



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 2

Neiva, 15 de junio de 2023

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

CARLOS MARIO CLAROS CACHAYA, con C.C. No. 1075307265,

JUAN FELIPE RIVERA PEÑA, con C.C. No. 1075320009,

Autor(es) de la tesis y/o trabajo de grado

Titulado **DISEÑO DE UNA APLICACIÓN MÓVIL PARA MONITOREAR EL CICLO BIOLÓGICO DEL CULTIVO DE ARROZ EN EL DEPARTAMENTO DEL HUILA**

presentado y aprobado en el año 2023 como requisito para optar al título de

INGENIERO ELECTRONICO;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



**UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS**



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

2 de 2

EL AUTOR/ESTUDIANTE:

Firma:

Carlos M. Claros

EL AUTOR/ESTUDIANTE:

Firma:

Felipe Rivera

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS				   	
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO					
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA 1 de 5

TÍTULO COMPLETO DEL TRABAJO: DISEÑO DE UNA APLICACIÓN MÓVIL PARA MONITOREAR EL CICLO BIOLÓGICO DEL CULTIVO DE ARROZ EN EL DEPARTAMENTO DEL HUILA

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
CLAROS CACHAYA	CARLOS MARIO
RIVERA PEÑA	JUAN FELIPE

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
QUINTERO POLANCO	JESUS DAVID

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre

PARA OPTAR AL TÍTULO DE: INGENIERO ELECTRONICO

FACULTAD: INGENIERIA

PROGRAMA O POSGRADO: ELECTRONICA

CIUDAD: Neiva

AÑO DE PRESENTACIÓN: 2023

NÚMERO DE PÁGINAS: 59

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas ☒ Fotografías ☒ Grabaciones en discos ☒ Ilustraciones en general___ Grabados___ Láminas___
 Litografías___ Mapas___ Música impresa___ Planos___ Retratos___ Sin ilustraciones___ Tablas o Cuadros ☒

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO					
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA 2 de 5

SOFTWARE requerido y/o especializado para la lectura del documento: Sistema operativo Android 5.0 y posteriores.

MATERIAL ANEXO:

PREMIO O DISTINCIÓN *(En caso de ser LAUREADAS o Meritoria):*

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>	<u>Español</u>	<u>Inglés</u>
1. APP		6. LAZY LOADING	
2. API		7. MÉTODO	
3. FRAMEWORK			
4. HTTP			
5. JSON			

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El arroz es una de las principales fuentes de alimentación a nivel mundial y su producción es una actividad económica importante en muchos países. Sin embargo, la producción de arroz puede ser vulnerable a diversos factores que afectan el rendimiento de las plantaciones. En Colombia, el arroz es cultivado en varios departamentos, pero la mayoría del cereal se importa, lo que aumenta el costo de la canasta familiar (*). Para mejorar la producción de arroz principalmente al norte del departamento del Huila y con ánimos de mitigar la importación, se desarrolla la aplicación “Siembra Arroz”, la cual monitorea el control del proceso de crecimiento de las plantaciones de arroz, proporcionando información detallada sobre el tipo de suelo, riego, fertilización, control de plagas y enfermedades, y otros aspectos. La aplicación también ayuda a los agricultores a hacer un seguimiento del rendimiento de sus cultivos y de los costos asociados con el cultivo, lo que les permite evaluar el rendimiento económico y tomar decisiones informadas. El proceso de desarrollo de la aplicación incluye el diseño de la interfaz de usuario, la implementación de la lógica de la aplicación y las pruebas para garantizar su correcto funcionamiento.



Además, la aplicación ayudará a los agricultores a hacer un seguimiento del rendimiento de sus cultivos y de los costos asociados con el cultivo, lo que les permitirá maximizar sus ganancias. La aplicación incluirá una serie de funcionalidades que permitirán a los usuarios registrar y visualizar los datos de crecimiento de las plantas, así como recibir alertas en caso de detectar problemas. El proceso de desarrollo de la aplicación incluirá el diseño de la interfaz de usuario, la implementación de la lógica de la aplicación y las pruebas para garantizar su correcto funcionamiento.

ABSTRACT: (Máximo 250 palabras)



Rice is one of the main sources of food worldwide and its production is an important economic activity in many countries. However, rice production can be vulnerable to various factors that affect plantation yields. In Colombia, rice is grown in several departments, but Android the grain is imported, which increases the Android the family food basket (*). To improve rice production mainly in the north of the department of Huila and with the aim of mitigating imports, the “Siembra Arroz” application was developed, which monitors the control of the growth process of rice plantations, providing detailed information on soil type, irrigation, fertilization, pest and disease control, and other aspects. The application also helps farmers track their crop yields and costs associated with the crop, enabling them to evaluate economic performance and make informed decisions. The application development process includes designing the user interface, implementing the application logic, and testing to ensure proper operation.

The app helps farmers keep track of their crop yields and the costs associated with growing them, allowing them to maximize their profits. The application includes a series of functionalities that allow users to record and view plant growth data, as well as receive alerts for each of the recommended controls. The app development process includes the Android UI, implementation of the app logic, and testing to ensure proper functionality.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	5 de 5

APROBACION DE LA TESIS

Nombre Presidente Jurado:

Firma:

Nombre Jurado: YAMIL ARMANDO CERQUERA ROJAS



Firma:

Nombre Jurado: JOHAN JULIÁN MOLINA MOSQUERA

Firma:

**DISEÑO DE UNA APLICACIÓN MÓVIL PARA MONITOREAR EL CICLO BIOLÓGICO DEL
CULTIVO DE ARROZ EN EL DEPARTAMENTO DEL HUILA**

Carlos Mario Claros Cachaya

Juan Felipe Rivera Peña

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA – HUILA
2023**

**DISEÑO DE UNA APLICACIÓN MÓVIL PARA MONITOREAR EL CICLO BIOLÓGICO DEL
CULTIVO DE ARROZ EN EL DEPARTAMENTO DEL HUILA**

Carlos Mario Claros Cachaya

Juan Felipe Rivera Peña

Trabajo de Grado Para Optar al Título de Ingeniero Electrónico

**Director y proponente de la idea:
Msc. Jesús David Quintero Quintero Polanco**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA – HUILA
2023**

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Neiva, 3 de mayo de 2023

DEDICATORIA

A mis padres Mario Claros y Blanca Cachaya que con mucho esfuerzo y sacrificio me han apoyado a lo largo de toda mi vida y mi pregrado, a mi esposa Daniela e hija Victoria que son mi motor de cada día, mis hermanos Juan David y Carolina, y cuñado Luis Quintero. Siempre estuvieron conmigo para salir adelante. A toda mi familia que en un momento u otro han estado junto a mi para hacer esto posible.

Carlos Mario Claros Cachaya

Esta dedicatoria va a mis padres, quienes siempre han sido mi fuente de inspiración y apoyo incondicional a lo largo de mi vida; La ética, valores humanos, que han ayudado a la integridad aceptable ante esta sociedad en la cual me integro; gracias por creer en mí y por enseñarme el valor del trabajo duro y la perseverancia. A mi pareja, por su amor, paciencia y comprensión durante este largo proceso de superación y triunfo en mí.

Juan Felipe Rivera Peña

AGRADECIMIENTOS

A mis padres, hermanos y esposa sin ellos no había sido posible este logro. Al ingeniero Jesus David Quintero, por ese apoyo como docente y tutor de este proyecto de grado, a la universidad Sur colombiana y cada uno de los docentes que nos ayudaron a lo largo de la carrera.

Carlos Mario Claros Cachaya

A mis amigos y compañeros de estudio, por su colaboración y compañerismo en los momentos difíciles, y finalmente agradezco a mi maestro y director de tesis por su orientación y asesoramiento experto, fortaleciendo las capacidades investigativas y conocimientos que se fueron desarrollando en cada aspecto.

Juan Felipe Rivera Peña

TABLA DE CONTENIDO

pág.

1. INTRODUCCIÓN	13
2. OBJETIVOS	14
2.1 OBJETIVO GENERAL	14
2.2 OBJETIVOS ESPECÍFICOS	14
3. MARCO TEÓRICO	15
3.1 CICLO BIOLÓGICO	15
3.2 SISTEMA OPERATIVO ANDROID	15
3.3 MODELO VISTA - VISTA MODELO	15
3.4 FIRESTORE	16
3.5 KOTLIN	17
4. CICLO BIOLÓGICO DEL CULTIVO DEL ARROZ	19
4.1.1 Fase Vegetativa	19
4.1.2 Fase Reproductiva	20
4.1.3 Fase de Maduración	20
4.1.4 Labores culturales relacionadas con cada etapa	21
4.2 VISITA TÉCNICA A DIFERENTES CULTIVOS DE ARROZ	23
4.2.1 Máquinas agrícolas	23
4.3 CULTIVOS DE ARROZ	26
5. HERRAMIENTAS UTILIZADAS	36
5.1 ESTRUCTURA DE LA BASE DE DATOS	36
5.1.1 Creación de tablas de datos	37
5.2 ARQUITECTURA DEL SISTEMA	37
5.3 INTERFAZ GRÁFICA	38
5.4 LOGO	39
6. DESARROLLO DEL APLICATIVO MÓVIL	41
6.1 CONFIGURACIÓN	41
6.1.1 Build.grade	41
6.1.2 AndroidManifest.xml	41
6.2 CREACION DE MODULOS EN KOTLIN	41
6.2.1 Módulo de login	41
6.2.2 Módulo de registro de cultivos	41
6.2.3 Módulo del clima	42
6.2.4 Módulo de notificaciones	42

6.2.5 Broadcast receiver	43
6.3 VISTAS	43
6.3.1 Login	44
6.3.2 Registro de cultivos	44
6.3.3 Registrar un cultivo	45
6.3.4 Clima	47
6.3.5 Recomendaciones agrícolas	48
7. CONCLUSIONES	49
8. RECOMENDACIONES	50
9. BIBLIOGRAFÍA	51
10. ANEXOS	52

LISTA DE FIGURAS

pág.

Figura 1.	16	Figura 2.	18
	19	Figura 3.	20
	20	Figura 4.	21
	23	Figura 5.	24
	25	Figura 6.	26
	26	Figura 7.	27
	27	Figura 8.	28
	28	Figura 9.	29
	30	Figura 10.	30
	30	Figura 11.	31
	35	Figura 12.	36
	38	Figura 13.	38
	39	Figura 14.	39
	40	Figura 15.	41
	41	Figura 16.	41
	42	Figura 17.	43
	43	Figura 18.	44
	44	Figura 19.	45
	46	Figura 20.	46
	47	Figura 21.	47

LISTA DE TABLAS

pág.

Tabla 1. 31
33
Tabla 2.32
Tabla 4.

3.
36

LISTA DE ANEXOS

pág.

Anexo A.	51	Anexo B.	52
----------	----	----------	----

GLOSARIO

APP: abreviatura de la palabra inglesa Application. Es una aplicación de software que realiza funciones específicas y trabaja en diversos dispositivos como teléfonos, relojes inteligentes, tablets e incluso televisores.

API: Interfaz de programación de aplicaciones que contiene un conjunto de funciones y procedimientos para realizar una o varias funciones con el fin de ser utilizadas por otro software.

FRAMEWORK: esquema de trabajo para desarrollar e implementar software en diversos lenguajes de programación.

HTTP: protocolo de transferencia de hipertexto el cual permite la transferencia de datos y navegación en la web.

JSON: notación de objetos de JavaScript, formato de intercambio de datos que permite transferir información entre cliente y servidor.

LAZY LOADING: patrón de carga diferida que permite entregar recursos de una página web a medida que el usuario los solicite.

MÉTODO: función que contiene un conjunto de instrucciones, las cuales se ejecutan cuando se llame a dicha función.

RESUMEN

El arroz es una de las principales fuentes de alimentación a nivel mundial y su producción es una actividad económica importante en muchos países. Sin embargo, la producción de arroz puede ser vulnerable a diversos factores que afectan el rendimiento de las plantaciones. En Colombia, el arroz es cultivado en varios departamentos, pero la mayoría del cereal se importa, lo que aumenta el costo de la canasta familiar (*). Para mejorar la producción de arroz principalmente al norte del departamento del Huila y con ánimos de mitigar la importación, se desarrolla la aplicación “Siembra Arroz”, la cual monitorea el control del proceso de crecimiento de las plantaciones de arroz, proporcionando información detallada sobre el tipo de suelo, riego, fertilización, control de plagas y enfermedades, y otros aspectos. La aplicación también ayuda a los agricultores a hacer un seguimiento del rendimiento de sus cultivos y de los costos asociados con el cultivo, lo que les permite evaluar el rendimiento económico y tomar decisiones informadas. El proceso de desarrollo de la aplicación incluye el diseño de la interfaz de usuario, la implementación de la lógica de la aplicación y las pruebas para garantizar su correcto funcionamiento.

Además, la aplicación ayudará a los agricultores a hacer un seguimiento del rendimiento de sus cultivos y de los costos asociados con el cultivo, lo que les permitirá maximizar sus ganancias. La aplicación incluirá una serie de funcionalidades que permitirán a los usuarios registrar y visualizar los datos de crecimiento de las plantas, así como recibir alertas en caso de detectar problemas. El proceso de desarrollo de la aplicación incluirá el diseño de la interfaz de usuario, la implementación de la lógica de la aplicación y las pruebas para garantizar su correcto funcionamiento.

(*) MINISTERIO DE AGRICULTURA Y DESARROLLO RURAL EN COLOMBIA. Informe sobre producción de arroz en Colombia. 2020.

ABSTRACT

Rice is one of the main sources of food worldwide and its production is an important economic activity in many countries. However, rice production can be vulnerable to various factors that affect plantation yields. In Colombia, rice is grown in several departments, but Android the grain is imported, which increases the Android the family food basket (*). To improve rice production mainly in the north of the department of Huila and with the aim of mitigating imports, the “Siembra Arroz” application was developed, which monitors the control of the growth process of rice plantations, providing detailed information on soil type, irrigation, fertilization, pest and disease control, and other aspects. The application also helps farmers track their crop yields and costs associated with the crop, enabling them to evaluate economic performance and make informed decisions. The application development process includes designing the user interface, implementing the application logic, and testing to ensure proper operation.

The app helps farmers keep track of their crop yields and the costs associated with growing them, allowing them to maximize their profits. The application includes a series of functionalities that allow users to record and view plant growth data, as well as receive alerts for each of the recommended controls. The app development process includes the Android UI, implementation of the app logic, and testing to ensure proper functionality.

(*) MINISTERIO DE AGRICULTURA Y DESARROLLO RURAL EN COLOMBIA. Informe sobre producción de arroz en Colombia. 2020.

1. INTRODUCCIÓN

En el ámbito agrícola, el control y monitoreo del proceso productivo de un cultivo es de vital importancia para obtener una cosecha de alta calidad y rentable. En este sentido, el uso de la tecnología puede ser una herramienta clave para mejorar y optimizar la gestión del ciclo de vida del cultivo.

El presente trabajo tiene como objetivo general diseñar e implementar un aplicativo móvil que permita el monitoreo del ciclo biológico del cultivo de arroz, desde su germinación hasta la recolección, con el fin de llevar un control detallado y eficiente de todo el proceso productivo.

Para lograr este objetivo, se realizará una recolección exhaustiva de la información del ciclo de vida del arroz, con la asesoría de un ingeniero agrónomo experto en el tema, brindado por molinos ORF. A partir de esta información, se procederá a diseñar el aplicativo móvil que permitirá la creación del ciclo de vida del arroz y su posterior monitoreo. Además, se verificará la funcionalidad del aplicativo móvil con algunos agricultores que tengan algún tipo de vínculo con el molino ORF. Finalmente, se elaborará el documento final del proyecto desarrollado.

La implementación de este aplicativo móvil permitirá una gestión más eficiente y detallada del proceso productivo del cultivo de arroz, generando un mayor beneficio económico y una mejora en la calidad de la cosecha. Asimismo, este trabajo representa una contribución importante al sector agrícola, al utilizar la tecnología para mejorar y optimizar la gestión de los cultivos.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e implementar un aplicativo móvil que permita monitorear el ciclo biológico del cultivo de arroz, para llevar un control de cosecha desde la germinación hasta la recolección.

2.2 OBJETIVOS ESPECÍFICOS

- Recolectar la información del ciclo de vida del arroz con la asesoría de un ingeniero agrónomo experto en el tema, brindado por molinos ORF.
- Una vez recolectada la información se procede a realizar el diseño del aplicativo móvil, para la creación del ciclo de vida del arroz.
- Realizar el documento final del proyecto desarrollado.

3. MARCO TEÓRICO

3.1 CICLO BIOLÓGICO

El ciclo biológico se refiere a los cambios y etapas que experimenta un organismo desde su nacimiento hasta su muerte. De acuerdo con Urry¹, estas etapas pueden incluir la reproducción y la maduración, así como cambios fisiológicos y de comportamiento a medida que el organismo crece y se desarrolla.

El ciclo biológico puede variar según la especie y puede incluir diferentes etapas, como la fase larval, juvenil y adulta. En los organismos que se reproducen sexualmente, el ciclo biológico también puede incluir la formación y fusión de células sexuales para producir una nueva generación.

En algunos organismos, como las plantas, el ciclo biológico puede incluir diferentes formas, como una etapa vegetativa y una etapa reproductiva.

3.2 SISTEMA OPERATIVO ANDROID

Android entendido en el autor Frank², es una plataforma de software para dispositivos móviles que incorpora un sistema Operativo y aplicaciones de base. También mezcla herramientas y aplicaciones que están vinculadas a una distribución Linux para dispositivos móviles y es de código abierto, donde se pueden crear aplicaciones para plataformas usando el SDK de Android para dispositivos móviles.

3.3 MODELO VISTA - VISTA MODELO

MVVM (Model-View-ViewModel)³ es un patrón de arquitectura de software utilizado en el desarrollo de aplicaciones informáticas que se basa en la separación de preocupaciones. Este patrón de diseño se utiliza comúnmente en el desarrollo de aplicaciones móviles y de escritorio, así como en el desarrollo web.

¹URRY, LISA. Et. Campbell Biology. 11th edition. College, Oakland, California, 2016. ISBN-13: 978-0134093413, ISBN-10: 0134093410

²FRANK, D. Et. (en línea) Jualiaca, Perú. Monografías. 2014. (Sistema Operativo Android). (12 de Junio de 2019). Obtenido de Monografías. Recuperado de: [<https://www.monografias.com/trabajos101/sistema-operativo-android/sistema-operativo-android>]

³MVVM. Android: ¿Qué es MVC, MVP y MVVM? [en línea]. PyM. 2017. Recuperado de: [<https://programacionymas.com/blog/android-mvc-mvp-mvvm>]

El patrón MVVM se compone de tres componentes principales:

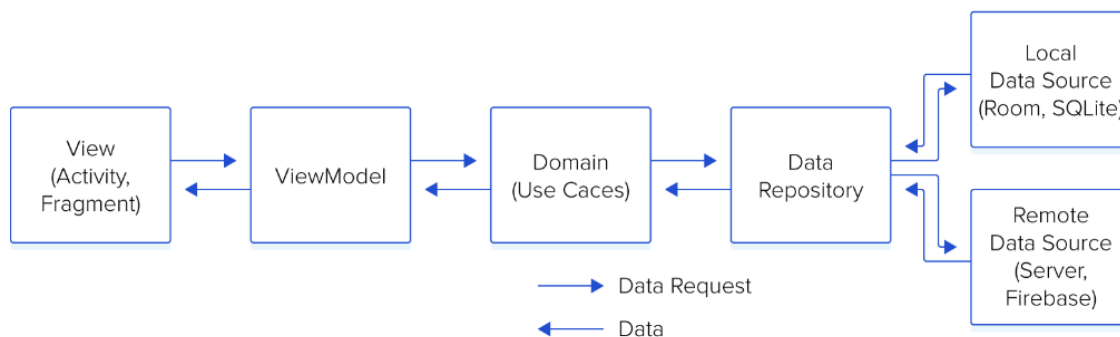
Modelo: representa la capa de datos de la aplicación y define cómo se accede y se manipula la información.

Vista: representa la interfaz de usuario de la aplicación, que permite a los usuarios interactuar con la información y realizar acciones.

ViewModel: actúa como un intermediario entre el Modelo y la Vista. Es responsable de traducir los datos del Modelo en un formato que la Vista pueda entender y también de traducir las acciones del usuario en comandos que el Modelo pueda procesar.

En el patrón MVVM, la Vista se comunica con el ViewModel a través de enlaces de datos (bindings), que permiten que los datos se actualicen automáticamente en la Vista cuando se producen cambios en el ViewModel. El ViewModel, a su vez, se comunica con el Modelo para recuperar o actualizar los datos. Todo esto se puede observar de una forma más clara en la Figura 1.

Figura 1. Diagrama vista controlador (MVC).



Fuente: Autoría propia

3.4 FIRESTORE

Firestore¹ es un servicio de base de datos en la nube, desarrollado por Google, que permite a los desarrolladores crear y gestionar bases de datos NoSQL en tiempo real para aplicaciones web y móviles. Firestore es parte de la plataforma Firebase de Google, que proporciona una variedad de servicios para el desarrollo de aplicaciones móviles y web.

Firestore es una base de datos NoSQL (No Relacional) que se basa en el modelo

de documentos y colecciones. En lugar de almacenar los datos en tablas como lo hace una base de datos relacional, Firestore almacena los datos en documentos que contienen campos y valores. Los documentos se organizan en colecciones, lo que permite a los desarrolladores estructurar los datos de manera lógica y coherente.

¹Firestore. Aplicaciones web. Firebase. 2014. [Consultado en 2022]. Recuperado de [\[https://firebase.google.com/\]](https://firebase.google.com/)

Algunas de las características clave de Firestore incluyen:

- Escalabilidad: Firestore es escalable horizontalmente, lo que significa que puede manejar grandes cantidades de datos y un alto volumen de tráfico de usuarios.
- Tiempo real: Firestore actualiza los datos en tiempo real, lo que permite a los usuarios ver los cambios realizados por otros usuarios de manera inmediata.
- Sincronización offline: Firestore admite el almacenamiento en caché de datos para permitir la sincronización offline.
- Seguridad: Firestore proporciona una amplia variedad de herramientas de seguridad para proteger los datos de los usuarios, como la autenticación de usuarios y la autorización de acceso.
- Integración con otras herramientas de Firebase: Firestore se integra con otras herramientas de Firebase, como el servicio de autenticación de usuarios, el servicio de almacenamiento en la nube y el servicio de mensajería en la nube.

Firestore es una herramienta potente para el desarrollo de aplicaciones web y móviles que requieren una base de datos en tiempo real y escalable. Ofrece características avanzadas de seguridad, integración con otras herramientas de Firebase y sincronización offline, lo que lo convierte en una opción popular para los desarrolladores de aplicaciones.

3.5 KOTLIN

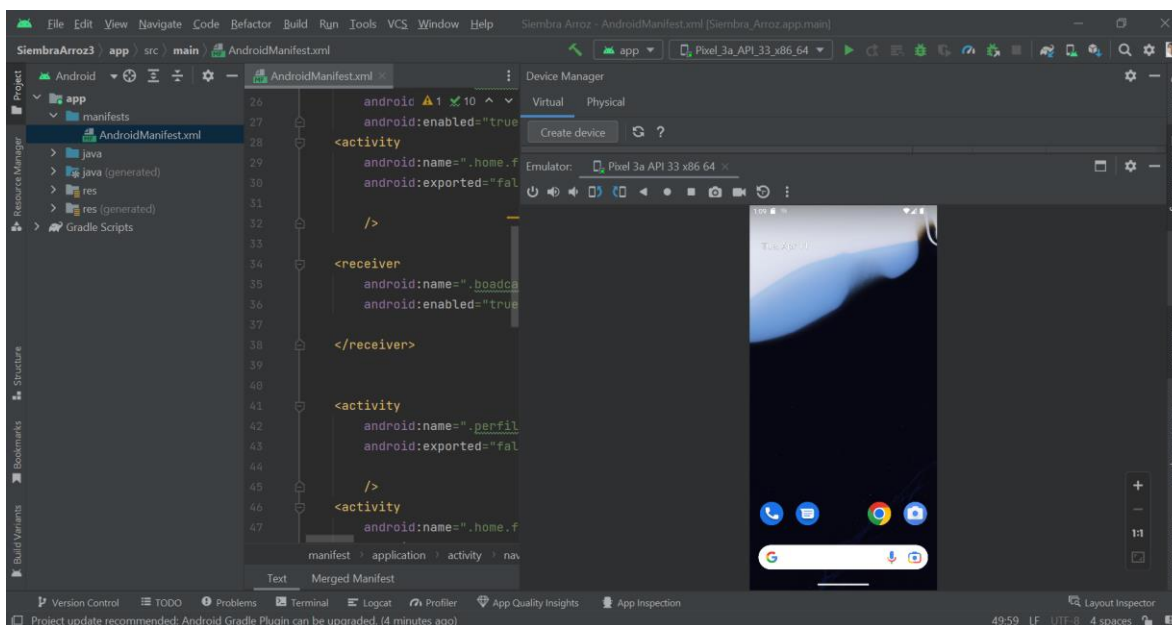
Kotlin¹ es un lenguaje de programación de tipo estático que se ejecuta en la máquina virtual de Java (JVM) y también se puede compilar en JavaScript o código nativo. Fue desarrollado por JetBrains, la compañía detrás del entorno de desarrollo integrado (IDE) IntelliJ IDEA, con el objetivo de mejorar la productividad de Java y proporcionar una alternativa a Java en la plataforma Android. Kotlin es conciso, expresivo y seguro, y tiene muchas funciones que facilitan la escritura de código limpio y fácil de mantener, como seguridad nula, funciones de extensión y lambdas. También es totalmente interoperable con Java, por lo que el código Java existente se puede usar en proyectos de Kotlin y viceversa. Kotlin ha ganado popularidad en

los últimos años y se usa ampliamente para el desarrollo de aplicaciones de Android, la programación del lado del servidor y otros tipos de aplicaciones.

¹KOTLIN. Lenguaje de programación Kotlin. JetBrains. Kotlinlang.org. 2016. [Consultado en 2022] Recuperado de <https://kotlinlang.org/>

En Android Studio que es el entorno de programación de Kotlin se puede crear diferentes tipos de dispositivos virtuales, modificar el sistema operativo y diferentes características de este como se observa en la figura 2.

Figura 2. Entorno de programación.



Fuente: Autoría propia

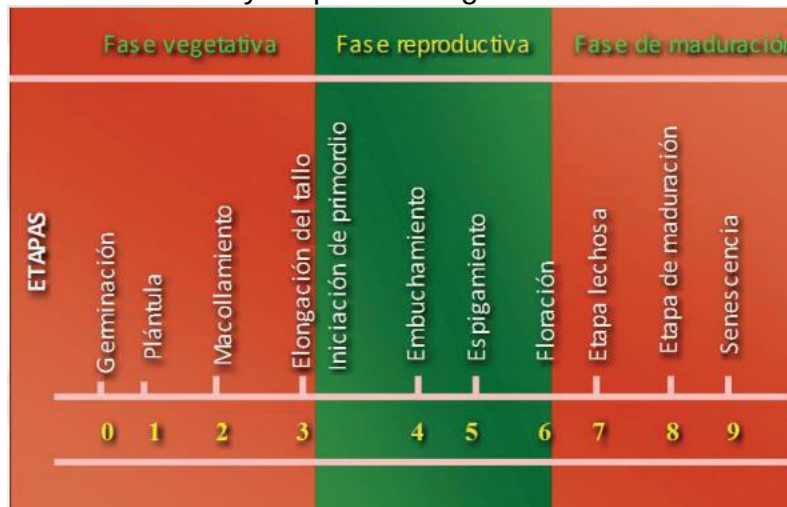
4. CICLO BIOLÓGICO DEL CULTIVO DEL ARROZ

La agricultura sostenible juega un papel importante en la conservación de los recursos naturales, la reducción de las emisiones de gases de efecto invernadero, la prevención de la pérdida de biodiversidad y el cuidado de los paisajes valiosos. Mantener una buena calidad del suelo, reducir la degradación y la erosión del suelo son los principales beneficios de adoptar prácticas agrícolas sostenibles.

Gracias a molinos ORF el cual brindo un ingeniero agrónomo experto en el tema, que tras varias horas de asesoría e investigaciones se logra recolectar y simplificar toda la información respecto al ciclo biológico del cultivo del arroz.

De acuerdo con Olmos¹ Et. el ciclo de vida del cultivo del arroz puede ser dividido en tres grandes fases: Vegetativa, Reproductiva y de Maduración. A su vez, estas fases de crecimiento pueden ser divididas en etapas fenológicas, las cuales determinan el estado de desarrollo del cultivo. La incidencia de clima y el manejo agronómico serán determinantes para que la planta pueda expresar todo su potencial. El resumen de estas etapas puede observarse en la figura 3.

Figura 3. Fases de crecimiento y etapas fenológicas del cultivo del arroz



Fuente: Autoría propia

¹Olmos, S. Cartilla fisiológica del arroz. Cátedra de Cultivos II Facultad de Ciencias Agrarias, UNNE. Corrientes, 2006 [consultado en 2018] tomado de [\[https://www.acpaarrozcorrientes.org.ar/academico/Apunte-MORFOLOGIA.pdf\]](https://www.acpaarrozcorrientes.org.ar/academico/Apunte-MORFOLOGIA.pdf)

4.1.1 Fase Vegetativa

Se inicia con la germinación de la semilla y termina con la diferenciación del primordio floral. En ella se determina el número de macollas que tendrá el cultivo (figura 4), del cual depende el número de panículas por unidad de área. La duración de esta fase puede variar, de acuerdo a la variedad y al clima. En términos generales, tiene una duración entre 35 y 50 días. La diferencia en el ciclo de vida entre una u otra variedad, está determinada por la mayor o menor duración de la fase vegetativa.

Figura 4. Macollamiento de la planta del arroz

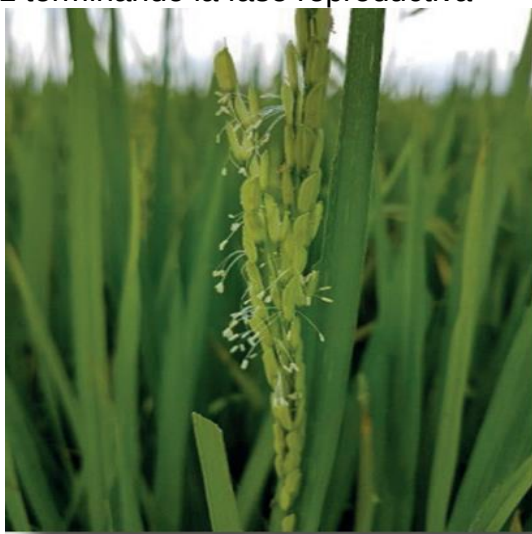


Fuente: Autoría propia

4.1.2 Fase Reproductiva

Se inicia con la diferenciación del primordio floral y termina con la floración del cultivo (figura 5). En esta fase se determina el número de espiguillas que tendrá cada panícula. Su duración oscila entre 30 y 35 días. Esta fase solo es visible cuando abrimos la planta e internamente vemos su desarrollo.

Figura 5. Planta de arroz terminando la fase reproductiva



Fuente: Autoría propia

4.1.3 Fase de Maduración

Inicia con la etapa de floración y termina con la madurez del grano. En esta fase se determina cuántas espiguillas se convertirán en granos efectivos y el peso individual de cada uno de ellos. Su duración oscila entre 30-45 días, dependiendo de la variedad y las condiciones climáticas

Figura 6. Fase de maduración del cultivo del arroz



Fuente: Autoría propia

4.1.4 Labores culturales relacionadas con cada etapa

Etapa 0

- La duración del proceso de germinación se ve influenciada por la humedad del suelo y la profundidad de siembra.
- La correcta realización de las prácticas de preparación y adecuación de suelos resultan fundamentales para que la semilla de arroz encuentre un medio adecuado para su germinación.
- Un suelo técnicamente bien preparado y adecuado facilita el manejo del agua de riego para evitar pérdidas de semilla, ya sea por excesos o déficit de agua.
- El diagnóstico físico del suelo es una herramienta que permite definir el tipo de preparación para cada lote específico.

Etapa 1

- Distribución oportuna y balanceada del riego para uniformizar el cultivo.
- Control temprano de malezas, en premergencia o postemergencia temprana.
- Primera fertilización del cultivo.
- Monitoreo de insectos fitófagos del suelo y comedores de follaje.

Etapa 2

- Es recomendable el riego intermitente durante esta etapa, evitando la inundación y el estrés por sequía.
- Si es necesario, control de malezas en postmediana (20-25 días después de la germinación-ddg). No se deben realizar aplicaciones tardías de herbicidas.

- Monitoreo de insectos como chupadores, comedores de follaje y barrenadores del tallo.
- Segunda nutrición balanceada del cultivo de acuerdo al análisis de suelos y el requerimiento de la variedad.
- Monitoreo de enfermedades.

Etapas 3

- Previo a esta etapa se debe realizar una fertilización rica en nitrógeno y potasio, según el análisis de suelo y los requerimientos de la variedad sembrada.
- Monitoreo de enfermedades y de insectos comedores de follaje y barrenadores.
- Buen manejo del agua riego: suelo debe permanecer húmedo.
- No se deben realizar aplicaciones de herbicidas cerca a esta etapa de desarrollo; en caso de presencia, realice el control manual de malezas y del arroz maleza.
- Condiciones de buena radiación solar y temperaturas adecuadas favorecen su desarrollo.

Etapas 4

- Monitoreo de enfermedades e insectos fitófagos chupadores y barrenadores.
- El suelo debe permanecer con muy buen contenido de humedad. Durante esta etapa la planta de arroz es muy sensible al estrés por falta de agua.
- Se deben evitar aplicaciones excesivas de nitrógeno que favorezcan crecimiento exagerado del follaje de cultivo.
- Condiciones de alta radiación solar y temperaturas frescas durante la noche favorecen esta etapa de desarrollo.

Etapas 5

- Se debe cuidar el manejo del agua del riego, ya que una condición de estrés podría afectar significativamente la productividad.
- Monitorear la presencia de insectos o enfermedades.
- Revisar los registros de los factores del clima y su influencia en el desarrollo.
- En caso de aplicación de pesticidas para protección de la panícula, se debe seleccionar muy bien los productos y evitar las incompatibilidades.

Etapas 6

- Control preventivo y curativo de enfermedades fungosas: para protección de la espiga y las hojas bandera y dos siguientes.
- El suelo debe permanecer saturado o con lámina de agua permanente.
- Monitoreo y evaluación de insectos como el chinche de la espiga y grillos.

Etapas 7

- Se debe cuidar el manejo del agua del riego, ya que una condición de estrés aumenta el vaneamiento y disminuye el peso de granos individuales, afectando significativamente la productividad.
- Evitar la inundación que pueda causar volcamiento a la planta aun sin llegar a la maduración de los granos.
- Evitar aquellas épocas de fuertes vientos y lluvias ocasionan volcamiento

Etapas 8

- Se debe suspender el suministro de agua unos días antes de la cosecha para facilitar el movimiento de la combinada. El número de días depende de la capacidad de retención de humedad del suelo.
- Es de gran importancia calibrar la cosechadora para evitar pérdidas significativas de rendimiento

Etapas 9

Las plantas del arroz se desarrollan continuamente desde su germinación hasta su muerte. Olmos² Et. señala que la senescencia es el último proceso del desarrollo que indica la finalización de la organización y funciones o la madurez de la planta.

²Olmos, S. Cartilla fisiológica del arroz. Cátedra de Cultivos II Facultad de Ciencias Agrarias, UNNE. Corrientes, 2006 [consultado en 2018] tomado de [\[https://www.acpaarrozcorrientes.org.ar/academico/Apuntes-MORFOLOGIA.pdf\]](https://www.acpaarrozcorrientes.org.ar/academico/Apuntes-MORFOLOGIA.pdf)

4.2 VISITA TÉCNICA A DIFERENTES CULTIVOS DE ARROZ

Gracias al ingeniero Jesus David Quintero docente de la universidad Surcolombiana, se realizaron visitas a diferentes cultivos de arroz en el municipio de Yaguara, donde se encontraron cultivos en diferentes etapas fenológicas, maquinaria agrícola e insumos agrícolas que los agricultores de la zona utilizan.

4.2.1 Máquinas agrícolas

El suelo además de ser el soporte físico de la planta de arroz, es el sustrato que provee los nutrientes durante su respectivo crecimiento y desarrollo del cultivo. Aunque también, es en el suelo donde se desarrollan otros factores adversos al cultivo, como las malezas, insectos, hongos, bacterias y otros. Teniendo esto en claro, entonces el objetivo principal de la preparación de tierras es entre otros; destruir las malezas presentes, incorporar la materia orgánica en el suelo (como residuos de la cosecha anterior y de las malezas) y contribuir a mejorar la estructura (mullir o reducir el tamaño de los terrones) en la capa arable, a fin de que la semilla sea colocada en un medio apropiado para la respectiva germinación en el suelo. Los arados de disco (figura 7) es uno de los principales equipos agrícolas que se utilizan para la preparación del terreno.

Figura 7. Arados de disco



Fuente: Autoría propia

Las acaballonadoras son equipos agrícolas utilizados para formar camas elevadas para cultivos. Estas máquinas suelen ser arrastradas por tractores y cuentan con múltiples cuchillas o discos que trabajan juntos para mover la tierra desde el centro del campo hacia los bordes, creando hileras elevadas de tierra. Los lechos elevados creados por acaballonadoras ayudan a mejorar el drenaje, la aireación y la fertilidad del suelo, lo que puede conducir a mayores rendimientos y cultivos más saludables.

Hay diferentes tipos de acaballonadoras disponibles en el mercado, como las versiones remolcadas, semi remolcadas o montadas, y vienen en varios tamaños para adaptarse a las diferentes necesidades agrícolas.

Figura 8. Acaballonadoras



Fuente: Autoría propia

La cosechadora de arroz, un tipo de máquina agrícola diseñada para cosechar eficientemente el arroz de los campos. Las cosechadoras de arroz pueden variar en tamaño y complejidad, pero generalmente cuentan con un mecanismo de corte que corta los tallos de arroz en la base, un mecanismo para recolectar las plantas de arroz cortadas y un sistema de trilla que separa los granos de arroz del material vegetal.

Figura 9. Cortadora de arroz



Fuente: Autoría propia

4.3 CULTIVOS DE ARROZ

Se encontraron cultivos en diferentes etapas fenológicas, cultivos desde la siembra hasta cuando ya están listos para la recolección.

Las semillas cuando se plantan en un entorno adecuado, las semillas de arroz germinan y se convierten en plantas de arroz, que eventualmente producen nuevas semillas de arroz, completando el ciclo de vida. La correcta realización de las prácticas de preparación y

adecuación de suelos resultan fundamentales para que la semilla de arroz encuentre un medio adecuado para su germinación.

Figura 10. Terreno preparado



Fuente: Autoría propia

En la figura 10. Se observa un terreno preparado, con surcos listo para sembrar las semillas de arroz

Figura 11. Semillas de arroz



Fuente: Autoría propia

Existen diferentes variedades de arroz en el departamento del Huila, la selección de semillas de arroz de alta calidad, asegurándose que sean sanas, libres de enfermedades y plagas. Se deben remojar las semillas durante 24 a 48 horas lo cual ayudará a ablandar la cubierta de la

semilla y acelerar la germinación, después se drena el agua y se enjuaga las semillas varias veces con agua limpia.

Figura 12. Terreno con arroz chispeado



Fuente: Autoría propia

Se escarba en la tierra y se encuentran semillas por encima y debajo de la tierra, en este caso se quema la maleza y se comienza a filtrar el agua.

Figura 13. Arroz sembrado



Fuente: Autoría propia

En la figura 14. Se observa un cultivo de arroz con 5 o 10 días de sembrado, en la etapa de plántula, en promedio de cada 5 días se emite una nueva hoja.

Figura 14. Estado de plántula



Fuente: Autoría propia

Cultivos con aproximadamente 20 días, listo para el primer control de maleza.

Figura 15. Cultivo con malezas



Fuente: Autoría propia

Cultivo en la etapa de elongación del tallo, esta etapa de desarrollo marca el final de la fase vegetativa y el inicio de la fase reproductiva.

Figura 16. Elongación del tallo



Fuente: Autoría propia

Desarrollo de la panícula – embuchamiento. Las espiguillas forman, junto con el raquis, la inflorescencia que crece dentro de la vaina de la hoja bandera, causando un abultamiento llamado comúnmente “embuchamiento”

Figura 17. Desarrollo de la panícula



Fuente: Autoría propia

Se abre la panícula y se observa el arroz en formación.

Figura 18. Panícula abierta



Fuente: Autoría propia

Figura 19. Grano lechoso



Fuente: Autoría propia

En esta etapa el agricultor está a la espera que la máquina cortadora de arroz pase por el cultivo y recolecte los granos.

Figura 20. Grano maduro



Fuente: Autoría propia

En las oficinas de la organización Roa Florhuila S.a en Neiva (ORF S.A), variedades de cultivos más sembrados en el departamento del Huila y el ciclo del cultivo en días (figura 22.).

Tabla 1. Variedades más sembradas en el Huila y ciclo de cultivos en días

Variedad	Ciclo del cultivo en días
----------	---------------------------

Fedearroz 67	125 a 130
Fedearroz2000	125 a 130
Fedearroz 2020	105 a 110
Gualanday	110 a 115
Ibis	115 a 125
Sikuani	120 a 125
Satiri	105 a 110
Tucano	105 a 105
Sh27	115 a 120
Yacua	105 a 110

Fuente: Ingeniero agrónomo ORF

Gracias a diferentes especialistas en el área, ingenieros, agricultores, administrativos asociados al molino y a la unidad técnica de insumos ROA Neiva se relacionó el cuadro de costos de producción de arroz por hectárea (gravedad) (Anexo A), donde se logra segmentar los controles e insumos que normalmente se le aplican al ciclo del cultivo del arroz. Se relacionan cada uno de estos controles en la tabla 3.

Tabla 2. Controles en el ciclo del arroz

	Detalle	Unidad	Cantidad	Producto
Pre-emergencia	preergente-oxidacion	lt	3	ronstar 38
	coadyuvante	lt	0.3	pegasso
Control 1	control maleza	lt	10	propanil adama
	control maleza-sello	lt	1.3	clomazone
	control maleza	lt	0.4	tordon
	control insecticida	lt	0.5	cipermetrina
	coadyuvante	lt	0.3	pegasso
Control 2	control maleza	lt	0.35	bysoiribac 400
	control maleza-sello	lt	3	butaclor
Control primera abonada	urea	bulto	2	urea
	cloruro de potacio	bulto	1	cloruro de potacio
	totem	bulto	1	elementos secundarios
	nitroxtend + mg	bulto	2	nitroxtend + mg

Control 4 - segunda abonada	sam granular	bulto	1	sulfato de amonio
	cloruro de potacio	bulto	1	cloruro de potacio
	coljap zinc	bulto	1	sulfato de zinc

Control 5 - tercera abonada	nitroxted + mg	bulto	2	nitroxted + mg
	fase 3	bulto	2	fase 3

Cntrl 6 - tercera abonada	amidas	bulto	2	amidas
	cloruro de potacio	bulto	1	cloruro de potacio

Control 7 - aplicación de 1a espiga	coadyudante	lt	0.3	pegasso
	insecticida	lt	0.2	thiametoxan + lamda
	fungicida	lt	0.8	nativo
	fungicida	lt	1	kazugamicida
	foliar	gr	1	btimin
	corrector ph	kg	0.1	cosmoagua

Control 8 - aplicación de 1a espiga	coadyuvante	lt	0.3	pegasso
	insecticida	lt	0.2	imidacloprid
	insecticida	lt	0.3	cipermetrina
	bactericida	kg	2	mancozeb
	fungicida	lt	0.7	carbedazim
	corrector ph	kg	0.1	cosmoagua

Control 9- recolección

Fuente: ORF

Con la información recolectada, se calcula un promedio de la fecha en que se debe hacer cada uno de estos controles (Tabla 3). Se relaciona esta información para cada una de las variedades y se toma como ejemplo un cultivo en promedio de 120 días como pueden ser Sikuani, Ibis, Sh27 y un cultivo en promedio de 105 días como pueden ser Fedearroz 2020, Satiri, Tucano o Yacua.

Tabla 3. Controles

	Control	Días de sembrado del cultivo	
1	Control maleza 1	25	20
2	Primera abonada	30	27

3	Control maleza 2	30	30
4	Segunda abonada	45	40
5	Tercera abonada	60	50
6	Cuarta abonada	75	60
7	Aplicación de 1 espiga	85	70
8	Aplicación de 2 espiga	90	80
9	Recolección	120	105

Fuente: Ingeniero agrónomo ORF

Se relaciona esta información en la base de datos del aplicativo móvil, la cual va ser tomada en cuenta en cada uno de los tiempos en los que se debe realizar el respectivo control, para así enviar las diferentes notificaciones y recomendaciones de cada uno de los controles y etapas del cultivo.

5. HERRAMIENTAS UTILIZADAS

Para el desarrollo del proyecto se utilizará el entorno de programación Android Studio con el lenguaje de programación Kotlin, debido a que este es más conciso y es el lenguaje oficial de Google para Android desde el año 2017 y Firestore lo cual permitirá guardar la información en la nube. Se busca realizar un sistema de información en el cual se puedan registrar usuarios, donde se ingresen los datos del lote donde va a realizar la siembra, enviar notificaciones, alertas del cultivo, recomendaciones dinámicas, análisis fitosanitarios de plagas y enfermedades, llevando al usuario a la incorporación de buenas prácticas.

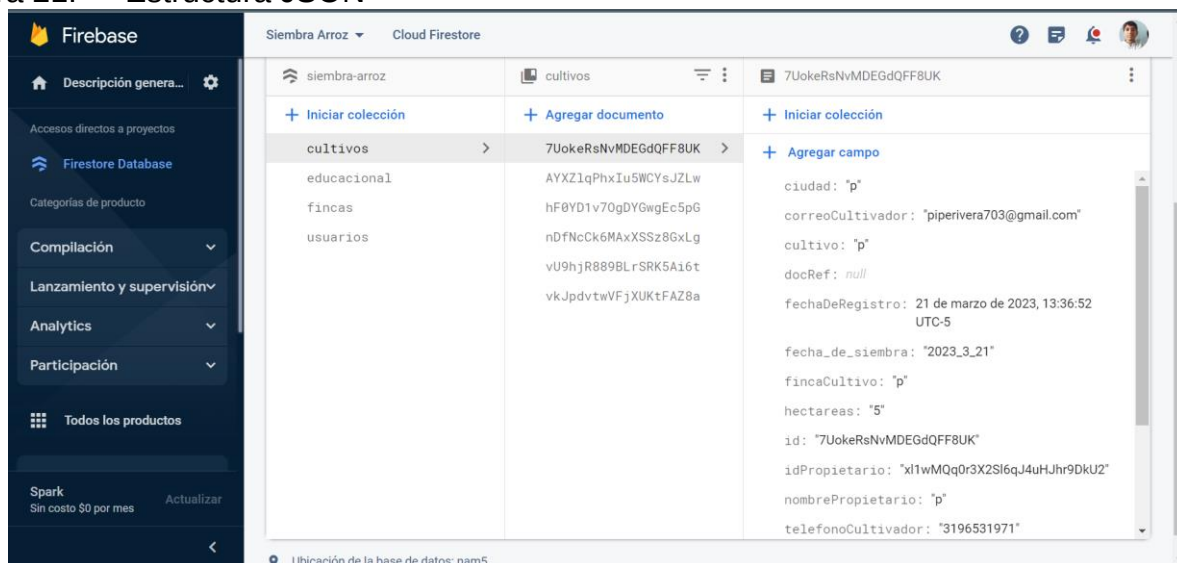
El aplicativo debe permitir al usuario registrar cada una de las fincas, el cual debe notificar al usuario en cada una de las fechas de control que se deben hacer al cultivo del arroz. En Firestore se guardará la información de cada finca y la mostrará en cada una de las alertas.

5.1 ESTRUCTURA DE LA BASE DE DATOS

En la base de datos de Firebase se almacenan los datos como objetos JSON, muy diferente a la base de datos SQL porque no se utilizan tablas ni registros, se crea un árbol JSON trabajando con claves y valores y creando nodos para tener la mejor percepción para enviar y leer los datos en la Realtime Database de Firebase.

En la figura 21. Se muestra un ejemplo de cómo está la base de datos creada para que se vayan almacenando los datos del usuario; donde se relaciona la fecha para notificar cada uno de los controles.

Figura 21. Estructura JSON



Fuente: Autoría propia

5.1.1 Creación de tablas de datos

La tabla 4. Hace referencia a los datos que debe ingresar el usuario, aquí se contiene la información básica para la creación del cultivo: Fecha de siembra (servirá para empezar el contador de los días en que deben hacerse los respectivos controles), Nombre del lote, Variedad(se relacionaron los 10 tipos de variedad), Tipo de siembra(teniendo en cuenta si es directo o chispeado), vereda, teléfono del propietario, fecha de siembra, nombre de la finca, cantidad de hectáreas sembradas(según las hectáreas, se recomendará la cantidad de insumos necesarios), ciudad y propietario.

Tabla 4. Campos por defecto del registro de cultivos

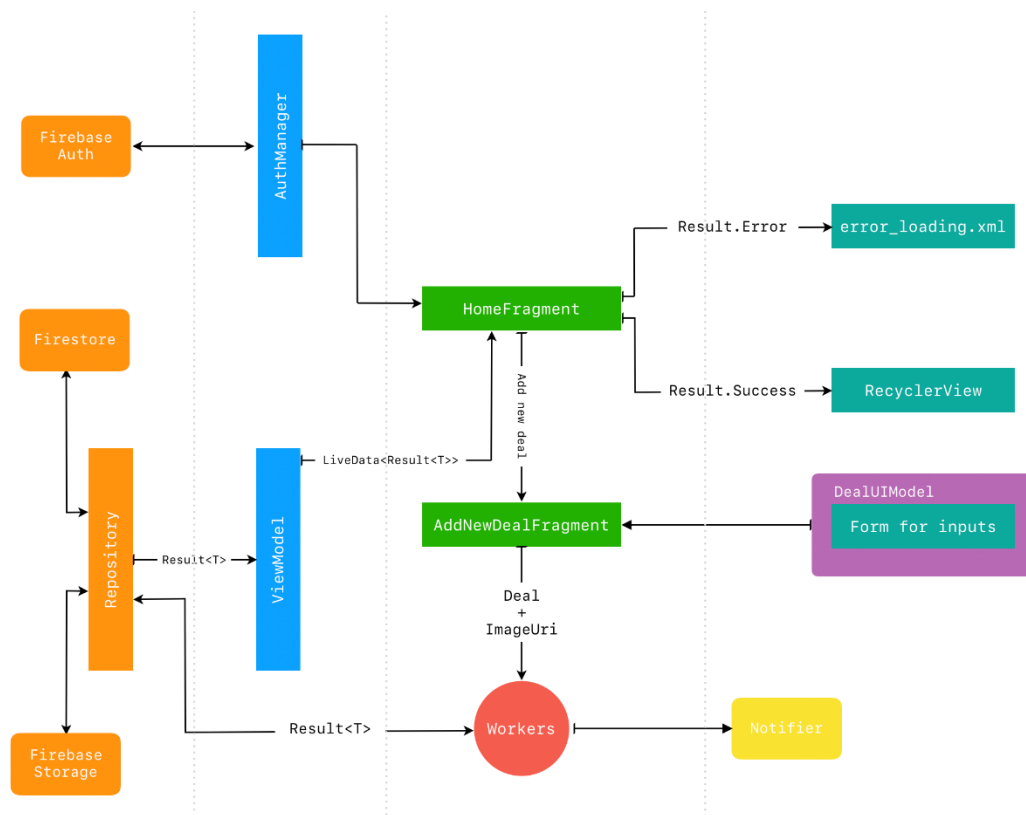
Campo:	Tipo de dato:
Nombre del lote	String
Variedad	String
Tipo de siembra	Boolean
Vereda	String
Teléfono del propietario	Int
Fecha de siembra	Int
Nombre de la finca	String
Cantidad de hectáreas sembradas	Int
Ciudad	String
Propietario	String

Fuente: Autoría propia

5.2 ARQUITECTURA DEL SISTEMA

En la figura 22, se representa un diagrama de bloques con la arquitectura del sistema MVMV y las conexiones entre sí.

Figura 22. Arquitectura del sistema



Fuente: Autoría propia

Como se puede observar, el diagrama de bloques representa cómo está conformada la conexión entre la aplicación móvil y el servidor web. La aplicación móvil se desarrolló en lenguaje Kotlin con el patrón MVVM que ofrece el entorno de trabajo Android Studio y el servidor web se desarrolló en la consola de firebase, utilizando el servicio de autenticación y almacenamiento en tiempo real, siendo nuestro backend del sistema.

En el entorno de desarrollo se muestra el diagrama de flujo donde se conecta cada bloque que conforma este patrón, de acuerdo a la petición http gestionada por cada usuario, ya sea por el método Get, del que hablaremos más adelante, también las conexiones del controlador (view) con las vistas (templates) y con el modelo, realizando consultas a la base de datos con operaciones de lectura y escritura, para así generar la respuesta http al usuario.

5.3 INTERFAZ GRÁFICA

Una interfaz de programación de aplicaciones XML (API) proporciona un conjunto de clases, métodos y funciones que se pueden utilizar para leer, escribir y manipular documentos XML. En Kotlin, estas interfaces se pueden definir utilizando bibliotecas como DOM, SAX o StAX.

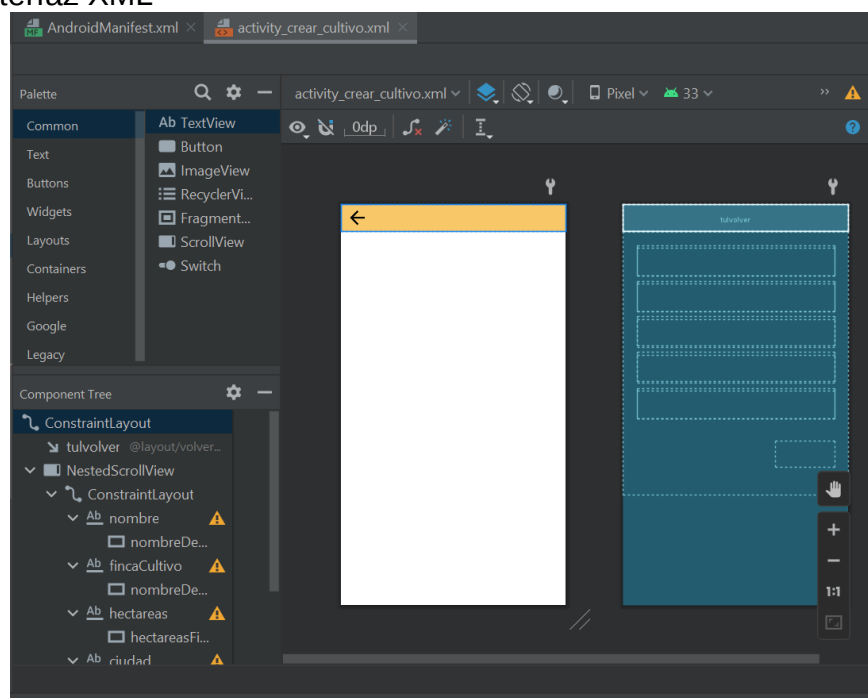
Para crear una interfaz gráfica en XML en Kotlin, necesitas seguir los siguientes pasos:

Crea un archivo XML en la carpeta "res/layout" de tu proyecto de Android Studio. Puedes hacer esto haciendo clic con el botón derecho en la carpeta "layout" y seleccionando "New > Layout Resource File". Dale un nombre a tu archivo y asegúrate de que el tipo de recurso sea "Layout".

Abre el archivo XML que acabas de crear y agrega los elementos de interfaz que necesites, como botones, textviews, edittexts, etc. Para hacer esto, puedes utilizar la vista de diseño o la vista de código. Asigna un ID único a cada elemento de interfaz. Esto te permitirá referirte a ellos desde el código Kotlin. En tu archivo Kotlin, declara las variables para cada elemento de interfaz que necesites utilizar.

En la fig. 23 Se tienen la vista de diseño, en la cual se crea la interfaz desde cero para el aplicativo móvil.

Figura 23. Interfaz XML



Fuente: Autoría propia

5.4 LOGO

Se diseñó el siguiente logo para representar la aplicación móvil en el programa. El logo contiene el nombre del proyecto en "Siembra Arroz" y la figura de una espiga de arroz para simbolizar la planta del arroz.

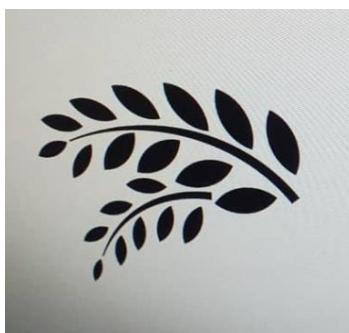
Figura 24. Desarrollo del logo



Fuente: Autoría propia

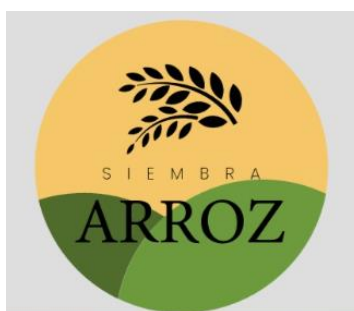
Se tuvo en cuenta que se debe hacer en formato SVG (Scalable Vector Graphics) la identidad de la app, la cual se verá en miniatura y en cualquier tamaño de manera profesional.

Figura 25. Logo en miniatura.



Fuente: Autoría propia

Figura 26. Logo



Fuente: Autoría propia

6. DESARROLLO DEL APLICATIVO MÓVIL

6.1 CONFIGURACIÓN

6.1.1 Build.grade

Se importan las librerías del proyecto, tales como retrofit para realizar el consumo del API mediante http, gson para convertir json a objetos y viceversa y demás librerías predeterminadas de java para Android Studio. Además de recalcar que la aplicación es desarrollada en la versión 33 de SDK.

6.1.2 AndroidManifest.xml

Se realiza la configuración principal del proyecto, como la declaración de las Activities (Login, registro de cultivos y la actividad de diagnóstico en las diferentes etapas, se configuran los permisos de acceso a internet y además se ajusta el nombre y logo de la aplicación.

6.2 CREACION DE MODULOS EN KOTLIN

Contienen la lógica para el manejo de las peticiones del sistema, y la lógica utilizada para la visualización de las vistas y demás procesos que se realicen en éstas.

6.2.1 Módulo de login

Para este fragmento se utiliza Firebase Authentication de la librería de autenticación de Google que permite obtener el formulario de Login, simplificando mucho más este proceso y dando seguridad a nuestra app. En el anexo B se encontrará todo el código del Fragmento de Login.

Figura 27. Módulo de Login

```
private fun crearRegistroUsuarioDB(){  
    val correo = auth.currentUser?.email  
    val nameUser = auth.currentUser?.displayName  
    val date = Date()  
    val datos = Usuario(auth.uid.toString(), correo, nameUser, date)  
    db.collection(Constants.USUARIO).document(auth.uid.toString()).set(datos)
```

Fuente: Autoría propia

6.2.2 Módulo de registro de cultivos

Maneja la lógica proveniente de la vista Registro, valida los datos del formulario, los almacena en formato Json y se comunica con la base de datos para realizar el guardado de los datos, además de realizar la renderización del código en Firebase para visualizar el contenido en la vista.

Figura 28. Módulo de registro de cultivos

```
@SuppressLint("SetTextI18n", "SimpleDateFormat")
override fun onBindViewHolder(holder: ViewHolder, position: Int) {
    val binding = ItemCardCultivoBinding.bind(holder.layout)
    val post = dataset[position]

    val sdf = SimpleDateFormat(pattern: "dd/M/yyyy hh:mm a")

    try {
        binding.id.text = "Id cultivo: ${post.id}"
        binding.nombre.text = "Nombre de la variedad: ${post.cultivo}"
        binding.propietario.text = "Propietario: ${post.nombrePropietario}"
        binding.finca.text = "Finca: ${post.fincaCultivo}"
        binding.hectareas.text = "Hectareas: ${post.hectareas}"
        binding.ciudad.text = "Ciudad: ${post.ciudad}"
        binding.fecha.text = "Fecha de registro: ${post.fechaDeRegistro?.let { sdf.format(it) }}"
    } catch (e: Exception) {
        e.printStackTrace()
    }
}
```

Fuente: Autoría propia

6.2.3 Módulo del clima

Este cálculo se hace usando el API de Weather API y no requiere muchas líneas de código, simplemente se hace el llamado al Get correspondiente y con los parámetros requeridos como se muestra en el siguiente fragmento de código.

Figura 29. Módulo del clima

```
import com.example.siembrarroz.modelosedatos.Clima
import com.example.siembrarroz.modelosedatos.Main
import retrofit2.Response
import retrofit2.http.*

interface ClimaService {

    @GET()
    suspend fun apiClima(@Url url: String, @Query("lat") lat: Double?,
        @Query("lon") lon: Double?, @Query("apiKey") apiKey: String?): Response<Clima>
}
```

Fuente: Autoría propia

6.2.4 Módulo de notificaciones

El número de días para las notificaciones se trabajará en milisegundos a partir de la hora y fecha donde se registre el cultivo.

Figura 30. Cálculo de notificaciones

```

} else if(diaD < diaC && variedadArroz == "Sikuani"){
    val resta = diaC - diaD
    val calculo = 23 - resta
    val resultado = calculo * 86400000

    milisegundos = resultado.toLong()

}

val intent = Intent(applicationContext, AlarmNotification::class.java)
intent.putExtra( name: "idCultivo", idCultivo)
intent.putExtra( name: "idEtapa",
    value: "Ya puedes prepararte para el primer control de maleza de tu lote de cultivo: $nombre de la variedad: $variedadArroz")
val pendingIntent = PendingIntent.getBroadcast(
    applicationContext,
    AlarmNotification.NOTIFICATION_ID,
    intent,
    flags: PendingIntent.FLAG_IMMUTABLE or PendingIntent.FLAG_UPDATE_CURRENT
)

val alarmManager = getSystemService(Context.ALARM_SERVICE) as AlarmManager
alarmManager.setExact(AlarmManager.RTC_WAKEUP, triggerAtMillis: Calendar.getInstance().timeInMillis + milisegundos, pendingIntent)
}

```

Fuente: Autoría propia

6.2.5 Broadcast receiver

Es la manera en que el aplicativo móvil se comunica con el teléfono y envía las diferentes notificaciones en cada uno de los controles.

Figura 31. Broadcast receiver

```

private fun createSimpleNotification(context: Context) {
    val intent = Intent(context, MainActivity::class.java).apply { this: Intent
        flags = Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK
    }

    val flag = if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) PendingIntent.FLAG_IMMUTABLE else 0
    val pendingIntent: PendingIntent = PendingIntent.getActivity(context, requestCode: 0, intent, flag)

    val notification = NotificationCompat.Builder(context, MainActivity.MY_CHANNEL_ID)
        .setSmallIcon(R.drawable.svg_espiga)
        .setContentTitle(titulo)
        .setContentText(contexto)
        .setStyle(
            NotificationCompat.BigTextStyle()
                .bigText(contexto)
        )
        .setContentIntent(pendingIntent)
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)
        .build()

    val manager = context.getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
    manager.notify(NOTIFICATION_ID, notification)
}

```

Fuente: Autoría propia

6.3 VISTAS

Se realizó la creación de 3 vistas principales para el aplicativo móvil, comenzando con la vista de Login, la cual tiene un único botón, el cual nos permite entrar o registrarse con un correo electrónico de Google, una vista de registro de cultivos con su respectivo formulario y botón para guardar la información. Una segunda vista con los diferentes datos climatológicas y un último fragmento con las recomendaciones con las labores recomendadas en cada una de las etapas.

6.3.1 Login

Se accede al aplicativo por medio de un correo de Google, para así poder conectarse a Firebase y guardar la respectiva información en la nube.

Figura 32. Login vista

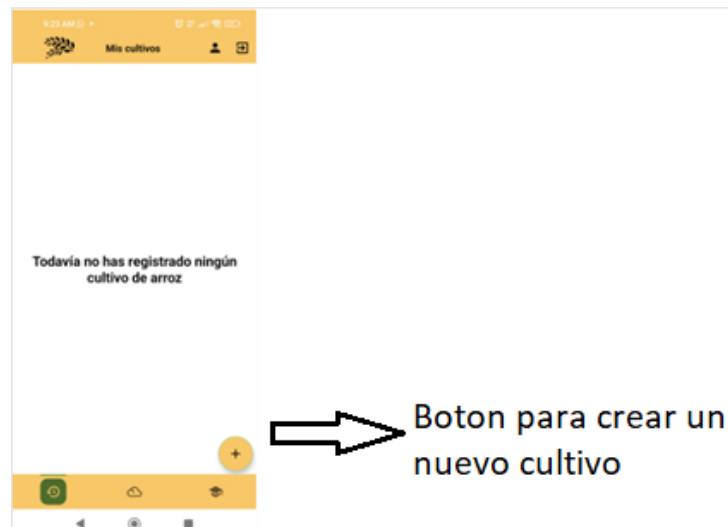


Fuente: Autoría propia

6.3.2 Registro de cultivos

En este fragmento es donde se registran los cultivos, es el menú principal el cual tiene un botón circular para crear un nuevo cultivo (Figura 33.), en este fragmento aparecerán todos los cultivos registrados (Figura 34).

Figura 33. Vista Fragmento principal



Fuente: Autoría propia

Figura 34. Vista Cultivo registrado



Fuente: Autoría propia

6.3.3 Registrar un cultivo

Al dar clic en el botón + del fragmento principal, aparecerá un nuevo fragmento donde el usuario registrará los datos del cultivo (Figura 35.), El usuario tendrá que registrar el nombre del lote, elegir entre los diferentes tipos de variedades (Figura 36), tipo de siembra (Figura 37.), vereda, teléfono del propietario, fecha de siembra, nombre de la finca, cantidad de hectáreas sembradas, ciudad y propietario

Figura 35. Vista registro del cultivo

9:34 AM

← Publicar nuevo cultivo

Nombre del lote

Variedad
Fedearroz 67

Tipo de siembra
Directo

Vereda

Teléfono del propietario

SELECCIONAR FECHA DE SIEMBRA

Nombre de la finca

Cantidad de hectareas sembradas

Ciudad

Propietario

PUBLICAR

Fuente: Autoría propia

Es la opción de variedad aparecerá una lista desplegable de los diferentes tipos de variedades de arroz más sembrados en el departamento, ligados cada uno de estos con el ciclo del cultivo en días.

Figura 36. Vista tipos de variedades de arroz

9:41 AM

← Publicar nuevo cultivo

Nombre del lote

Variedad
Fedearroz 67

Fedearroz 67

Fedearroz 2000

Fedearroz 2020

Gualanday

Ibis

Sikuani

Satiri

Tucano

Sh27

Yacua

Figura 37. Vista tipo de siembra

9:41 AM

← Publicar nuevo cultivo

Nombre del lote

Variedad
Fedearroz 67

Tipo de siembra
Directo

Directo
Chispeado

SELECCIONAR FECHA DE SIEMBRA

Nombre de la finca

Cantidad de hectareas sembradas

Ciudad

Propietario

Fuente: Autoría propia

6.3.4 Clima

En este fragmento se encontrará la información respecto a la latitud, longitud, clima, con el fin de que el agricultor pueda tener en cuenta esta información.

Figura 38. Vista datos del clima

9:35 AM

El clima

DATOS DEL CLIMA

Latitud: 2.9189
Longitud: -75.2688
Temperatura: 25.90 °C
Humedad: 78 %

Fuente: Autoría propia

6.3.5 Recomendaciones agrícolas

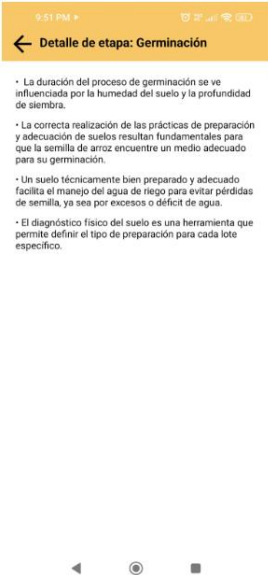
En este fragmento se dividen en 9 contenedores, se relaciona la información a cada una de las 9 etapas con las labores recomendadas.

Figura 39. Vista recomendaciones



Fuente: Autoría propia

Figura 40. Vista recomendaciones etapa de germinación



Fuente: Autoría propia

7. CONCLUSIONES

Luego de realizar la respectiva investigación acerca del ciclo biológico del arroz y las necesidades que requiere el sector arrocerero del departamento del Huila, y enfocarla al pequeño agricultor, se definió que la aplicación debería llevar elementos puntuales en su contenido y estructura, tal como teoría básica acerca del proceso de cultivo, nuevas y buenas prácticas, control de cultivos con fechas de cada labor y hasta costos para el uso de insumos.

Indagando acerca de los diferentes agricultores, se realiza un aplicativo móvil accesible, totalmente gratuito para que el pequeño agricultor potencialice sus cultivos y crezca en el aspecto económico y social, por esta razón dentro de los temas que se desarrollaron en la aplicación están las diferentes técnicas y buenas prácticas y en general como desarrollar desde la siembra hasta la recolección.

Con este aplicativo se dan sugerencias, no debe ser tomado como recomendaciones exactas, ya que pueden existir diferentes factores que puedan afectar el cultivo y no se puede determinar con exactitud o dar una fecha exacta para cada uno de los controles sugeridos, para lo cual requiere una supervisión constante. Se pretende que sea una ayuda, donde se pueda llevar con exactitud cada uno de los tiempos del cultivo.

En el desarrollo de la base de datos se pensó en una plataforma en la nube que contara con una estructura no relacional y disponible de manera gratuita para los requisitos del aplicativo. Por lo tanto, aprovechando del aprendizaje obtenido en el área de desarrollo de aplicaciones, en donde se conoció y se decidió la plataforma Firebase de Google, como la herramienta para implementar nuestra base de datos, esto gracias a las ventajas y servicios que nos permitió ofrecer para el aplicativo, tales como: el alojamiento y disposición de los datos e información en tiempo real, además de un sistema de autenticación de los usuarios que brinda una mayor seguridad y protección de los datos. Para hacer de su desarrollo algo más sencillo se utilizó la documentación introductoria y tutoriales informativos con los que cuenta la plataforma de Google para Firebase.

8. RECOMENDACIONES

El aplicativo se hizo para que se pudiera masificar en cualquier otro cultivo, la arquitectura desarrollada permite aumentar sus funcionalidades fácilmente.

Se debe tener en cuenta que existen diferentes factores climatológicos, condiciones del terreno, plagas, calidad de los granos que pueden afectar el debido desarrollo del cultivo del arroz, por lo cual el aplicativo no sería capaz de determinar.

Ya que se ha desarrollado la aplicación para el sistema operativo Android, sería de gran ayuda y más aún para poder lograr un alcance a la mayoría de usuarios, se debe ampliar el diseño de la aplicación para el sistema operativo a IOS base para así poder utilizar la app en diferentes plataformas.

Se pueden realizar varias mejoras que permitan tener un control más preciso, se había pensado conectar el aplicativo también con ORF para que expertos puedan también seguir cada uno de los cultivos de los agricultores, uso de sensores, cámaras, los cuales envíen información en tiempo real y se pueda monitorear de manera remota. Esto no se realizó en este proyecto debido a que se buscaba desarrollar un aplicativo que no generara ningún costo y así todos los agricultores pudieran beneficiarse.

9. BIBLIOGRAFÍA

Firestore. Aplicaciones web. Firebase. 2014. [Consultado en 2022]. Recuperado de [\[https://firebase.google.com/\]](https://firebase.google.com/)

FRANK, D. Et. (en línea) Jualiaca, Perú. Monografias. 2014. (Sistema Operativo Android). (12 de Junio de 2019). Obtenido de Monografias. Recuperado de: [\[https://www.monografias.com/trabajos101/sistema-operativo-android/sistema-operativo-android\]](https://www.monografias.com/trabajos101/sistema-operativo-android/sistema-operativo-android)

KOTLIN. Lenguaje de programación Kotlin. JetBrains. Kotlinlang.org. 2016. [Consultado en 2022] Recuperado de [\[https://kotlinlang.org/\]](https://kotlinlang.org/)

MINISTERIO DE AGRICULTURA Y DESARROLLO RURAL EN COLOMBIA. Informe sobre producción de arroz en Colombia. 2020.

MVVM. Android: ¿Qué es MVC, MVP y MVVM? [en línea]. PyM. 2017. Recuperado de: [\[https://programacionymas.com/blog/android-mvc-mvp-mvvm\]](https://programacionymas.com/blog/android-mvc-mvp-mvvm)

Olmos, S. Cartilla fisiológica del arroz. Cátedra de Cultivos II Facultad de Ciencias Agrarias, UNNE. Corrientes, 2006 [consultado en 2018] tomado de [\[https://www.acpaarrozcorrientes.org.ar/academico/Apunte-MORFOLOGIA.pdf\]](https://www.acpaarrozcorrientes.org.ar/academico/Apunte-MORFOLOGIA.pdf)

URRY, LISA. Et. Campbell Biology. 11th edition. College, Oakland, California, 2016. ISBN-13: 978-0134093413, ISBN-10: 0134093410

10. ANEXOS

Anexo A. Cuadro de costos de producción de arroz por hectárea (gravedad)

ORF S.A. UNIDAD TECNICA INSUMOS ROA NEIVA ZONA HUILA NORTE Fecha Actualización: ENERO 26/2023							
CUADRO DE COSTOS DE PRODUCCION DE ARROZ POR HECTAREA (GRAVEDAD)							
	DETALLE	UNIDAD	CANTIDAD	PRODUCTO	VR. UNITARIO	VR. TOTAL	%
A. COSTOS DIRECTOS							
1. PREPARACIÓN DEL TERRENO							
	3 Pasos de Rastra	Ha	3.0		\$ 85,000	\$ 255,000	1.98%
	1 Pasos Land Plane	Ha	1.0		\$ 110,000	\$ 110,000	0.85%
	Caballoneo - Taipa y Laser	Ha	1.0		\$ 110,000	\$ 110,000	0.85%
	SUBTOTAL					\$ 475,000	3.69%
2. SIEMBRA							
	Semilla	Kg	200.0	\$ 238,372	\$ 4,767	\$ 953,488	7.40%
	Siembra - Sembradora	Ha	1.0		\$ 120,000	\$ 120,000	0.93%
	SUBTOTAL					\$ 1,073,488	8.34%
3. CONTROL DE MALEZAS							
	Quemado	Lt	0.30	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Glifosato	Lt	3.0	TouchDown	\$ 63,169	\$ 189,507	1.47%
	Glifosato Granulado	Kg	0.0	Glifosato arrasador 757 X 1 KI	\$ 64,153	\$ 0	0.00%
	Acompañante	Lt	0.6	Profamina Amino 720	\$ 24,481	\$ 14,689	0.11%
	Corrector PH	Kg	0.1	CosmoAgua	\$ 40,110	\$ 4,011	0.03%
	Paqueno	Lt	0.30	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Glifosato	Lt	4.0	Round Up Activo	\$ 27,500	\$ 110,000	0.85%
	Corrector PH	Kg	0.1	CosmoAgua	\$ 40,110	\$ 4,011	0.03%
	Pre-Emergente	Lt	3.0	Ronstar 38	\$ 110,600	\$ 331,800	2.58%
	Coadyuvante	Lt	0.30	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Post-Tenjurado	Lt	10.0	Propanil Adama	\$ 24,986	\$ 249,860	1.94%
	Control de Maleza - Sello	Lt	1.3	Clomazone	\$ 41,090	\$ 53,417	0.41%
	Control de Maleza	Lt	0.4	Tordon	\$ 34,420	\$ 13,768	0.11%
	Control insecticida	Lt	0.5	Opermetrina	\$ 25,574	\$ 12,787	0.10%
	Coadyuvante	Lt	0.30	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Post-Mediana	Lt	0.35	Byspiribac 400	\$ 206,148	\$ 72,152	0.56%
	Control de Maleza - Sello	Lt	3.0	Butaclor	\$ 44,535	\$ 133,605	1.04%
	Aplicaciones terrestres	Aplicación	5.0		\$ 50,000	\$ 250,000	1.94%
	Despalillo	Jornal	5.0		\$ 50,000	\$ 250,000	1.94%
	SUBTOTAL					\$ 2,719,114	13.35%
4. FERTILIZACIÓN							
	Primer abonada	Bulto	2.0	Fosfato monoamónico	\$ 238,587	\$ 477,174	3.70%
	Nutrientes	Bulto	1.0	Elementos Menores	\$ 128,962	\$ 128,962	1.00%
	Cloruro de Potasio	Bulto	0.5	Cloruro de Potasio	\$ 220,380	\$ 110,190	0.86%
	Urea	Bulto	2.0	Urea	\$ 204,620	\$ 409,240	3.18%
	Cloruro de Potasio	Bulto	1.0	Cloruro de Potasio	\$ 220,380	\$ 220,380	1.71%
	Tolam	Bulto	1.0	Elementos Secundarios	\$ 61,957	\$ 61,957	0.48%
	Nitroxeno + Mg	Bulto	2.0	Nitroxeno + Mg	\$ 228,768	\$ 457,536	3.55%
	SAM Granular	Bulto	1.0	Sulfato de Amonio	\$ 111,538	\$ 111,538	0.87%
	Cloruro de Potasio	Bulto	1.0	Cloruro de Potasio	\$ 220,380	\$ 220,380	1.71%
	Calap Zinc	Bulto	1.0	Sulfato de Zinc	\$ 222,478	\$ 222,478	1.73%
	Nitroxeno + Mg	Bulto	2.0	Nitroxeno + Mg	\$ 228,768	\$ 457,536	3.55%
	Fase 3	Bulto	2.0	Fase 3	\$ 193,207	\$ 386,414	3.00%
	Amidas	Bulto	2.0	Amidas	\$ 252,628	\$ 505,256	3.92%
	Cloruro de Potasio	Bulto	1.0	Cloruro de Potasio	\$ 220,380	\$ 220,380	1.71%
	Aplicación de Fertilizantes	Aplicación	5.0		\$ 50,000	\$ 250,000	1.94%
	SUBTOTAL					\$ 4,239,421	32.92%
5. CONTROL SANITARIO							
	Aplicación preventiva - 45 Dias	Lt	0.3	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Fungicida	Lt	0.3	Difenoconazole	\$ 115,717	\$ 34,715	0.27%
	Fungicida	40 gr	2.00	Italy	\$ 17,765	\$ 35,530	0.28%
	Insecticida	Lt	0.3	Eigun	\$ 178,962	\$ 53,689	0.42%
	Corrector PH	Kg	0.1	CosmoAgua	\$ 40,110	\$ 4,011	0.03%
	Aplicación preventiva - 65 Dias	Lt	0.3	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Insecticida	Lt	0.3	Fulminator	\$ 73,202	\$ 21,961	0.17%

	Fungicida	Lt	0.6	Azoxystrobin	\$ 119,344	\$ 71,606	0.56%
	Fungicida	Kg	0.30	Tricyclazole	\$ 59,016	\$ 17,705	0.14%
	Corrector PH	Kg	0.1	CosmoAgua	\$ 40,110	\$ 4,011	0.03%
Aplicación 1a Espiga	Coadyuvante	Lt	0.3	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Insecticida	Lt	0.2	Thiametoxan + Lambda	\$ 60,109	\$ 12,022	0.09%
	Fungicida	Lt	0.8	Notivo	\$ 140,360	\$ 112,288	0.87%
	Fungicida	Lt	1.0	Kazugamicina	\$ 16,000	\$ 16,000	0.12%
	Foliar	Gr	1.00	Brimin	\$ 12,262	\$ 12,262	0.10%
	Corrector PH	Kg	0.1	CosmoAgua	\$ 40,110	\$ 4,011	0.03%
Aplicación 2a Espiga	Coadyuvante	Lt	0.3	Pegasso	\$ 24,590	\$ 7,377	0.06%
	Insecticida	Lt	0.2	Imidacloprid	\$ 62,025	\$ 12,405	0.10%
	Insecticida	Lt	0.3	Cipermetrina	\$ 25,574	\$ 7,672	0.06%
	Bactericida	Kg	2.0	Mancozeb	\$ 25,027	\$ 50,054	0.39%
	Fungicida	Lt	0.7	Carbendazim	\$ 27,869	\$ 19,508	0.15%
	Corrector PH	Kg	0.1	CosmoAgua	\$ 40,110	\$ 4,011	0.03%
Aplicaciones terrestres		Aplicación	4.0		\$ 28,000	\$ 112,000	0.87%
Aplicaciones Aereas		Vuelo	1.0			\$ 45,000	0.35%
SUBTOTAL						\$ 725,424	5.63%
6. COSECHA							
Recolección a granel + Zorreo		Bulto	100.0		\$ 4,000	\$ 400,000	3.11%
Regador + Instalé		Ha	1.0		\$ 400,000	\$ 400,000	3.11%
Transporte		Ton	6.87		\$ 40,000	\$ 274,800	2.13%
SUBTOTAL						\$ 1,074,800	8.35%
7. OTROS							
Limpieza de acequias		Jornal	1.0		\$ 45,000	\$ 45,000	0.35%
SUBTOTAL						\$ 45,000	0.35%
8. COSTOS INDIRECTOS							
8. GASTOS GENERALES							
Arriendo		Ha	1.0		\$ 1,500,000	\$ 1,500,000	11.65%
Agua		Ha	1.0		\$ 500,000	\$ 500,000	3.88%
Asistencia técnica		Ha	1.0		\$ 40,000	\$ 40,000	0.31%
Financiación 2.2%	9,352,247.8	Mes	4.5		\$ 205,749	\$ 925,873	7.19%
Administración 4 %	9,352,247.8				0.04	\$ 374,090	2.90%
Imprevistos 2%(Aliment,transp,implemt)	9,352,247.8				0.02	\$ 187,045	1.45%
SUBTOTAL						\$ 3,527,007	27.39%
TOTAL COSTOS DIRECTOS						\$ 9,352,248	72.61%
TOTAL COSTOS INDIRECTOS						\$ 3,527,007	27.39%
TOTAL COSTOS DEL CULTIVO						\$ 12,879,255	100.00%
INGRESOS		Bulto	110.0			\$ 13,636,700	
RENTABILIDAD		%	1.0			\$ 88%	
GANANCIA						\$ 757,445	
PUNTO DE EQUILIBRIO		Bulto	103.9	51.9			

PRECIO CARGA	\$ 263,000
(-) FOMENTO - 0.8%	\$ 1,265
(-) RETENCIÓN - 1.8%	\$ 3,795
(-) DESCUENTO DE LABORATORIO - 1.8%	
(-) BOLSA AGROPECUARIA - 0.4348%	\$ -
(-) MODALIDAD DE PAGO - 3.7%	\$ -
TOTAL PRECIO CARGA NETO	\$ 247,940
PRECIO BULTO NETO	\$ 123,970

2a. Alvaro campo toru, 3112575731.

Arda Patricia Reyes 3165340916

```

package com.example.siembraarroz.login

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.core.content.ContextCompat
import com.example.siembraarroz.R
import com.example.siembraarroz.clases.Constantes
import com.example.siembraarroz.databinding.ActivityMainBinding
import com.example.siembraarroz.home.fragments.activity.HomeActivity
import com.example.siembraarroz.modelosedatos.Usuario
import com.google.android.gms.auth.api.signin.GoogleSignIn
import com.google.android.gms.auth.api.signin.GoogleSignInClient
import com.google.android.gms.auth.api.signin.GoogleSignInOptions
import com.google.android.gms.common.api.ApiException
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.GoogleAuthProvider
import com.google.firebase.firestore.FirebaseFirestore
import java.util.*

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private val auth = FirebaseAuth.getInstance()
    private val db = FirebaseFirestore.getInstance()
    lateinit var googleSignInClient: GoogleSignInClient
    private lateinit var fusedLocationProviderClient: FusedLocationProviderClient

    private val REQUEST_CODE = 100

    companion object {
        const val MY_CHANNEL_ID = "myChannel"
        var latitud = 0.0
        var longitud = 0.0
    }

    private val laun =

```

```

registerForActivityResult(ActivityResultContracts.StartActivityForResult()){ act->
    if (act.resultCode == Activity.RESULT_OK) {
        val data = act.data
        val task = GoogleSignIn.getSignedInAccountFromIntent(data)
        try {
            // Google Sign In was successful, authenticate with Firebase
            val account = task.getResult(ApiException::class.java)!!
            //Log.d("TAG", "firebaseAuthWithGoogle:" + account.id)
            firebaseAuthWithGoogle(account.idToken!!)

        } catch (e: ApiException) {
            // Google Sign In failed, update UI appropriately
            //Log.w("TAG", "Google sign in failed", e)
            // ...
        }
    }
}

override fun onCreate(savedInstanceState: Bundle?) {
    //splashScreen
    setTheme(R.style.Theme_SiembraArroz)
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)
    fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this)

    obtenerUbicacion()
    validarSesion()
    // Configure Google Sign In
    val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.default_web_client_id))
        .requestEmail()
        .build()

    googleSignInClient = GoogleSignIn.getClient(this, gso)

    //boton de google
    binding.botonGog.setOnClickListener {
        signIn()
    }
}

private fun obtenerUbicacion(){
    if(ContextCompat.checkSelfPermission(this,
    Manifest.permission.ACCESS_COARSE_LOCATION) ==

```

```

        PackageManager.PERMISSION_GRANTED){
fusedLocationProviderClient.lastLocation.addOnSuccessListener { locacion ->
    if(locacion != null){
        latitud = locacion.latitude
        longitud = locacion.longitude
    }
}

    } else{
        solicitarPermiso()
    }
}

    private fun solicitarPermiso(){
        when{
            ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_COARSE_LOCATION) ==
                PackageManager.PERMISSION_GRANTED ->{
                obtenerUbicacion()
            }
        }
    }

    shouldShowRequestPermissionRationale(Manifest.permission.ACCESS_COARSE_LOCATION)->{
        Toast.makeText(applicationContext,
            "El permiso de ubicación fue rechazado anteriormente en este dispositivo, ve a
            los ajustes para habilitarlo",
            Toast.LENGTH_LONG).show()
    } else -> {

        requestPermissions(arrayOf(Manifest.permission.ACCESS_COARSE_LOCATION),
            REQUEST_CODE)
    }
}

    override fun onRequestPermissionsResult(
        requestCode: Int,
        permissions: Array<out String>,
        grantResults: IntArray
    ) {
        when(requestCode){
            REQUEST_CODE->{
                if(grantResults.isNotEmpty() && grantResults[0] ==
                    PackageManager.PERMISSION_GRANTED){
                    obtenerUbicacion()
                }
            } else->{
                super.onRequestPermissionsResult(requestCode, permissions, grantResults)
            }
        }
    }
}

```

```

    }
    }

}

private fun validarSesion(){
    if (auth.currentUser?.uid != null){
        starApp()
    }
}

//iniciar sesion con google
private fun signIn() {
//con esto se cierra la sesión por defecto del antiguo usuario de google
    googleSignInClient.signOut()
    val signInIntent = googleSignInClient.signInIntent
    laun.launch(signInIntent)

}

private fun firebaseAuthWithGoogle(idToken: String) {
    val credential = GoogleAuthProvider.getCredential(idToken, null)
    auth.signInWithCredential(credential)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                buscarUsuario()
                Toast.makeText(
                    this,
                    "Iniciando...",
                    Toast.LENGTH_LONG
                ).show()
                starApp()

            } else {

                Toast.makeText(this, "Authentication Failed.", Toast.LENGTH_SHORT).show()

            }

            // ...
        }
        .addOnFailureListener {
            Toast.makeText(
                this,
                "Error",
                Toast.LENGTH_SHORT
            ).show()
        }
    }
}

```

```

    }
    private fun starApp() {
        val intent = Intent(this, HomeActivity::class.java)
        startActivity(intent)
        finish()
    }

    private fun buscarUsuario(){
        auth.currentUser?.uid.let {
            db.collection(Constants.USUARIO).document(auth.uid.toString()).get()
                .addOnSuccessListener {
                    if (!it.exists()){
                        crearRegistroUsuarioDB()
                    }
                }
        }
    }

    private fun crearRegistroUsuarioDB(){
        val correo = auth.currentUser?.email
        val nameUser = auth.currentUser?.displayName
        val date = Date()
        val datos = Usuario(auth.uid.toString(), correo, nameUser, date)
        db.collection(Constants.USUARIO).document(auth.uid.toString()).set(datos)
    }
}

```