
2.2. PRODUCT BACKLOG	9
2.2.1. PRODUCT BACKLOG ORGANIZADO EN SPRINTS.....	10
2.3. SPRINT BACKLOG	11
2.3.1. Sprint 1.....	11
2.3.2. Sprint 2.....	13
2.3.3. Sprint 3.....	17
2.3.4. Sprint 4.....	19
2.3.5. Sprint 5.....	21
2.4. CASOS DE PRUEBA POR SPRINT.....	23
2.4.1. Casos de prueba Sprint 1.....	23
2.4.2. Casos de prueba Sprint 2.....	28
2.4.3. Casos de prueba Sprint 3.....	33
2.4.4. Casos de prueba Sprint 4.....	37
2.4.5. Casos de prueba Sprint 5.....	41
2.5. EJECUCIÓN Y RESULTADOS DE PRUEBAS POR SPRINT.....	44
2.5.1. Sprint 1.....	44
2.5.2. Sprint 2.....	48
2.5.3. Sprint 3.....	54
2.5.4. Sprint 4.....	59
2.5.5. Sprint 5.....	63
3. CARACTERÍSTICAS A SER PROBADAS	67
4. CARACTERÍSTICAS A NO SER PROBADAS	71
5. CRITERIOS DE APROBACIÓN Y FALLO	76
6. CRITERIOS DE SUSPENSIÓN Y REANUDACIÓN.....	78
7. TAREAS DE LAS PRUEBAS.....	79

8.	NECESIDADES AMBIENTALES.....	82
8.1.	HARDWARE	82
8.2.	PLANEACIÓN DE COSTOS	82

INDICE DE TABLAS

Tabla 1. Bitácora de cambios del plan de pruebas	1
Tabla 2. Formato preliminar	2
Tabla 3. Documentación a entregar	8
Tabla 4. Product backlog	9
Tabla 5. Product Backlog organizado en Sprint	10
Tabla 6. Tareas Sprint 1	11
Tabla 7. Tareas Sprint 2	13
Tabla 8. Tareas Sprint 3	17
Tabla 9. Tareas Sprint 4	19
Tabla 10. Tareas Sprint 5	21
Tabla 11. Características a ser probadas Sprint 1	67
Tabla 12. Características a ser probadas Sprint 2	68
Tabla 13. Características a ser probadas Sprint 3	69
Tabla 14. Características a ser probadas Sprint 4	70
Tabla 15. Características a ser probadas Sprint 5	71
Tabla 16. Características a no ser probadas Sprint 1	72
Tabla 17. Características a no ser probadas Sprint 2	72
Tabla 18. Características a no ser probadas Sprint 3	74
Tabla 19. Características a no ser probadas Sprint 4	75
Tabla 20. Características a no ser probadas Sprint 5	75
Tabla 21. Criterios de aprobación y fallo.....	76
Tabla 22. Criterios de suspensión y reanudación.....	78
Tabla 23. Tareas de las pruebas	79
Tabla 24. Necesidades ambientales de hardware.....	82
Tabla 25. Planeación de costos	82

INDICE DE FIGURAS

Figura 1. Script testImageAnalysis	44
Figura 2. Script testClearResults	45
Figura 3. Script testSpecificIdentification	45
Figura 4. Script testOpenPlagasActivity	46
Figura 5. Script DataTable	47
Figura 6. Resultados de las pruebas Sprint 1	47
Figura 7. Script verifyEnteredLocationDisplayed	48
Figura 8. Script verifyLocationSearch	49
Figura 9. Script verifyDeviceLocationIsDisplayed	49
Figura 10. Script testMonitoringFormSubmission	50
Figura 11. Script testFabButtonVisibility	51
Figura 12. Script testLoginMessageVisibility	51
Figura 13. Script addMarker	52
Figura 14. Resultados de las pruebas Sprint 2	53
Figura 15. Script loadMarkers	54
Figura 16. Script filterMarkers	55
Figura 17. Script testLoginPerformance	56
Figura 18. Script testRegistrationSuccess	57
Figura 19. Script testRegistrationFailure	57
Figura 20. Resultados de pruebas Sprint 3	58
Figura 21. Script testHistoryTableContents	59
Figura 22. Script testMapLoading	60
Figura 23. Script testProfileFragmentElements	61
Figura 24. Script testProfileFragmentSaveChanges	61
Figura 25. Resultados de pruebas Sprint 4	62
Figura 26. Script chartData	63
Figura 27. Script testCustomDialogFragment	64
Figura 28. Script testSplashActivity	65
Figura 29. Resultados de pruebas Sprint 5	66

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

1. INTRODUCCIÓN

El presente documento describe el plan de pruebas a ejecutar para la aplicación Placuix, diseñada para la detección de plagas y enfermedades en los cultivos de granos básicos. El contenido del plan se compone de aspectos relacionados con los entregables, la documentación, las características que se probarán y no se probarán, los criterios de aprobación y fallo, los criterios de suspensión y reanudación, las tareas de las pruebas y las necesidades ambientales.

1.1.OBJETIVO GENERAL

Verificar que la aplicación Placuix para la detección de plagas y enfermedades en los cultivos de granos básicos se comporte de acuerdo con las especificaciones, de manera precisa y confiable.

1.2.ESTRATEGIA DE PRUEBAS

Realizar pruebas exhaustivas de funcionalidad en los componentes claves de la aplicación, identificando y corrigiendo errores que puedan existir.

1.3.ALCANCE

Este documento de plan de pruebas sirve como guía para llevar a cabo de manera organizada las diversas actividades de pruebas que se realizarán durante el desarrollo de la aplicación, extendiéndose a todos los aspectos críticos.

1.4.PROPOSITO

El propósito de este documento es establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación del plan de pruebas de Placuix, abarcando la definición de responsabilidades para cada una de las tareas de prueba, la asignación de recursos necesarios y los requisitos que deben ser considerados en el esfuerzo de pruebas.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2. ENTREGABLES

2.1.DOCUMENTACIÓN A ENTREGAR

Tabla 3
Documentación a entregar

DOCUMENTO	PERSONA QUIEN ENTREGA	PERSONA QUIEN RECIBE	FECHA PLANEADA	FECHA DE ENTREGA
Product Backlog organizado en Sprints	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	02/10/2023	07/10/2023
Características a ser probadas y no probadas del Sprint 1	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	05/10/2023	07/10/2023
Características a ser probadas y no probadas del Sprint 2	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	05/10/2023	07/10/2023
Características a ser probadas y no probadas del Sprint 3	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	05/10/2023	07/10/2023
Características a ser probadas y no probadas del Sprint 4	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	05/10/2023	7/10/2023
Características a ser probadas y no probadas del Sprint 5	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	05/10/2023	07/10/2023
Casos de pruebas, Ejecución y resultados de pruebas del Sprint 1 al 5	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	25/10/2023	30/10/2023
Informe final de pruebas	Br. Jeyson Guevara Br. Ruddy Mendoza Br. Ramón González	Lic. Carmen Anielka Reyes Téllez	31/10/2023	03/11/2023

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.2.PRODUCT BACKLOG

A continuación, se presenta una tabla que detalla las historias de usuario, asignando puntuaciones de prioridad en una escala de 1 a 5 y puntuaciones de impacto en una escala de 1 a 10. Estas puntuaciones ayudan a calcular la importancia de cada historia y estimar el tiempo necesario para su implementación.

Tabla 4
Product backlog

ID	Historia de usuario	Prioridad	Impacto	Importancia	Tiempo estimado
H00	Detección de plagas y enfermedades mediante imágenes	5	10	50	21 días
H01	Información sobre plagas y enfermedades	5	10	50	8 días
H02	Clima actual	4	7	28	5 días
H03	Pronóstico del tiempo	4	8	32	5 días
H04	Registro de plagas y enfermedades	4	8	32	2 días
H05	Historial de plagas y enfermedades	3	7	21	3 días
H06	Mapa de zonas de cultivos de granos básicos	3	6	18	1 día
H07	Mapa de plagas y enfermedades detectadas	4	7	28	6 días
H08	Foro agrícola	4	8	32	8 días
H09	Perfil del agricultor	3	6	18	10 días
H10	Calificación de la aplicación	3	5	15	3 días
H11	Registro de agricultor	4	7	28	2 días
H12	Visualización de calificaciones de la aplicación	3	6	18	2 días
H13	Cantidad de usuarios registrados	3	5	15	2 días
H14	Marcadores en el mapa de plagas y enfermedades	4	8	32	3 días
H15	Plagas y enfermedades detectadas	5	9	45	4 días
H16	Inicio de sesión	4	8	32	1 día
H17	Dashboard de plagas y enfermedades detectadas	3	9	27	1 día
H18	Splash Screen	2	5	10	1 día
H19	Manual de introducción	4	6	24	1 día

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.2.1. PRODUCT BACKLOG ORGANIZADO EN SPRINTS

Aquí se presentan las historias de usuario organizadas en orden descendente de importancia, lo que facilita su agrupación en Sprints para la planificación y desarrollo del proyecto.

Tabla 5

Product Backlog organizado en Sprint

ID	Historia de usuario	Importancia	Sprint programado	Fecha de inicio	Fecha de fin
H00	Detección de plagas y enfermedades mediante imágenes	50	1	14/08/23	03/09/23
H01	Información sobre plagas y enfermedades	50	1	14/08/23	21/08/23
H15	Plagas y enfermedades detectadas	45	1	22/08/23	25/08/23
H03	Pronóstico del tiempo	32	2	04/09/23	08/09/23
H04	Registro de plagas y enfermedades	32	2	11/09/23	12/09/23
H08	Foro agrícola	32	2	13/09/23	20/09/23
H14	Marcadores en el mapa de plagas y enfermedades	32	2	21/09/23	23/09/23
H16	Inicio de sesión	32	2	24/09/23	24/09/23
H02	Clima actual	28	3	25/09/23	29/09/23
H07	Mapa de plagas y enfermedades detectadas	28	3	01/10/23	05/10/23
H11	Registro de agricultor	28	3	06/10/23	07/10/23
H17	Dashboard de plagas y enfermedades detectadas	27	3	08/10/23	08/10/23
H19	Manual de introducción	24	4	09/10/23	09/10/23
H05	Historial de plagas y enfermedades	21	4	10/10/23	11/10/23
H06	Mapa de zonas de cultivos de granos básicos	18	4	12/10/23	12/10/23
H09	Perfil del agricultor	18	4	13/10/23	22/10/23
H12	Visualización de calificaciones de la aplicación	18	5	23/10/23	24/10/23
H10	Calificación de la aplicación	15	5	25/10/23	27/10/23
H13	Cantidad de usuarios registrados	15	5	28/10/23	28/10/23
H18	Splash Screen	10	5	29/10/23	29/10/23

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.3.SPRINT BACKLOG

Es la lista de tareas que se eligen del Product Backlog en la planificación del sprint actual. Estas tareas son esenciales para llevar a cabo el desarrollo de un incremento del producto.

2.3.1. Sprint 1

A continuación, se describe el inicio del primer sprint, que tiene una duración de 3 semanas y abarca 3 historias de usuario junto con un total de 28 tareas, que incluyen las pruebas de software necesarias.

Tabla 6
Tareas Sprint 1

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H00	Detección de plagas y enfermedades	Diseñar la interfaz de usuario de la función de detección de plagas y enfermedades	3 hrs	Ramón González
		Crear prototipos interactivos para evaluar la usabilidad y la experiencia del usuario	3 hrs	Ramón González
		Diseñar elementos visuales claros para representar los resultados de detección	2 hrs	Ramón González
		Colaborar con el investigador para recopilar y organizar datos sobre plagas y enfermedades en la agricultura para incorporarlos en el diseño	2 hrs	Ramón González
		Implementar la funcionalidad de acceso a la cámara del dispositivo y carga de imágenes desde la galería.	2 hrs	Jeyson Guevara
		Desarrollar el algoritmo de detección de plagas y enfermedades	2 hrs	Jeyson Guevara
		Entrenar e integrar el modelo de aprendizaje de plagas y enfermedades agrícolas para referencias de detección.	12 hrs	Jeyson Guevara
		Programar la lógica de presentación de resultados detallados	1 hrs	Jeyson Guevara
		Recopilar datos e información sobre plagas y enfermedades agrícolas comunes en los cultivos de granos básicos	10 hrs	Ruddy Mendoza

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

H01	Información sobre plagas y enfermedades	Colaborar con el diseñador para asegurarse de que la interfaz de usuario incluya la información necesaria para la detección.	2 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario clara y fácil de navegar para acceder a la lista de plagas y enfermedades en los cultivos de maíz, frijol y sorgo.	4 hrs	Ramón González
		Crear un diseño de pantalla que muestre las descripciones detalladas de las plagas y enfermedades.	2 hrs	Ramón González
		Diseñar y crear la base de datos	4 hrs	Jeyson Guevara
		Implementar la funcionalidad que permita a los usuarios acceder a la lista de plagas y enfermedades en los cultivos específicos.	3 hrs	Jeyson Guevara
		Desarrollar la lógica para mostrar descripciones detalladas de cada plaga y enfermedad en la interfaz de usuario.	4 hrs	Jeyson Guevara
		Programar la funcionalidad que presente recomendaciones claras y prácticas de manejo integrado para el control y la prevención de plagas y enfermedades.	2 hrs	Jeyson Guevara
		Recopilar datos sobre las plagas y enfermedades que afectan a los cultivos de maíz, frijol y sorgo	4 hrs	Ruddy Mendoza
		Proporcionar recomendaciones basadas en prácticas de manejo integrado para el control y la prevención de cada plaga o enfermedad.	3 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario intuitiva y eficiente que permita a las instituciones acceder a la tabla de plagas y enfermedades detectadas.	3 hrs	Ramón González
H15	Plagas y enfermedades detectadas	Crear un diseño de pantalla que muestre de manera organizada la información detallada sobre cada plaga o enfermedad	2 hrs	Ramón González
		Diseñar elementos de interfaz que permitan a los usuarios ordenar y filtrar la tabla para buscar información específica.	1 hrs	Ramon González
		Creación y diseño de la base de datos	1 hrs	Jeyson Guevara

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Implementar la funcionalidad que permita a las instituciones acceder a la tabla de plagas y enfermedades detectadas a través de la aplicación web.	3 hrs	Jeyson Guevara
Programar la lógica que recopile y muestre información detallada sobre cada plaga o enfermedad	3 hrs	Jeyson Guevara
Desarrollar la funcionalidad de ordenar y filtrar la tabla para que los usuarios puedan buscar información específica de manera eficiente.	1 hrs	Jeyson Guevara
Colaborar con el diseñador y el desarrollador para definir los campos de información detallada que deben incluirse en la tabla.	1 hrs	Ruddy Mendoza
Proporcionar retroalimentación y guía sobre la mejor manera de presentar los datos en la tabla.	1 hrs	Ruddy Mendoza
Pruebas de software	3 hrs	Jeyson Guevara

2.3.2. Sprint 2

El segundo sprint tiene una duración de 3 semanas y consta de 5 historias de usuario y 41 tareas en total, incluyendo las pruebas de software correspondientes.

Tabla 7
Tareas Sprint 2

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H03	Pronóstico del tiempo	Diseñar una interfaz de usuario amigable que permita a los usuarios ver el pronóstico del tiempo.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre claramente el pronóstico a corto plazo y a largo plazo para la ubicación actual o una ubicación específica.	1 hrs	Ramón González
		Diseñar elementos visuales que representen los datos de temperatura, humedad, velocidad del viento, nubosidad y estado del día de manera comprensible.	1 hrs	Ramón González

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

H04	Registro de plagas y enfermedades	Implementar la funcionalidad que obtenga automáticamente la ubicación actual del agricultor o permita la búsqueda de ubicaciones específicas.	2 hrs	Jeyson Guevara
		Programar la obtención de datos meteorológicos a partir de la API.	1 hrs	Jeyson Guevara
		Desarrollar la lógica para presentar el pronóstico del tiempo a corto plazo (próximas 24 horas) y a largo plazo (próximos 5 días).	3 hrs	Jeyson Guevara
		Integrar la funcionalidad de presentación de detalles sobre temperatura, humedad, velocidad del viento, nubosidad y estado del día.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y detalles clave a mostrar en la interfaz de usuario.	1 hrs	Ruddy Mendoza
		Proporcionar orientación sobre la presentación efectiva de datos meteorológicos para que sean útiles para la planificación agrícola.	1 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario intuitiva y fácil de usar para el formulario de registro de plagas y enfermedades.	1 hrs	Ramón González
		Crear un diseño de formulario que incluya campos solicitados	1 hrs	Ramón González
		Diseñar elementos visuales que guíen al agricultor a través del proceso de registro de manera efectiva.	1 hrs	Ramón González
		Implementar la funcionalidad del formulario de registro, incluyendo la creación de campos y la lógica de validación.	2 hrs	Jeyson Guevara
		Programar la lógica para registrar la información ingresada por el agricultor	1 hrs	Jeyson Guevara
		Integrar un sistema de almacenamiento de datos para guardar la información registrada de manera segura.	2 hrs	Jeyson Guevara

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

H08	Foro agrícola	Desarrollar la funcionalidad de envío de la información a la base de datos de plagas y enfermedades detectadas.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos clave que deben incluirse en el formulario de registro.	1 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario atractiva y fácil de usar para el foro agrícola en la aplicación móvil.	2 hrs	Ramón González
		Crear un diseño de pantalla que permita a los agricultores crear y comentar publicaciones	2 hrs	Ramón González
		Diseñar elementos visuales que faciliten la navegación por las publicaciones y los comentarios de otros agricultores.	1 hrs	Ramón González
		Implementar la funcionalidad del foro agrícola	3 hrs	Jeyson Guevara
		Programar la lógica que permita a los agricultores ver las publicaciones realizadas por otros usuarios	2 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que permita a los agricultores comentar las publicaciones de otros usuarios utilizando mensajes de texto	2 hrs	Jeyson Guevara
		Diseñar y crear la base de datos	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la creación de publicaciones y comentarios.	1 hrs	Ruddy Mendoza
H14	Marcadores en el mapa de plagas y enfermedades	Diseñar una interfaz de usuario en la aplicación web que permita agregar marcadores en el mapa de plagas y enfermedades de manera intuitiva.	1 hrs	Ramón González
		Crear un diseño de pantalla que incluya un campo para ingresar el nombre de la plaga o enfermedad detectada al agregar un marcador.	1 hrs	Ramón González

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

H16	Inicio de sesión	Diseñar elementos visuales que indiquen claramente cómo colocar el marcador de manera precisa en el mapa interactivo.	1 hrs	Ramón González
		Implementar la funcionalidad que permita a las instituciones agregar marcadores en el mapa interactivo desde la aplicación web	2 hrs	Jeyson Guevara
		Programar la lógica que permita a los usuarios agregar el nombre de la plaga o enfermedad detectada al colocar un marcador en el mapa.	2 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que garantice la colocación precisa del marcador en la ubicación deseada en el mapa interactivo.	1 hrs	Jeyson Guevara
		Integrar una base de datos que almacene la información de ubicaciones de marcadores junto con los nombres de plagas o enfermedades detectadas	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y detalles clave al agregar marcadores en el mapa.	1 hrs	Ruddy Mendoza
		Diseñar una interfaz de inicio de sesión que sea clara y fácil de usar.	1 hrs	Ramón González
		Crear un diseño de formulario de inicio de sesión	1 hrs	Ramón González
		Diseñar elementos visuales que guíen a los usuarios a través del proceso de autenticación de manera efectiva.	1 hrs	Ramón González
		Implementar la funcionalidad del formulario de inicio de sesión, incluyendo la validación de campos y la autenticación de usuarios.	1 hrs	Jeyson Guevara
		Programar la lógica que permita verificar la autenticidad de las credenciales ingresadas por los usuarios.	1 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que muestre las opciones correspondientes una vez que un usuario ha iniciado sesión con éxito.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y	1 hrs	Ruddy Mendoza

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

elementos clave en el formulario de inicio de sesión.			
Pruebas de software	4 hrs	Jeyson Guevara	

2.3.3. Sprint 3

El segundo sprint tiene una duración de 3 semanas y consta de 4 historias de usuario y 26 tareas en total, incluyendo las pruebas de software correspondientes.

Tabla 8
Tareas Sprint 3

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H02	Clima actual	Diseñar una interfaz de usuario que permita ver el clima actual de manera clara y fácil de entender.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre de manera efectiva los datos climáticos	1 hrs	Ramón González
		Diseñar elementos visuales que reflejen el estado del día de forma intuitiva (soleado, nublado, lluvioso, etc.).	1 hrs	Ramón González
		Implementar la funcionalidad que permita obtener automáticamente la ubicación actual del agricultor o permitirle buscar una ubicación específica.	2 hrs	Jeyson Guevara
		Programar la lógica para obtener datos precisos y actualizados regularmente sobre el clima	2 hrs	Jeyson Guevara
		Desarrollar la funcionalidad para mostrar esta información en tiempo real en la interfaz de usuario.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave relacionados con la visualización de datos climáticos.	1 hrs	Ruddy Mendoza
H07	Mapa de plagas y enfermedades	Diseñar una interfaz de usuario atractiva y fácil de usar para el mapa interactivo de plagas y enfermedades.	2 hrs	Ramón González
		Crear un diseño de pantalla que muestre de manera efectiva las áreas donde se han	2 hrs	Ramón González

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

		detectado plagas y enfermedades en los cultivos.		
H11	Registro de agricultor	Diseñar elementos visuales que permitan a los usuarios hacer zoom y explorar el mapa para obtener detalles específicos de ubicación.	1 hrs	Ramón González
		Implementar la funcionalidad del mapa interactivo	3 hrs	Jeyson Guevara
		Programar la lógica para permitir a los usuarios hacer zoom y explorar el mapa para obtener detalles de ubicación	1 hrs	Jeyson Guevara
		Desarrollar las herramientas de búsqueda y filtro que permitan a los usuarios encontrar información sobre plagas y enfermedades en áreas específicas.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización del mapa de plagas y enfermedades.	1 hrs	Ruddy Mendoza
		Diseñar una interfaz de usuario clara y atractiva que permita acceder a la opción de registro desde la pantalla de inicio de sesión de la aplicación.	1 hrs	Ramón González
		Crear un diseño de formulario de registro que incluya campos para recopilar la información necesaria.	1 hrs	Ramón González
		Diseñar elementos visuales que guíen a los usuarios a través del proceso de registro de manera efectiva.	1 hrs	Ramón González
		Implementar la funcionalidad que permita acceder a la opción de registro desde la pantalla de inicio de sesión de la aplicación.	2 hrs	Jeyson Guevara
		Programar la lógica para validar y procesar la información ingresada por los Agricultores en el formulario de registro.	1 hrs	Jeyson Guevara
		Desarrollar la funcionalidad que envíe una confirmación de registro exitoso una vez que el proceso de registro esté completo.	1 hrs	Jeyson Guevara
		Integrar un sistema de autenticación que permita a los Agricultores iniciar sesión en	1 hrs	Jeyson Guevara

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

		sus cuentas después de registrarse para acceder a las funcionalidades de la aplicación.		
H17	Dashboard de plagas y enfermedades detectadas	Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en el formulario de registro.	1 hrs	Ruddy Mendoza
		Diseño de gráficos por cultivo y departamento	4 hrs	Ramón González
		Investigación de librerías de implementación de gráficos	2 hrs	Ruddy Mendoza
		Desarrollo de la funcionalidad del dashboard	6 hrs	Jeyson Guevara
		Pruebas de software	3 hrs	Jeyson Guevara

2.3.4. Sprint 4

El cuarto sprint tiene una duración de 2 semanas y consta de 4 historias de usuario y 24 tareas en total, incluyendo las pruebas de software correspondientes.

Tabla 9
Tareas Sprint 4

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H19	Manual de introducción	Diseño de las pantallas del manual	8 hrs	Ramón González
		Describir las funcionalidades principales	2 hrs	Ruddy Mendoza
		Desarrollo de la lógica del manual	10 hrs	Jeyson Guevara
		Funcionalidad de deslizamiento e intuitividad	4 hrs	Jeyson Guevara
H05	Historial de plagas y enfermedades	Diseñar una interfaz de usuario que permita ver el historial de plagas y enfermedades en un formato de tabla de manera clara y organizada.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre la tabla de registros de plagas y enfermedades de forma eficiente.	1 hrs	Ramón González

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

		Diseñar elementos visuales que faciliten la navegación y la interacción con la tabla, incluyendo opciones de ordenamiento y filtrado.	2 hrs	Ramón González
		Implementar la funcionalidad que permita cargar y mostrar la tabla de registros de plagas y enfermedades en la interfaz de usuario.	1 hrs	Jeyson Guevara
		Programar la lógica para obtener y presentar la información detallada de cada entrada en la tabla	2 hrs	Jeyson Guevara
		Desarrollar las opciones de ordenamiento y filtrado	2 hrs	Jeyson Guevara
		Integrar base de datos que mantenga actualizado el historial de plagas y enfermedades	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización de la tabla de historial.	1 hrs	Ruddy Mendoza
H06	Mapa de zonas de cultivos de granos básicos	Diseñar una interfaz de usuario que permita acceder al mapa interactivo de zonas de cultivos de granos básicos en Nicaragua de manera clara y atractiva.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre el mapa de forma eficiente.	1 hrs	Ramón González
		Implementar la funcionalidad que permita cargar y mostrar el mapa interactivo de zonas de cultivos de granos básicos en Nicaragua en la interfaz de usuario.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización del mapa de zonas de cultivos de granos básicos.	1 hrs	Ruddy Mendoza
H09	Perfil del agricultor	Diseñar una interfaz de usuario que permita visualizar y actualizar su información de perfil de manera clara y atractiva.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre la sección de edición de perfil en la configuración de la aplicación.	1 hrs	Ramón González

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Diseñar elementos visuales que permita cambiar imagen de perfil, editar nombre de usuario, profesión y ubicación.	1 hrs	Ramón González
Implementar la funcionalidad que permita acceder a la sección de edición de perfil desde la configuración de la aplicación.	2 hrs	Jeyson Guevara
Programar la lógica para cargar y cambiar la imagen de perfil, así como para editar el nombre de usuario, profesión y ubicación.	1 hrs	Jeyson Guevara
Desarrollar la funcionalidad que guarde los cambios realizados en el perfil y proporcione una confirmación de que han sido guardados correctamente.	1 hrs	Jeyson Guevara
Colaborar con el diseñador y el desarrollador para definir los campos y elementos clave en la visualización y edición del perfil.	1 hrs	Ruddy Mendoza
Pruebas de software	3 hrs	Jeyson Guevara

2.3.5. Sprint 5

El quinto sprint tiene una duración de 1 semana y consta de 4 historias de usuario y 17 tareas en total, incluyendo las pruebas de software correspondientes.

Tabla 10
Tareas Sprint 5

ID	Historia de usuario	Tareas	Tiempo	Tarea tomada por:
H12	Visualización de calificaciones de la aplicación	Diseñar una interfaz de usuario que permita ver las calificaciones de la aplicación de manera clara y efectiva.	1 hrs	Ramón González
		Crear un diseño de pantalla que muestre las calificaciones en forma de gráfico de pastel.	1 hrs	Ramón González
		Diseñar elementos visuales que indiquen claramente si una calificación fue considerada "útil" o "no útil"	1 hrs	Ramón González
		Implementar la funcionalidad que permita acceder y visualizar las calificaciones de la aplicación.	2 hrs	Jeyson Guevara

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

H10	Calificación de la aplicación	Programar la lógica para mostrar las calificaciones en un gráfico de pastel, con indicadores claros de "útil" o "no útil".	2 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir la presentación adecuada de las calificaciones de la aplicación.	1 hrs	Ruddy Mendoza
		Diseñar una ventana emergente que invite a calificar la aplicación de manera atractiva y no intrusiva.	1 hrs	Ramón González
		Crear elementos visuales claros para las opciones de calificación, como "útil" o "no útil".	1 hrs	Ramón González
		Implementar la funcionalidad de la ventana emergente que aparecerá para invitar a calificar la aplicación.	1 hrs	Jeyson Guevara
		Programar la lógica para registrar la calificación seleccionada en la base de datos de la aplicación.	1 hrs	Jeyson Guevara
		Colaborar con el diseñador y el desarrollador para definir la apariencia adecuada de la ventana emergente de calificación.	1 hrs	Ruddy Mendoza
		Diseñar elementos visuales que permitan mostrar de manera clara y efectiva el número de usuarios registrados en la interfaz.	1 hrs	Ramón González
		Programar la lógica para obtener el número total de usuarios registrados y mostrarlo claramente en un elemento de texto en la interfaz.	1 hrs	Jeyson Guevara
		Proporcionar orientación sobre cómo presentar la información de manera efectiva para que pueda ser interpretada fácilmente.	1 hrs	Ruddy Mendoza
H18	Splash Screen	Diseño del splash screen	1 hrs	Ramón González
		Desarrollo de la lógica del splash screen	2 hrs	Jeyson Guevara
		Pruebas del software	3 hrs	Jeyson Guevara

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.4.CASOS DE PRUEBA POR SPRINT

2.4.1. Casos de prueba Sprint 1

Test Case ID	P01	Test Case Description	El proceso de selección de una imagen de la galería y la ejecución de un análisis de imagen. Luego, verifica que el resultado se muestre en el fragmento.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	Imágenes en la galería		
2	El usuario ha seleccionado una imagen				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Hacer clic en el botón de la galería	Se espera que la galería se abra	Galería abierta	Pass
2	Seleccionar una imagen de la galería	Se espera que la imagen seleccionada se muestre correctamente	Imagen carga correctamente	Pass
3	Completar el análisis de imagen	Se espera que el modelo analice la imagen	El modelo analiza la imagen	Pass
4	Verificar que el resultado se muestre	Se espera que el resultado se muestre	El resultado se muestra en la interfaz	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P02	Test Case Description	Verifica que el resultado se muestre en el fragmento, también verifica que el resultado no esté vacío (es decir, que no sea una cadena vacía).	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	Imágenes en la galería		
2	El usuario ha seleccionado una imagen				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Hacer clic en el botón de la galería	Se espera que la galería se abra	Galería abierta	Pass
2	Seleccionar una imagen de la galería	Se espera que la imagen seleccionada se muestre correctamente	Imagen carga correctamente	Pass
3	Completar el análisis de imagen	Se espera que el modelo analice la imagen	El modelo analiza la imagen	Pass
4	Verificar que el resultado se muestre	Se espera que el resultado se muestre	El resultado se muestra en la interfaz	Pass
5	Verificar que el resultado no esté vacío	Se espera que el resultado no esté vacío	Cadena de texto en pantalla	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P03	Test Case Description	Verifica que la identificación específica ("Roya (Puccinia sorghi)") esté presente en los resultados.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	Imágenes en la galería		
2	El usuario ha seleccionado una imagen				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Hacer clic en el botón de la galería	Se espera que la galería se abra	Galería abierta	Pass
2	Seleccionar una imagen de la galería	Se espera que la imagen seleccionada se muestre correctamente	Imagen carga correctamente	Pass
3	Completar el análisis de imagen	Se espera que el modelo analice la imagen	El modelo analiza la imagen	Pass
4	Verificar que el resultado se muestre	Se espera que el resultado se muestre	El resultado se muestra en la interfaz	Pass
5	Verificar que la identificación sea específica	Se espera que la identificación específica esté presente	Nombre de la enfermedad se muestra correctamente	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P04, P05	Test Case Description	Verificar si al hacer clic en un "CardView" de un cultivo (por ejemplo, el cultivo de maíz), la aplicación inicia correctamente la actividad "PlagasActivity" con el cultivo correcto (en este caso, "maíz").	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	CardView Maíz		
2	El usuario ha seleccionado el cardView del maíz				
3	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Clic en el CardView de un cultivo	Se espera que se haga clic en el cultivo (maíz)	Selección del cardview correcto	Pass
2	Verificación de la actividad PlagasActivity	Se espera que la actividad se inicie con "maíz"	Abre la interfaz con las plagas y enfermedades del maíz	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P06	Test Case Description	Verificar que la aplicación web muestre una tabla con información sobre las plagas y enfermedades detectadas por los usuarios en la aplicación móvil.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	DataTable		
2	El usuario ha ingresado a la aplicación con un tipo de cuenta institución.				
3	El usuario ha navegado hacia la pestaña de plagas y enfermedades detectadas.				
4	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Dar clic a la pestaña de navegación del tab panel llamada "Plagas y enfermedades detectadas"	Se espera que se muestre la tabla con las plagas y enfermedades detectadas.	Se muestra la interfaz con la tabla de las plagas y enfermedades	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.4.2. Casos de prueba Sprint 2

Test Case ID	P07, P08	Test Case Description	Verificar que el usuario puede hacer clic en el icono de búsqueda y escribir una ubicación en el campo de búsqueda, así como obtener la ubicación actual del dispositivo.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	----------------	--------------------	------------	---	------

S #	Prerequisites:	S #	Test Data
1	Cargar la aplicación	1	locationTextView=" Juigalpa"
2	La actividad "Home" de la aplicación está cargada		
3	Tener conexión a internet		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ingresar "Juigalpa" en el campo de búsqueda	"Juigalpa" se muestra en el campo de búsqueda.	Campo de búsqueda con el texto Juigalpa	Pass
2	Hacer clic en el icono de búsqueda	El campo de búsqueda se activa y se puede escribir en él.	Icono de búsqueda activado	Pass
3	Cerrar el teclado virtual	El teclado virtual se cierra.	El teclado se cerró	Pass
4	Verificar la ubicación mostrada en la interfaz	La ubicación se muestra como "Juigalpa".	Se muestra Juigalpa como ubicación en la interfaz	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P09	Test Case Description	Comprobar que la búsqueda de ubicación devuelve datos precisos y que se muestran en la interfaz de usuario de acuerdo a lo esperado.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	locationTextView=" Juigalpa"		
2	La actividad "Home" de la aplicación está cargada	2	temperatureTextView=" 29°C"		
3	Tener conexión a internet	3	humidityTextView= "74%"		
		4	windTextView= "7 km/h"		
		5	cloudinnesTextView= "40%"		
		6	weatherIconImageView=""		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ingresar "Juigalpa" en el campo de búsqueda	"Juigalpa" se muestra en el campo de búsqueda.	Campo de búsqueda con el texto Juigalpa	Pass
2	Hacer clic en el icono de búsqueda	El campo de búsqueda se activa y se puede escribir en él.	Icono de búsqueda activado	Pass
3	Cerrar el teclado virtual	El teclado virtual se cierra.	El teclado se cerró	Pass
4	Verificar la ubicación mostrada en la interfaz	La ubicación se muestra como "Juigalpa".	Se muestra Juigalpa como ubicación en la interfaz	Pass
5	Verificar la temperatura mostrada en la interfaz	La temperatura se muestra como "29°C".	Se muestra 29°C como temperatura	Pass
6	Verificar la humedad mostrada en la interfaz	La humedad se muestra como "74%".	Se muestra 74% de humedad	Pass
7	Verificar la velocidad del viento mostrada en la interfaz	La velocidad del viento se muestra como "7 km/h".	Se muestra 7 km/h de velocidad del viento	Pass
8	Verificar la nubosidad mostrada en la interfaz	La nubosidad se muestra como "40%".	Se muestra una nubosidad del 40%	Pass
9	Verificar la visibilidad del icono del clima en la interfaz	El icono del clima está visible.	Se muestra el icono del clima en el momento	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P10	Test Case Description	Verificar el envío del formulario de monitoreo con los campos de textos proporcionados.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	cropNameEditText=" Nombre del cultivo"		
2	El usuario ha ingresado a la aplicación con un tipo de cuenta agricultor	2	pestNameEditText=" Nombre de la plaga"		
3	Carga la pantalla del formulario	3	contactoEditText=" Número de contacto"		
4	Tener conexión a internet	4	cityAutoCompleteTextView=" Nombre de la Ciudad"		
		5	submitButton		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar la actividad MainActivity.	La actividad MainActivity se inicia con éxito.	La actividad se muestra correctamente en pantalla.	Pass
2	Rellenar el campo cropNameEditText con "Nombre del Cultivo".	El campo cropNameEditText contiene "Nombre del Cultivo".	Se ingresa el Nombre del cultivo correctamente.	Pass
3	Rellenar el campo pestNameEditText con "Nombre de la Plaga".	El campo pestNameEditText contiene "Nombre de la Plaga".	Se ingresa el Nombre de la Plaga correctamente	Pass
4	Reemplazar el contenido del campo contactoEditText con "Número de Contacto".	El campo contactoEditText contiene "Número de Contacto".	Se ingresa el Número de Contacto correctamente	Pass
5	Rellenar el campo cityAutoCompleteTextView con "Nombre de la Ciudad".	El campo cityAutoCompleteTextView contiene "Nombre de la Ciudad".	Se ingresa el Nombre de la Ciudad correctamente	Pass
6	Hacer clic en el botón de envío submitButton.	La aplicación procesa el formulario y realiza la acción de envío.	Se envía el formulario con los datos ingresados correctamente	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P11	Test Case Description	Comprobar la visibilidad de elementos específicos en la interfaz de usuario de publicaciones. El propósito es asegurarse de que estos elementos estén presentes y visibles para los usuarios al momento de crear una publicación en el foro.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	fab		
2	La actividad "MainActivity" de la aplicación está cargada	2	login message text		
3	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar la actividad MainActivity.	La actividad MainActivity se inicia con éxito.	La actividad se muestra correctamente en la aplicación.	Pass
2	Verificar la visibilidad del botón FAB (fab)	El botón FAB está visible en la interfaz.	Se muestra el botón flotante para crear publicaciones	Pass
3	Iniciar la actividad MainActivity.	La actividad MainActivity se inicia con éxito.	La actividad se vuelve a reiniciar y se muestra correctamente en la aplicación.	Pass
4	Verificar la visibilidad del mensaje de inicio de sesión.	El mensaje de inicio de sesión es visible en la interfaz.	Se muestra otro botón con el texto de que inicie sesión para crear publicaciones.	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P12	Test Case Description	Verificar que en el mapa interactivo de plagas y enfermedades se agreguen correctamente los marcadores junto con el nombre de la plaga o enfermedad detectada	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S	Test Data		
1	Cargar la aplicación	1	markerName="Gusano cogollero"		
2	El usuario accedió a la aplicación web con un tipo de cuenta institución	2	markerCategory="Plaga"		
3	Navega a pestaña del mapa interactivo	3	markerCrop="Maíz"		
4	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Rellenar el campo markerName con "Gusano cogollero"	Campo markerName con el texto Gusano cogollero	Campo de texto con su valor correcto	Pass
2	Seleccionar el campo markerCategory con "Plaga"	Select markerCategory con la opción Plaga	Select con la opción seleccionada	Pass
3	Seleccionar el campo markerCrop con "Maíz"	Select markerCrop con la opción Maíz	Select con la opción seleccionada	Pass
4	Hacer clic en el botón addMarker	Ventana emergente pidiendo que se ubique el marcador en el mapa	Aparece una ventana con el mensaje esperado	Pass
5	Seleccionar la posición del marcador en el mapa	Marcador se coloca en el mapa	El marcador agregado aparece en el mapa	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.4.3. Casos de prueba Sprint 3

Test Case ID	P13	Test Case Description	Verificar que el mapa interactivo de plagas y enfermedades detectadas muestra clara y precisamente las áreas donde se han detectado.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	Db.collection(markers)		
2	El usuario accedió a la aplicación web con un tipo de cuenta institución				
3	Navega a pestaña del mapa interactivo				
4	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Abrir la pestaña del mapa de plagas y enfermedades	Se muestra la pestaña de Mapa de plagas y enfermedades	Se obtuvo el resultado esperado	Pass
2	Visualizar el mapa con sus marcadores	Se muestra el mapa con los marcadores en las ubicaciones que fueron agregados	Se obtuvo el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P14	Test Case Description	Comprobar que el mapa de plagas y enfermedades incluye un filtro para mostrar los marcadores agregados según la categoría seleccionada	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	filterPlaga=true		
2	El usuario accedió a la aplicación web con un tipo de cuenta institución	2	filterEnfermedad=true		
3	Navega a pestaña del mapa interactivo				
4	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Seleccionar la casilla de filterPlaga	Muestra los marcadores pertenecientes a la categoría Plaga	Se muestra el resultado esperado	Pass
2	Seleccionar la casilla de filterEnfermedad	Muestra los marcadores pertenecientes a la categoría Enfermedad	Muestra el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P15	Test Case Description	Verificar el proceso de registro en la aplicación, tanto en el escenario de éxito como en el escenario de falla, manejando correctamente los datos ingresados por el usuario y responda de acuerdo a las condiciones de registro.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	regName=" TestUser", ""		
2	El usuario no ha iniciado sesión	2	regMail=" test@example.com", "invalid-email"		
3	Navega a la interfaz de registro de usuario	3	regPassword=" password", "password"		
4	Tener conexión a internet	4	regPassword2=" password", "different-password"		
		5	regBtn		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar la actividad RegisterActivity.	La actividad RegisterActivity se inicia con éxito.	Muestra la interfaz de registro	Pass
2	Simular la entrada de datos válidos en los campos de registro.	El sistema permite el registro exitoso y redirige o muestra un mensaje de éxito.	Se agregan los datos de entrada de simulación	Pass
3	Realizar un clic en el botón de registro.	La aplicación procesa el registro con éxito y redirige o muestra un mensaje de éxito.	Se envían los datos de simulación y se registra correctamente	Pass
4	Iniciar la actividad RegisterActivity	La actividad RegisterActivity se inicia con éxito.	Reinicializa la interfaz de registro	Pass
5	Simular la entrada de datos incorrectos en los campos de registro.	El sistema detecta errores en los datos y muestra mensajes de error.	Se agregan los datos de entrada de simulación	Pass
6	Realizar un clic en el botón de registro.	El sistema detecta errores en los datos y muestra mensajes de error.	Se envían los datos de simulación y muestra un error	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P16	Test Case Description	Evaluar el rendimiento del proceso de inicio de sesión en la actividad LoginActivity de la aplicación Android.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	login_mail=" correo@gmail.com"		
2	El usuario ya tiene una cuenta creada	2	login_password=" correoito"		
	Tener conexión a internet	3	loginBtn		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar la medición del tiempo registrando el momento de inicio.	La medición del tiempo se inicia con éxito.	Se comienza a contar el tiempo	Pass
2	Introducir "correo@gmail.com" en el campo de correo electrónico.	El campo de correo electrónico contiene "correo@gmail.com".	Se ingresa correctamente el valor de prueba	Pass
3	Establecer la contraseña como "correoito" en el campo de contraseña.	El campo de contraseña contiene "correoito".	Se ingresa correctamente el valor de prueba	Pass
4	Hacer clic en el botón de inicio de sesión (loginBtn).	La aplicación procesa el inicio de sesión con éxito y redirige o muestra un mensaje de éxito.	Redirige a otra pantalla que solo se puede acceder al iniciar sesión correctamente	Pass
5	Detener la medición del tiempo registrando el momento de finalización.	La medición del tiempo se detiene con éxito.	Se detiene el contador de tiempo	Pass
6	Calcular el tiempo transcurrido durante el proceso de inicio de sesión.	El tiempo transcurrido se calcula correctamente.	Se muestra en la prueba el tiempo transcurrido	Pass
7	Comparar el tiempo transcurrido con el umbral de tiempo aceptable (5 segundos).	El tiempo transcurrido es igual o menor al umbral aceptable.	El tiempo transcurrido fue de 3 segundos	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.4.4. Casos de prueba Sprint 4

Test Case ID	P17	Test Case Description	Verificar que la tabla de historial en la actividad HistoryActivity se haya cargado correctamente y que contenga las columnas esperadas	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	cropNameTextView=" Cultivo"		
2	El usuario ha iniciado sesión en la aplicación.	2	pestNameTextView=" Plaga/Enfermedad"		
		3	locationTextView= "Ubicación"		
		4	timeDetectedTextView= "Hora"		
		5	dateDetectedTextView= "Fecha"		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar la actividad HistoryActivity utilizando la regla ActivityTestRule.	La actividad HistoryActivity se inicia con éxito.	Se obtuvo el resultado esperado	Pass
2	Esperar a que la actividad se cargue completamente.	La actividad se carga completamente sin errores.	Se obtuvo el resultado esperado	Pass
3	Utilizar un ViewAssertion para verificar que la vista de la tabla (tableLayout) se haya cargado.	La tabla de historial se carga satisfactoriamente.	Se obtuvo el resultado esperado	Pass
4	Verificar que cropNameTextView contiene el texto "Cultivo."	La vista de texto "Cultivo" se encuentra en cropNameTextView.	Se obtuvo el resultado esperado	Pass
5	Verificar que pestNameTextView contiene el texto "Plaga/Enfermedad."	La vista de texto "Plaga/Enfermedad" se encuentra en pestNameTextView.	Se obtuvo el resultado esperado	Pass
6	Verificar que locationTextView contiene el texto "Ubicación."	La vista de texto "Ubicación" se encuentra en locationTextView.	Se obtuvo el resultado esperado	Pass
7	Verificar que timeDetectedTextView contiene el texto "Hora."	La vista de texto "Hora" se encuentra en timeDetectedTextView.	Se obtuvo el resultado esperado	Pass
8	Verificar que dateDetectedTextView contiene el texto "Fecha."	La vista de texto "Fecha" se encuentra en dateDetectedTextView.	Se obtuvo el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P18	Test Case Description	Verificar que el WebView en la actividad MAGActivity se carga correctamente y se muestra en la interfaz de usuario.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	webview		
2	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar la actividad MAGActivity utilizando la regla ActivityTestRule.	La actividad MAGActivity se inicia con éxito.	Se muestra la interfaz del mapa de granos básicos	Pass
2	Utilizar Espresso para verificar que el WebView (identificado por R.id.webview) se carga y se muestra en la interfaz de usuario.	El WebView se carga correctamente y se muestra en la interfaz de usuario.	El mapa interactivo de granos básicos carga correctamente	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P19	Test Case Description	Verificar si los elementos de la interfaz de usuario se cargan correctamente en el fragmento ProfileFragment	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	userNameEditText		
2	El usuario ha iniciado sesión	2	professionEditText		
3	Tener conexión a internet	3	cityEditText		
		4	profileImageView		
		5	saveChangesButton		
		6	profileImageView		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Verificar si el campo de nombre de usuario se muestra.	El campo de nombre de usuario (userNameEditText) se muestra en la interfaz de usuario.	Se obtuvo el resultado esperado	Pass
2	Verificar si el campo de profesión se muestra.	El campo de profesión (professionEditText) se muestra en la interfaz de usuario.	Se obtuvo el resultado esperado	Pass
3	Verificar si el campo de ciudad se muestra.	El campo de ciudad (cityEditText) se muestra en la interfaz de usuario.	Se obtuvo el resultado esperado	Pass
4	Verificar si la imagen de perfil se muestra.	La imagen de perfil (profileImageView) se muestra en la interfaz de usuario.	Se obtuvo el resultado esperado	Pass
5	Verificar si el botón de guardar cambios se muestra.	El botón de guardar cambios (saveChangesButton) se muestra en la interfaz de usuario.	Se obtuvo el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P20	Test Case Description	Verificar el proceso de actualización del nombre de usuario en el fragmento ProfileFragment	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	userNameEditText= “NuevoNombre”		
2	El usuario ha iniciado sesión	2	saveChangesButton		
3	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Ingresa un nuevo nombre en el campo de nombre de usuario.	El nuevo nombre se ingresa en el campo de nombre de usuario.	Se obtuvo el resultado esperado	Pass
2	Hacer clic en el botón de guardar cambios.	La aplicación procesa la solicitud de guardar cambios.	Se obtuvo el resultado esperado	Pass
3	Esperar un momento para que aparezca el "Toast".	Se espera que aparezca un "Toast" con el mensaje "Nombre de usuario actualizado con éxito."	Se obtuvo el resultado esperado	Pass
4	Verificar si aparece un "Toast" con el mensaje específico.	El "Toast" con el mensaje "Nombre de usuario actualizado con éxito." se muestra correctamente.	Se obtuvo el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.4.5. Casos de prueba Sprint 5

Test Case ID	P21	Test Case Description	Verificar que en la aplicación web se muestran las calificaciones de la Aplicación realizada por los usuarios, incluyendo si la calificaron como "útil" o "no útil" mediante un gráfico de pastel.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	chartData		
2	El usuario ha iniciado sesión con un tipo de cuenta institución.				
3	Tener conexión a internet				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Verificar que se muestra un gráfico de pastel con la calificación Útil y No útil divididas	Se muestra un gráfico de pastel con 2 colores divididos, uno para la calificación de Útil y otro para No Útil.	Se obtuvo el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P22	Test Case Description	Verificar el comportamiento del fragmento de diálogo personalizado (CustomDialogFragment) en la aplicación por medio del cual el usuario evalúa con una calificación.	
Test type	Caja negra	Testing Method	Comportamiento de interfaz de usuario	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Cargar la aplicación	1	openDialogButton		
2	El usuario ha iniciado sesión	2	btnNoUtil		
3	Tener conexión a internet	3	custom_dialog_layout		
		4	btnUtil		

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Hacer clic en el botón que abre el diálogo.	El diálogo personalizado se abre en respuesta al clic en el botón.	Se obtuvo el resultado esperado	Pass
2	Realizar una prueba de "No Útil" haciendo clic en el botón correspondiente.	El diálogo registra una acción de "No Útil." El diálogo se cierra automáticamente.	Se obtuvo el resultado esperado	Pass
3	Verificar que el diálogo se haya cerrado después de hacer clic en "No Útil."	El diálogo no se muestra en la interfaz de usuario, ya que se ha cerrado.	Se obtuvo el resultado esperado	Pass
4	Realizar una prueba de "Útil" haciendo clic en el botón que abre el diálogo nuevamente.	El diálogo se abre nuevamente en respuesta al clic en el botón.	Se obtuvo el resultado esperado	Pass
5	Hacer clic en el botón "Útil."	El diálogo registra una acción de "Útil." El diálogo se cierra automáticamente.	Se obtuvo el resultado esperado	Pass
6	Verificar que el diálogo se haya cerrado después de hacer clic en "Útil."	El diálogo registra una acción de "Útil." El diálogo se cierra automáticamente.	Se obtuvo el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Test Case ID	P23	Test Case Description	Verificar el comportamiento del fragmento de diálogo personalizado (CustomDialogFragment) en la aplicación por medio del cual el usuario evalúa con una calificación.	
Test type	Caja blanca	Testing Method	Cobertura de sentencia	
Created By	Jeyson Guevara	Reviewed By	Version	1.2

QA Tester's Log

Tester's Name	Jeyson Guevara	Date Tested	29/10/2023	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:	S #	Test Data		
1	Dependencias de Robolectric configuradas	1	splashActivity		
2	Actividades SplashActivity y ManualActivity en el proyecto				
3	Actividades definidas en AndroidManifest.xml				
4	Versión de Android configurada en @Config				
5	Aserciones reflejando comportamiento esperado				
6	Tiempo en milisegundos adecuado				

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Iniciar la prueba de SplashActivity.	La actividad se inicia con éxito.	Se obtuvo el resultado esperado	Pass
2	Verificar que la actividad SplashActivity no sea nula.	La actividad actual es SplashActivity.	Se obtuvo el resultado esperado	Pass
3	Simular la ejecución del Handler con un retraso de 3 segundos.	La actividad ManualActivity se inicia después del retraso de 3 segundos.	Se obtuvo el resultado esperado	Pass
4	Verificar que la actividad ManualActivity se inicie después del retraso	La actividad actual es ManualActivity.	Se obtuvo el resultado esperado	Pass

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.5.EJECUCIÓN Y RESULTADOS DE PRUEBAS POR SPRINT

En las siguientes imágenes se muestran los scripts para realizar las pruebas de la aplicación móvil y web, así como los resultados.

2.5.1. Sprint 1

A continuación, se muestran los scripts correspondientes a las pruebas funcionales realizadas sobre el módulo DetectionFragment de la aplicación móvil según el **caso de prueba P01**.

Figura 1

Script testImageAnalysis

```
package com.example.foro;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.assertion.ViewAssertions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.rule.ActivityTestRule;

import com.example.foro.Activities.MainActivity;

import org.junit.Rule;
import org.junit.Test;

public class DetectionFragmentTest {

    @Rule
    public ActivityTestRule<MainActivity> activityRule = new ActivityTestRule<>(MainActivity.class);

    @Test
    public void testImageAnalysis() {
        // Simular hacer clic en el botón de la galería para seleccionar una imagen
        Espresso.onView(ViewMatchers.withId(R.id.button)).perform(ViewActions.click());

        // Esperar a que se abra la galería (puede variar según la galería del dispositivo)

        // Simular seleccionar una imagen de la galería (ajusta el identificador de la vista)
        Espresso.onView(ViewMatchers.withId(R.id.imageView)).perform(ViewActions.click());

        // Esperar a que se complete el análisis de imagen (puedes ajustar esto según tu código)

        // Verificar el resultado en el fragmento
        Espresso.onView(ViewMatchers.withId(R.id.result))
            .check(ViewAssertions.matches(ViewMatchers.isDisplayed())); // Asegurarse de que el resultado se muestre
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 2 corresponde a las pruebas funcionales realizadas sobre el módulo DetectionFragment de la aplicación móvil según el **caso de prueba P02**.

Figura 2

Script testClearResults

```
@Test
public void testClearResults() {
    // Simular hacer clic en el botón de la galería para seleccionar una imagen
    Espresso.onView(ViewMatchers.withId(R.id.button)).perform(ViewActions.click());

    // Esperar a que se abra la galería (puede variar según la galería del dispositivo)

    // Simular seleccionar una imagen de la galería (ajusta el identificador de la vista)
    Espresso.onView(ViewMatchers.withId(R.id.imageView)).perform(ViewActions.click());

    // Esperar a que se complete el análisis de imagen (puedes ajustar esto según tu código)

    // Verificar el resultado en el fragmento
    Espresso.onView(ViewMatchers.withId(R.id.result))
        .check(ViewAssertions.matches(ViewMatchers.isDisplayed())); // Asegurarse de que el resultado se muestre
    // Verificar que el resultado sea claro (por ejemplo, que no esté vacío)
    Espresso.onView(ViewMatchers.withId(R.id.result))
        .check(ViewAssertions.matches(ViewMatchers.withText(org.hamcrest.Matchers.not(org.hamcrest.Matchers.isEmptyString()))));
}
```

La figura 3 corresponde a las pruebas funcionales realizadas sobre el módulo DetectionFragment de la aplicación móvil según el **caso de prueba P03**.

Figura 3

Script testSpecificIdentification

```
@Test
public void testSpecificIdentification() {
    // Simular hacer clic en el botón de la galería para seleccionar una imagen
    Espresso.onView(ViewMatchers.withId(R.id.button)).perform(ViewActions.click());

    // Esperar a que se abra la galería (puede variar según la galería del dispositivo)

    // Simular seleccionar una imagen de la galería (ajusta el identificador de la vista)
    Espresso.onView(ViewMatchers.withId(R.id.imageView)).perform(ViewActions.click());

    // Esperar a que se complete el análisis de imagen (puedes ajustar esto según tu código)

    // Verificar el resultado en el fragmento
    Espresso.onView(ViewMatchers.withId(R.id.result))
        .check(ViewAssertions.matches(ViewMatchers.isDisplayed())); // Asegurarse de que el resultado se muestre
    // Verificar que la identificación específica esté presente en los resultados
    Espresso.onView(ViewMatchers.withId(R.id.result))
        .check(ViewAssertions.matches(ViewMatchers.withText("Roya (Puccinia sorghi)")); // Ajusta al resultado esperado
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 4 corresponde a las pruebas funcionales realizadas sobre el módulo DashboardFragment de la aplicación móvil según el **caso de prueba P04 y P05**.

Figura 4

Script testOpenPlagasActivity

```
package com.example.foro;

import android.content.Intent;
import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.intent.Intents;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.ext.junit.runners.AndroidJUnit4;
import androidx.test.rule.ActivityTestRule;
import com.example.foro.Activities.Home;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
public class PlagasActivityTest {

    1 usage
    @Rule
    public ActivityTestRule<Home> activityRule = new ActivityTestRule<>(Home.class);

    @Before
    public void setup() { Intents.init(); }

    @Test
    public void testOpenPlagasActivity() {
        // Realiza un clic en el CardView de un cultivo (por ejemplo, maíz)
        Espresso.onView(ViewMatchers.withId(R.id.cardViewMaiz)).perform(ViewActions.click());

        // Verifica que la actividad PlagasActivity se inicie con el cultivo correcto (maíz)
        Intent intent = new Intent();
        intent.putExtra("cultivo", "maiz");
        activityRule.launchActivity(intent);

        // Limpia las intenciones después de la prueba
        Intents.release();
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 5 corresponde a las pruebas funcionales realizadas sobre el módulo Monitoreo de la aplicación web según el **caso de prueba P06**.

Figura 5

Script DataTable

```
// Inicializa la tabla DataTable y la hace responsiva
const table = $("#example").DataTable({
  language: { ...
},

  pageLength: 5,
  lengthMenu: [ ...
],
  responsive: true,
  dom: "lBfrtip",
  buttons: [ ...
],
});

// Escuchar el evento "draw.dt" para capturar cuando se dibuja la tabla
table.on("draw.dt", function () {
  // Obtener los datos del DataTable y mostrarlos en la consola
  console.log(table.data().toArray());
});
});
```

En la figura 6 se muestra el resultado de la ejecución de pruebas donde se puede observar que hubo una cobertura de código del 100% en cuanto a los componentes probados.

Figura 6

Resultados de las pruebas Sprint 1

Status	3 passed	3 tests, 2 m 13 s 626 ms
Filter tests:		
Tests	Duration	LGE LM-X430
✓ Test Results	1 m 45 s	3/3
✓ DetectionFragmentTest	1 m 45 s	3/3
✓ testImageAnalysis	36 s	✓
✓ testSpecificIdentification	34 s	✓
✓ testClearResults	34 s	✓
Status	1 passed	1 test, 43 s 577 ms
Filter tests:		
Tests	Duration	LGE LM-X430
✓ Test Results	1 s	1/1
✓ PlagasActivityTest	1 s	1/1
✓ testOpenPlagasActivity	1 s	✓
<pre> (6) [Array(6), Array(6), Array(6), Array(6), Array(6), Array(6)] 0: (6) ['Gusano cogollero', 'Maíz', 'Chontales', '+50584903928', '2023-10-21', '09:11'] 1: (6) ['Mosaico dorado', 'Frijol', 'Chontales', '+50584903928', '2023-10-21', '09:11'] 2: (6) ['Roya', 'Maíz', 'Managua', '+50584903926', '2023-10-21', '09:24'] 3: (6) ['Mancha zonada', 'Sorgo', 'Managua', '89636254', '2023-10-21', '09:24'] 4: (6) ['Chicharrita verde', 'Frijol', 'Jinotega', '85236954', '2023-10-21', '10:11'] 5: (6) ['Mancha zonada', 'Sorgo', 'Managua', '96524821', '2023-10-21', '10:11'] length: 6 [[Prototype]]: Array(0) </pre>		

Nota. Figura tomada de Android Studio y consola de Google Chrome.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.5.2. Sprint 2

La figura 7 corresponde a las pruebas funcionales realizadas sobre el módulo DashboardFragment de la aplicación móvil según el **caso de prueba P07**.

Figura 7

Script verifyEnteredLocationDisplayed

```
package com.example.foro;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.assertion.ViewAssertions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.ext.junit.rules.ActivityScenarioRule;
import androidx.test.ext.junit.runners.AndroidJUnit4;
import com.example.foro.Activities.Home;

import org.hamcrest.Matchers;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
public class LocationSearchTest {

    // El siguiente Rule se utiliza para iniciar la actividad Home en la que se encuentra el fragmento DashboardFragment.
    @Rule
    public ActivityScenarioRule<Home> activityRule = new ActivityScenarioRule<>(Home.class);

    // Realiza un clic en el icono de búsqueda e ingresa una ubicación en el campo de búsqueda.
    @Before
    public void clickSearchIconAndEnterLocation() {
        Espresso.onView(ViewMatchers.withId(R.id.searchIconImageView)).perform(ViewActions.click());
        Espresso.onView(ViewMatchers.withId(R.id.cityEditText)).perform(ViewActions.typeText(stringToBeTyped: "Juigalpa"));
        Espresso.onView(ViewMatchers.withId(R.id.cityEditText)).perform(ViewActions.closeSoftKeyboard());
    }

    // Realiza una prueba para asegurarse de que la ubicación ingresada se refleje en el campo de ubicación.
    @Test
    public void verifyEnteredLocationIsDisplayed() {
        Espresso.onView(ViewMatchers.withId(R.id.locationTextView))
            .check(ViewAssertions.matches(ViewMatchers.withText("Juigalpa")));
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Las figuras 8 y 9 corresponden a las pruebas funcionales realizadas sobre el módulo DashboardFragment de la aplicación móvil según el **caso de prueba P08 y P09**.

Figura 8

Script verifyLocationSearch

```
// Realiza una prueba para asegurarse de que la búsqueda de ubicación se realiza correctamente.
@Test
public void verifyLocationSearch() {
    // Suponiendo que el campo de ubicación muestra "NombreDeLaCiudad" después de la búsqueda
    // Puedes verificar que la ubicación sea igual a lo que ingresaste.
    Espresso.onView(ViewMatchers.withId(R.id.locationTextView))
        .check(ViewAssertions.matches(ViewMatchers.withText("Juigalpa")));

    // Verificar que el TextView contenga una cadena específica
    Espresso.onView(ViewMatchers.withId(R.id.temperatureTextView))
        .check(ViewAssertions.matches(ViewMatchers.withText(Matchers.containsString( substring: "29°C"))));

    Espresso.onView(ViewMatchers.withId(R.id.humidityTextView))
        .check(ViewAssertions.matches(ViewMatchers.withText(Matchers.containsString( substring: "74%"))));

    Espresso.onView(ViewMatchers.withId(R.id.windTextView))
        .check(ViewAssertions.matches(ViewMatchers.withText(Matchers.containsString( substring: "7 km/h"))));

    Espresso.onView(ViewMatchers.withId(R.id.cloudinessTextView))
        .check(ViewAssertions.matches(ViewMatchers.withText(Matchers.containsString( substring: "40%"))));
    // También puedes verificar el icono del clima si se muestra en la interfaz de usuario.
    Espresso.onView(ViewMatchers.withId(R.id.weatherIconImageView)).check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
}
```

Figura 9

Script verifyDeviceLocationIsDisplayed

```
// Realiza una prueba para asegurarse de que la ubicación del dispositivo se refleje en el campo de ubicación.
@Test
public void verifyDeviceLocationIsDisplayed() {
    // Suponiendo que la ubicación del dispositivo se muestra en el campo de ubicación.
    // Puedes verificar que la ubicación sea igual a la ubicación real del dispositivo.
    // Por ejemplo, si la ubicación del dispositivo es "Mi Ubicación Actual", verifica eso.

    Espresso.onView(ViewMatchers.withId(R.id.locationTextView))
        .check(ViewAssertions.matches(ViewMatchers.withText("Juigalpa")));
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 10 corresponde a las pruebas funcionales realizadas sobre el módulo MonitoringFragment de la aplicación móvil según el **caso de prueba P10**.

Figura 10

Script testMonitoringFormSubmission

```
package com.example.foro;

import android.content.Context;

import androidx.test.core.app.ApplicationProvider;
import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.rule.ActivityTestRule;
import androidx.test.ext.junit.runners.AndroidJUnit4;

import com.example.foro.Activities.MainActivity;

import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
public class MonitoringFragmentTest {

    @Rule
    public ActivityTestRule<MainActivity> activityRule = new ActivityTestRule<>(MainActivity.class);

    1 usage
    private Context context;

    @Before
    public void setup() { context = ApplicationProvider.getApplicationContext(); }

    @Test
    public void testMonitoringFormSubmission() {
        // Rellenar los campos del formulario
        Espresso.onView(ViewMatchers.withId(R.id.cropNameEditText)).perform(ViewActions.typeText(stringToBeTyped: "Nombre del Cultivo"));
        Espresso.onView(ViewMatchers.withId(R.id.pestNameEditText)).perform(ViewActions.typeText(stringToBeTyped: "Nombre de la Plaga"));
        Espresso.onView(ViewMatchers.withId(R.id.contactoEditText)).perform(ViewActions.replaceText(stringToBeSet: "Número de Contacto"));
        Espresso.onView(ViewMatchers.withId(R.id.cityAutoCompleteTextView)).perform(ViewActions.typeText(stringToBeTyped: "Nombre de la Ciudad"));

        // Hacer clic en el botón de envío
        Espresso.onView(ViewMatchers.withId(R.id.submitButton)).perform(ViewActions.click());
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Las figuras 11 y 12 corresponden a las pruebas funcionales realizadas sobre el módulo HomeFragment de la aplicación móvil según el **caso de prueba P11**.

Figura 11

Script testFabButtonVisibility

```
package com.example.foro;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.assertion.ViewAssertions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.rule.ActivityTestRule;

import com.example.foro.Activities.MainActivity;
import org.junit.Rule;
import org.junit.Test;

public class HomeFragmentTest {

    @Rule
    public ActivityTestRule<MainActivity> activityTestRule = new ActivityTestRule<>(MainActivity.class);

    @Test
    public void testFabButtonVisibility() {
        // Verificar que el FAB esté visible
        Espresso.onView(ViewMatchers.withId(R.id.fab))
            .check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
    }
}
```

Figura 12

Script testLoginMessageVisibility

```
@Test
public void testLoginMessageVisibility() {
    // Verificar que el mensaje de inicio de sesión esté visible
    Espresso.onView(ViewMatchers.withId(R.id.login_message_text))
        .check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
}
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 13 corresponde a las pruebas funcionales realizadas sobre el módulo Monitoreo de la aplicación web según el **caso de prueba P12**.

Figura 13

Script addMarker

```
// Función para agregar un marcador
function addMarker() {
    var name = document.getElementById("markerName").value;
    var category = document.getElementById("markerCategory").value;
    var crop = document.getElementById("markerCrop").value;

    if (name && category && crop) {
        alert(
            "Haz clic en el mapa para agregar el marcador en la ubicación deseada."
        );

        var markerClickHandler = function (e) {
            var marker = L.marker([e.latlng.lat, e.latlng.lng]).addTo(map);

            // Ahora, dentro de la función .then(), definimos docRef
            db.collection("markers")
                .add({
                    name: name,
                    category: category,
                    crop: crop,
                    latitude: e.latlng.lat,
                    longitude: e.latlng.lng,
                })
                .then(function (docRef) {
                    // Definimos docRef dentro de esta función
                    console.log("Marcador guardado con ID: ", docRef.id + " con el nombre de: ", name);

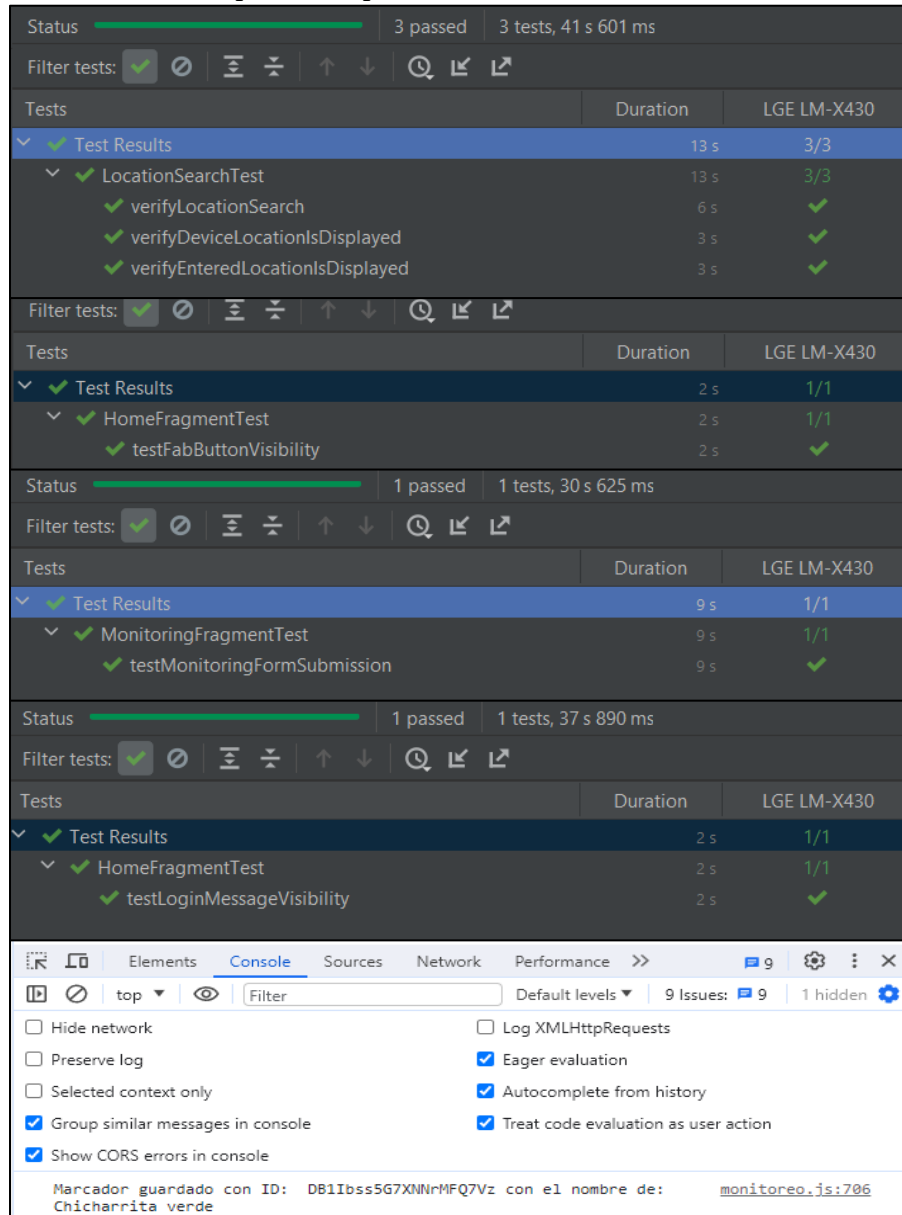
                    // Almacenamos el marcador en un objeto con el ID de Firebase como clave
                    marker.category = category;
                    markers[docRef.id] = marker; // Agregamos el nuevo marcador al array

                    // Agregar un console.assert para verificar que el nuevo marcador se agregó correctamente
                    console.assert(markers[docRef.id] === marker, "Error: Nuevo marcador no se agregó correctamente.");
                });
        };
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

En la figura 14 imagen se muestra el resultado de la ejecución de pruebas del sprint 2 donde se puede observar que hubo una cobertura de código del 100% en cuanto a los componentes probados.

Figura 14
Resultados de las pruebas Sprint 2



Nota. Figura tomada de Android Studio y consola de Google Chrome.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.5.3. Sprint 3

La figura 15 corresponde a las pruebas funcionales realizadas sobre el módulo Monitoreo de la aplicación web según el **caso de prueba P13**.

Figura 15

Script loadMarkers

```
// Función para cargar los marcadores desde Firestore
function loadMarkers() {
  db.collection("markers")
    .get()
    .then(function (querySnapshot) {
      querySnapshot.forEach(function (doc) {
        var markerData = doc.data();
        var marker = L.marker([
          markerData.latitude,
          markerData.longitude,
        ]).addTo(map);
        marker.bindPopup(
          "Nombre: " +
            markerData.name +
            "<br>Categoría: " +
            markerData.category +
            "<br>Cultivo: " +
            markerData.crop +
            "<br><button onclick=\"deleteMarker('" +
              doc.id +
              "\">Eliminar</button>"
        );
        marker.category = markerData.category;
        markers[doc.id] = marker; // Almacena el marcador en un objeto con el ID como clave
      });
      // Mostrar el nombre del marcador en la consola
      console.log("Nombre del marcador: " + markerData.name);
    });
  filterMarkers();
})
.catch(function (error) {
  console.error("Error al cargar marcadores desde Firestore: ", error);
});
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 16 corresponde a las pruebas funcionales realizadas sobre el módulo Monitoreo de la aplicación web según el **caso de prueba P14**.

Figura 16

Script filterMarkers

```
document
  .getElementById("filterEnfermedad")
  .addEventListener("change", function () {
    filterMarkers();
  });

function filterMarkers() {
  var showPlaga = document.getElementById("filterPlaga").checked;
  var showEnfermedad = document.getElementById("filterEnfermedad").checked;

  console.log("showPlaga:", showPlaga);
  console.log("showEnfermedad:", showEnfermedad);

  for (var i in markers) {
    var marker = markers[i];
    var category = marker.category;

    console.log("Marker category:", category);

    // Verifica si el marcador debe mostrarse o ocultarse según las categorías seleccionadas
    if (
      (showPlaga && category === "Plaga") ||
      (showEnfermedad && category === "Enfermedad")
    ) {
      map.addLayer(marker); // Muestra el marcador
    } else {
      map.removeLayer(marker); // Oculta el marcador
    }
  }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 17 corresponde a las pruebas funcionales realizadas sobre el módulo LoginActivity de la aplicación móvil según el caso de prueba P16.

Figura 17

Script testLoginPerformance

```
package com.example.foro;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.ext.junit.runners.AndroidJUnit4;
import androidx.test.ext.junit.rules.ActivityScenarioRule;

import com.example.foro.Activities.LoginActivity;

import org.junit.Assert;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
public class LoginActivityPerformanceTest {

    @Rule
    public ActivityScenarioRule<LoginActivity> activityRule =
        new ActivityScenarioRule<>(LoginActivity.class);

    @Test
    public void testLoginPerformance() {
        // Iniciar la medición del tiempo
        long startTime = System.currentTimeMillis();

        // Usar la acción replaceText para establecer el texto directamente en el campo de contraseña
        Espresso.onView(ViewMatchers.withId(R.id.login_mail)).perform(ViewActions.typeText(stringToBeTyped: "correo@gmail.com"));
        Espresso.onView(ViewMatchers.withId(R.id.login_password)).perform(ViewActions.replaceText(stringToBeSet: "correito"));
        Espresso.onView(ViewMatchers.withId(R.id.loginBtn)).perform(ViewActions.click());

        // Detener la medición del tiempo
        long endTime = System.currentTimeMillis();

        // Calcular el tiempo transcurrido
        long elapsedTime = endTime - startTime;

        // Puedes establecer un umbral de tiempo aceptable y afirmar que se cumpla
        long umbralAceptable = 5000; // 5 segundos
        Assert.assertTrue(message: "Tiempo de inicio de sesión fuera de lo aceptable", condition: elapsedTime <= umbralAceptable);
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 18 y figura 19 corresponden a las pruebas funcionales realizadas sobre el módulo RegisterActivity de la aplicación móvil según el caso de prueba P15.

Figura 18

Script testRegistrationSuccess

```
package com.example.foro;

import androidx.test.ext.junit.rules.ActivityScenarioRule;
import androidx.test.ext.junit.runners.AndroidJUnit4;

import com.example.foro.Activities.RegisterActivity;

import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.matcher.ViewMatchers;

@RunWith(AndroidJUnit4.class)
public class RegisterActivityTest {

    // Regla para iniciar la actividad bajo prueba
    @Rule
    public ActivityScenarioRule<RegisterActivity> activityRule =
        new ActivityScenarioRule<>(RegisterActivity.class);

    @Test
    public void testRegistrationSuccess() {
        // Simula la entrada de datos en los campos de registro y realiza un clic en el botón de registro
        Espresso.onView(ViewMatchers.withId(R.id.regName)).perform(ViewActions.typeText( stringToBeTyped: "TestUser"));
        Espresso.onView(ViewMatchers.withId(R.id.regMail)).perform(ViewActions.typeText( stringToBeTyped: "test@example.com"));
        Espresso.onView(ViewMatchers.withId(R.id.regPassword)).perform(ViewActions.typeText( stringToBeTyped: "password"));
        Espresso.onView(ViewMatchers.withId(R.id.regPassword2)).perform(ViewActions.typeText( stringToBeTyped: "password"));
        Espresso.onView(ViewMatchers.withId(R.id.regBtn)).perform(ViewActions.click());
    }
}
```

Figura 19

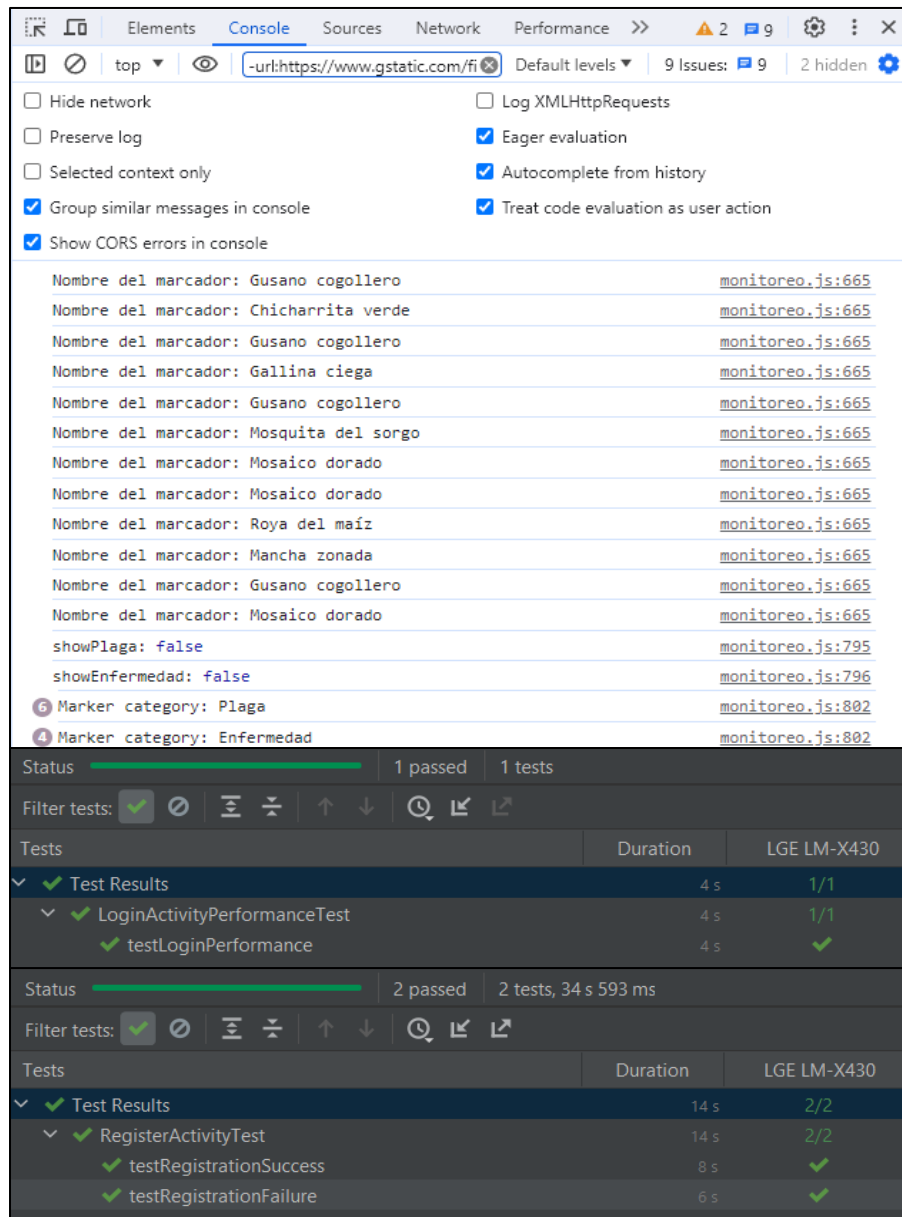
Script testRegistrationFailure

```
@Test
public void testRegistrationFailure() {
    // Simula la entrada de datos incorrectos y realiza un clic en el botón de registro
    Espresso.onView(ViewMatchers.withId(R.id.regMail)).perform(ViewActions.typeText( stringToBeTyped: "invalid-email"));
    Espresso.onView(ViewMatchers.withId(R.id.regName)).perform(ViewActions.typeText( stringToBeTyped: ""));
    Espresso.onView(ViewMatchers.withId(R.id.regPassword)).perform(ViewActions.typeText( stringToBeTyped: "password"));
    Espresso.onView(ViewMatchers.withId(R.id.regPassword2)).perform(ViewActions.typeText( stringToBeTyped: "different-password"));
    Espresso.onView(ViewMatchers.withId(R.id.regBtn)).perform(ViewActions.click());
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

En la figura 20 se muestra el resultado de la ejecución de pruebas del sprint 3 donde se puede observar que hubo una cobertura de código del 100% en cuanto a los componentes probados.

Figura 20
Resultados de pruebas Sprint 3



Nota. Figura tomada de Android Studio y consola de Google Chrome.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.5.4. Sprint 4

La figura 21 corresponde a las pruebas funcionales realizadas sobre el módulo HistoryActivity de la aplicación móvil según el **caso de prueba P17**.

Figura 21

Script testHistoryTableContents

```
package com.example.foro;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.NoMatchingViewException;
import androidx.test.espresso.ViewAssertion;
import androidx.test.rule.ActivityTestRule;

import org.junit.Rule;
import org.junit.Test;

import static androidx.test.espresso.matcher.ViewMatchers.withId;
import static androidx.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.CoreMatchers.allOf;

import android.view.View;

import com.example.foro.Activities.HistoryActivity;

public class HistoryActivityTest {

    @Rule
    public ActivityTestRule<HistoryActivity> activityTestRule =
        new ActivityTestRule<>(HistoryActivity.class);

    @Test
    public void testHistoryTableContents() {
        // Espera a que la actividad se cargue completamente
        Espresso.onView(withId(R.id.tableLayout)).check(new ViewAssertion() {
            @Override
            public void check(View view, NoMatchingViewException noViewException) {
                // Verifica que la tabla se haya cargado
                // A continuación, verifica las vistas de texto en la tabla
                Espresso.onView(allOf(withId(R.id.cropNameTextView), withText("Cultivo")));
                Espresso.onView(allOf(withId(R.id.pestNameTextView), withText("Plaga/Enfermedad")));
                Espresso.onView(allOf(withId(R.id.locationTextView), withText("Ubicación")));
                Espresso.onView(allOf(withId(R.id.timeDetectedTextView), withText("Hora")));
                Espresso.onView(allOf(withId(R.id.dateDetectedTextView), withText("Fecha")));
            }
        });
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 22 corresponde a las pruebas funcionales realizadas sobre el módulo MAGActivity de la aplicación móvil según el **caso de prueba P18**.

Figura 22

Script testMapLoading

```
package com.example.foro;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.assertion.ViewAssertions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.ext.junit.runners.AndroidJUnit4;
import androidx.test.rule.ActivityTestRule;
import com.example.foro.Activities.MAGActivity;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
public class MAGActivityTest {

    @Rule
    public ActivityTestRule<MAGActivity> mActivityRule = new ActivityTestRule<>(MAGActivity.class);

    @Test
    public void testMapLoading() {
        // Verifica que el WebView se carga correctamente
        Espresso.onView(ViewMatchers.withId(R.id.webview))
            .check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 23 corresponde a las pruebas funcionales realizadas sobre el módulo ProfileFragment de la aplicación móvil según el **caso de prueba P19**.

Figura 23

Script testProfileFragmentElements

```
package com.example.foro;

import android.os.SystemClock;
import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.assertion.ViewAssertions;
import androidx.test.espresso.intent.rule.IntentsTestRule;
import androidx.test.espresso.matcher.RootMatchers;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.ext.junit.runners.AndroidJUnit4;

import com.example.foro.Activities.MainActivity;

import org.hamcrest.Matchers;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
public class ProfileFragmentTest {

    1 usage
    @Rule
    public IntentsTestRule<MainActivity> mActivityRule = new IntentsTestRule<>(MainActivity.class);

    @Test
    3 public void testProfileFragmentElements() {
        // Verificar si los elementos de la interfaz de usuario se cargan correctamente en el fragmento.
        Espresso.onView(ViewMatchers.withId(R.id.userNameEditText)).check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
        Espresso.onView(ViewMatchers.withId(R.id.professionEditText)).check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
        Espresso.onView(ViewMatchers.withId(R.id.cityEditText)).check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
        Espresso.onView(ViewMatchers.withId(R.id.profileImageView)).check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
        Espresso.onView(ViewMatchers.withId(R.id.saveChangesButton)).check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
        Espresso.onView(ViewMatchers.withId(R.id.profileImageView)).check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
    }
}
```

La figura 24 corresponde a las pruebas funcionales realizadas sobre el módulo ProfileFragment de la aplicación móvil según el **caso de prueba P20**.

Figura 24

Script testProfileFragmentSaveChanges

```
@Test
public void testProfileFragmentSaveChanges() {
    // Ingresar un nuevo nombre en el campo de nombre.
    Espresso.onView(ViewMatchers.withId(R.id.userNameEditText))
        .perform(ViewActions.replaceText(stringToBeSet: "NuevoNombre"), ViewActions.closeSoftKeyboard());

    // Hacer clic en el botón de guardar cambios.
    Espresso.onView(ViewMatchers.withId(R.id.saveChangesButton)).perform(ViewActions.click());

    // Espera un momento para que aparezca el Toast.
    SystemClock.sleep(ms: 2000); // Ajusta el tiempo según sea necesario.

    // Verifica si aparece un Toast con el mensaje específico.
    Espresso.onView(ViewMatchers.withText("Nombre de usuario actualizado con éxito.")).
        inRoot(RootMatchers.withDecorView(
            Matchers.not(Matchers.is(mActivityRule.getActivity().getWindow().getDecorView()))
        ))
        .check(ViewAssertions.matches(ViewMatchers.isDisplayed()));
}
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

En la figura 25 se muestra el resultado de la ejecución de pruebas del sprint 4 donde se puede observar que hubo una cobertura de código del 100% en cuanto a los componentes probados.

Figura 25

Resultados de pruebas Sprint 4

Status	1 passed	1 tests, 23 s 832 ms
Filter tests:		
Tests	Duration	LGE LM-X430
✓ Test Results	1 s	1/1
✓ HistoryActivityTest	1 s	1/1
✓ testHistoryTableContents	1 s	✓
Status	1 passed	1 tests, 36 s 777 ms
Filter tests:		
Tests	Duration	LGE LM-X430
✓ Test Results	2 s	1/1
✓ MAGActivityTest	2 s	1/1
✓ testMapLoading	2 s	✓
Status	2 passed	2 tests, 59 s 57 ms
Filter tests:		
Tests	Duration	LGE LM-X430
✓ Test Results	25 s	2/2
✓ ProfileFragmentTest	25 s	2/2
✓ testProfileFragmentElements	2 s	✓
✓ testProfileFragmentSaveChanges	22 s	✓

Nota. Figura tomada de Android Studio.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

2.5.5. Sprint 5

La figura 26 corresponde a las pruebas funcionales realizadas sobre el módulo Admin de la aplicación web según el **caso de prueba P21**.

Figura 26

Script chartData

```
// Obtén los datos de experiencia (Útil y No útil) desde tu tabla de Firebase
const data = snapshot.val();

// Cuenta la cantidad de registros por experiencia
let utilCount = 0;
let noUtilCount = 0;

for (const key in data) {
  const experiencia = data[key].experiencia;
  if (experiencia === "Útil") {
    utilCount++;
  } else if (experiencia === "No Útil") {
    noUtilCount++;
  }
}

// Muestra los valores en la consola
console.log("Util Count:", utilCount);
console.log("No Util Count:", noUtilCount);

// Obtén el elemento canvas y crea el gráfico de pastel
const canvas = document.getElementById("experienciaChart");
const ctx = canvas.getContext("2d");

const chartData = {
  labels: ["Útil", "No Útil"],
  datasets: [
    {
      data: [utilCount, noUtilCount],
      backgroundColor: ["#287233", "#4B3621"],
    },
  ],
};

new Chart(ctx, {
  type: "pie",
  data: chartData,
  options: {
    responsive: true,
    title: {
      display: true,
      text: "Distribución de Experiencia",
      fontSize: 16,
    },
  },
});
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 27 corresponde a las pruebas funcionales realizadas sobre el módulo ProfileFragment de la aplicación móvil según el caso de prueba P22.

Figura 27

Script testCustomDialogFragment

```
package com.example.foro;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.assertion.ViewAssertions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.ext.junit.rules.ActivityScenarioRule;
import androidx.test.ext.junit.runners.AndroidJUnit4;
import androidx.test.filters.LargeTest;
import com.example.foro.Activities.MainActivity;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
@LargeTest
public class CustomDialogFragmentTest {

    @Rule
    public ActivityScenarioRule<MainActivity> activityRule = new ActivityScenarioRule<>(MainActivity.class); // Reemplaza con el nombre de tu actividad principal.

    @Test
    public void testCustomDialogFragment() {
        // Abre el diálogo
        Espresso.onView(ViewMatchers.withId(R.id.openDialogButton)) // Reemplaza con el ID de tu botón para abrir el diálogo.
            .perform(ViewActions.click());

        // Realiza una prueba de "No Útil"
        Espresso.onView(ViewMatchers.withId(R.id.btnNoÚtil)) // Reemplaza con el ID de tu botón "No Útil".
            .perform(ViewActions.click());

        // Verifica que se haya cerrado el diálogo después de hacer clic en "No Útil"
        Espresso.onView(ViewMatchers.withId(R.layout.custom_dialog_layout)) // Reemplaza con el ID de tu diseño de diálogo.
            .check(ViewAssertions.doesNotExist());

        // Realiza una prueba de "Útil"
        Espresso.onView(ViewMatchers.withId(R.id.openDialogButton)) // Reemplaza con el ID de tu botón para abrir el diálogo.
            .perform(ViewActions.click());
        Espresso.onView(ViewMatchers.withId(R.id.btnÚtil)) // Reemplaza con el ID de tu botón "Útil".
            .perform(ViewActions.click());

        // Verifica que se haya cerrado el diálogo después de hacer clic en "Útil"
        Espresso.onView(ViewMatchers.withId(R.layout.custom_dialog_layout)) // Reemplaza con el ID de tu diseño de diálogo.
            .check(ViewAssertions.doesNotExist());
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

La figura 28 corresponde a las pruebas funcionales realizadas sobre el módulo SplashActivity de la aplicación móvil según el **caso de prueba P22**.

Figura 28

Script testSplashActivity

```
package com.example.foro;

import ...

@RunWith(RobolectricTestRunner.class)
@Config(sdk = 29) // Especifica la versión de Android que deseas utilizar
public class SplashActivityTest {

    4 usages
    private SplashActivity splashActivity;

    @Before
    public void setup() {
        splashActivity = Robolectric.setupActivity(SplashActivity.class);
    }

    @Test
    public void testSplashActivity() {
        assertNotNull(splashActivity);

        // Simular la ejecución de la tarea postDelayed en el Handler
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                // Coloca aquí las aserciones que desees verificar después del retraso.
                // Por ejemplo, puedes verificar si se inicia la actividad esperada.
                Intent startedIntent = shadowOf(splashActivity).getNextStartedActivity();
                Intent expectedIntent = new Intent(splashActivity, ManualActivity.class);
                assertEquals(expectedIntent.getComponent(), startedIntent.getComponent());
            }
        }, delayMillis: 3000); // 3000 ms (3 segundos)

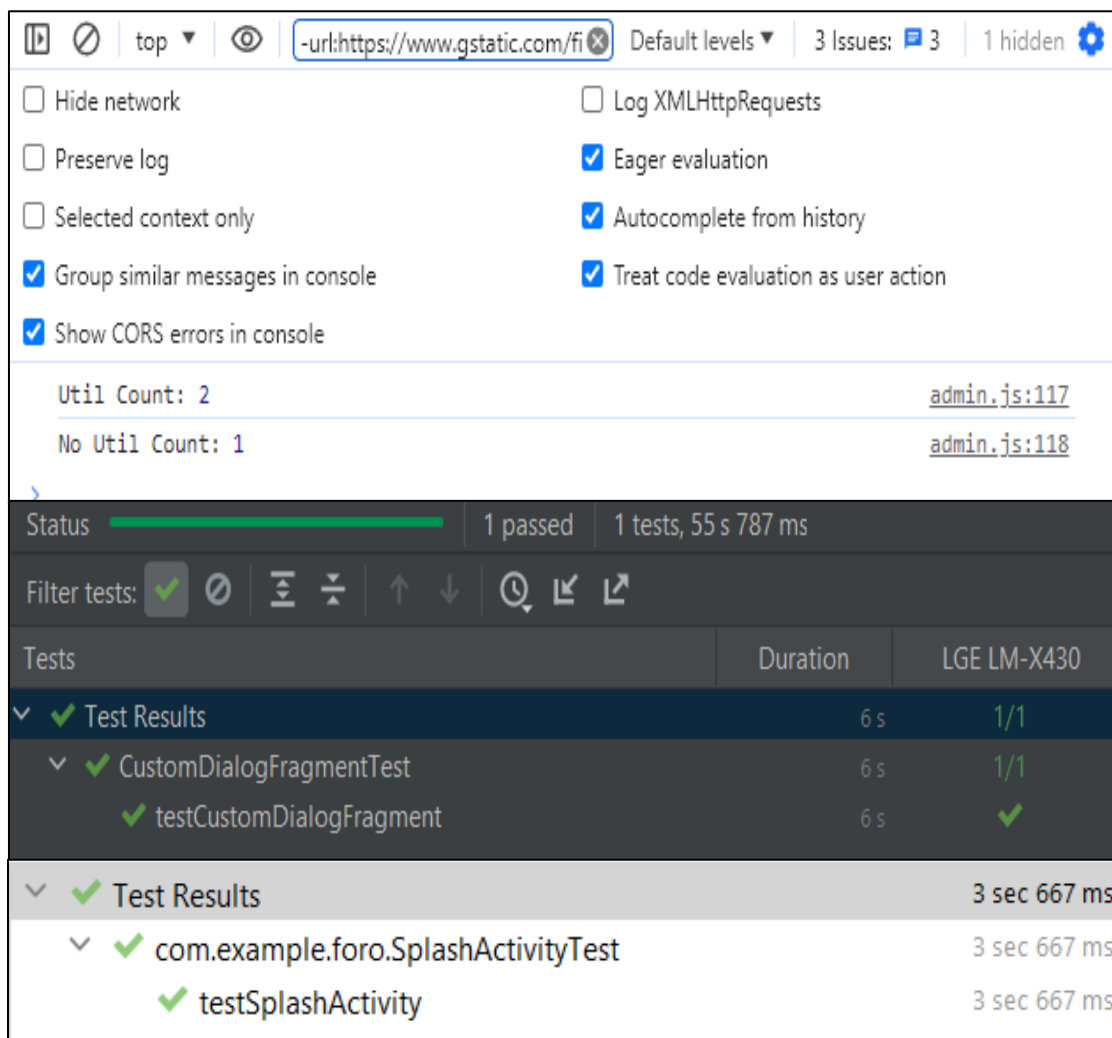
        // Avanzar el bucle de loop para ejecutar las tareas pendientes
        ShadowLooper.runMainLooperToNextTask();
    }
}
```

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

En la figura 29 se muestra el resultado de la ejecución de pruebas del sprint 5 donde se puede observar que hubo una cobertura de código del 100% en cuanto a los componentes probados.

Figura 29

Resultados de pruebas Sprint 5



Nota. Figura tomada de Android Studio y consola de Google Chrome.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

3. CARACTERÍSTICAS A SER PROBADAS

3.1. Características a ser probadas Sprint 1

Tabla 11

Características a ser probadas Sprint 1

CARACTERÍSTICA	DESCRIPCIÓN	MÓDULO	CÓDIGO DE PRUEBA
Análisis de imágenes	La aplicación debe ser capaz de analizar imágenes de los cultivos del usuario.	-Componente DetectionFragment	P01
Resultados claros	La aplicación debe proporcionar resultados claros que identifiquen la presencia de plagas o enfermedades.	-Componente DetectionFragment	P02
Identificación específica	Los resultados deben incluir el nombre específico de la plaga o enfermedad detectada.	-Componente DetectionFragment	P03
Acceso a la lista de plagas/enfermedades	La aplicación debe proporcionar una lista de plagas y enfermedades específicas que afectan a los cultivos de maíz, frijol y sorgo.	-Componente DashboardFragment	P04
Descripciones detalladas	Debe incluir descripciones detalladas de cada plaga o enfermedad, incluyendo su descripción física y los daños que causan a los cultivos.	-Componente DashboardFragment	P05

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Acceso a la tabla de detecciones	La tabla debe incluir información detallada sobre cada plaga o enfermedad, como el nombre, la fecha de detección, la ubicación, el cultivo afectado y el número de contacto del agricultor.	-Componente Monitoreo	P06
----------------------------------	---	-----------------------	-----

3.2.Características a ser probadas Sprint 2

Tabla 12

Características a ser probadas Sprint 2

CARACTERISTICA	DESCRIPCIÓN	MODULO	CÓDIGO DE PRUEBA
Obtener automáticamente la ubicación	La aplicación debe obtener la ubicación actual del agricultor o permitirle buscar una ubicación específica de manera precisa.	-Componente DashboardFragment	P07
Pronóstico a corto plazo	La aplicación proporciona un pronóstico del tiempo a corto plazo (próximas 24 horas) para la ubicación seleccionada.	-Componente DashboardFragment	P08
Detalles del pronóstico	La información del pronóstico incluye detalles sobre la temperatura, la humedad, la velocidad del viento, el nivel de nubosidad y el estado general del día.	-Componente DashboardFragment	P09
Registro de plagas y enfermedades	El formulario de registro de plagas y enfermedades incluye campos para ingresar información completa, como	-Componente MonitoringFragment	P10

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

	nombre del cultivo afectado, nombre de la plaga o enfermedad detectada, fecha y hora de detección, ubicación y número de contacto.		
Creación de publicaciones en el foro	Se muestra un botón flotante en caso de haber iniciado sesión para crear publicaciones. En caso contrario, muestra un mensaje para iniciar sesión.	-Componente HomeFragment	P11
Agregar marcadores en el mapa	Agregar marcadores en el mapa interactivo de plagas y enfermedades junto con el nombre de la plaga o enfermedad detectada.	-Componente Monitoreo	P12

3.3.Características a ser probadas Sprint 3

Tabla 13
Características a ser probadas Sprint 3

CARACTERISTICA	DESCRIPCIÓN	MODULO	CÓDIGO DE PRUEBA
Mostrar zonas de plagas y enfermedades detectadas	El mapa interactivo muestra clara y precisamente las áreas donde se han detectado plagas y enfermedades en los cultivos.	-Componente Monitoreo	P13
Herramientas de búsqueda y filtro	El mapa incluye herramientas de búsqueda y filtro para encontrar información sobre plagas y enfermedades en áreas específicas.	-Componente Monitoreo	P14

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Acceso a la opción de registro	Los usuarios puedan acceder a la opción de registro desde la pantalla de inicio de sesión de la aplicación.	-Componente RegisterActivity	P15
Inicio de sesión después del registro	Los usuarios pueden iniciar sesión en su cuenta después del registro para acceder a las funcionalidades de la aplicación.	-Componente LoginActivity	P16

3.4.Características a ser probadas Sprint 4

Tabla 14
Características a ser probadas Sprint 4

CARACTERISTICA	DESCRIPCIÓN	MODULO	CÓDIGO DE PRUEBA
Ver historial de plagas y enfermedades	Mostrar un historial de plagas y enfermedades en formato de tabla, con detalles organizados.	-Componente HistoryActivity	P17
Mapa interactivo de zonas de cultivos de granos básicos	Mostrar un mapa interactivo que muestre las zonas de cultivos de granos básicos en Nicaragua.	-Componente MAGActivity	P18
Visualizar y actualizar el perfil del agricultor	Permitir a los agricultores ver y actualizar su información de perfil.	-Componente ProfileFragment	P19
Confirmación de cambios en el perfil	Confirmación de que los cambios en el perfil han sido guardados correctamente.	-Componente ProfileFragment	P20

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

3.5.Características a ser probadas Sprint 5

Tabla 15

Características a ser probadas Sprint 5

CARACTERISTICA	DESCRIPCIÓN	MODULO	CÓDIGO DE PRUEBA
Visualización de calificaciones de la aplicación	Muestra las calificaciones de la aplicación proporcionadas por otros usuarios, incluyendo si la calificaron como "útil" o "no útil" mediante un gráfico de pastel.	-Componente Admin	P21
Ventana emergente de calificación	Mostrar una ventana emergente que invite a los agricultores a calificar la aplicación, con opciones para calificarla como "útil" o "no útil".	-Componente ProfileFragment	P22
Splash Screen	La aplicación móvil muestra una pantalla de presentación con su logo al abrirla.	-Componente SplashActivity	P23

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

4. CARACTERÍSTICAS A NO SER PROBADAS

4.1. Características a no ser probadas Sprint 1

Tabla 16

Características a no ser probadas Sprint 1

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Funcionalidad de la cámara/galería del dispositivo	Abre la cámara del dispositivo del usuario o la galería de fotos.	Esto depende del dispositivo del usuario y no de la aplicación.	Si la cámara o la galería de fotos del dispositivo del usuario no funcionan correctamente, esto podría afectar la capacidad del usuario para cargar imágenes en la aplicación.

4.2. Características a no ser probadas Sprint 2

Tabla 17

Características a no ser probadas Sprint 2

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Pronóstico a largo plazo	La aplicación proporciona un pronóstico del tiempo a largo plazo (próximos 5 días) para la ubicación seleccionada.	El pronóstico a corto plazo es más crítico para la toma de decisiones inmediatas.	La falta de esta característica podría afectar la capacidad de planificar actividades agrícolas a largo plazo, pero los agricultores aún pueden basarse en el pronóstico a corto plazo para

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

			tomar decisiones inmediatas.
Precisión en la ubicación del marcador	Agregar un marcador que se coloque de manera precisa en la ubicación deseada en el mapa interactivo.	Si bien la precisión es importante, pequeñas imprecisiones en la ubicación del marcador pueden corregirse en futuros sprints. La funcionalidad básica del marcador sigue siendo funcional.	Si los marcadores no se colocan con precisión en el mapa, podría dificultar el seguimiento y control de plagas y enfermedades, pero es posible corregirlo en futuros sprints.
Validación de campos del formulario	Asegurarse de que los campos del formulario de inicio de sesión se validen correctamente para garantizar la autenticación.	Los posibles errores de validación no amenazan gravemente la funcionalidad general de la aplicación. Los usuarios pueden volver a intentar el inicio de sesión si se produce un error de validación.	La falta de validación de campos podría permitir que los usuarios ingresen información incorrecta durante el inicio de sesión,
Inicio de sesión persistente	Comprobar si la aplicación valida si se ha iniciado sesión al cargar la aplicación y muestra las opciones correspondientes.	No es esencial en esta etapa inicial, ya que los usuarios pueden volver a iniciar sesión cuando abren la aplicación.	La falta de inicio de sesión persistente podría requerir que los usuarios inicien sesión cada vez que abran la aplicación, lo que podría resultar en una experiencia de usuario menos conveniente,

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

4.3.Características a no ser probadas Sprint 3

Tabla 18

Características a no ser probadas Sprint 3

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Obtener automáticamente la ubicación	La aplicación debe obtener la ubicación actual del agricultor o permitirle buscar una ubicación específica de manera precisa.	Existe la misma funcionalidad en el Sprint 2, por lo que ya fue probada	La precisión de la ubicación puede verse afectada por factores externos imprevistos.
Detalles del pronóstico	La información del pronóstico incluye detalles sobre la temperatura, la humedad, la velocidad del viento, el nivel de nubosidad y el estado general del día.	Existe la misma funcionalidad en el Sprint 2, por lo que ya fue probada	Problemas inesperados en la obtención o presentación de datos climáticos debido a cambios en las fuentes de datos o en la aplicación misma.
Funcionalidad de zoom en el mapa	El mapa permite hacer zoom y explorar para obtener detalles específicos de ubicación.	El mapa interactivo se mostrará inicialmente en una vista de nivel de zoom predeterminada que proporciona una visión adecuada de las áreas afectadas por plagas y enfermedades	La falta de zoom podría limitar la capacidad de los usuarios para obtener detalles específicos de ubicación, lo que podría considerarse una limitación en la experiencia del usuario.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

4.4.Características a no ser probadas Sprint 4

Tabla 19
Características a no ser probadas Sprint 4

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Mostrar áreas de cultivos de granos básicos	Muestra clara y precisamente las áreas donde se cultivan granos básicos como maíz, frijol y sorgo en Nicaragua.	El mapa se muestra a través de un WebView y no es desarrollado internamente, ya que se obtiene de la página del MAG.	Existe un riesgo potencial si el MAG realiza cambios en su mapa o en la forma en que se presenta la información sin notificar a la aplicación móvil. Esto podría afectar la experiencia del usuario si las características de visualización, zoom o búsqueda del mapa se ven afectadas de manera inesperada.
Funcionalidad de zoom en el mapa	Permite hacer zoom y explorar el mapa para obtener detalles específicos de ubicación.		
Herramientas de búsqueda y filtro	Incluye herramientas de búsqueda y filtro para encontrar información sobre cultivos en áreas específicas.		

4.5.Características a no ser probadas Sprint 5

Tabla 20
Características a no ser probadas Sprint 5

CARACTERÍSTICA	DESCRIPCIÓN	JUSTIFICACIÓN	RIESGO
Indicador claro de si la calificación fue "útil" o "no útil"	Acompaña cada calificación con un indicador claro de si el usuario la consideró "útil" o "no útil".	No se prueba por separado ya que está implícita en la historia principal y su función esencial es mostrar las calificaciones y su utilidad.	Se considera de bajo riesgo debido a su estrecha relación con la funcionalidad principal y su naturaleza estándar en la presentación de calificaciones en aplicaciones similares.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

5. CRITERIOS DE APROBACIÓN Y FALLO

Tabla 21
Criterios de aprobación y fallo

ID CRITERIO	DESCRIPCIÓN	APROBACIÓN	FALLO
C01	Se evalúa el proceso de actualización del nombre de usuario en su perfil.	El nombre se actualiza con éxito y se muestra el "Toast" correspondiente.	El nombre no se actualiza correctamente o no se muestra el "Toast" esperado.
C02	Se evalúa la calificación asignada por un usuario a la aplicación móvil.	Una vez obtenido el valor de la variable, si la calificación retornada es "Útil" y "No Útil", la calificación es válida.	La calificación retornada es diferente a "Útil" y "No Útil"
C03	Se evalúa que una imagen haya sido analizada por el modelo de aprendizaje automático.	Habiendo seleccionado una imagen desde galería o tomado una imagen con la cámara del dispositivo móvil, se verificará que la aplicación retorne el nombre y el nivel de precisión con la coincidencia identificada.	La aplicación retorna un nombre de plaga o enfermedad que no coincide con el modelo de aprendizaje automático.
C04	Se evalúa que al seleccionar la tarjeta de un cultivo se muestre información.	Una vez seleccionada la tarjeta de un cultivo (Frijol, Sorgo, Maíz), se verificará que muestre una lista 5 plagas y 5 enfermedades de dicho cultivo.	- La cantidad de plagas es diferente de 5 - La cantidad de enfermedades es diferente de 5
C05	Se evalúa que los datos ingresados por el	Una vez llenado los formularios y enviados, se	Los campos de tipo obligatorio no fueron llenados y los datos no

**APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y
ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS**

	usuario en formularios sean correctos.	verificará que los campos de tipo obligatorio sean llenados y que los datos cumplan con los requerimientos.	cumplen los requisitos
C06	Se evalúa que el usuario pueda crear una cuenta correctamente en la aplicación móvil.	Una vez enviado el formulario de registro de usuario, se verificará que aparezca en la base de datos.	La cuenta del usuario no aparece en la base de datos.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

6. CRITERIOS DE SUSPENSIÓN Y REANUDACIÓN

Tabla 22

Criterios de suspensión y reanudación

CRITERIO DE SUSPENSIÓN	CRITERIO DE REANUDACIÓN
Existencia de defectos que impidan la ejecución de más casos de prueba.	Corrección y verificación exitosa de los defectos. Se reanudarán las pruebas una vez que se haya solucionado el defecto y se haya confirmado que no afecta negativamente a otros casos de prueba.
Porcentaje de casos de prueba fallidos supera el 20%.	Análisis y corrección de los casos fallidos, seguido de una nueva ejecución de los casos de prueba. Las pruebas se reanudarán una vez que el porcentaje de casos fallidos sea inferior al 20%.
Cambio en los requisitos del software o especificaciones técnicas.	Actualización de los casos de prueba para reflejar los nuevos requisitos o especificaciones. Se reanudarán las pruebas una vez que los casos de prueba hayan sido ajustados de acuerdo con los nuevos requisitos.
Recurso crítico no disponible (por ejemplo, hardware o software necesario).	Disponibilidad del recurso crítico necesario. Las pruebas se reanudarán cuando el recurso necesario esté nuevamente disponible y operativo.
Cambio significativo en el entorno de prueba (por ejemplo, configuraciones).	Restauración del entorno de prueba a su estado anterior. Las pruebas se reanudarán después de que se haya restaurado el entorno de prueba a su estado original.
Nueva versión del software o sistema bajo prueba disponible.	Verificación de la nueva versión y actualización de los casos de prueba si es necesario. Las pruebas se reanudarán después de que se haya verificado la nueva versión y se hayan adaptado los casos de prueba según sea necesario.

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

7. TAREAS DE LAS PRUEBAS

Tabla 23
Tareas de las pruebas

TAREA	DESCRIPCIÓN	FECHA INICIO	FECHA FIN	DURACIÓN (HRS)	RESPONSABLE	ROL
Definir casos de prueba	Escribir los casos de prueba de caja negra utilizando la plantilla seleccionada	21/08/23	28/08/23	48	Jeyson Guevara	Desarrollo
Prueba de componentes	Comprobar que los componentes de la aplicación se han renderizado	29/08/23	03/09/23	24	Jeyson Guevara	Desarrollo
Prueba de funciones	Comprobar la ejecución de funciones necesarias de los módulos y componentes.	29/08/23	03/09/23	48	Jeyson Guevara	Desarrollo
Preparar el entorno de ejecución de pruebas	Instalar y configurar las herramientas seleccionadas para realizar las pruebas	04/09/23	08/09/23	12	Jeyson Guevara	Desarrollo

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

Definir las suites de prueba	Diseñar la estructura para la organización de scripts de prueba	04/09/23	08/09/23	10	Jeyson Guevara	Desarrollo
Codificar los scripts de prueba	Definir los archivos con la extensión .java en los que pueden estar escritos las pruebas	09/09/23	15/10/23	48	Jeyson Guevara	Desarrollo
	Escribir pruebas utilizando el framework de Java llamado Espresso					
	Definir los comandos de ejecución					
	Configurar Espresso para que permita obtener la cobertura de código probado					
Ejecución de los scripts de prueba	Correr los comandos de prueba en la	16/10/23	29/10/23	36	Jeyson Guevara	Desarrollo

**APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y
ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS**

terminal						
Corrección de fallas encontradas	Consiste en corregir los errores encontrados a través de las pruebas de funciones específicas.	16/10/23	29/10/23	18	Jeyson Guevara	Desarrollo

APLICACIÓN PARA LA DETECCIÓN DE PLAGAS Y ENFERMEDADES EN CULTIVOS DE GRANOS BÁSICOS

8. NECESIDADES AMBIENTALES

8.1.HARDWARE

Tabla 24

Necesidades ambientales de hardware

DISPOSITIVO	MARCA	CARACTERISTICAS	¿TENEMOS EL EQUIPO?
Computadora portátil	Lenovo	• Pantalla 15.6” HD • 16 GB RAM • Procesador AMD RYZEN 7 7000 SERIES	Si
Teléfono móvil	LG K40s	• Pantalla 6.1” HD • 2 GB RAM • Resolución 720 x 1520	Si

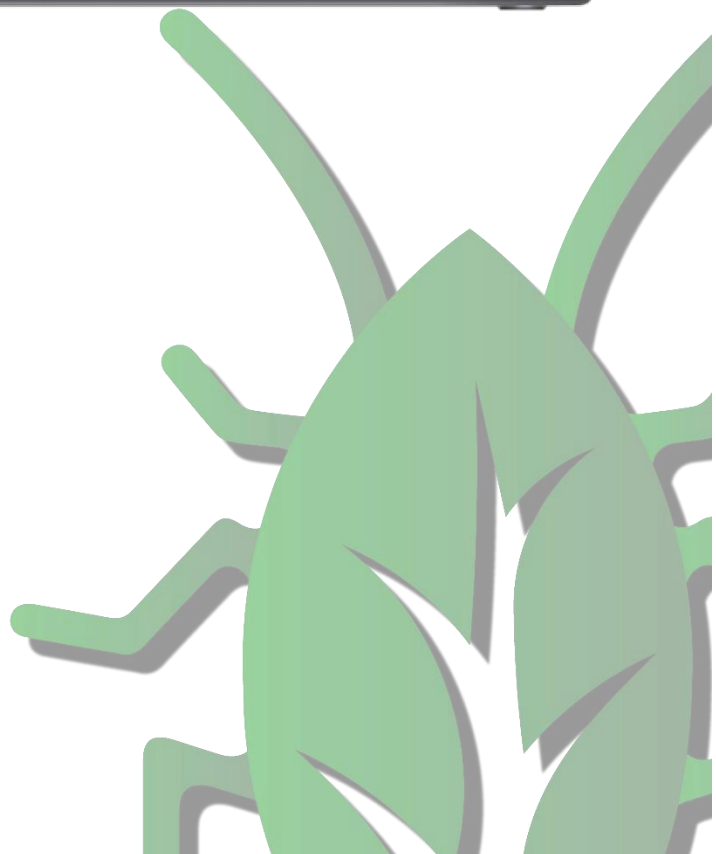
8.2.PLANEACIÓN DE COSTOS

Tabla 25

Planeación de costos

ID RECURSO	TIPO DE RECURSO	FORMA DE ADQUISICIÓN	CANTIDAD	COSTO DE RECURSO X UNIDAD	DÍA(S)	TOTAL
01	Tester	Contratación del servicio	-	C\$398.82	14	C\$5,583.48
02	Internet	Contratación del servicio	-	C\$36.51	50	C\$1,825.5
03	Energía eléctrica	Contratación del servicio	-	C\$5	50	C\$250
04	Computadora	Compra	1	C\$450(Depreciación)	122	C\$54,900
05	Mobiliario de oficina	Compra	-	C\$4000	-	C\$4000
06	Papelería	Compra	-	C\$400	-	C\$400
Total						C\$66,958.98

Participación en eventos nacionales



Índice

I. Introducción.....	1
----------------------	---

Índice de figuras

Figura 1. Primer lugar en Categoría Avanzado en Hackaton Nicaragua 2023.....	2
Figura 2. Reconocimientos de Hackaton Nicaragua 2023.....	2
Figura 3. Segundo lugar en Aplicaciones Digitales para la Agricultura y Ganadería	3
Figura 4. Reconocimiento de Aplicaciones Digitales para la Agricultura y Ganadería 2023	3
Figura 5. Estand de Placuix en el evento de la Muestra nacional de proyectos de innovación.....	4
Figura 6. Participación en Primera muestra nacional de proyectos de innovación 2023	4

I. Introducción

En el proceso de desarrollo de Placuix, el equipo de trabajo no solo ha centrado sus esfuerzos en la excelencia técnica, sino también en la participación activa en eventos nacionales que promueven la innovación y el desarrollo tecnológico. Esta sección de anexos, titulada "Participación en Eventos Nacionales", destaca el compromiso del proyecto en diversos escenarios competitivos donde se ha enfrentado a la evaluación de expertos y profesionales del campo.

En particular, Placuix ha alcanzado notables logros en competiciones de alto nivel, posicionándose como un referente en el ámbito tecnológico y agrícola. Destacamos con orgullo la obtención del primer lugar en la categoría avanzada en el Hackathon Nicaragua 2023, dedicado a la Inteligencia Artificial para la Educación y el Desarrollo, evento organizado por el Instituto Nacional Tecnológico (INATEC). Esta distinción no solo valida la calidad técnica del proyecto, sino también su capacidad para aportar soluciones significativas a problemáticas relevantes en la sociedad.

Asimismo, el proyecto ha demostrado su valía al adjudicarse el segundo lugar en la competición de Aplicaciones Digitales para la Agricultura y Ganadería, evento auspiciado por el Instituto Nicaragüense de Tecnología Agropecuaria (INTA). Este reconocimiento subraya la pertinencia de Placuix en el contexto agrícola, destacando su potencial para mejorar la eficiencia y sostenibilidad en el sector.

Además, Placuix ha participado activamente en la Primera Muestra Nacional de Proyectos de Innovación, consolidándose como parte fundamental del panorama innovador en el país.

A través de esta sección de anexos, se presenta una recopilación visual de los momentos destacados de estas competiciones, capturando la esencia del compromiso y la dedicación que han llevado al proyecto a alcanzar estos reconocimientos.

Figura 1

Primer lugar en Categoría Avanzado en Hackaton Nicaragua 2023



Nota. El nombre del equipo con el que se participó fue Legends_Nica, conformado por: Ramón González, Mayra Suarez, Jeyson Guevara, Ruddy Mendoza, Marchely Fonseca.

Figura 2

Reconocimientos de Hackaton Nicaragua 2023



Figura 3

Segundo lugar en Aplicaciones Digitales para la Agricultura y Ganadería 2023



Nota. La competencia fue realizada en CDT-INTA Posoltega, Chinandega.

Figura 4

Reconocimiento de Aplicaciones Digitales para la Agricultura y Ganadería 2023



Figura 6

Participación en Primera muestra nacional de proyectos de innovación 2023



Figura 5

Estand de Placuix en el evento de la Muestra nacional de proyectos de innovación

