

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

Tema: “Solución Informática para la gestión del agua en cultivos bajo invernadero”

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

AUTOR: Mallamas Mites David Anderson.

TUTOR: Ing. Yandún Velasteguí Marco Antonio MSc.

Tulcán, 2025.

CERTIFICADO DEL TUTOR

Certifico que el estudiante Mallamas Mites David Anderson con el número de cédula 0401844170 ha desarrollado el Trabajo de Integración Curricular: "Solución Informática para la gestión del agua en cultivos bajo invernadero"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva

Ing. Yandún Velasteguí Marco Antonio MSc.

TUTOR

Tulcán, octubre de 2025

AUTORÍA DE TRABAJO

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Mallamas Mites David Anderson con cédula de identidad número 0401844170 respectivamente declaro que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.



Mallamas Mites David Anderson

AUTOR

Tulcán, octubre de 2025

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo Mallamas Mites David Anderson declaro ser autor de los criterios emitidos en el Trabajo de Integración Curricular: "Solución Informática para la gestión del agua en cultivos bajo invernadero" y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.



Mallamas Mites David Anderson

AUTOR

Tulcán, octubre de 2025

AGRADECIMIENTO

A mi compañera de vida, Eliana Cerón.

Este trabajo ha llevado mucho esfuerzo, que de a poco se lo ha logrado poder desarrollar, agradeciendo a mi familia por tanto apoyo que me brindaron, por la motivación que me supieron brindar de tal manera que me hicieron centrarme en mis objetivos.

Expresando un agradecimiento eterno a mi tutor de tesis, MSc. Marco Yandún gracias por ser mi mentor para el desarrollo de esta investigación, como también agradeciendo a mis docentes que, tras este trayecto, se logró adquirir conocimientos en clases, no solo académicos sino también valores éticos como futuros profesionales.

Mi reconocimiento a la Unidad Educativa Jorge Martínez Acosta por abrirme sus puertas desde el principio a fin, brindándome todas las facilidades necesarias para acceder a su invernadero.

A mis compañeros de carrera, por los momentos compartidos y el apoyo mutuo durante estos años de formación académica.

Y finalmente agradecer a la Universidad Politécnica Estatal del Carchi, por la acogida que me brindó como mi segundo hogar.

DEDICATORIA

Se lo dedico a mis padres, Aníbal Mallamas y Rocío Mites, por su sacrificio, amor, acompañamiento y confianza inquebrantable. Gracias por inculcarme valores sólidos desde pequeño y enseñarme a perseverar con humildad y esfuerzo; todo lo que soy hoy se lo debo a ustedes.

A mis docentes de la UPEC, por compartir su conocimiento y servir de inspiración a lo largo de mi formación universitaria. Sus enseñanzas fueron la semilla que germinó en mí el deseo de aprender, crecer y aportar al desarrollo de nuestra sociedad.

A mis familiares, por estar siempre presentes, brindando su apoyo y cariño en los momentos más importantes de mi vida.

ÍNDICE

RESUMEN	15
ABSTRACT	16
INTRODUCCIÓN	17
I. EL PROBLEMA	18
1.1. PLANTEAMIENTO DEL PROBLEMA	18
1.2. FORMULACIÓN DEL PROBLEMA	19
1.3. JUSTIFICACIÓN	19
1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN	20
1.4.1. Objetivo General	20
1.4.2. Objetivos Específicos	20
1.4.3. Preguntas de Investigación	20
II. FUNDAMENTACIÓN TEÓRICA	22
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	22
2.2. MARCO TEÓRICO	24
2.2.1. Solución Informática.....	24
2.2.2. Fases fenológicas del cultivo bajo invernadero	25
2.2.3. Tecnologías Habilitadoras Clave.....	27
2.2.4. Inteligencia Artificial Aplicada	27
2.2.5. Visión Artificial.....	31
2.2.6. Redes Neuronales	32
2.2.7. Predicción de necesidades hídricas.....	33
2.2.8. Internet de las cosas (IoT).....	33
2.2.9. Sistemas Multi-Agente	33
2.2.10. Microcontroladores (ESP32)	35
2.2.11. Sensores para sistema de riego	36
2.2.12. Metodologías de Desarrollo.....	37

2.2.13. Metodología Ágil	37
2.2.14. Adaptación de PXP al Contexto Agrícola-Tecnológico	39
2.2.15. Herramientas de Desarrollo	41
2.2.16. Lenguajes de Programación	41
2.2.17. Lenguajes y Entornos de Desarrollo	42
2.2.18. Base de Datos	45
2.2.19. Integración de Tecnologías en Arquitectura Multi-Agente	47
2.2.20. Gestión del agua de riego en cultivos bajo invernadero	47
2.2.21. Importancia del riego en ambientes controlados	48
III. METODOLOGÍA	49
3.1. ENFOQUE METODOLÓGICO	49
3.1.1. Enfoque	49
3.1.2. Tipo de Investigación.....	49
3.2. IDEA A DEFENDER	50
3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES	50
3.4. MÉTODOS UTILIZADOS	52
3.4.1. Métodos.....	52
3.4.2. Técnicas.....	52
3.5. ANÁLISIS ESTADÍSTICO	53
3.5.1. Población	53
IV. RESULTADOS Y DISCUSIÓN	54
4.1. RESULTADOS	54
4.1.1. Resultados de la encuesta.....	54
4.1.2. Fundamentación Teórica del Censo	54
4.1.3. Caracterización de la Población	54
4.1.4. Aplicación del Censo en el Estudio	55
4.1.5. Control de Calidad del Censo	55
4.1.6. Aspectos Éticos y de Confidencialidad.....	55

4.1.7. Observación.....	56
4.1.8. Formula de Spearman.....	57
4.1.9. Correlaciones más significativas.....	58
4.2. PROPUESTA	58
4.2.1. Fase de planificación	59
4.2.2. Levantamiento de Procesos	59
4.2.3. Fase de diseño del prototipo	65
4.2.4. Red Neuronal - Sistema de Riego.....	66
4.2.5. Diagrama de flujo tflite.....	67
4.2.7. Fase de Desarrollo.....	81
4.2.8. Fase de Pruebas.....	91
4.2.9. Fase de Validación final y Retroalimentación	95
4.2.10. Tablas de Validación en Campo y Prototipo.....	105
4.2.11. Testeo del Sistema (Prototipo vs invernadero real)	106
4.2.12. Comparación de confianza por fase de cultivo	110
4.2.13. Media aritmética de confianza entre (prototipo vs invernadero real)...	111
4.3. DISCUSIÓN	112
V. CONCLUSIONES Y RECOMENDACIONES	114
5.1. CONCLUSIONES	114
5.2. RECOMENDACIONES	114
VI. REFERENCIAS BIBLIOGRÁFICAS.....	116
VII. ANEXOS.....	120

ÍNDICE DE TABLAS

Tabla 1. fases del cultivo bajo invernadero	26
Tabla 2. Metodologías de Desarrollo	37
Tabla 3. Comparación de Metodologías de Software	38
Tabla 4. Implementación de PXP en el Proyecto de Riego Inteligente	40
Tabla 5. Framework de aplicaciones móviles	42
Tabla 6. Framework ventajas y desventajas	42
Tabla 7. Comparativa de Bases de Datos	46
Tabla 8. Operación de variables	51
Tabla 9. Historias de Usuario	59
Tabla 10. Requerimientos Funcionales	61
Tabla 11. Requerimientos no Funcionales	63
Tabla 12. Registros de Validación en Campo (Tomate – Solanum lycopersicum) ...	106
Tabla 13. Registros de Validación en Prototipo Controlado del Sistema de Riego ..	106
Tabla 14. Validación Dual del Sistema	112

ÍNDICE DE FIGURAS

Figura 1. Cuadro de resultados	57
Figura 2. Correlaciones	58
Figura 3. Red Neuronal- Sistema de Riego (Agente IA de cultivo)	66
Figura 4. Diagrama de flujo model_unquant.fflite	67
Figura 5. General del ecosistema multi-agente	69
Figura 6. Base de datos, RTDB	70
Figura 7. Diagrama de Conexiones del Sistema IoT de Riego Inteligente	71
Figura 8. Diagrama Esquemático del Circuito de Control de Riego	71
Figura 9. Diagrama Iniciar sesión	73
Figura 10. Registrar usuario	73
Figura 11. Visualización de datos.....	74
Figura 12. Lectura de sensores	74
Figura 13. Fase de cultivo	75
Figura 14. Tiempo de cosecha.....	75
Figura 15. Diseño de pantalla para login	76
Figura 16. Diseño de pantalla para registro de usuario	76
Figura 17. Diseño de pantalla para menú	77
Figura 18. Diseño de pantalla para asistente agronómico	77
Figura 19. Diseño de pantalla para control de riego.....	78
Figura 20. Diseño de pantalla para monitoreo invernadero.....	78
Figura 21. Diseño de pantalla para manual de usuario	79
Figura 22. Diseño de pantalla para créditos.....	79
Figura 23. Diagrama de procesos del sistema.....	80
Figura 24. Variables entorno del sistema.....	81
Figura 25. Autenticación de usuarios	82
Figura 26. Teachable machine modelo h5 y label.....	82
Figura 27. Librerías pub dev	82
Figura 28. Pantalla de Login	83
Figura 29. Pantalla de Registro	83
Figura 30. Menú Principal.....	84
Figura 31. AgroPredictor - Predicción de Cosecha.....	85
Figura 32. Asistente Virtual Agronómico.....	86

Figura 33. Control de Riego Inteligente	87
Figura 34. Smart Agriculture - Monitoreo de Sensores.....	88
Figura 35. Manual de Usuario	89
Figura 36. Créditos.....	89
Figura 37. Sistema de Notificaciones.....	90
Figura 38. Selección de dispositivo Android o iOS.....	91
Figura 39. Selección del dispositivo y aplicación móvil para la prueba de usabilidad	92
Figura 40. Estado Running en el dispositivo Infinix X6788 en modo automático.....	92
Figura 41. Resultado del test de AgroPredict con aprobación (Pass) para su uso	93
Figura 42. Resultados de consumo de CPU, RAM, batería y red	93
Figura 43. Modelo e información del móvil donde se realizó test de la APK.....	93
Figura 44. Información de compatibilidad con dispositivos móviles Android.....	94
Figura 45. Detección de permisos otorgados desde Flutter a la apk.....	94
Figura 46. Prueba de stress realizada a la APK en Test Lab	94
Figura 47. Resultados de rendimiento en diferentes procesos y periodos de tiempo .	95
Figura 48. Resultados de consumo de CPU, RAM y red de la APK	95
Figura 49. Interfaz de acceso login de AgroPredict.....	96
Figura 50. Interfaz de registro de usuario en AgroPredict.....	97
Figura 51. Menú principal de la aplicación AgroPredict.....	98
Figura 52. Interfaz de AgroPredic con predicción automática de cosecha.	99
Figura 53. Interfaz del asistente virtual agronómico respondiendo una consulta.....	100
Figura 54. Panel de control del sistema de riego inteligente.	101
Figura 55. Panel de monitoreo en tiempo real del módulo Smart Agriculture.	102
Figura 56. Interfaz del Manual de Usuario AgroPredict.....	103
Figura 57. Interfaz de créditos y detalles institucionales.	104
Figura 58. Ejemplo de notificaciones enviadas por WhatsApp Business API.....	105
Figura 59. fase desarrollo prototipo	107
Figura 60. fase desarrollo invernadero real	107
Figura 61. fase de Floración prototipo.....	108
Figura 62. fase de floración invernadero real.....	108
Figura 63. fase de producción prototipo	109
Figura 64. fase de producción invernadero real.....	109
Figura 65. desconocido invernadero real.....	110

Figura 66. Resultados nivel de confianza entre prototipo e invernadero real.....	110
Figura 67. Media aritmética prototipo.....	111
Figura 68. Media aritmética invernadero real	111
Figura 69. prototipo maqueta invernadero	146
Figura 70. invernadero de la UE Jorge Martínez Acosta	146
Figura 71. técnico invernadero	147
Figura 72. Cultivos bajo etapa desarrollo.....	147
Figura 73. Cultivo etapa de floración.....	148
Figura 74. cultivo etapa de producción	148
Figura 75. técnico en invernadero mantenimiento	148

ÍNDICE DE ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC	120
Anexo 2. Certificado del abstract por parte del centro de idiomas.....	121
Anexo 3. Carta de acceso a invernadero institución.....	123
Anexo 4. Encuesta.....	124
Anexo 5. Entrevista	127
Anexo 6. Manual de Usuario	129
Anexo 7. Manual Técnico	136
Anexo 8. Fotografías	146

RESUMEN

La investigación aborda la problemática de la gestión ineficiente del agua en cultivos bajo invernadero en la Unidad Educativa "Jorge Martínez Acosta", en la provincia del Carchi. Para solucionar esta problemática, se desarrolla AgroPredict, un sistema inteligente de gestión hídrica basado en una arquitectura multi-agente que incorpora inteligencia artificial, visión artificial y tecnologías IoT. El sistema emplea sensores ESP32 que monitorean en tiempo real variables ambientales como temperatura y humedad del suelo. Utiliza visión artificial para detectar automáticamente las fases fenológicas del cultivo (Desarrollo, Floración, Producción, Desconocido) con una precisión del 99.5% en prototipo y 96.3% en condiciones reales. Además, el control automatizado del riego se ajusta según fases de cultivo y condiciones, garantizando eficiencia hídrica y salud del cultivo. La solución incluye una aplicación móvil desarrollada con Flutter con módulos para monitoreo, control manual y automático del riego, asistente virtual agronómico con Gemini AI para diagnósticos y recomendaciones, y notificaciones automatizadas vía WhatsApp. La validación en campo muestra tiempos de respuesta rápidos, alta precisión y robustez del sistema. AgroPredict contribuye a la agricultura de precisión en ámbitos educativos y comerciales, promoviendo un uso responsable y eficiente del agua.

Palabras clave: inteligencia artificial, IoT agrícola, gestión hídrica, visión artificial, agricultura sostenible.

ABSTRACT

This research addresses the issue of inefficient water management in greenhouse crops at the "Jorge Martínez Acosta" Educational Unit, located in the province of Carchi. To solve this problem, AgroPredict was developed, an intelligent water management system based on a multi-agent architecture that incorporates Artificial Intelligence (AI), Computer Vision, and IoT technologies. The system uses ESP32 sensors to monitor real-time environmental variables such as soil temperature and humidity. It employs computer vision to automatically detect the crop's phenological phases (Development, Flowering, Production, Unknown) with a precision of 99.5% in the prototype and 96.3% under real-world conditions. Furthermore, the automated irrigation control adjusts dynamically according to the crop phases and ambient conditions, ensuring both water efficiency and crop health. The solution includes a mobile application developed with Flutter, featuring modules for monitoring, manual and automatic irrigation control, a virtual agronomic assistant powered by Gemini AI for diagnostics and recommendations, and automated notifications via WhatsApp. Field validation shows fast response times, high precision, and robustness of the system. AgroPredict contributes to precision agriculture in both educational and commercial settings, promoting responsible and efficient water use.

Keywords: artificial intelligence, agricultural IoT, water management, artificial vision, sustainable agriculture.

INTRODUCCIÓN

La agricultura moderna enfrenta el desafío crítico de producir más alimentos con menos recursos, especialmente en lo que respecta al uso eficiente del agua. En Ecuador, la agricultura es la principal fuente de ingresos económicos, la gestión inadecuada del recurso hídrico en cultivos bajo invernadero resulta en pérdidas que pueden alcanzar hasta el 50% del agua utilizada, comprometiendo tanto la sostenibilidad ambiental como la viabilidad económica de las explotaciones agrícolas.

La provincia del Carchi, reconocida por su vocación agrícola y ubicación estratégica en la zona norte del Ecuador, no es ajena a esta problemática. En particular, la Unidad Educativa "Jorge Martínez Acosta" de San Gabriel, que cuenta con un invernadero educativo para la formación práctica de sus estudiantes de bachillerato técnico agrícola, enfrenta desafíos significativos en la gestión adecuada del agua debido a la persistencia de métodos manuales tradicionales y la ausencia de herramientas tecnológicas de monitoreo y control.

Esta necesidad es la razón por la que se realiza el presente trabajo de titulación, sugiriendo la creación y ejecución de una solución de software integral llamada "AgroPredict", que combina tecnologías nuevas e inteligencia artificial, en combinación con sistemas multi-agente y el Internet de las Cosas (IoT) con el fin de transformar la forma en que se maneja el agua en entornos controlados de sembrado. El objetivo de esta investigación no es únicamente optimizar el empleo del recurso hídrico, sino además convertir el invernadero educativo en un laboratorio viviente de agricultura de precisión, en la cual los alumnos tengan la posibilidad de formarse en el manejo de tecnologías de punta que se utilizan en el sector agrícola.

I. EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La tecnología agrícola está transformando la manera en que cultivamos, promoviendo métodos más eficaces y sostenibles. Esto es fundamental para que los agricultores tengan éxito en la actualidad. Desde técnicas tradicionales y verificadas como la rotación de cultivos hasta innovaciones avanzadas como el uso de imágenes satelitales para observar campos agrícolas y programas informáticos especializados, la agrotecnología es fundamental para garantizar la viabilidad de la agricultura moderna, ayudando a los agricultores a mejorar su productividad y a cultivar de forma más sostenible (Sergieieva, 2025).

En Ecuador, la mala gestión del agua y la falta de prácticas agrícolas sostenibles ponen en riesgo la producción alimentaria del país. La agricultura, siendo la principal actividad económica, corre el serio peligro de agotar los acuíferos, contaminar las fuentes hídricas y promover la deforestación. Estas prácticas insostenibles no solo merman la productividad agrícola, sino que también amenazan directamente la seguridad alimentaria. Esto impacta profundamente tanto a las comunidades rurales, que dependen intrínsecamente de la tierra, como a las zonas urbanas, donde la elevación constante de productos en sus precios agrava aún más la desigualdad social.

En la provincia de Carchi, al norte de Ecuador, Posee urgencia en solventar este problema se siente aún más. A pesar de su gran potencial agrícola y el uso de invernaderos, la región enfrenta serios desafíos debido a la variabilidad climática y a las arraigadas prácticas tradicionales de riego. La mala gestión del agua, tanto en cultivos locales como en sistemas de invernadero, ha resultado en un uso ineficiente de este recurso vital. Esto se debe, en gran parte, a la persistencia de métodos de riego obsoletos y a la notable falta de infraestructura para su captación y almacenamiento (INIAP, 2023). La consecuencia directa es una limitación en el desarrollo económico local y un agravamiento de la inseguridad alimentaria: las pérdidas de cultivos, la reducción de ingresos para los agricultores y la escasez de

alimentos nutritivos impactan directamente a miles de familias, especialmente en las áreas rurales.

En la Unidad Educativa "Jorge Martínez Acosta" de San Gabriel, cantón Montúfar, la gestión del agua en su invernadero educativo enfrenta desafíos importantes. Los alumnos, quienes actúan como técnicos responsables de este espacio, carecen de los instrumentos tecnológicos requeridos para mejorar el aprovechamiento de este recurso esencial. Como consecuencia directa, se estima una pérdida de hasta el 50% del agua. Esta falta de eficiencia no solo afecta de manera negativa el desarrollo de las cosechas, sino que también restringe las posibilidades de que los alumnos obtengan un aprendizaje práctico en métodos sostenibles de agricultura. Lo que se obtiene es una vivencia educativa. una gestión ineficaz de los recursos institucionales y menos enriquecedora.

1.2. FORMULACIÓN DEL PROBLEMA

¿De qué forma una solución informática que utiliza inteligencia artificial puede optimizar la gestión y uso eficiente del agua en cultivos bajo invernadero, mejorando la productividad agrícola y reduciendo el desperdicio de recursos hídricos en la Unidad Educativa "Jorge Martínez Acosta" de San Gabriel, Cantón Montúfar Provincia de Carchi?

1.3. JUSTIFICACIÓN

En el mundo agrícola de hoy, es fundamental contar con un software que asista a manejar el agua en los invernaderos. Estos sistemas de computación proveen una respuesta exacta y eficaz a la demanda en aumento de administrar el recurso hídrico de forma inteligente en los invernaderos. La complejidad del ambiente de cultivo bajo invernadero, con condiciones controladas y fluctuaciones micro climáticas, requiere una tecnología especializada capaz de adecuarse y mejorar la utilización del agua de en una forma particular para cada clase de planta cultivada. Los cultivos que se dan bajo invernadero, donde es fundamental controlar el ambiente, las soluciones informáticas posibilitan la supervisión en tiempo real de las condiciones del terreno y de la climatología. Para establecer las necesidades, estos datos en tiempo real son esenciales, riego para las plantas. Estos programas informáticos hacen uso de sensores y algoritmos sofisticados para compilar información acerca de la temperatura, la humedad y otros aspectos elementos de la tierra. Esta información es empleada para modificar automáticamente los sistemas de riego, asegurando

que las plantas obtengan la cantidad requerida de agua en el instante apropiado, mejorando de esta manera su rendimiento y crecimiento. Estas soluciones informáticas, además de la exactitud en el riego, brindan un beneficio estratégico al ofrecer análisis de datos minuciosos y predecibles. Los que se dedican a la agricultura tienen acceso a análisis e informes que les asisten en el entendimiento de las tendencias de consumo de agua, detectar los modelos estacionales y anticipar las necesidades futuras necesidades de agua de sus cultivos. Esta información mejora la toma de decisiones, sino que también permite una planificación a largo plazo, asegurando la sostenibilidad de las operaciones agrícolas en el contexto del manejo adecuado del agua.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

Desarrollar una solución informática con inteligencia artificial para la gestión del agua en cultivos bajo invernadero en la Unidad Educativa 'Jorge Martínez Acosta', ubicada en la ciudad de San Gabriel, cantón Montúfar, provincia del Carchi.

1.4.2. Objetivos Específicos

- Identificar las fases fenológicas del cultivo de tomate (*Solanum lycopersicum*) bajo invernadero para mejorar la gestión hídrica del sistema de riego
- Construir un prototipo con medición climática que optimice el suministro de agua según fases específicas de la planta, incluyendo un sistema de notificaciones automatizadas vía WhatsApp para informar sobre el estado del sistema.
- Codificar software con arquitectura multi-agente para detección de fases del cultivo, monitoreo IoT, control remoto y asistencia virtual agronómica.
- Validar el sistema mediante pruebas en campo vs prototipo, verificando la precisión de nivel de confianza, tiempos de respuesta, eficiencia hídrica, robustez y el correcto análisis de los datos técnicos del cultivo para la gestión del agua.

1.4.3. Preguntas de Investigación

¿Cuáles son las fases fenológicas del cultivo de tomate (*Solanum lycopersicum*) bajo invernadero y sus requerimientos hídricos específicos?

¿Qué variables climáticas debe integrar el prototipo IoT con notificaciones WhatsApp para optimizar el suministro hídrico por fase fenológica?

¿Cómo estructurar la arquitectura multi-agente para integrar detección fenológica, monitoreo IoT, control remoto y asistencia agronómica virtual?

¿Qué nivel de confianza, y eficiencia hídrica presenta el sistema desarrollado frente a invernadero real vs prototipo?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

(Quilismal & Jimenez, 2023) En su estudio Sistema Automatizado para el Control de Riego y Fumigación en Plantaciones de Physalis Peruviana (Uvilla), el enfoque principal de esta investigación fue la construcción de un sistema que buscara entender y mejorar el cultivo de la uvilla. Para esto se empleó una metodología mixta. Por un lado, se aplicaron encuestas a los agricultores de Carchi que siembran esta planta, en donde se buscó entender su producción manual y la falta de tecnología para un riego automatizado. Por el otro lado, se recolectaron datos precisos de los sensores de humedad de suelo y ambiente. A partir de esto, proponemos un diseño diferente a los que existen en el mercado y que se basa en la automatización de riego de un cultivo con la integración de software y hardware libre, que se complementara con una aplicación móvil que se conectara de forma inalámbrica a los sensores para que el agricultor monitoree el riego. Los datos de los sensores se almacenarán en la nube y el agricultor podrá obtenerlos en tiempo real.

(Jimenez F. , 2020) En su Investigación "Modelo basado en agentes inteligentes como soporte a la gestión del riego en cultivos agrícolas" fue el crear un sistema de riego para cultivos agrícolas de campo, en donde se colocó sensores ambientes en un terreno y estos se comunicaban entre sí. Pretendía realizar algo más que una simple simulación para ejecutar un escenario real en el Distrito ubicado en Colombia donde se realizó este proceso y estudio para poder integrarlo. Para esto, construimos un agente de riego que, bajo criterios técnicos, determina y aplica la cantidad de agua que los cultivos requieren en el momento preciso. También el modelo que se basó en agentes, en función al estado de los cultivos, los aspectos sociales de riego y asentamiento de los agricultores vecinos, asigna el agua del sistema de riego. Los resultados que se obtuvieron después de analizar este sistema fue que cumplía con los parámetros que se deseó manteniendo así la humedad adecuada para el cultivo, evitando el déficit de agua en riego.

(Cadena, 2020) En el estudio titulado "Diseño de un sistema para el control de riego mediante técnicas de aprendizaje automático aplicadas a la agricultura de precisión en la Granja La Pradera de la Universidad Técnica del Norte", se desarrolló una propuesta destinada a implementar un sistema de riego automatizado basado en inteligencia artificial. El proyecto tuvo como objetivo optimizar el riego de cultivos de alfalfa utilizando redes neuronales artificiales que aprendieran de las condiciones ambientales registradas. El principio de funcionamiento consistió en recopilar datos de variables como humedad, temperatura y radiación para entrenar la red neuronal, la cual posteriormente tomaría decisiones en tiempo real sobre la necesidad de irrigación. Este enfoque surgió al observar que el método manual de riego por inundación, comúnmente empleado en La Pradera, resultaba poco eficiente, con un consumo excesivo de agua y una distribución irregular. Con la aplicación del sistema inteligente, se busca mejorar la eficiencia del uso hídrico y aumentar la producción, evaluando los resultados mediante pruebas experimentales comparativas entre el nuevo método automatizado y el sistema manual tradicional.

(Jimenez G. , 2024) En la investigación titulada "Sistema de riego inteligente de corto alcance para jardines basado en visión artificial", se desarrolló una herramienta de riego de alta eficiencia capaz de identificar el césped y aplicar agua únicamente en las áreas que realmente lo requieren. A diferencia de otros proyectos similares, esta propuesta no se centra en redes de aspersores o robots móviles, sino en alcanzar un nivel superior de precisión en la irrigación. Para ello, se implementó un modelo entrenado con el algoritmo YOLOv5, encargado de reconocer las zonas cubiertas de césped; además, mediante Python se procesaron los contornos tanto del terreno como de los objetos presentes. El sistema emplea diversos componentes electrónicos que optimizan su funcionamiento: un sensor de humedad FC-28 para medir las condiciones del suelo, una cámara ESP32-CAM para la captura visual, El control central está a cargo de la ESP32. La base de datos se almacena localmente con uso de internet. También se realizó la creación de una app móvil desarrollada con Android estudio para su uso de controlar, el riego logrando realizar la supervisión del sistema, ofreciendo así una solución integral que combina visión artificial, electrónica e IoT. Los ensayos realizados mostraron una precisión del 95,6% en la detección del césped, confirmando su eficacia durante las pruebas prácticas en entornos reales y evidenciando su contribución al uso racional del agua en jardines.

(Calapaqui & Chiguano, 2022) El presente proyecto técnico tiene como finalidad diseñar un sistema inteligente de cultivo para flores de clavel, implementado en el invernadero de la “Finca Consuelito”, ubicada en Cotopaxi, Ecuador. Este desarrollo integra tecnologías de redes de sensores inalámbricos y visión artificial para mejorar la eficiencia agronómica del proceso. La arquitectura del sistema comprende cuatro subsistemas principales: el primero gestiona el riego de manera automática en función de la humedad del suelo detectada por los nodos sensores; el segundo se encarga del manejo de los actuadores —bomba y válvulas— mediante un reloj RTC, la red de sensores y la cámara ESP32-CAM, garantizando un control preciso de activaciones y cortes; el tercero emplea la herramienta Teachable Machine para reconocer posibles signos de enfermedad en las plantas y emitir alertas preventivas; y el cuarto presenta los datos registrados en tiempo real a través de la aplicación BLYNK. En conjunto, estos componentes conforman una solución integral orientada a optimizar el cultivo de claveles con un enfoque eficiente, automatizado y sostenible.

2.2. MARCO TEÓRICO

2.2.1. Solución Informática

Son herramientas de software diseñadas para optimizar procesos de trabajo, automatizar tareas repetitivas y liberar el capital humano para actividades de mayor valor. Estas pueden ser personalizadas para necesidades específicas de una organización o estandarizadas (SaaS) para diversos sectores. Su implementación representa un paso clave en la transformación digital y la informatización global, como lo demuestra el enfoque en el gobierno electrónico y el desarrollo de aplicaciones móviles en Cuba para fortalecer capacidades tecnológicas en el sector agropecuario, promoviendo la soberanía alimentaria mediante la transferencia de conocimiento científico y tecnológico (Chirino, Diaz, Aguilera, & Chiroles, 2022).

Importancia:

- **Transformación Digital:** Son fundamentales para la modernización y digitalización de empresas y procesos.
- **Optimización de Procesos:** Mejoran la eficiencia operativa al automatizar tareas y facilitar flujos de trabajo.
- **Liberación de Capital Humano:** Permiten que el personal se enfoque en tareas de mayor valor agregado y creatividad.

- Toma de Decisiones Basada en Datos: Proporcionan información clave para decisiones estratégicas y oportunas.

Ventajas:

- Reducción de Errores y Costos: Minimizan fallos humanos y optimizan el uso de recursos.
- Agilización y Simplificación: Aceleran procesos y hacen la gestión de información más sencilla.
- Mejora de la Comunicación: Facilitan la interacción interna y externa de manera eficiente.
- Respuesta Rápida: Permiten adaptarse con celeridad a los cambios del entorno o del mercado.
- Acceso a Información y Conocimiento: Facilitan la gestión y transferencia de datos y saber.

Desventajas

- Inversión Inicial: Requieren un capital significativo para su adquisición e implementación.
- Seguridad de Datos: Implican un riesgo potencial y la necesidad de proteger la información.
- Dependencia Excesiva: Pueden generar una fuerte dependencia tecnológica que dificulta la adaptabilidad sin ellas.
- Generación de Residuos Electrónicos: Contribuyen al problema de la basura tecnológica.
- Necesidad de Capacitación: Requieren que el personal aprenda a usarlas eficazmente.




2.2.2. Fases fenológicas del cultivo bajo invernadero

Los cultivos bajo invernadero atraviesan diferentes etapas fenológicas que marcan su ciclo productivo. Estas fases son fundamentales para comprender el desarrollo de las plantas y establecer estrategias de manejo agronómico apropiadas. En el caso específico del tomate (*Solanum lycopersicum*), como en la mayoría de los cultivos

hortícolas de invernadero, el ciclo de producción se divide principalmente en tres fases críticas: desarrollo vegetativo, floración y producción.

- La fase de desarrollo vegetativo representa el período inicial del cultivo, donde la planta concentra sus recursos en establecer su estructura. Durante esta etapa, se evidencia la formación de hojas verdes pequeñas y un crecimiento activo del follaje. La planta construye su arquitectura vegetal y consolida su sistema radicular, sentando las bases necesarias para las siguientes etapas reproductivas y productivas.
- La fase de floración marca el inicio del período reproductivo del cultivo. En esta etapa comienzan a aparecer las flores que posteriormente darán origen a los frutos. Una polinización adecuada es esencial durante este período para asegurar un buen cuajado de frutos y, en consecuencia, una producción satisfactoria.
- La fase de producción corresponde al período de fructificación, momento en el cual los frutos verdes inician su proceso de maduración progresiva hasta alcanzar su color rojo característico. Esta fase resulta determinante para la calidad y el rendimiento final de la cosecha, requiriendo un manejo preciso del riego y la nutrición para optimizar el desarrollo del cultivo.

Tabla 1. fases del cultivo bajo invernadero

Código	Fase del cultivo	Referencia	Descripción	Características
1	<ul style="list-style-type: none"> • DESARROLLO 		Fase vegetativa inicial	<ul style="list-style-type: none"> • Hojas verdes pequeñas
2	<ul style="list-style-type: none"> • FLORACIÓN 		Fase reproductiva	<ul style="list-style-type: none"> • Aparición de flores
3	<ul style="list-style-type: none"> • PRODUCCIÓN 		Fase de fructificación	<ul style="list-style-type: none"> • Frutos verdes y rojos

2.2.3. Tecnologías Habilitadoras Clave

Son innovaciones fundamentales del sistema de riego por goteo automatizado en invernadero requiere la integración de diversas tecnologías habilitadoras que permiten el monitoreo, control y gestión eficiente del recurso hídrico. Estas tecnologías conforman la base técnica del sistema propuesto y posibilitan la toma de decisiones en tiempo real para optimizar el suministro de agua según las necesidades del cultivo, más eficiente y sostenible (Mendez, 2020).

Existen varias tecnologías consideradas habilitadas digitales. Seguidamente, se mencionan algunas de las más significativas:

- Sensores de Humedad del Suelo
- Sensores Ambientales
- Válvulas Solenoides Electrónicas
- Sistemas de Control y Procesamiento
- Infraestructura de Riego por Goteo

2.2.4. Inteligencia Artificial Aplicada

Busca emplear la inteligencia artificial para solucionar problemas concretos y crear valor en el mundo real. Es la inteligencia artificial que observamos y experimentamos diariamente, que se encuentra en sectores como las finanzas, el transporte y la salud. La inteligencia artificial puesta en práctica genera sistemas informáticos que pueden aprender de la experiencia, adaptarse, razonar y decidir mediante métodos como el procesamiento del lenguaje natural y el aprendizaje automático. Su éxito se manifiesta en un aumento de la eficacia, una disminución de los costos y unos resultados más favorables, logrando que la tecnología simule el pensamiento humano para nuestro beneficio (Casco, 2024)

- Gemini 1.5 Flash

Se caracteriza por su arquitectura transformer optimizada que permite procesamiento multimodal con capacidad de analizar y comprender múltiples tipos de datos de entrada incluyendo texto, imágenes, audio y video de manera simultánea, lo que lo hace especialmente útil para sistemas IoT que generan datos heterogéneos. Soporta hasta 1 millón de tokens de contexto mediante su ventana de contexto extendida, permitiendo procesar grandes cantidades de datos históricos de sensores y mantener coherencia en análisis prolongados. Está optimizado para

generar respuestas en milisegundos logrando baja latencia, característica crítica para sistemas de monitoreo en tiempo real que requieren análisis y decisiones inmediatas. Presenta eficiencia computacional al estar diseñado con un balance óptimo entre rendimiento y consumo de recursos, resultando en menores costos operativos comparado con modelos más pesados como Gemini Pro o Ultra. La integración con Gemini 1.5 Flash se realiza mediante la API de Google AI, que permite enviar peticiones HTTP con datos estructurados y recibir respuestas JSON. El modelo puede procesar prompts que incluyen datos de sensores, históricos y consultas específicas, retornando análisis, predicciones o acciones recomendadas que posteriormente pueden automatizarse o notificarse a usuarios.

- Keras

Biblioteca de código abierto para Python que proporciona una interfaz de alto nivel para el desarrollo y entrenamiento de redes neuronales, actualmente integrada como API oficial de TensorFlow consolidándose como estándar de facto para prototipado rápido y desarrollo de modelos de aprendizaje profundo. Se fundamenta en tres principios arquitectónicos centrados en facilidad de uso mediante una API diseñada para humanos minimizando la carga cognitiva con interfaces consistentes e intuitivas, presenta modularidad donde los modelos se construyen mediante bloques componibles como capas, optimizadores y funciones de pérdida que pueden combinarse libremente, y ofrece extensibilidad permitiendo crear componentes personalizados manteniendo compatibilidad con el ecosistema completo. El formato .h5 basado en HDF5 (Hierarchical Data Format version 5) es el estándar utilizado por Keras para almacenar modelos entrenados, conteniendo la arquitectura del modelo con definición completa de la topología de red incluyendo tipos de capas, número de neuronas, funciones de activación y conexiones entre capas, almacena los pesos entrenados como valores numéricos de todos los parámetros del modelo obtenidos durante el proceso de entrenamiento, incluye la configuración del optimizador con información sobre el algoritmo de optimización utilizado y sus hiperparámetros, y guarda el estado del entrenamiento con metadata sobre el proceso incluyendo número de épocas, función de pérdida y métricas alcanzadas.

- Firebase Realtime Database:

Sistema de gestión de datos diseñado para sincronizar información instantáneamente entre múltiples clientes conectados, a diferencia de bases de datos tradicionales donde los clientes deben solicitar activamente actualizaciones, las RTDB implementan un modelo push donde los cambios se propagan automáticamente. Firebase Realtime Database, desarrollada por Google, es la implementación de RTDB más utilizada en proyectos IoT, caracterizándose por sincronización automática donde todos los clientes conectados (aplicaciones web, móviles, dispositivos IoT) reciben actualizaciones en milisegundos cuando los datos cambian, modelo de datos JSON con estructura de árbol flexible basada en documentos JSON permitiendo jerarquías anidadas y relaciones complejas sin esquemas rígidos, persistencia offline donde los clientes pueden leer y escribir datos incluso sin conexión a internet sincronizándose automáticamente al restablecer conectividad, y protocolo WebSocket utilizando conexiones persistentes bidireccionales que eliminan la latencia del polling tradicional. Las reglas de Firebase constituyen un lenguaje declarativo que define permisos de acceso y validación de datos escritas en formato JSON que evalúan automáticamente cada operación de lectura y escritura, incluyendo componentes de autenticación (.read y .write) que controlan quién puede acceder a qué datos basándose en el estado de autenticación del usuario permitiendo implementar permisos granulares por ruta de datos, validación (.validate) que verifica que los datos escritos cumplan con requisitos específicos de estructura, tipo y contenido antes de permitir la operación, e indexación (.indexOn) que optimiza consultas especificando qué campos deben indexarse para búsquedas eficientes. Las reglas utilizan variables predefinidas como auth para información del usuario autenticado, data para estado actual de los datos antes de la operación, newData para nuevo estado propuesto de los datos, y \$variable para capturar segmentos dinámicos de ruta, disponiendo de operadores de comparación, operadores lógicos, funciones de validación (hasChildren, exists, contains) y acceso a métodos de timestamp. En arquitecturas IoT, Firebase RTDB funciona como hub central que recibe datos de dispositivos donde los microcontroladores ESP32 escriben lecturas de sensores directamente en rutas específicas usando autenticación por tokens, distribuye actualizaciones donde dashboards, aplicaciones móviles y servicios backend reciben instantáneamente nuevas mediciones sin polling, almacena históricos manteniendo series temporales de datos para análisis, graficación y auditoría, y coordina dispositivos permitiendo control remoto escribiendo comandos

que los dispositivos escuchan y ejecutan, mientras las reglas aseguran que dispositivos solo puedan escribir en sus propias rutas, usuarios autenticados puedan leer datos relevantes y toda información cumpla con formatos esperados, previniendo corrupción de datos y accesos no autorizados.

- Arduino

Ecosistema de hardware y software open-source que democratiza el desarrollo de sistemas embebidos, su entorno de programación simplificado (Arduino IDE) utiliza un lenguaje basado en C/C++ con abstracciones que facilitan el control de hardware. El ESP32, microcontrolador desarrollado por Espressif Systems, soporta la plataforma Arduino mediante un core específico que permite programarlo usando el Arduino IDE, esta compatibilidad combina hardware potente con procesador dual-core de 32 bits a 240 MHz, 520 KB de RAM y conectividad WiFi y Bluetooth integrada, junto con la simplicidad Arduino mediante funciones familiares como `digitalWrite()`, `analogRead()` y `Serial.print()` adaptadas al hardware ESP32, además de bibliotecas especializadas que proporcionan acceso a funcionalidades avanzadas (WiFi, HTTPClient, WebServer, TLS) mediante bibliotecas optimizadas. Un programa Arduino (sketch) para ESP32 se estructura en declaraciones e includes donde se importan bibliotecas necesarias y se definen constantes globales (pines, credenciales, URLs), función `setup()` ejecutada una sola vez al iniciar que configura hardware (`pinMode`, `Serial.begin`), establece conexiones (`WiFi.begin`) e inicializa periféricos, función `loop()` ejecutada repetitivamente que contiene la lógica principal del programa (lectura de sensores, procesamiento, comunicación), y funciones auxiliares con código modularizado para tareas específicas como envío de datos, procesamiento de respuestas o manejo de errores. Las bibliotecas clave para IoT incluyen `WiFi.h` que gestiona conectividad WiFi permitiendo escaneo de redes, conexión, verificación de estado y obtención de dirección IP, `HTTPClient.h` que facilita peticiones HTTP/HTTPS (GET, POST, PUT, DELETE) con soporte para headers personalizados, autenticación y manejo de respuestas, `ArduinoJson.h` para serialización y deserialización de JSON esencial para intercambio de datos estructurados con APIs, y `FirebaseESP32.h` como cliente específico para integración con Firebase que abstrae complejidad de autenticación, paths y operaciones CRUD. La conexión del ESP32 con WhatsApp Graph API v18.0 requiere autenticación mediante tokens de acceso permanentes incluidos en headers HTTP Authorization como Bearer tokens, construcción de payload creando objetos JSON con estructura específica (`messaging_product`, `recipient_type`, `to`, `type`, `content`),

petición POST enviando mensaje mediante `HTTPClient.post()` a la URL del endpoint de mensajería, y manejo de respuestas procesando códigos HTTP y respuestas JSON para confirmar entrega o detectar errores, permitiendo que el ESP32 actúe como cliente que ante condiciones específicas (alerta de sensor, evento programado) construye y envía peticiones HTTPS directamente a los servidores de WhatsApp, habilitando notificaciones push sin intermediarios en casos de emergencia.

2.2.4.1. Inteligencia Artificial en la agricultura

Dado el aumento en la demanda de alimentos y la necesidad de una agricultura más eficiente, (IA) se ha convertido en algo esencial. Esta tecnología no solamente disminuye la dependencia de la mano de obra, sino que cambia la administración agrícola y de plantaciones. A través de la evaluación de datos, que incluye pronósticos e imágenes obtenidas por satélites meteorológicos, la inteligencia artificial facilita el monitoreo en tiempo real de la salud de las cosechas. Detectar plagas y enfermedades, prever rendimientos. Además, proporciona Orientación esencial para los agricultores en cuanto a qué cultivar, cómo maximizar el empleo de prever los precios y los pesticidas, para ayudarles a afrontar los retos de la agricultura cada vez más compleja. Es claro que la inteligencia artificial es un instrumento poderoso para incrementar la eficiencia y la productividad, una fuerza de transformación que ya está promoviendo diferentes campos como el transporte, las finanzas y la salud, y que las empresas tienen que seguir fomentando en el sector agrícola. (Javaid, 2023).

2.2.5. Visión Artificial

Es un campo elemental de la Inteligencia Artificial que habilita a las computadoras para "observar" y entender imágenes del mundo real. Desde sus inicios en la década de los sesenta, ha progresado hasta transformarse en una herramienta fundamental para identificar patrones, empleada en la vigilancia y la seguridad, Robótica y reconocimiento de rostros. Opera al procesar cada píxel de una imagen. digital a través de cálculos matemáticos, llamados filtros, que simulan la Visión humana, aunque con una exactitud sin igual. Incorporada en distintas industrias, No es una tecnología independiente; al transmitir y recibir datos, se vuelve un elemento esencial. de la cadena productiva, dándole a las máquinas la capacidad de examinar su entorno y realizar elecciones inteligentes (Borella, 2022).

2.2.5.1. Visión artificial en la agricultura

Mediante datos obtenidos de cámaras, drones y dispositivos IoT, esta tecnología identifica patrones y proporciona información práctica para mejorar la eficiencia operativa. Integrándose con herramientas modernas como la IA y el IoT, permite a los agricultores optimizar procesos clave como el monitoreo de cultivos, la gestión del ganado y la clasificación de productos. Sus aplicaciones no solo reducen la mano de obra y los costos, sino que también maximizan la producción y mejoran la precisión en todas las tareas agrícolas (Lumenalta, 2025).

- Monitoreo mejorado de cultivos
- Predicción mejorada del rendimiento
- Manejo rentable de plagas y malezas
- Gestión eficiente del ganado
- Agricultura de precisión
- Optimización laboral
- Mejoras en la sostenibilidad

2.2.5.2. Técnicas de Visión Artificial

Las técnicas de visión artificial posibilitan que las computadoras analicen imágenes. Las máquinas analizan el mundo visual para diferentes usos al procesar los píxeles a través de algoritmos, lo que les permite detectar patrones y distinguir objetos. (Lumenalta, 2025).

- Detección de bordes
- Segmentación de imágenes
- Reconocimiento de objetos
- Reconstrucción 3D
- Estereovisión

2.2.6. Redes Neuronales

Son soluciones informáticas que replican el cerebro humano, formando un subtipo fundamental para el aprendizaje automático y un eje de la inteligencia artificial. Debido a su estructura interconectada, procesan datos de manera eficaz y son sobresalientes en la identificación de patrones y solución de problemas complejos que los sistemas convencionales no son capaces de tratar. Esta tecnología está promoviendo una interrupción sin precedentes, provocando transformaciones rápidas en campos como el mercado de trabajo, el marketing, la investigación

médica y el consumo de contenido muestran cómo Los dispositivos obtienen habilidades que antes eran exclusivas de los seres humanos (Ortiz, 2025).

Las redes neuronales se pueden clasificar en base a su estructura:

- Monocapa
- Multicapa
- Convolucional
- Concurrentes
- Radiales

2.2.7. Predicción de necesidades hídricas

La agricultura contemporánea está incorporando sistemas de riego predictivos basados en inteligencia artificial y aprendizaje automático, ante el incremento de la demanda de agua y la falta de recursos. Estos instrumentos examinan en tiempo real grandes cantidades de datos, desde el clima hasta la humedad del suelo mediante sensores IoT, para predecir con exactitud cuándo y cuánta agua requiere cada cultivo. Esto posibilita una administración optimizada y automatizada del riego, lo que disminuye el uso de energía y agua, reduce los costos al mínimo y mejora notablemente la productividad de las cosechas. En términos generales, estos sistemas convierten la administración hídrica en una práctica agrícola más eficiente y sostenible (Guiles & Ahmet, 2023).

2.2.8. Internet de las cosas (IoT)

Una red de aparatos físicos que colectan e intercambian información, desde electrodomésticos y sistemas industriales, incrementa su capacidad al fusionarse con la visión artificial. Esta combinación posibilita que se procesen imágenes y videos en tiempo real, proporcionando a los objetos la habilidad de examinar su entorno entre la visión artificial, el software y los sensores crea una inteligencia en tiempo real. que mejora el desempeño de los dispositivos, lo que proporciona un amplio rango de aplicaciones novedosas en áreas como la vigilancia, la medicina y la educación, mejorando de manera significativa la calidad de vida. Para llevarla a cabo, se necesita una infraestructura que administre de manera efectiva el alto volumen de datos e ilustraciones (Alvear, Rosero, Peluffo, & Pijal, 2023).

2.2.9. Sistemas Multi-Agente

Es una arquitectura de software en la que numerosas entidades independientes, llamados agentes colaboran para lograr un objetivo compartido en un ambiente con dividido. Estos agentes, que pueden ir desde programas de software hasta Los robots o sensores inteligentes no solamente cooperan entre sí, sino que además interactúan con su entorno y entre ellos de forma adaptable y flexible. Su fortaleza se encuentra en la habilidad para solucionar problemas complejos de manera distribuida, utilizando la la especialización de cada agente y la inteligencia colectiva para encontrar soluciones fuertes y eficaces que serían difíciles de lograr con un solo sistema (Anwar, 2025).

Tres rasgos fundamentales caracterizan a cada agente en un sistema multi-agente:

- Autonomía: Cada elemento del sistema puede actuar y tomar decisiones de manera autónoma, teniendo en cuenta las circunstancias del entorno y las metas definidas.
- Interactividad: Las entidades del sistema sostienen una comunicación continua, comparten datos importantes lo que facilita la toma de decisiones en colaboración de decisiones y fortalece la eficacia operativa.
- Adaptabilidad: El sistema se adapta de manera dinámica frente a cambios. Nuevas entradas de datos o externas, alterando sus acciones en tiempo real para dar una respuesta apropiada a las transformaciones en los términos del ambiente.

2.2.9.1. Como funcionan los sistemas multi-agente

Los sistemas multi-agente funcionan como un grupo dinámico de entidades inteligentes, que trabajan juntas en un ambiente común para lograr una meta compartida, a pesar de su autonomía. Su funcionamiento se fundamenta en la interacción continua y en la toma de decisiones de manera coordinada entre estos agentes. Cada persona percibe su medio ambiente, procesa la información y actúa, comunicándose con los demás para intercambiar conocimientos, negociar y adaptarse. Esta interacción distribuida posibilita que el sistema total solucione problemas complejos que no podría resolver un solo agente, utilizando la especialización y la inteligencia colectiva para alcanzar resultados sólidos y eficaces. (Anwar, 2025).

- **Percepción y detección:** Los agentes, a través de sensores o fuentes de entrada, adquieren datos del ambiente que les posibilitan reconocer las condiciones y los cambios en el entorno donde funcionan.
- **Toma de decisiones:** Con base en los datos recogidos, los agentes aplican reglas lógicas, algoritmos de inteligencia artificial o modelos de aprendizaje automático para definir sus próximas acciones.
- **Comunicación y coordinación:** Los diferentes agentes se interconectan intercambiando información, negociando o sincronizando actividades con el propósito de alcanzar los objetivos globales del sistema.
- **Ejecución de acciones:** Cada agente lleva a cabo tareas específicas determinadas por su proceso de decisión, influyendo de forma directa en el comportamiento general del conjunto.
- **Aprendizaje y adaptación:** Ciertos agentes implementan métodos de aprendizaje continuo, lo que les permite optimizar su desempeño y ajustar sus estrategias de manera progresiva en función de la experiencia adquirida.

2.2.10. Microcontroladores (ESP32)

Desarrollado por la empresa china Espressif Systems. Su gran aceptación se debe a que integra en un solo chip capacidades avanzadas de conectividad Wi-Fi y Bluetooth, junto con una potente capacidad de procesamiento. Esto lo convierte en una opción ideal y muy eficiente para proyectos que requieren comunicación inalámbrica, bajo consumo de energía y la capacidad de interactuar con diversos dispositivos y sensores (Peña, 2024).

2.2.10.1. Microcontrolador ESP32 aplicado a la agricultura

Se ha consolidado como una pieza clave para la innovación en la agricultura, gracias a su naturaleza económica y de bajo consumo, además de integrar conectividad Wi-Fi y Bluetooth. Estas características lo hacen idóneo para desarrollar dispositivos del Internet de las Cosas (IoT) que optimizan las operaciones agrícolas. Su versatilidad permite la creación de soluciones inteligentes para el monitoreo de cultivos, la gestión del ganado y la automatización de sistemas de riego, facilitando así una agricultura más eficiente y conectada (Peña, 2024).

- **Sensores de humedad del suelo:** las placas ESP32 se pueden utilizar para construir sensores que miden el contenido de humedad del suelo y envían los datos de forma inalámbrica a un centro central para su análisis.

- Monitoreo de cultivos: las placas ESP32 se pueden utilizar para construir cámaras y algoritmos de procesamiento de imágenes que rastrean el crecimiento de los cultivos y predicen su rendimiento.
- Seguimiento del ganado: Las etiquetas inalámbricas que hacen seguimiento a la posición y al desplazamiento del ganado, supervisan su salud y ponen en marcha procedimientos automáticos de ordeño y alimentación se pueden elaborar con las placas ESP32.
- Control de riego: Los sensores que miden la humedad del terreno y los datos climáticos pueden ser contruidos con las placas ESP32, y estos utilizan dicha información para regular los sistemas de riego y mejorar el uso del agua.

2.2.11. Sensores para sistema de riego

Un sensor de humedad es un aparato electrónico que, mediante la conversión de los datos en una señal eléctrica, nos posibilita cuantificar el nivel de humedad en cualquier ambiente. Se utilizan sensores combinados de temperatura y humedad ambiental, como los modelos DHT11 y DHT22, que permiten monitorear variables climáticas del invernadero que influyen en la demanda hídrica. Actualmente también existen sensores combinados multiparamétricos aptos para sistemas IoT inteligentes en agricultura de precisión.

Integración con Válvulas Solenoides: estos sensores guían la activación de válvulas solenoides eléctricas, las cuales funcionan sustancialmente a 24 V CA, adecuadas para abrir y cerrar el flujo de agua en ciclos cortos (pulsos) sincronizados con las mediciones de humedad y condiciones de cultivo.

Válvula Rivulis V2000: es ideal para sistemas de riego por goteo en invernadero es la con las siguientes características:

- Disponible en tamaños de 2" a 6" con opciones angulares y rectas
- Control máximo del caudal y presión con mecanismo de pistón
- Construida en PVC resistente a químicos, rayos UV y condiciones exigentes
- Diafragma protegido contra fugas y golpes de ariete
- Compatible con activación remota y temporizadores de 24 V CA
- Minimiza pérdidas de presión y alta durabilidad
- Utilizada mundialmente en agricultura de precisión

2.2.12. Metodologías de Desarrollo

Las metodologías de desarrollo de software son el núcleo de la programación colaborativa, asegurando que el trabajo se ejecute con eficacia, productividad y organización. Estas metodologías, lejos de ser meros formalismos, han progresado hasta transformarse en el fundamento esencial para desarrollar software de gran calidad. Las metodologías ágiles son las que predominan hoy en día, y casi todas las compañías del sector las han implementado. Por lo tanto, para cualquier profesional que quiera sobresalir en el desarrollo de software, es esencial conocerlas. Si bien en algunos proyectos todavía se utilizan las metodologías tradicionales, implementar un enfoque estructurado es esencial para triunfar en el competitivo mundo de la programación (Santander, Santander, 2020).

Tabla 2. Metodologías de Desarrollo

Criterio	Metodología Ágil	Metodología Tradicional
Enfoque	Iterativo e incremental, basado en entregas continuas.	Secuencial y estructurado, con fases bien definidas
Flexibilidad	Alta flexibilidad, permite cambios durante el desarrollo	Rígida, difícil de modificar una vez definido el alcance
Planificación	Se adapta de forma continua según las necesidades del proyecto	Se realiza al inicio y se sigue estrictamente
Tiempos de entrega	Corto, se realizan entregas parciales y frecuentes	Largo, la entrega del producto ocurre al final del desarrollo
Participación del cliente	Constante, el cliente participa en todas las fases del desarrollo	Limitada, generalmente solo al inicio y al final del proyecto
Gestión de riesgos	Se detectan y solucionan de manera continua a lo largo del desarrollo	Se identifican al inicio del proyecto, pero los cambios pueden ser costosos.
Documentación	Mínima, prioriza la funcionalidad sobre la documentación extensa.	Extensa, se requiere una planificación y documentación detallada.
Control de calidad	Se integra en cada iteración del desarrollo	Se realiza al final del proyecto
Ejemplos de Metodologías	Scrum, PXP, Kanban, RAD	Cascada, Modelo en V, Modelo Espiral

2.2.13. Metodología Ágil

La metodología ágil es un método dinámico que mejora la entrega de proyectos al operar en ciclos breves y constantes. Esta metodología posibilita la entrega de progresos concretos en cada etapa culminada, en vez de esperar hasta el final. Esto no solo hace que el proceso sea más eficaz, sino que también permite una

adaptación continua a las variaciones, lo que garantiza una mejor alineación del resultado final con las expectativas y necesidades, proporcionando valor de manera más rápida y flexible (Zendesk, 2023).

- Scrum: Una de las más destacadas en el método ágil, es una forma astuta de gestionar proyectos que combina la filosofía de la agilidad con un modelo específico de gestión. Es como tener un equipo de trabajo que se adapta con rapidez, brinda valor de manera continua y aprende sobre la marcha (Lomeli, 2023).
- Kanban: Emplea tableros visuales para administrar proyectos. Estos tableros ilustran el flujo de trabajo en columnas que simbolizan fases, como "Por hacer", "En progreso" y "Terminado". Las tareas se trasladan entre columnas hasta que finalizan, y están representadas por tarjetas. Esto posibilita que los equipos tengan una visualización clara y eficaz de su trabajo y carga (Martins, 2025).
- PXP: Es una metodología ágil que prioriza velocidad, simplicidad y colaboración. Con ciclos de desarrollo cortos y menos documentación, PXP permite a los desarrolladores adaptarse rápidamente a cambios y mejorar la eficiencia mediante revisiones y pruebas constantes (Raeburn, 2025).

Tabla 3. Comparación de Metodologías de Software

Nº	Característica	PXP	Scrum	Kanban
1	Adaptabilidad a estrategias de desarrollo	✓	✓	✓
2	Enfoque en la calidad del producto	✓	✓	✓
3	Comunicación constante dentro del equipo	✓	✓	✓
4	Capacidad de respuesta rápida a cambios	✓	✓	✓
5	Planificación abierta y flexible	✓	✓	✓
6	Eliminación de interrupciones en el flujo			✓
7	Gestión eficiente de flujos de trabajo			✓
8	Uso de listas de control		✓	✓
9	Establecimiento de límites claros			✓
10	Colaboración constante con el cliente	✓	✓	
11	Adaptabilidad a cualquier tipo de proyecto	✓	✓	✓
12	Aplicabilidad en grandes empresas		✓	✓
13	Retroalimentación con cliente y equipo	✓	✓	✓
14	Enfoque en las necesidades del cliente	✓	✓	
15	Prioridad a los requisitos como éxito	✓	✓	
16	Velocidad y flexibilidad en gestión de proyectos	✓	✓	✓
17	División de proyectos en fases	✓	✓	
18	Trabajo en sprints o iteraciones cortas	✓	✓	
	Total, de características presentes:	13	15	12

2.2.13.1. Metodologías Utilizadas

Su objetivo es desarrollar y gestionar proyectos con eficiencia, flexibilidad y control,

se basa en la comunicación, reutilización del código desarrollado y realimentación (Ginzo, 2022).

- Planificación: El proceso de planificación parte del desarrollo de historias de usuario, las cuales se priorizan y dividen en componentes más pequeños o versiones parciales. Estas se revisan periódicamente, normalmente cada dos semanas, coincidiendo con el cierre de cada iteración. El propósito es garantizar que el producto de software evolucione de forma funcional, verificable y listo para pruebas, con miras a su publicación progresiva.
- Diseño: El diseño se basa en mantener el código lo más simple posible, generando los elementos mínimos necesarios para obtener un prototipo funcional. Cuando el enfoque es orientado a objetos, se emplean herramientas como las tarjetas CRC (Clase-Responsabilidad-Colaboración), que permiten estructurar las relaciones y responsabilidades dentro del sistema.
- Codificación: La programación se realiza en parejas frente al mismo equipo de trabajo, práctica que favorece la colaboración y el control de calidad. Las duplas pueden rotarse para asegurar que el código mantenga coherencia y sea comprensible para cualquier desarrollador, logrando así un producto uniforme, organizado y bien estructurado.
- Pruebas: En proyectos ágiles o de corta duración, es fundamental que las pruebas se lleven a cabo con frecuencia y de manera automatizada. Asimismo, se promueve que el cliente participe, ya que puede llevar a cabo comprobaciones directas, proponer modificaciones y analizar versiones intermedias del software.
- Lanzamiento: Esta fase se alcanza cuando todas las historias de usuario o versiones parciales han sido validadas con éxito, cumpliendo las especificaciones establecidas. El resultado final es un software funcional, confiable y alineado con las necesidades del cliente, listo para su integración con el producto principal.

2.2.14. Adaptación de PXP al Contexto Agrícola-Tecnológico

- La implementación de PXP en este proyecto requirió adaptaciones específicas para atender la naturaleza híbrida hardware-software y el contexto agrícola:
- Ciclos híbridos de desarrollo: tanto como para hardware y software permitiendo así poder conectar entre los dos, y tener unos buenos registros de operaciones exitosas (logs con estado "pass").

- Personal Xp: desarrollo para una sola persona, desarrollando por agentes sin desviarse de tema para mantener el enfoque y la eficiencia en el avance del proyecto.
- Validación en contexto real: se realizó en contexto real, aplicando historias de usuario específicamente orientadas a la fenología del cultivo, garantizando que las funcionalidades desarrolladas respondieran adecuadamente a las necesidades agronómicas y operativas detectadas.

Tabla 4. Implementación de PXP en el Proyecto de Riego Inteligente

Fase PXP	Actividades Realizadas	Artefactos Generados	Participantes Involucrados
Planificación	<ul style="list-style-type: none"> - Censo con técnicos encargados del invernadero para identificar necesidades de riego - Creación de historias de usuario sobre detección de fases y control de riego - Ordenamiento de funcionalidades según dependencias técnicas - Estimación colaborativa de tiempos de desarrollo - Creación de arquitectura distribuida en cuatro agentes 	<ul style="list-style-type: none"> - Catálogo de historias de usuario - Calendario de entregas - Plan de iteraciones - Diagrama preliminar de arquitectura multi-agente 	<ul style="list-style-type: none"> - técnicos invernadero - Equipo de desarrollo - Ingenieros agrónomos asesor
Diseño	<ul style="list-style-type: none"> - Diseño del esquema de datos para Firebase - Elaboración de interfaces para sistema de monitoreo y app móvil - Definición de protocolos de comunicación entre componentes - Desarrollo del sistema de visión artificial 	<ul style="list-style-type: none"> - Diagramas de arquitectura modular - Modelo relacional y no-relacional - Prototipos de interfaz en Tkinter y Flutter - Documentación de protocolos JSON 	<ul style="list-style-type: none"> - Programador principal - sistemas embebidos Microcontrolador IoT agrícola
Codificación	<ul style="list-style-type: none"> - Integración continua entre subsistemas mediante pruebas diarias - Programación modular con interfaces claras entre agentes - Refactorización de componentes críticos - Construcción de pruebas unitarias para cada función crítica 	<ul style="list-style-type: none"> - Repositorio Git con integración continua - Código fuente documentado - Pruebas automatizadas - Scripts de comunicación entre agentes 	<ul style="list-style-type: none"> - Programador Python Para visión computacional - Experto en aprendizaje automático
Pruebas	<ul style="list-style-type: none"> - Simulación de comunicación entre agentes en entorno controlado - Validación de detección de fases con plantas reales - Pruebas de resistencia para componentes IoT 	<ul style="list-style-type: none"> - Batería de pruebas automatizadas - Informes de precisión de modelos - Métricas de fiabilidad de comunicaciones - Registro de condiciones límite 	<ul style="list-style-type: none"> - Control de calidad - técnicos encargados colaboradores - Asesor en cultivos de invernadero
Implementación	<ul style="list-style-type: none"> - Instalación incremental de componentes en invernadero piloto 	<ul style="list-style-type: none"> - Manual de usuario ilustrado 	<ul style="list-style-type: none"> - Técnico instalador - Agricultor Usuarios - soporte técnico

- Ajuste y calibración de sensores según condiciones locales	- Protocolos de mantenimiento preventivo
- Capacitación práctica a usuarios finales	- Documentación técnica de interfaces
- Monitoreo durante primeras semanas de operación	- Sistema de respaldo y recuperación

2.2.15. Herramientas de Desarrollo

Son programas esenciales que optimizan la creación, prueba y gestión de proyectos. Incluyen entornos de desarrollo (IDE), editores de código y sistemas de control de versiones, clave para la productividad y calidad del software. El proceso a menudo comienza con el modelado, usando diagramas UML para definir funcionalidades entre desarrolladores y clientes. Estas herramientas facilitan la colaboración y son cruciales para un desarrollo eficiente y sin fallos, especialmente en equipos dispersos (Caiza, Guaman, & Lopez, 2024).

- GitHub: Es una plataforma indispensable para los desarrolladores, que permite la colaboración y el control de versiones en proyectos de software. Es una opción muy popular para manejar cualquier proyecto de código abierto o privado, ya que, con una comunidad masiva, hace más sencillo seguir los cambios en el código y proporciona instrumentos sólidos para trabajar en equipo, lo que optimiza la eficiencia y el flujo de trabajo (Fernandez, 2023).
- Visual Studio Code: Es la interfaz para poder crear código ya sea de diferentes lenguajes de programación usando Frameworks, valorado siendo ligero, rápido y personalizable. Destaca por su capacidad de extensión, autocompletado inteligente, depuración integrada y control de versiones con Git, lo que lo hace una herramienta versátil y accesible para cualquier desarrollador (Cimas, 2022).

2.2.16. Lenguajes de Programación

Es el modo en que convertimos ideas en algoritmos complejos, los cuales se combinan para crear todas las aplicaciones y páginas web que empleamos todos los días, desde YouTube hasta WhatsApp. A pesar de que las máquinas solo comprenden un lenguaje binario compuesto por ceros y unos, estos lenguajes posibilitan que redactemos código de forma clara y eficaz; funcionan como el puente que regula la conducta lógica y física de los dispositivos para llevar a cabo funciones concretas (Santander, Santander, 2021).

Tabla 5. Framework de aplicaciones móviles

Framework	Flutter	React Native	Ionic	PhoneGap	NativeScript
Lenguajes soportados	Dart	JavaScript, TypeScript	JavaScript, TypeScript, HTML	HTML, CSS, JavaScript	JavaScript, TypeScript
Plataformas compatibles	iOS, Android, Web, Desktop (Windows, macOS, Linux)	iOS, Android	iOS, Android, Web	iOS, Android	iOS, Android
Tipo de aplicación	Nativo/ Multiplataforma	Nativo/ Multiplataforma	Híbrido	Híbrido	Nativo/ Multiplataforma
Rendimiento	Alto (cerca del nativo)	Medio-Alto	Medio	Bajo-Medio	Medio
Popularidad	Muy alta (en rápido crecimiento)	Alta	Alta	Media-Baja	Media

Tabla 6. Framework ventajas y desventajas

Framework	Ventajas	Desventajas
Flutter	Alto rendimiento, Desarrollo rápido, Diseño atractivo y personalizable	Tamaño de archivo grande, Menos bibliotecas nativas
React Native	Amplia comunidad, Código reutilizable, Buen rendimiento	Dependencia de módulos nativos, Menos personalización de UI
Ionic	Fácil desarrollo multiplataforma, Amplio soporte de plugins	Rendimiento inferior al nativo, Dependencia de WebView
PhoneGap	Basado en tecnologías web estándar, Desarrollo sencillo	Bajo rendimiento, Limitaciones en personalización
NativeScript	Acceso directo a APIs nativas, Soporte para Angular y Vue.js	Menos comunidad, Menos documentación disponible

2.2.17. Lenguajes y Entornos de Desarrollo

Para construir sistemas agrícolas inteligentes, es esencial seleccionar los entornos de desarrollo y los lenguajes de programación correctos, garantizando que las soluciones sean eficaces, sencillas de mantener y aptas para expandirse. Por ejemplo, en este proyecto se eligió una arquitectura distribuida, lo que implicó utilizar diferentes tecnologías particulares para cada componente del sistema (Gomstyn & Jonker, 2023).

2.2.17.1. Python y Spyder con Interfaz Tkinter

La importancia de Python en el desarrollo de sistemas agrícolas inteligentes radica en su balance entre la facilidad de utilización y la potencia para el procesamiento de datos e IA combinando diferentes elementos. El marco Tkinter, para interfaces

gráficas se puede crear usando el estándar de Python, lo que permite desarrollar aplicaciones funcionales y multiplataforma que ven nativas en cualquier sistema operativo. hace más sencillo integrar componentes heterogéneos gracias a su versatilidad y su capacidad de trabajar con diversas plataformas (Amos, 2024).

El entorno de desarrollo Spyder nos brinda herramientas para programación científica en Python, incluyendo:

- Editor con resaltado de sintaxis y análisis de código
- Consola interactiva para pruebas incrementales
- Explorador de variables para inspección de estructuras de datos
- Herramientas de depuración y perfilado
- Integración con bibliotecas científicas como NumPy, SciPy y Matplotlib

Tkinter, una biblioteca que viene incluida con Python, es una excelente opción para diseñar interfaces gráficas que funcionen en cualquier sistema operativo. Esta herramienta posibilita que hagamos aplicaciones de monitoreo y control fácilmente, incluso para empleos en entornos de campo, porque necesita pocos recursos del sistema, lo cual es un beneficio esencial para trabajar con eficiencia (Amos, 2024).

En Agente 1 (Sistema de Visión y Predicción), se usa Python-Spyder-Tkinter así permitió implementar:

- Procesamiento de imágenes mediante OpenCV
- Integración con modelos TensorFlow/Keras
- Comunicación con bases de datos Firebase y MySQL
- Interfaz de usuario para monitoreo en tiempo real
- Gestión de hilos para procesamiento concurrente

2.2.17.2. Flutter/Dart

Flutter, un marco de interfaz de usuario versátil y de código abierto creado por Google, es una potente herramienta para crear aplicaciones que operan en varias plataformas, incluidas Android, iOS, Windows, Linux, macOS y la web. Flutter, que se basa en el lenguaje Dart, es conocido por su habilidad de garantizar una experiencia de usuario consistente y de gran calidad sin tener en cuenta el dispositivo. Esta adaptabilidad no solo acelera el desarrollo, sino que lo hace perfecto para

aplicaciones que requieren trabajar con una gran variedad de dispositivos (Shapovalov, 2023).

Dart, Es un lenguaje de programación creado por Google que es open source y fue diseñado para proveer a los programadores una herramienta con análisis de tipo estático y orientada a objetos. Se distingue por rasgos como el tipado estático opcional, la gestión de memoria automática (recolección de basura) y la compatibilidad con programación asíncrona. Estas características lo convierten en la opción ideal para desarrollar interfaces de usuario que interactúan de manera eficaz con servicios externos y responden con rapidez, por lo que es un elemento clave para el desarrollo de aplicaciones móviles sólidas y ágiles. (Martinez, 2021).

Estas herramientas ayudaron a crear el Agente 4 (Aplicación Móvil), permitiendo implementar:

- Interfaces de usuario adaptativas para diferentes dispositivos
- Comunicación en tiempo real con Firebase Real time Database
- Integración con APIs de servicios web (Gemini)
- Gestión eficiente de recursos en dispositivos móviles
- Manejo de entrada multimodal (texto, voz, imágenes)

2.2.17.3. Arduino IDE para ESP32

El ESP32 es una solución que se destaca por su bajo consumo energético y costo. Se trata de un "System on Chip" eficaz, desarrollado por Espressif Systems, que resulta perfecto para incorporar la conectividad en diferentes proyectos. El entorno hace que su programación se vuelva considerablemente más sencilla. de desarrollo Arduino IDE. Al proporcionar un lenguaje basado en C++ y un entorno intuitivo para trabajar, este sistema ha transformado el desarrollo de sistemas embebidos al facilitar el acceso directo a las capacidades del hardware, lo que hace posible que los expertos y los principiantes implementen sus ideas de manera más eficaz y accesible. (Lozano, 2021).

La adaptación del IDE Arduino para la plataforma ESP32 permitió aprovechar su simplicidad manteniendo acceso a las capacidades avanzadas del microcontrolador, incluyendo:

- Gestión de conectividad WiFi y Bluetooth

- Implementación de clientes HTTPS para comunicación segura
- Manejo de protocolo MQTT para comunicación IoT
- Control de pines GPIO para sensores y actuadores
- Implementación de modos de bajo consumo

Esta combinación de tecnologías fue ideal para el Agente 3 (Sistema de Control IoT). Con un código eficiente y sencillo de mantener, logramos controlar electroválvulas, conectar con servicios en la nube e integrar sensores ambientales sin dificultad.

2.2.18. Base de Datos

Se refiere al registro o almacenamiento de un sistema de información en la era digital y el Big Data, lo que posibilita a todas las aplicaciones, sean de escritorio o móviles, procesar y guardar volúmenes amplios de datos. Su función primordial es funcionar como sistemas organizados para la administración de datos, lo que es esencial para el análisis y la operación de información en el contexto digital actual (Larque, 2023).

2.2.18.1. Tipos de base de Datos

- DB-NoSQL
- DB-objetos
- DB-grafos
- DB-memoria

2.2.18.2. Ventajas de mejorar el consumo de DB

- Velocidad: Consultas y operaciones más rápidas.
- Eficiencia: Mejor uso de recursos (CPU, RAM, disco).
- Escalabilidad: Soporta más usuarios y datos.
- Disponibilidad: Menos caídas o lentitud.
- Experiencia de usuario: Aplicaciones más fluidas.
- Productividad: Operaciones empresariales ágiles.
- Reducción de costos: Menos necesidad de hardware costoso.
- Toma de decisiones: Acceso rápido a información crucial.
- Ventaja competitiva: Agilidad en el mercado.

2.2.18.3. Comparativa de Tecnologías de Bases de Datos

La selección de tecnologías de almacenamiento y gestión de datos constituye un aspecto crítico en sistemas agrícolas inteligentes, que deben manejar información

heterogénea con diferentes patrones de acceso. La Tabla 3 presenta una comparativa de las principales alternativas evaluadas para este proyecto.

Tabla 7. Comparativa de Bases de Datos

Características	MySQL	Firestore Database	Realtime Database	MongoDB	SQLite
Modelo de datos	Relacional	NoSQL orientado a documentos JSON	NoSQL orientado a documentos BSON	Relacional	
Escalabilidad	Vertical con limitaciones en horizontal	Alta escalabilidad horizontal automática	Alta escalabilidad horizontal con sharding	Limitada (embebida en aplicación)	
Sincronización en tiempo real	No nativa, requiere polling	Nativa con listeners y WebSockets	Requiere implementación adicional	No soportada	
Operación offline	No soportada	Soportada con sincronización automática	Requiere implementación manual	Nativa (base local)	
Lenguaje de consulta	SQL estándar	Consultas basadas en REST y JSON	MongoDB Query Language	SQL con limitaciones	
Transacciones ACID	Completas	Parciales (consistencia eventual)	Disponibles desde v4.0	Completas	
Autenticación y autorización	Externa, requiere implementación	Integrada con Firebase Auth	Requiere configuración	A nivel de aplicación	
Hospedaje	Auto-hospedado o servicios cloud	Servicio gestionado por Google	Auto-hospedado o Atlas (cloud)	Embebido en aplicación	
Uso de recursos	Moderado a alto	Bajo en cliente, gestionado en cloud	Moderado a alto	Muy bajo	
Idoneidad IoT	Limitada para grandes volúmenes	Alta para actualizaciones frecuentes	Media, mejor para analítica	Alta para dispositivos limitados	

Tras el análisis comparativo, este proyecto implementó una arquitectura híbrida que combina:

2.2.18.4. Firestore Database

Para datos en tiempo real que requieren sincronización rápida entre componentes del sistema (sensores, estado de cultivo, notificaciones).

2.2.18.5. MySQL

Para registros históricos, configuración del sistema y datos transaccionales que requieren integridad referencial.

Esta combinación aprovecha las fortalezas complementarias de ambos sistemas: la capacidad de Firebase para sincronización en tiempo real y su modelo de datos flexible, junto con la consistencia transaccional y capacidades analíticas de MySQL.

2.2.19. Integración de Tecnologías en Arquitectura Multi-Agente

La diversidad de tecnologías empleadas responde a los requerimientos específicos de cada componente del sistema. Según González-Briones et al. (2018), la heterogeneidad tecnológica en sistemas multi-agente permite optimizar cada componente para su función particular, manteniendo interfaces estandarizadas para la interoperabilidad.

La arquitectura implementada establece un flujo de datos coherente entre componentes heterogéneos mediante:

- Serialización JSON para intercambio de datos entre agentes
- APIs RESTful para comunicación cliente-servidor
- WebSockets para notificaciones en tiempo real
- Esquemas de datos compartidos entre diferentes tecnologías

Esta integración tecnológica facilita el mantenimiento y extensión del sistema, permitiendo reemplazar o actualizar componentes individuales sin afectar al funcionamiento global de la solución.

2.2.20. Gestión del agua de riego en cultivos bajo invernadero

El cultivo en invernadero, a pesar de su eficiencia y rentabilidad, enfrenta un desafío crítico: su elevada y constante demanda de agua, que varía según la especie, tamaño, temperatura y condiciones atmosféricas. Es sorprendente que una planta necesite cientos de libras de agua para producir una sola libra de materia orgánica seca, y aún más, que más del 99% de esa agua se pierda en la atmósfera. La contaminación y la falta de agua, especialmente en áreas como la cuenca mediterránea, junto con esta notable ineficiencia, convierten en una prioridad primordial optimizar el manejo del agua para el riego en los invernaderos. Proyectos como los de la Estación Experimental Las Palmerillas, en Almería, que estudian cómo optimizar el uso del agua en cultivos de hortalizas, brindan experiencias importantes y son esenciales para garantizar la sostenibilidad agrícola, cuidar nuestros recursos naturales y garantizar la producción alimentaria futura. (Perez, 2021).

2.2.21. Importancia del riego en ambientes controlados

El riego, en particular en entornos controlados, se ha vuelto un elemento esencial para el éxito y la sostenibilidad de la agricultura moderna. Es fundamental administrar el agua de riego de manera precisa y eficaz en un entorno en el que la seguridad alimentaria a nivel mundial es prioritaria. Al valorar y preservar la calidad del agua, desde su composición química hasta sus propiedades sanitarias, se logra mejorar la producción agrícola y reducir los peligros relacionados. Una gestión apropiada de la calidad del agua de riego asegura no solo una mejora en la salud de las plantas y el rendimiento de los sistemas de riego, sino también un abastecimiento sostenible de alimentos para la población global en ascenso (Juarez, 2024).

2.2.22. Sostenibilidad del riego en invernaderos

La sostenibilidad del riego en invernaderos se refiere a la capacidad de mantener un uso responsable, equilibrado y duradero del recurso hídrico dentro de sistemas de cultivo protegidos, asegurando la producción agrícola sin comprometer el agua disponible para futuras generaciones ni afectar negativamente al medio ambiente. Esta sostenibilidad implica varios aspectos clave (ChatGPT, 2025).

- Uso eficiente del agua
- Gestión integral del recurso hídrico
- Recolección y reutilización de aguas lluvias o drenajes.
- Tecnologías de la información
- Impacto ambiental y conservación

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque

Esta investigación se fundamenta en un enfoque mixto, el cual integra estrategias de carácter cualitativo y cuantitativo tiene el propósito de adquirir la comprensión más directa del fenómeno estudiado. De acuerdo con (Otero Ortega, 2018), dicho enfoque combina recopilación de distintos datos, para conllevar a un análisis para poder interpretarlos de ambos tipos, permitiendo al investigador seleccionar los procedimientos que se ajusten mejor a las necesidades del estudio.

- Aspecto cualitativo: Esta dimensión se orienta a la observación detallada y al análisis de las posibles anomalías en los cultivos bajo invernadero, tales como variaciones en el desarrollo o crecimiento de las plantas. (Sánchez Flores, 2019), Según la investigación cualitativa parte del reconocimiento de una realidad dinámica, subjetiva y contextual, lo que permite interpretar fenómenos dentro de entornos complejos.
- Aspecto cuantitativo: Se enfoca en la recopilación de datos que puedan medirse y verificarse, como las condiciones ambientales específicas o la cantidad de agua utilizada en el sistema de riego (Hernández-Sampieri, Fernández-Collado, & Baptista Lucio, 2014), consiste en superponer la lógica empírico-deductiva, la cual se fundamenta en el empleo de métodos experimentales y en un uso estricto de instrumentos estadísticos.

3.1.2. Tipo de Investigación

Para llevar a cabo el presente estudio, se tomaron en cuenta diversos métodos que para su validez:

- Investigación exploratoria: Hizo posible reconocer las variables fundamentales vinculadas con el asunto, como la calidad del agua, sus características químicas y la sostenibilidad del sistema. Esta etapa temprana fue esencial para

- establecer una base conceptual firme, entendiendo el contexto agrícola actual y la importancia del problema tratado.
- Investigación de campo: Se trató de la realización de encuestas y entrevistas a personal especializado para obtener datos empíricos que ayuden a resolver el problema identificado en la formulación del problema.

Observación participante: Esta técnica posibilitó la interacción directa con el equipo que participaba en las labores agrícolas, lo que permitió una mejor comprensión de las dinámicas auténticas del entorno. Con el fin de recoger información específica sobre los invernaderos y de mejorar el análisis del fenómeno estudiado, se llevaron a cabo visitas técnicas a varios invernaderos de la provincia del Carchi.

3.2. IDEA A DEFENDER

El desarrollo de una solución informática para la gestión del agua optimizará el crecimiento y producción de cultivos bajo invernadero, evitando pérdidas considerables de agua en la Unidad Educativa 'Jorge Martínez Acosta'.

3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

3.3.1 Definición de Variables

Variable Independiente

- Solución Informática

Es la solución tecnológica perfecta para solucionar dificultades, mejorar procedimientos o cumplir objetivos de manera eficaz. Incluye componentes técnicos como datos, hardware y software, además de servicios esenciales como la integración, el soporte y la consultoría. Piensa en ella como esa "idea brillante" que facilita la optimización de tus operaciones, desde sistemas ERP hasta instrumentos de automatización (Kortas, 2025).

Variable Dependiente

- Control de gestión del agua en cultivos bajo invernadero

Este enfoque es vital porque impacta directamente en la cosecha, la salud del ecosistema y nuestra seguridad alimentaria. Ante diferentes cambios constantes los cambios climáticos y la creciente demanda de alimentos, es imperativo que los agricultores adopten prácticas más inteligentes. Precisamente, integrar información y tecnología de agricultura de precisión permite usar el agua justa.

Tabla 8. Operación de variables

Variables	Concepto	Dimensiones	Indicador	Técnicas	Instrumento
Independiente: Solución informática	Facilitar el trabajo al técnico encargado, optimizando la producción y evitando pérdidas masivas de agua de riego en los cultivos bajo invernadero mediante un sistema inteligente multi-agente.	Software Hardware Agentes IA Interfaces	<ul style="list-style-type: none"> • Funcionalidad • Eficiencia • Usabilidad • Confiabilidad • Portabilidad • Interoperabilidad 	Encuesta Observación Experimentación Análisis de datos	<ul style="list-style-type: none"> • Computador • ESP32 • Sensores IoT • Cámara USB
Dependiente: Control de Gestión del agua de riego en cultivos bajo invernadero	Se llevará a cabo a partir de la toma de decisiones basada en los registros de riego generados acorde a las fases del cultivo, utilizando modelos predictivos entrenados y sistema de monitoreo en tiempo real.	Temperatura Humedad Desarrollo Floración Producción Gestión de riego	<ul style="list-style-type: none"> • Eficiencia de Riego • Fase del Cultivo • Tiempo de Riego • Precisión de Predicción 	<ul style="list-style-type: none"> • Observación de sensores • Medición de imágenes • Validación cruzada de modelos 	<ul style="list-style-type: none"> • Formulario • Registro • Sensores (Humedad, Temperatura.) • Cámara USB • Modelos TensorFlow

3.4. MÉTODOS UTILIZADOS

3.4.1. Métodos

Método Inductivo y Deductivo

- Inductivo

Es aquel razonamiento que te lleva de lo particular a lo general. Funciona observando casos específicos y patrones repetidos, lo que te permite construir y extraer conclusiones amplias o teorías. Esencialmente, es como juntar varias piezas pequeñas para formar una idea grande, permitiéndote generar hipótesis y comprender fenómenos a partir de la suma de tus observaciones (Lopez & Paul, 2025).

- Deductivo

Se trata de un proceso lógico que se caracteriza por ir de lo general a lo específico, empleando principios previamente establecidos para examinar y entender situaciones individuales. Se trata de un camino directo y estructurado, el cual es apreciado por su precisión en áreas formales como las matemáticas, donde se descomponen verdades amplias para descubrir certezas específicas (Suarez, 2024).

3.4.2. Técnicas

Para recolectar datos de tipo cualitativo y cuantitativo, se emplearon encuestas y entrevistas.

3.4.2.1. Entrevista

Es un método de investigación empírica crucial en nuestro estudio. La vemos no solo como una técnica, sino como un diálogo directo y estratégico que nos posibilita explorar a fondo los puntos de vista individuales (Feria, Matilla, & Mantecon, 2020).

Se realizó una entrevista a los docentes encargados sobre el uso de tecnología en la gestión del riego en invernaderos, con el fin de recolectar información a través de una guía de preguntas sobre el proceso actual que se realizó en la Unidad Educativa Jorge Martínez Acosta en la ciudad de San Gabriel.

3.4.2.2. Encuesta

Es una potente herramienta de investigación empírica. Además de ser una simple

En términos de técnica, la percibimos como un procedimiento directo para recopilar información que nos facilita comprender en profundidad un fenómeno (Feria, Matilla, & Mantecon, 2020).

Se realizó una encuesta a todos los alumnos de formación técnica de la Unidad Educativa Jorge Martínez Acosta, ubicada en San Gabriel, con el fin de obtener información detallada acerca de sus vivencias y percepciones en relación con el control de gestión del agua para riego en cultivos bajo invernadero. Se alcanzaron respuestas valiosas mediante un cuestionario estructurado, que trató temas fundamentales en el funcionamiento de los procesos actuales y sus expectativas. Esta encuesta ofreció una perspectiva integral y representativa de lo que los alumnos saben y necesitan en este ambiente productivo y educativo.

3.5. ANÁLISIS ESTADÍSTICO

3.5.1. Población

Toma énfasis a un grupo o conjunto por el cual se pretende obtener accesibilidad para realizar un estudio. Constituye el universo total de inquilinos, eventos o complementos que poseen que nos interesen para realizar el estudio y que pertenecen al espacio específico donde se desarrolla nuestra investigación; de ella buscamos extraer toda la información necesaria para comprender a fondo un fenómeno o problema, y sobre la cual se aplicarán las conclusiones obtenidas (Condori, 2020).

En la presente investigación se consideró como población al total de los estudiantes técnicos de la Unidad Educativa Jorge Martínez Acosta de la ciudad de San Gabriel, compuesto por 164 alumnos. Dado que este número de integrantes se considera manejable desde el punto de vista estadístico, se optó por trabajar con la población completa. Esta decisión nos permitió realizar un estudio censal, lo que a su vez garantizó un mayor grado de precisión, evitando así la necesidad de aplicar fórmulas de muestreo y fortaleciendo el análisis cuantitativo al disponer de datos directos de todos los participantes involucrados. A partir de la información recopilada, se aplicaron procedimientos estadísticos descriptivos, como frecuencias, porcentajes y gráficos de distribución, con el objetivo de identificar patrones relevantes y establecer conclusiones fundamentadas en datos reales y completos del contexto educativo.

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Resultados de la encuesta

Según las encuestas efectuadas, la creación de una solución informática ayudará a mejorar notablemente el control manual de la gestión del agua en los cultivos bajo invernadero, logrando una optimización efectiva del recurso hídrico. Además, los hallazgos mostraron que la investigación en la Unidad Educativa Jorge Martínez Acosta, de San Gabriel, es factible y cuenta con el respaldo de alumnos técnicos y profesores responsables

4.1.2. Fundamentación Teórica del Censo

Según Hernández Sampieri, el censo es una técnica de investigación que supone recopilar información acerca de cada uno de los sujetos pertenecientes a una población específica. Esta técnica se diferencia de los métodos de muestreo al incluir a cada miembro de la población, eliminando así posibles sesgos y proporcionando datos completos y precisos.

La aplicación del censo en este estudio se justifica por tres criterios fundamentales establecidos por Hernández Sampieri:

- La población es relativamente pequeña (164 estudiantes)
- Se requiere un alto nivel de precisión en los resultados
- Es necesaria información exhaustiva sin riesgos de errores muestrales

4.1.3. Caracterización de la Población

La población objeto de estudio está constituida por 164 estudiantes de bachillerato técnico que:

- Tienen acceso directo al invernadero institucional
- Realizan prácticas agrícolas como parte de su formación

4.1.4. Aplicación del Censo en el Estudio

La aplicación del censo a los 164 estudiantes permite:

- Obtener una imagen precisa y detallada de las percepciones
- Eliminar sesgos asociados a técnicas de muestreo
- Garantizar la representatividad total de la población estudiantil

Fase Preparatoria

- Validación del instrumento de encuesta
- Coordinación con autoridades educativas
- Planificación logística para alcanzar al 100% de la población

Fase de Ejecución

- Aplicación sistemática de la encuesta a los 164 estudiantes
- Registro detallado de participación para garantizar cobertura total
- Documentación del proceso de recolección

Fase de Procesamiento

- Tabulación completa de todas las encuestas
- Análisis exhaustivo de datos recolectados
- Generación de información estadística integral

4.1.5. Control de Calidad del Censo

- Control de participación de cada técnico
- Registro de estudiantes pendientes
- Estrategias de seguimiento para casos especiales
- Revisión de completitud de encuestas
- Verificación de coherencia en respuestas
- Documentación de incidencias

4.1.6. Aspectos Éticos y de Confidencialidad

- Protección de datos personales de los estudiantes
- Consentimiento informado de participación
- Uso exclusivo de información para fines investigativos

4.1.7. Observación

Análisis de la correlación Spearman

- Correlaciones muy fuertes: obtuvimos una correlación bastante fuerte de la pregunta 7 (notificaciones en WhatsApp sobre el estado del riego) y la pregunta 10 (sistema de riego utilizado actualmente), con un coeficiente $\rho=0.955$, lo que indica una relación perfecta entre aquellas variables.
- Correlaciones fuertes: Hay una fuerte correlación entre la familiaridad con tecnología agrícola (Q2) y el sistema de riego utilizado (Q10): $\rho=0.638$
- La percepción sobre la utilidad de la automatización del riego (Q8) muestra una fuerte correlación con el sistema de riego utilizado (Q10): $\rho=0.635$
- También existe una fuerte relación entre la preferencia por notificaciones (Q7) y la percepción de utilidad de la automatización (Q8): $\rho=0.601$

Correlaciones moderadas:

- Todas las demás preguntas muestran correlaciones moderadas con el sistema de riego utilizado (Q10), con valores entre 0.548 y 0.598
- La familiaridad con tecnología agrícola (Q2) y la preferencia por notificaciones (Q7) muestran una correlación moderada: $\rho=0.584$

Interpretación de resultados

- El sistema de riego que utilizan actualmente los estudiantes (Q10) parece ser un factor determinante que influye en sus percepciones y preferencias sobre las tecnologías de riego.
- Los estudiantes que muestran mayor familiaridad con tecnologías agrícolas tienden a utilizar sistemas de riego más avanzados y tienen mayor interés en recibir notificaciones sobre el estado del riego.
- La matriz de correlación de Spearman revela asociaciones mucho más fuertes entre las variables que la correlación de Pearson, lo que indica que el método de Spearman es más apropiado para este tipo de datos de encuesta.

Diferencias con la correlación Pearson

- Trabaja con rangos en lugar de valores exactos, lo que es ideal para datos ordinales de encuestas
- Es más robusto ante valores atípicos
- No ejecuta relaciones, solo se basa en la monotonía.

- Detecta mejor las relaciones en datos que no siguen una distribución normal

Matriz de Correlación Spearman - Encuestas

- Este en cambio mide la fuerza entre dos variables, determinando así lo más importante, hasta llegar a lo menos importante, usando sus rangos en lugar de sus valores exactos. Es ideal para datos ordinales como los de encuestas.

Interpretación: Los valores oscilan entre -1 y +1, donde:

- +1 variable perfecta positiva: si una de las variables aumenta, la otra también tiende a aumentar
- -1 variable perfecta negativa: si una de las variables aumenta la otra tiende a disminuir
- 0 muestra que no se encuentra variables que se relacione

4.1.8. Formula de Spearman

$$p = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)}$$

Donde

- (p): Es la obtención del resultado que se obtiene entre las dos variables
- (di): Se obtiene al restar las observaciones de ambas variables en orden.
- (n): Numero de observaciones.
- ($\sum d_i^2$): Suma de los cuadrados en diferencias a sus rangos.

Variables	Q1: Año de bachillerato	Q2: Familiaridad con tecnología	Q3: Importancia de tecnología	Q4: Utilidad de sensores	Q5: Cámaras e IA para riego	Q6: Facilidad uso app móvil	Q7: Notificaciones WhatsApp	Q8: Utilidad automatización	Q9: Beneficio de IA predictiva	Q10: Sistema de riego actual
Q1: Año de bachillerato	1.000	0.281	0.127	0.175	0.075	0.127	0.507	0.163	0.033	0.556
Q2: Familiaridad con tecnología	0.281	1.000	0.241	0.208	0.268	0.149	0.601	0.284	0.160	0.655
Q3: Importancia de tecnología	0.127	0.241	1.000	0.223	0.236	0.224	0.577	0.205	0.125	0.609
Q4: Utilidad de sensores	0.175	0.208	0.223	1.000	0.139	0.081	0.540	0.264	0.172	0.584
Q5: Cámaras e IA para riego	0.075	0.268	0.236	0.139	1.000	0.209	0.551	0.257	0.150	0.592
Q6: Facilidad uso app móvil	0.127	0.149	0.224	0.081	0.209	1.000	0.533	0.178	0.180	0.575
Q7: Notificaciones WhatsApp	0.507	0.601	0.577	0.540	0.551	0.533	1.000	0.609	0.503	0.982
Q8: Utilidad automatización	0.163	0.284	0.205	0.264	0.257	0.178	0.609	1.000	0.161	0.640
Q9: Beneficio de IA predictiva	0.033	0.160	0.125	0.172	0.150	0.180	0.503	0.161	1.000	0.551
Q10: Sistema de riego actual	0.556	0.655	0.609	0.584	0.592	0.575	0.982	0.640	0.551	1.000

■ -1.0 ■ -0.5 0 ■ 0.5 ■ 1.0

Figura 1. Cuadro de resultados

4.1.9. Correlaciones más significativas

Variable 1	Variable 2	Coefficiente ρ	Fuerza	Descripción
Q7	Q10	0.962	Muy fuerte	Notificaciones WhatsApp - Sistema de riego actual
Q2	Q10	0.655	Moderada	Familiaridad con tecnología - Sistema de riego actual
Q8	Q10	0.640	Moderada	Utilidad automatización - Sistema de riego actual
Q7	Q8	0.609	Moderada	Notificaciones WhatsApp - Utilidad automatización
Q3	Q10	0.609	Moderada	Importancia de tecnología - Sistema de riego actual
Q2	Q7	0.601	Moderada	Familiaridad con tecnología - Notificaciones WhatsApp

Figura 2. Correlaciones

Interpretación de los resultados

- Correlación muy fuerte ($\rho = 0.962$): Entre notificaciones en WhatsApp (Q7) y sistema de riego actual (Q10). Esto sugiere que los estudiantes que utilizan sistemas de riego específicos tienen una fuerte preferencia por recibir notificaciones sobre el estado del riego.
- Correlaciones fuertes y moderadas con el sistema de riego (Q10): El sistema de riego actual muestra correlaciones positivas significativas con casi todas las demás variables. Esto indica que la experiencia con sistemas de riego específicos está fuertemente relacionada con las opiniones sobre la tecnología agrícola en general.
- Correlación entre año de bachillerato técnico y familiaridad ($\rho = 0.281$): Existe una correlación débil positiva entre el año de bachillerato y la familiaridad con tecnología agrícola, sugiriendo que los estudiantes de cursos superiores tienden a estar ligeramente más familiarizados con estas tecnologías.
- Efecto central de las notificaciones (Q7): La variable relacionada con las notificaciones por WhatsApp muestra correlaciones moderadas a fuertes con todas las demás variables, lo que sugiere que el interés en recibir notificaciones es un buen predictor de la receptividad general hacia la tecnología de riego.

4.2. PROPUESTA

Desarrollar una solución informática basada en tecnologías inteligentes para optimizar la gestión del agua en cultivos bajo invernadero, mediante la implementación de un sistema multi-agente denominado AgroPredict que integre monitoreo ambiental en tiempo real, control automatizado del riego, asistencia agronómica y predicción de necesidades hídricas, con la finalidad de minimizar el

consumo hídrico y mejorar el desarrollo de cultivos en la Unidad Educativa “Jorge Martínez Acosta”.

4.2.1. Fase de planificación

En la reunión inicial, se definieron los requisitos que debía cumplir la solución informática, identificando los objetivos principales de los cultivos bajo invernadero en la Unidad Educativa Jorge Martínez Acosta y sus necesidades en relación con el desarrollo del software. Se dio prioridad a elementos como la sencillez de uso, la funcionalidad y la adaptación a los procedimientos internos de administración del agua para riego. Se supone que el objetivo principal de este encuentro es adquirir información clave sobre cómo se controla el agua destinada al riego en la operación del invernadero. Los estudiantes técnicos y los maestros a cargo brindaron información acerca de los procedimientos vigentes, como la medición de humedad, la organización del riego y las técnicas utilizadas para analizar el consumo de agua. Para identificar zonas de mejora y desarrollar una solución informática optimizada que mejore la gestión del agua para riego, asegurando su adecuación a las demandas de los cultivos y a la sostenibilidad, es esencial este entendimiento.

4.2.2. Levantamiento de Procesos

Los resultados del levantamiento de procesos para controlar la gestión del agua de riego en los cultivos invernaderos de la Unidad Educativa Jorge Martínez Acosta se presentan a continuación.

Tabla 9. Historias de Usuario

Historia de Usuario 1	
Número	1
Nombre Historia	Sistema de Riego Inteligente Integrado
Prioridad	Crítica
Funcionalidad	<ul style="list-style-type: none"> • Activar riego automáticamente según fase del cultivo detectada • Enviar notificación WhatsApp inmediata al iniciar riego • Mostrar estado actual del riego en tiempo real
Criterio de aceptación	El sistema debe activar el riego en menos de 5 segundos y enviar notificación WhatsApp con 95% de éxito de entrega
Historia de Usuario 2	
Número	2
Nombre Historia	Control de Automatización Familiar
Prioridad	Alta

Funcionalidad	<ul style="list-style-type: none"> • Interfaz con iconos grandes y descriptivos • Manual integrado para nuevos usuarios
Criterio de aceptación	90% de estudiantes deben usar el sistema sin consultar manual y completar configuración en menos de 10 minutos
Historia de Usuario 3	
Número	3
Nombre Historia	Detección Automática de Fases de Cultivo
Prioridad	Crítica
Funcionalidad	<ul style="list-style-type: none"> • Detectar fases: DESARROLLO, FLORACIÓN, PRODUCCIÓN Y DESCONOCIDO • Mostrar nivel de confianza de la detección • Calcular fecha estimada de cosecha automáticamente
Criterio de aceptación	Precisión mayor al 85% en clasificación de fases del cultivo
Historia de Usuario 4	
Número	4
Nombre Historia	Monitoreo IoT en Tiempo Real
Prioridad	Alta
Funcionalidad	<ul style="list-style-type: none"> • Actualización automática al detectar estado del cultivo • Indicadores de estado con colores • Alertas automáticas por condiciones fuera de rango
Criterio de aceptación	Precisión de $\pm 0.5^{\circ}\text{C}$ en temperatura y $\pm 2\%$ en humedad con recomendaciones específicas de manejo
Historia de Usuario 5	
Número	6
Nombre Historia	Asistente Virtual Especializado
Prioridad	Media
Funcionalidad	<ul style="list-style-type: none"> • Responder consultas por texto, y voz • Capacidad de guardar conversaciones importantes
Criterio de aceptación	Respuestas especializadas en gestión del agua en español.
Historia de Usuario 6	
Número	7
Nombre Historia	Gestión de Datos Históricos
Prioridad	Alta
Funcionalidad	<ul style="list-style-type: none"> • Almacenar datos de humedad, temperatura y tiempo de riego • Respaldos automáticos en el historial
Criterio de aceptación	Integridad de datos mayor al 99.9% con acceso en la interfaz
Historia de Usuario 7	

Número	8
Nombre Historia	Autenticación Segura
Prioridad	Media
Funcionalidad	<ul style="list-style-type: none"> • Loggin con Google, email/contraseña y huella digital • Persistencia de sesión entre usos • Sincronización de datos entre dispositivos
Criterio de aceptación	Cumplimiento de estándares de seguridad con perfiles personalizados automáticos
Historia de Usuario 8	
Número	9
Nombre Historia	Navegación Intuitiva y Manual
Prioridad	Baja
Funcionalidad	<ul style="list-style-type: none"> • Menú principal con iconos descriptivos • Manual de usuario interactivo • Adaptación a diferentes tamaños de pantalla
Criterio de aceptación	Transiciones fluidas entre pantallas con ayuda contextual en cada módulo
Historia de Usuario 9	
Número	10
Nombre Historia	Sistema Confiable 24/7
Prioridad	Alta
Funcionalidad	<ul style="list-style-type: none"> • Disponibilidad mayor al 99% en condiciones normales • Reconexión WiFi-automática • Funcionamiento offline parcial con cache
Criterio de aceptación	Recuperación automática en menos de 30 segundos con alerta de conectividad

Tabla 10. Requerimientos Funcionales

REQUERIMIENTOS FUNCIONALES - AGROPREDICT	
Id requerimiento:	RF_01_00
Nombre:	Módulo de Predicción de Fases de Cultivo con IA
Prioridad:	Alta
Descripción:	Sistema de visión artificial que detecta automáticamente la fase actual del cultivo (DESARROLLO, FLORACIÓN, PRODUCCIÓN, DESCONOCIDO), predice tiempos de cosecha específicos para el cultivo
Entrada:	Imágenes de cultivo capturadas desde la cámara del dispositivo Intel, tipo de cultivo seleccionado (tomate, pimiento, pepino, fresa, brócoli, frutas exóticas)
Procesos:	Clasificación automática de fases, cálculo de predicción temporal basado en datos históricos por cultivo, actualización en Firebase Realtime Database
Salida:	Fase actual detectada con barra de progreso visual, fecha estimada de cosecha calculada automáticamente, recomendaciones específicas por cultivo y fase, porcentaje de confianza de la detección
Id requerimiento:	RF_02_00
Nombre:	Sistema de Monitoreo IoT en Tiempo Real con ESP32
Prioridad:	Alta
Descripción:	Recolección y visualización en tiempo real de datos ambientales desde sensores ESP32 (DHT11, FC-28) con actualización automática y alertas inteligentes
Entrada:	Señales de sensores DHT11 (temperatura) y FC-28 (humedad del suelo) transmitidas vía WiFi desde ESP32

Procesos:	Lectura analógica/digital calibrada, transmisión WiFi a Firebase Realtime Database, procesamiento de datos con rangos óptimos, generación de alertas automáticas
Salida:	Valores de temperatura (°C) y humedad (%) con indicadores de estado (Óptimo/Alto/Bajo), alertas automáticas por condiciones fuera de rango, recomendaciones de manejo específicas
Id requerimiento:	RF_03_00
Nombre:	Control Automatizado de Riego con Temporización Inteligente
Prioridad:	Alta
Descripción:	Sistema de control automatizado del riego basado en datos reales de CSV con duraciones específicas por fase de cultivo y tipo de planta, integrado con ESP32 para control de válvulas
Entrada:	Fase actual del cultivo detectada por IA, tipo de cultivo seleccionado, configuración manual desde la app Android
Procesos:	Cálculo de duración basado en datos CSV reales (tomate: 3-8 min según fase), activación de válvulas ESP32, temporización precisa con cuenta regresiva, actualización de estado en Firebase
Salida:	Control automático de válvulas de riego, temporizador visual con tiempo restante, notificaciones de inicio/fin de ciclo, registro de actividad en base de datos
Id requerimiento:	RF_04_00
Nombre:	Riego Manual con Control Directo
Prioridad:	Media
Descripción:	Sistema de control manual del riego desde la aplicación Android que permite activación/desactivación inmediata sin temporización para situaciones de emergencia o control directo
Entrada:	Comandos manuales del usuario desde la interfaz Android, estado actual del sistema de riego
Procesos:	Validación de comandos, envío de instrucciones directas a Firebase, actualización inmediata del estado ESP32, registro de actividad manual
Salida:	Control inmediato del sistema de riego, indicador visual de estado manual, confirmación de cambio en la interfaz
Id requerimiento:	RF_05_00
Nombre:	Asistente Virtual Agronómico con Gemini AI
Prioridad:	Media
Descripción:	Chatbot conversacional multimodal especializado en gestión del agua y diagnóstico de cultivos, integrado con Google Gemini API, Speech-to-Text y Text-to-Speech
Entrada:	Consultas de texto del usuario, comandos de voz procesados por speech-to-text, configuración de modo silencio/sonido
Procesos:	Procesamiento de lenguaje natural con Gemini API, análisis de imágenes para diagnóstico de enfermedades, consulta especializada en gestión del agua, síntesis de voz para respuestas
Salida:	Respuestas especializadas en español con diagnósticos de plantas, recomendaciones de riego personalizadas, respuestas por texto y voz, capacidad de guardar conversaciones
Id requerimiento:	RF_06_00
Nombre:	Sistema de Notificaciones WhatsApp Automatizadas
Prioridad:	Alta
Descripción:	Envío automático de notificaciones vía WhatsApp Business API con mensajes detallados sobre cambios de estado del riego, alertas del sistema e información del invernadero
Entrada:	Eventos de cambio de estado del sistema de riego, inicio del sistema ESP32, condiciones ambientales críticas
Procesos:	Formateo de mensajes con información detallada (estado, hora, ubicación), integración con WhatsApp Business API, mecanismo de reintentos automáticos, escape de caracteres especiales
Salida:	Mensajes WhatsApp formateados con emojis, timestamp preciso, información del invernadero y supervisión por GreenAI, confirmación de entrega
Id requerimiento:	RF_07_00

Nombre:	Autenticación Segura Multiplataforma
Prioridad:	Media
Descripción:	Sistema de autenticación segura con múltiples métodos: Google Sign-In, autenticación biométrica (huella digital) y correo/contraseña, integrado con Firebase Auth
Entrada:	Credenciales de Google del usuario, autenticación biométrica, datos de registro con email y contraseña
Procesos:	Autenticación mediante Google Sign-In API, verificación biométrica con LocalAuth, validación de credenciales, creación y gestión de perfiles en Firebase Auth
Salida:	Acceso seguro a la aplicación con persistencia de sesión, perfil de usuario personalizado, sincronización automática de datos entre dispositivos
Id requerimiento:	RF_08_00
Nombre:	Sistema de Navegación y Manual de Usuario Interactivo
Prioridad:	Baja
Descripción:	Interfaz de navegación principal con acceso a todos los módulos y manual de usuario interactivo con guías paso a paso para cada funcionalidad del sistema
Entrada:	Selección de módulos por el usuario, navegación entre pantallas, consultas de ayuda
Procesos:	Gestión de rutas de navegación, presentación de contenido de ayuda contextual, animaciones de transición, organización de contenido por categorías
Salida:	Navegación fluida entre módulos, manual interactivo con instrucciones detalladas, acceso rápido a funcionalidades principales
Id requerimiento:	RF_09_00
Nombre:	Sincronización de Tiempo NTP para ESP32
Prioridad:	Media
Descripción:	Sincronización automática con servidores NTP (pool.ntp.org, time.google.com) para mantener timestamps precisos en todas las operaciones del sistema ESP32
Entrada:	Servidores NTP primarios y backup, configuración de zona horaria GMT-5 (Ecuador/Colombia)
Procesos:	Configuración automática de tiempo con servidores primarios y backup, manejo de errores de sincronización, reintentos automáticos en caso de falla
Salida:	Timestamps precisos en formato ISO 8601 para notificaciones WhatsApp, hora local correcta para logs del sistema, sincronización temporal confiable

Tabla 11. Requerimientos no Funcionales

REQUERIMIENTOS NO FUNCIONALES - AGROPREDICT	
Id requerimiento:	RNF_01_00
Nombre:	Rendimiento Optimizado para Android
Prioridad:	Alta
Descripción:	El sistema debe mantener tiempos de respuesta óptimos y uso eficiente de recursos en dispositivos Android con especificaciones mínimas, optimizado para resoluciones estándar
Entrada:	Interacciones de usuario en la app Android, consultas a APIs externas, operaciones de base de datos
Procesos:	Optimización de memoria y CPU, gestión eficiente de recursos con animaciones controladas, cache inteligente para datos frecuentes, compresión de imágenes
Salida:	Salida: Tiempo de respuesta UI < 300ms, consultas Gemini API < 3s, sincronización Firebase < 2s, consumo de RAM < 100MB en dispositivos promedio
Id requerimiento:	RNF_02_00
Nombre:	Disponibilidad y Confiabilidad del Sistema IoT
Prioridad:	Alta
Descripción:	Garantizar operación continua del sistema ESP32 y app Android con recuperación automática ante fallos de conectividad WiFi y mecanismos de reconexión
Entrada:	Estados de conexión WiFi del ESP32, disponibilidad de Firebase, conectividad de APIs externas

Procesos:	Monitoreo continuo de conectividad con timeouts configurables, mecanismos de reconexión automática con reintentos programados, manejo inteligente de estados offline
Salida:	Disponibilidad del sistema > 99% en condiciones normales, recuperación automática en < 30 segundos, funcionamiento offline temporal con cache local
Id requerimiento:	RNF_03_00
Nombre:	Precisión de Integración con APIs de IA
Prioridad:	Alta
Descripción:	Las integraciones con Google Gemini API deben mantener alta precisión en detección de fases de cultivo y respuestas del asistente virtual con manejo robusto de errores
Entrada:	Consultas de texto del usuario, parámetros de configuración de modelo
Procesos:	Integración optimizada con Gemini 1.5 Pro, manejo de respuestas y errores de API, validación de resultados, configuración de temperatura y parámetros
Salida:	Precisión de detección de fases > 85%, respuestas del asistente contextualmente relevantes, tiempo de respuesta API < 3 segundos, manejo elegante de errores
Id requerimiento:	RNF_04_00
Nombre:	Compatibilidad Android Amplia
Prioridad:	Alta
Descripción:	La aplicación debe funcionar correctamente en dispositivos Android desde la versión 7.0+ con soporte completo de funcionalidades y adaptación de UI
Entrada:	Diferentes versiones de Android (7.0+), variedad de hardware, resoluciones de pantalla diversas, capacidades de sensores
Procesos:	Pruebas de compatibilidad extensivas, adaptación de interfaz responsive, optimización para diferentes especificaciones, manejo de permisos por versión
Salida:	Funcionamiento completo en Android 7.0+, interfaz adaptativa para múltiples resoluciones, soporte para diferentes tipos de sensores biométricos
Id requerimiento:	RNF_05_00
Nombre:	Seguridad de Datos y Comunicaciones
Prioridad:	Alta
Descripción:	Protección integral de datos del usuario, comunicaciones cifradas con Firebase y APIs externas, manejo seguro de credenciales y tokens
Entrada:	Datos personales del usuario, credenciales de autenticación, tokens de API, información de sensores IoT
Procesos:	Cifrado de comunicaciones HTTPS, autenticación segura con Google y Firebase, almacenamiento seguro de tokens, validación de entrada de datos
Salida:	Comunicaciones seguras con cifrado end-to-end, autenticación biométrica funcional, protección de tokens de API, cumplimiento de privacidad de datos
Id requerimiento:	RNF_06_00
Nombre:	Usabilidad y Experiencia de Usuario
Prioridad:	Media
Descripción:	Interfaz intuitiva y fácil de usar para estudiantes de bachillerato técnico agrícola con navegación simplificada y retroalimentación visual clara
Entrada:	Interacciones del usuario, patrones de uso, feedback de usabilidad, métricas de navegación
Procesos:	Diseño centrado en el usuario con animaciones fluidas, navegación intuitiva con breadcrumbs, feedback visual inmediato, ayuda contextual integrada
Salida:	Interfaz fácil de aprender con curva de aprendizaje < 30 minutos
Id requerimiento:	RNF_07_00
Nombre:	Escalabilidad del Sistema
Prioridad:	Media
Descripción:	Capacidad del sistema para manejar múltiples usuarios simultáneos y crecimiento en la cantidad de datos de sensores sin degradación significativa del rendimiento
Entrada:	Múltiples sesiones de usuario simultáneas, crecimiento exponencial de datos de sensores, incremento en consultas de API
Procesos:	Arquitectura escalable con Firebase, optimización de consultas de base de datos, gestión eficiente de recursos compartidos, cache distribuido

Salida:	Soporte para usuarios simultáneos, manejo de registros de sensores, tiempo de respuesta consistente bajo carga incrementada
Id requerimiento:	RNF_08_00
Nombre:	Mantenibilidad y Documentación
Prioridad:	Baja
Descripción:	Código bien estructurado y documentado para facilitar el mantenimiento, actualizaciones y extensiones futuras del sistema
Entrada:	Código fuente de múltiples módulos, documentación técnica, especificaciones de API
Procesos:	Arquitectura modular bien definida, comentarios de código comprensivos, estructura de archivos organizada
Salida:	Código mantenible con documentación completa, facilidad para agregar nuevos módulos, tiempo de desarrollo de nuevas funciones optimizado

4.2.3. Fase de diseño del prototipo

Arquitectura del sistema: El sistema integra un microcontrolador ESP32 que recopila datos del entorno del invernadero mediante sensores y controla los actuadores de riego.

El Agente IA de Cultivo analiza la información recibida, toma decisiones autónomas y coordina la acción de los demás agentes: el Agente Actuador de Riego, encargado de ejecutar las órdenes; el Agente de Notificación, que comunica alertas y reportes al agricultor; y el Agente de Monitoreo, que actualiza la aplicación AgroPredict con datos, predicciones y visualizaciones. El agrónomo aporta conocimiento y validación al sistema, generando un ciclo continuo de aprendizaje y mejora. La arquitectura se apoya en herramientas como TensorFlow, Python, Arduino y bases de datos, que permiten la comunicación, el análisis inteligente y la supervisión integral del cultivo.

4.2.4. Red Neuronal - Sistema de Riego

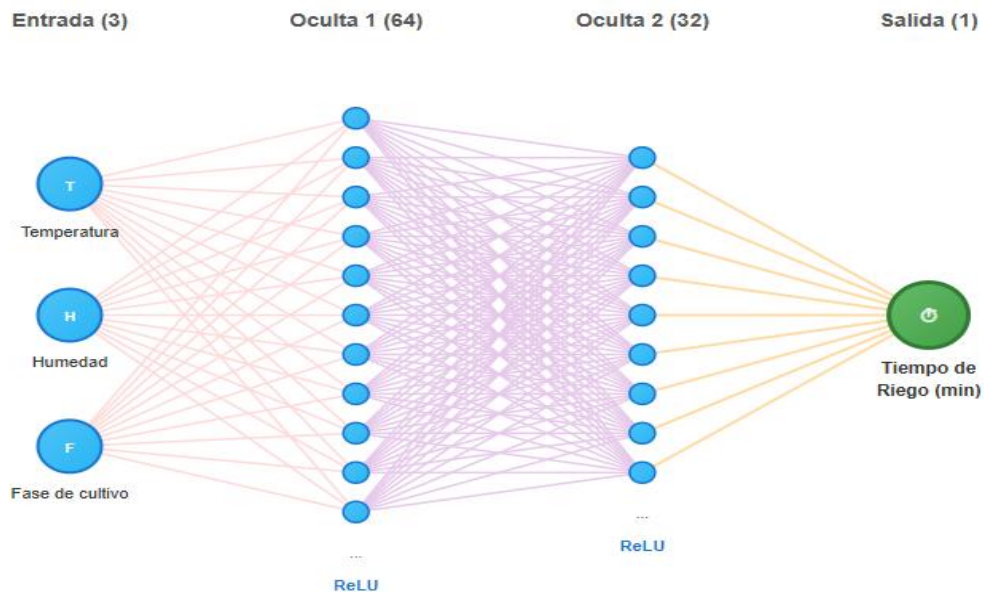


Figura 3. Red Neuronal- Sistema de Riego (Agente IA de cultivo)

Esta red neuronal implementada en el archivo `model.h5` de Keras predice automáticamente cuántos minutos debe durar el riego basándose en las condiciones actuales del cultivo. La información fluye de izquierda a derecha a través de cuatro capas conectadas, donde las líneas rosadas y naranjas representan las conexiones entre neuronas con diferentes pesos que determinan la importancia de cada relación.

- Capa de Entrada (3 neuronas)

La red recibe tres datos fundamentales: Temperatura (T) medida por el sensor DHT11 en grados Celsius, Humedad del suelo (H) medida por el sensor FC-28 en porcentaje de 0 a 100, y Fase de cultivo (F) detectada por la cámara y codificada numéricamente como 0 para DESCONOCIDO, 1 para DESARROLLO, 2 para FLORACIÓN y 3 para PRODUCCIÓN. Estos tres valores se convierten en un tensor de TensorFlow mediante `tf.convert_to_tensor([[temp, hum, phase_code]], dtype=tf.float32)` antes de ingresar a la red.

- Capa Oculta 1 (64 neuronas)

Contiene 64 neuronas donde cada una recibe las tres entradas y aplica una transformación matemática $z = w_1 * T + w_2 * H + w_3 * F + b$ seguida de la función de activación ReLU que convierte valores negativos en cero $ReLU(z) = \max(0, z)$,

permitiendo que la red detecte patrones básicos como "temperatura alta con humedad baja" o "fase de floración requiere más agua". Esta capa identifica 64 características diferentes combinando los datos de entrada de formas distintas.

- Capa Oculta 2 (32 neuronas)

Reduce la información a 32 neuronas que reciben las 64 salidas anteriores, aplicando nuevamente transformación lineal y activación ReLU para capturar patrones más complejos como "en floración con 30°C y 40% humedad se necesita riego prolongado" o "en desarrollo con humedad moderada el riego debe ser corto". Esta capa aprende reglas específicas de riego según combinaciones complejas de temperatura, humedad y fase de crecimiento.

- Capa de Salida (1 neurona)

Produce un único número que representa los minutos de riego recomendados, recibiendo las 32 salidas de la capa anterior y aplicando una transformación lineal final sin activación adicional porque necesitamos un valor continuo en minutos. La predicción se obtiene con `prediction = self.irrigation_model.predict(input_data, verbose=0)` y se extrae como `predicted_time = float(prediction[0][0])`, luego el sistema descarta predicciones menores a 0.3 minutos como innecesarias y limita el máximo a 30 minutos para evitar sobrieriego.

4.2.5. Diagrama de flujo tflite



Figura 4. Diagrama de flujo model_unquant.tflite

Este diagrama muestra los 4 pasos para crear el modelo model_unquant.tflite que detecta automáticamente las fases del cultivo usando visión por computadora.

- Inicio - Captura de Imágenes

Se toman fotografías del cultivo en diferentes etapas usando una cámara: plantas en desarrollo (crecimiento de hojas), plantas en floración (con flores visibles), plantas en producción (con frutos formados) y situaciones desconocidas (imágenes poco claras). Las fotos deben capturarse desde diferentes ángulos y con distinta iluminación para que el modelo aprenda bien.

- Organización de Imágenes en Carpetas

Las fotografías se organizan en cuatro carpetas según su categoría: carpeta "DESARROLLO" con fotos de crecimiento, carpeta "FLORACIÓN" con imágenes de flores, carpeta "PRODUCCIÓN" con fotos de frutos, y carpeta "DESCONOCIDO" o "?" con imágenes ambiguas. El nombre de cada carpeta se convierte automáticamente en la etiqueta que el modelo aprenderá a reconocer.

- Entrenamiento con Teachable Machine de Google

Se usa la plataforma gratuita teachablemachine.withgoogle.com que permite entrenar modelos sin programar, se selecciona "Image Project" y se cargan las cuatro carpetas organizadas donde la plataforma reconoce automáticamente cada clase. Al hacer clic en "Train Model" comienza el entrenamiento usando una red neuronal convolucional que aprende a diferenciar las características visuales de cada fase analizando colores, formas y patrones en las imágenes, el proceso toma varios minutos y muestra la precisión alcanzada en tiempo real hasta completarse exitosamente.

- Exportación del Modelo - TensorFlow Lite

Al finalizar el entrenamiento se selecciona "Export Model" y se elige formato "TensorFlow Lite" optimizado para dispositivos con recursos limitados. Se descargan dos archivos: `model_unquant.tflite` que contiene toda la red neuronal entrenada con pesos y configuración en formato comprimido para ejecutarse rápidamente, y `labels.txt` que lista las cuatro clases en orden (DESARROLLO, FLORACIÓN, PRODUCCIÓN, DESCONOCIDO) para interpretar los resultados numéricos del modelo. Estos archivos se colocan junto al script Python donde son cargados mediante `Interpreter(model_path='model_unquant.tflite')` permitiendo que el sistema procese cada imagen de la cámara en tiempo real, la redimensione, ejecute la inferencia en

milisegundos, y retorne qué fase detectó mostrándola en pantalla y guardándola en Firebase para control automático del riego.

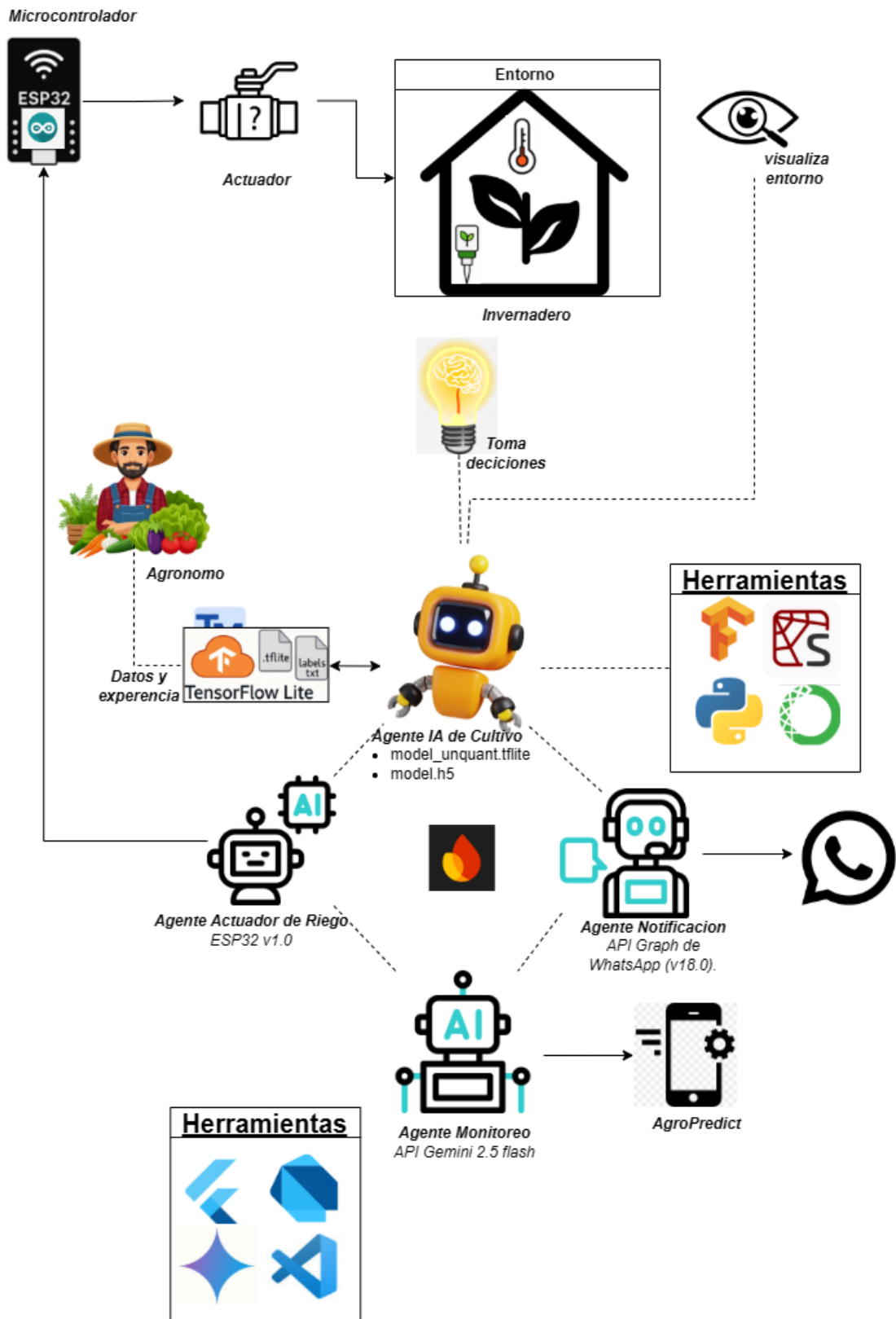


Figura 5. General del ecosistema multi-agente

4.2.5.1. Modelo base de Datos RTDB



Figura 6. Base de datos, RTDB

4.2.6. Modelo del circuito

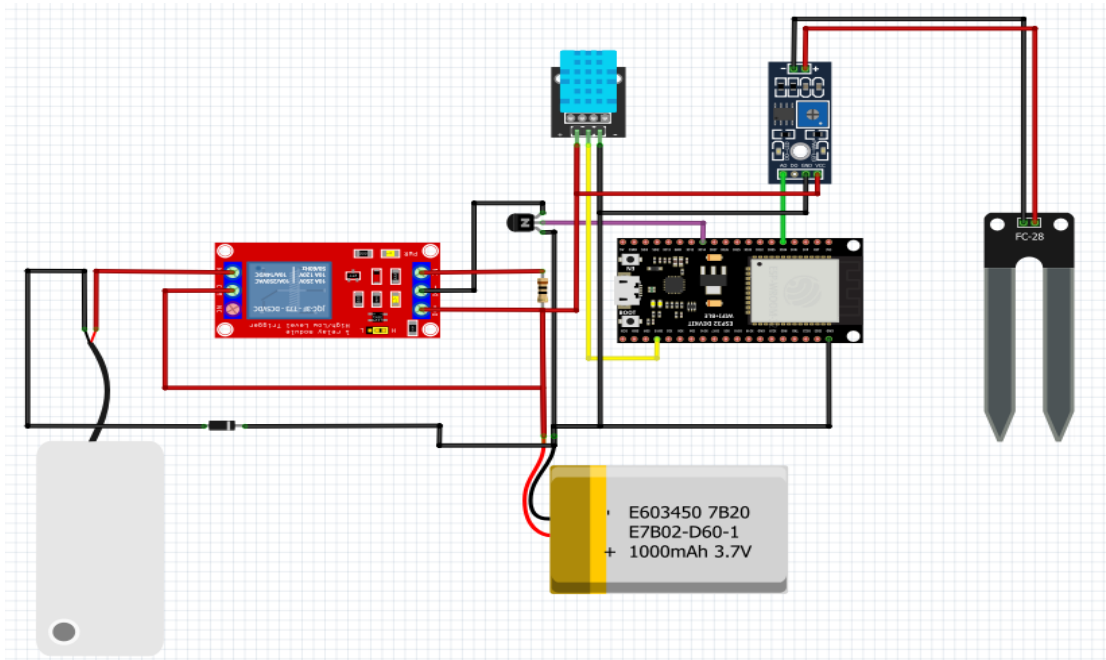


Figura 7. Diagrama de Conexiones del Sistema IoT de Riego Inteligente

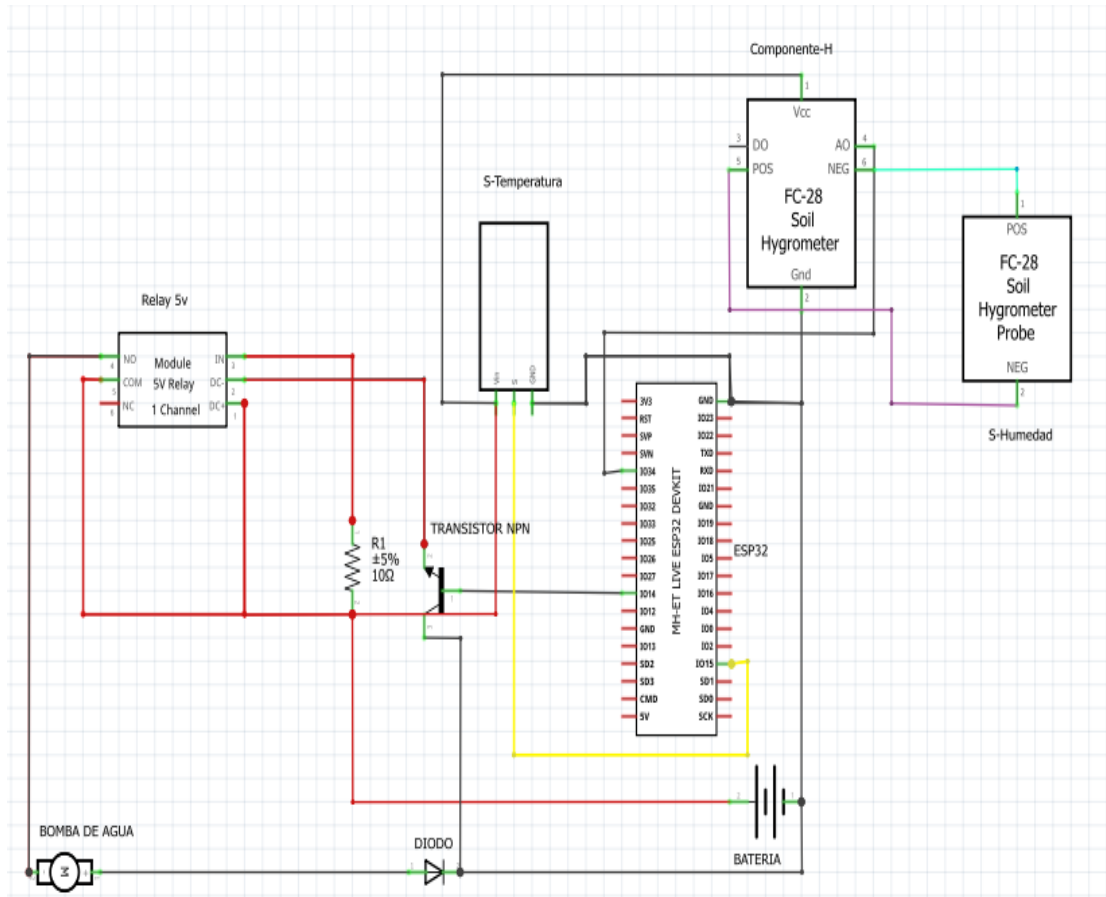


Figura 8. Diagrama Esquemático del Circuito de Control de Riego

El Agente ESP32 está configurado con múltiples conexiones de pines donde el GPIO 15 se conecta al sensor DHT11 para lectura de temperatura y humedad ambiental mediante protocolo digital propietario con alimentación de 3.3V, el GPIO 34 funciona como entrada analógica conectada al sensor FC-28 de humedad de suelo con resolución de 12 bits leyendo valores entre 2480 (mojado) y 4095 (seco) que son calibrados a porcentaje de humedad mediante promedio móvil de 5 lecturas para estabilidad, el GPIO 13 está configurado como entrada digital con pull-down interno conectado a un botón físico que funciona mediante interrupción por flanco de subida permitiendo control manual del sistema alternando el estado de riego, el GPIO 2 controla un LED indicador que se enciende cuando el riego está activo y se apaga cuando está inactivo funcionando como realimentación visual del sistema, el GPIO 14 controla el motor o bomba de agua mediante módulo relay recomendado para aislar y manejar voltajes mayores activándose sincronizadamente con el LED cuando el riego está en curso, mientras que el WiFi integrado del ESP32 se conecta a la red "respaldopro" estableciendo comunicación con Firebase Realtime Database para sincronizar el estado de riego verificado cada segundo, almacenar lecturas de sensores cada 5 segundos, obtener sincronización de tiempo mediante protocolo NTP y enviar notificaciones a WhatsApp Business API v18.0 cuando cambia el estado del sistema, creando así un flujo donde los sensores leen datos ambientales que se almacenan en Firebase, el estado de riego se puede modificar desde el botón físico o remotamente actualizando Firebase, este estado controla las salidas físicas del LED y motor, y cualquier cambio genera notificaciones automáticas vía WhatsApp informando hora, estado y ubicación del sistema supervisado por GreenAI.

4.2.6.1. Diagramas de caso de uso

Permiten observar de manera clara y concisa cómo se realizan las interacciones dentro del sistema. Con estos diagramas, podemos identificar y comprender cada caso que se presenta en la comunicación entre el usuario y el sistema, acatan funcionalidades que pueden ejecutarse. Esta representación gráfica facilita el entendimiento de la lógica del sistema, permitiendo visualizar quiénes son los actores involucrados, qué operaciones pueden realizar y cómo se relacionan entre sí. A continuación, se muestran los diagramas de casos de uso del sistema.

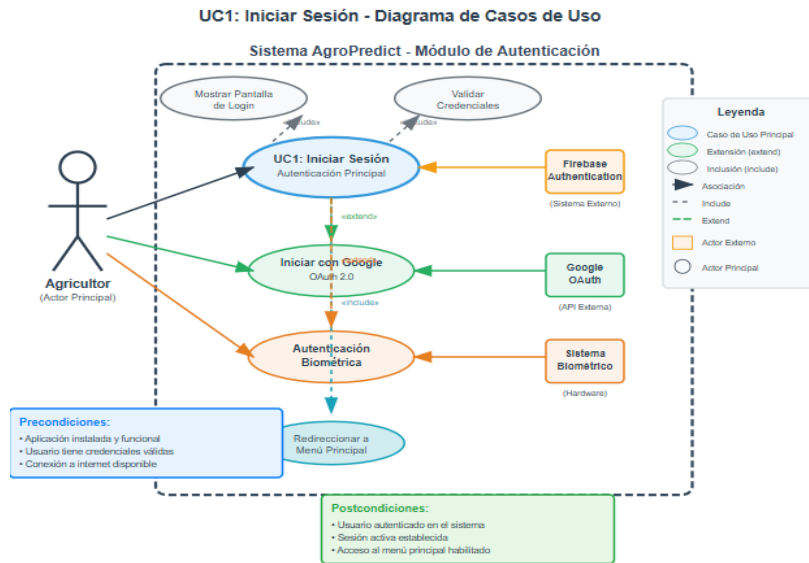


Figura 9. Diagrama Iniciar sesión

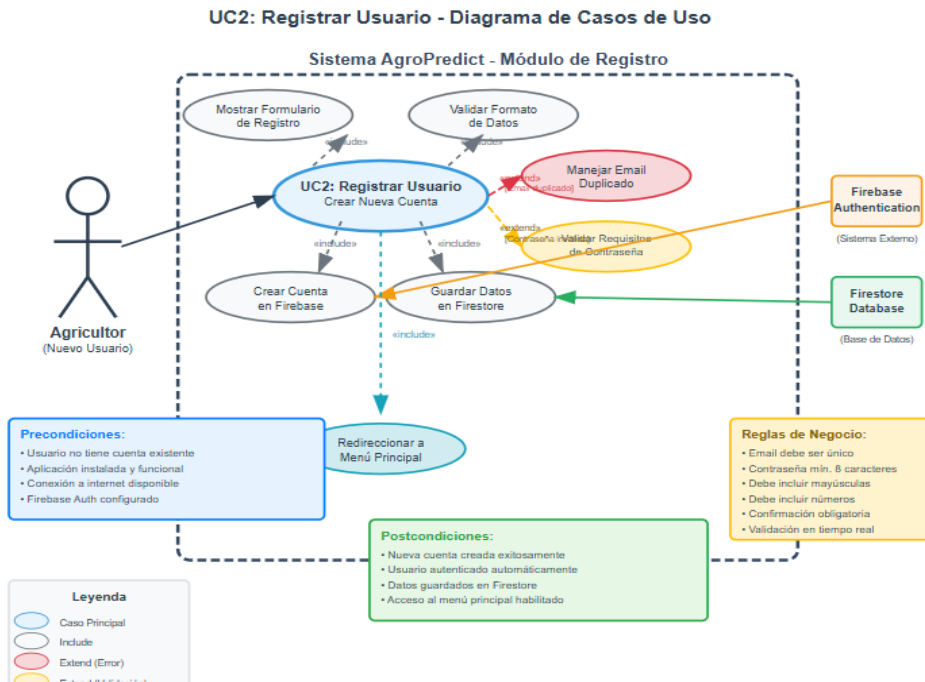


Figura 10. Registrar usuario

UC5: Visualizar Datos en Tiempo Real - Diagrama de Casos de Uso

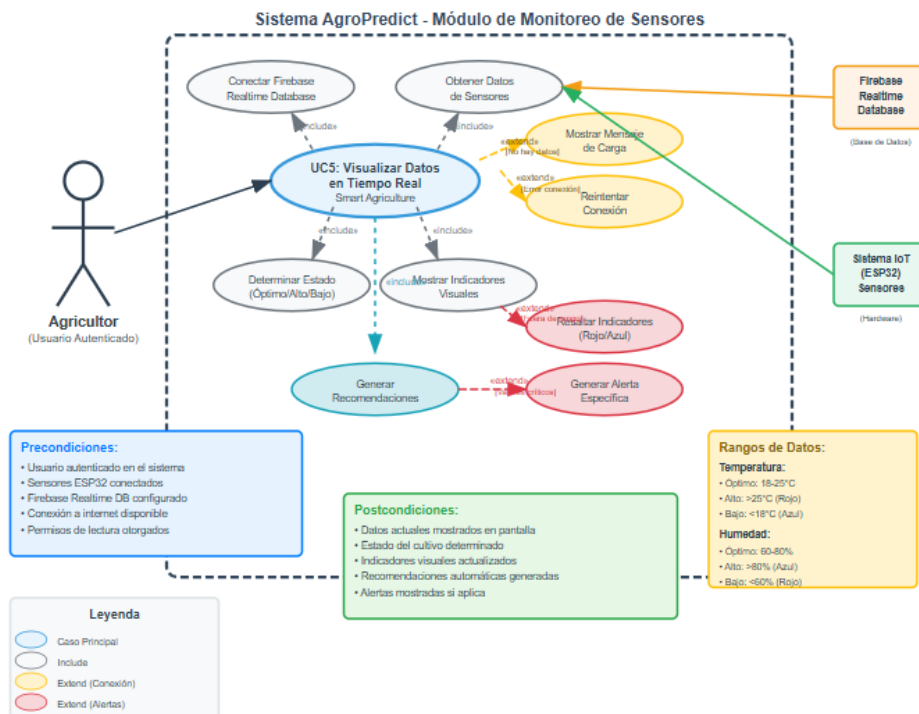


Figura 11. Visualización de datos

UC6: Actualizar Lecturas de Sensores - Diagrama de Casos de Uso

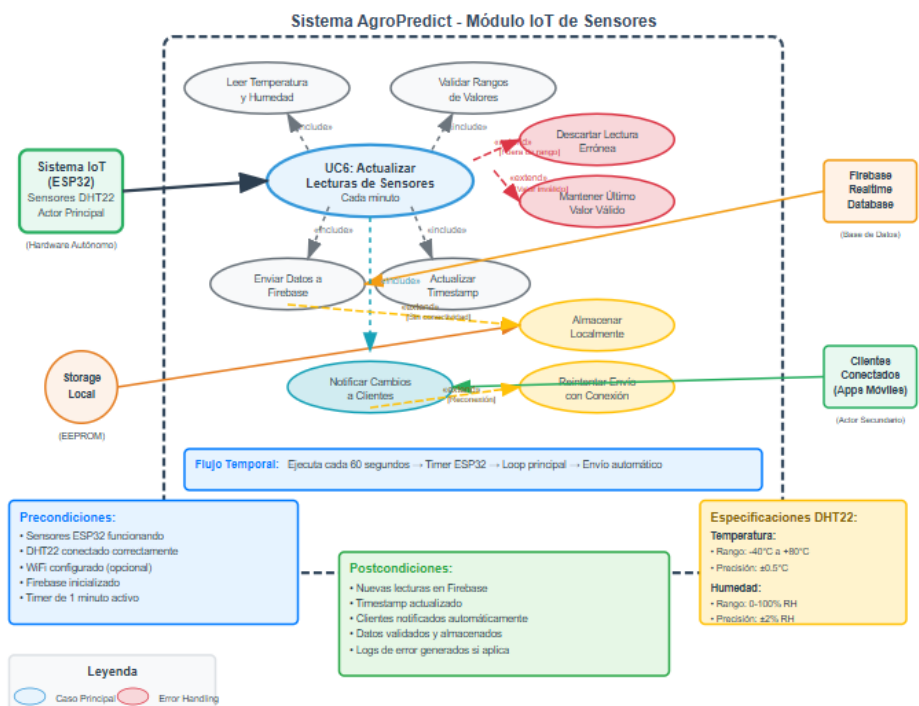


Figura 12. Lectura de sensores

UC9: Analizar Fase de Cultivo - Diagrama de Casos de Uso

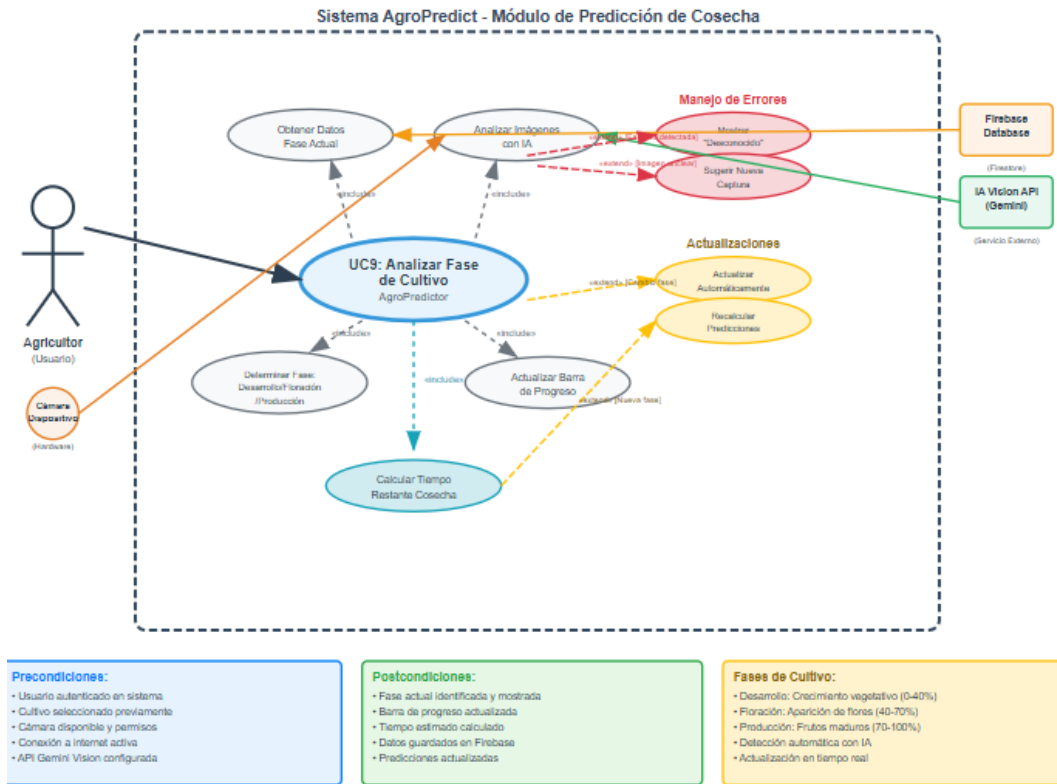


Figura 13. Fase de cultivo

UC10: Predecir Tiempo de Cosecha - Diagrama de Casos de Uso

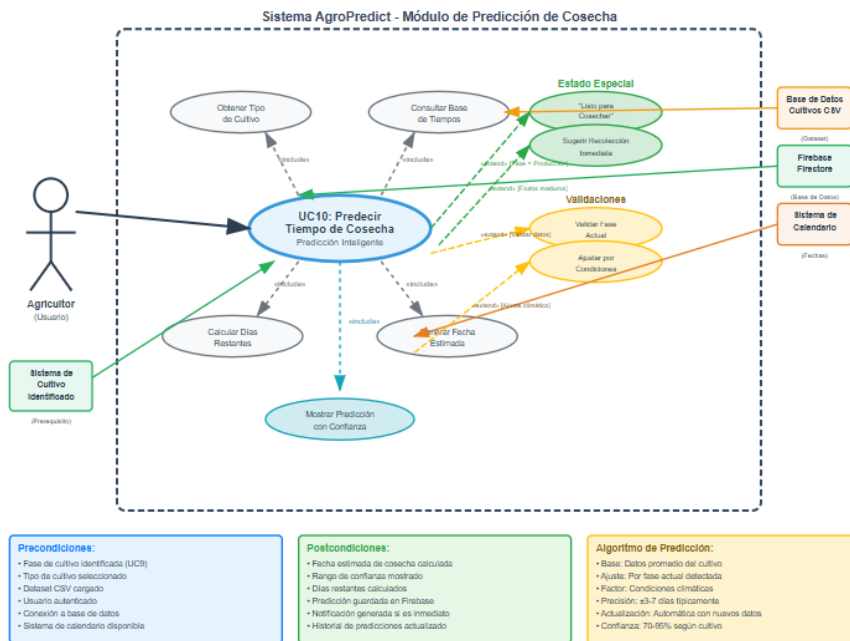


Figura 14. Tiempo de cosecha

4.2.6.2. Diseño de Interfaces



Figura 15. Diseño de pantalla para login



Figura 16. Diseño de pantalla para registro de usuario



Figura 17. Diseño de pantalla para menú

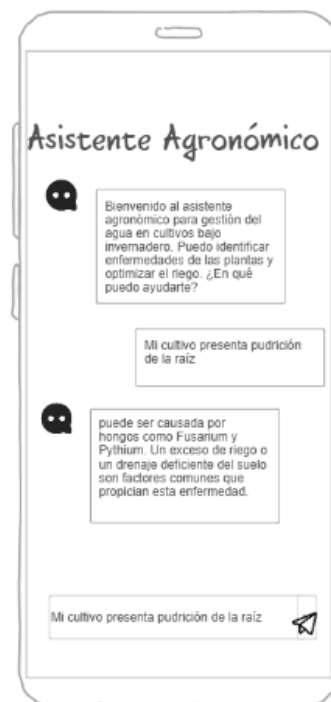


Figura 18. Diseño de pantalla para asistente agronómico



Figura 19. Diseño de pantalla para control de riego



Figura 20. Diseño de pantalla para monitoreo invernadero



Figura 21. Diseño de pantalla para manual de usuario



Figura 22. Diseño de pantalla para créditos

4.2.6.3. Diagrama de Procesos

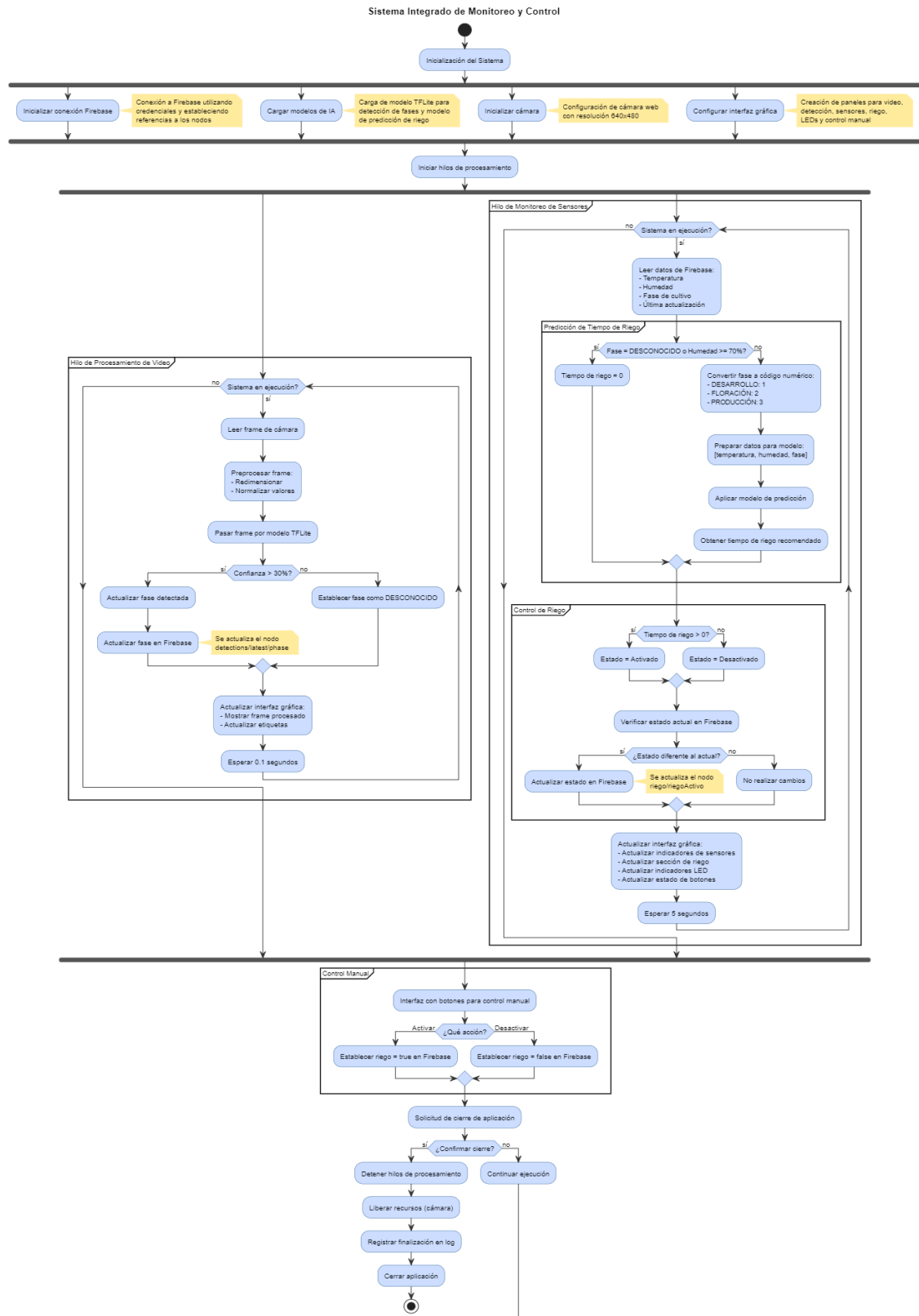


Figura 23. Diagrama de procesos del sistema

4.2.7. Fase de Desarrollo

La instalación del entorno de trabajo Visual Studio Code se realizó para comenzar a desarrollar la aplicación móvil, lo cual posibilitó más adelante la configuración del framework Flutter. Antes de eso, se configuró el SDK de Android para asegurar que fuera reconocido en el entorno de desarrollo.

Flutter es un marco de trabajo que fue desarrollado por Google para la creación de aplicaciones móviles con interfaces intuitivas y contemporáneas, haciendo uso del lenguaje de programación Dart. Su estructura se fundamenta en la utilización de widgets, que posibilitan el desarrollo de la interfaz de usuario (UI/UX) con flexibilidad y modularidad. Además, Flutter cuenta con un extenso repertorio de librerías que se pueden encontrar en Pub.dev y que facilitan la incorporación de varias funcionalidades.

Al instalar Flutter, el sistema crea automáticamente un repositorio que está organizado en carpetas; esto contribuye a la conservación de una estructura ordenada del proyecto. Además, fue preciso conectar la aplicación con Firebase; para eso se registró el proyecto en la plataforma. Esta conexión posibilita el uso de las herramientas proporcionadas por Firebase, entre ellas la autenticación, la base de datos en tiempo real y otras funciones fundamentales para que la aplicación desarrollada en Flutter funcione correctamente.

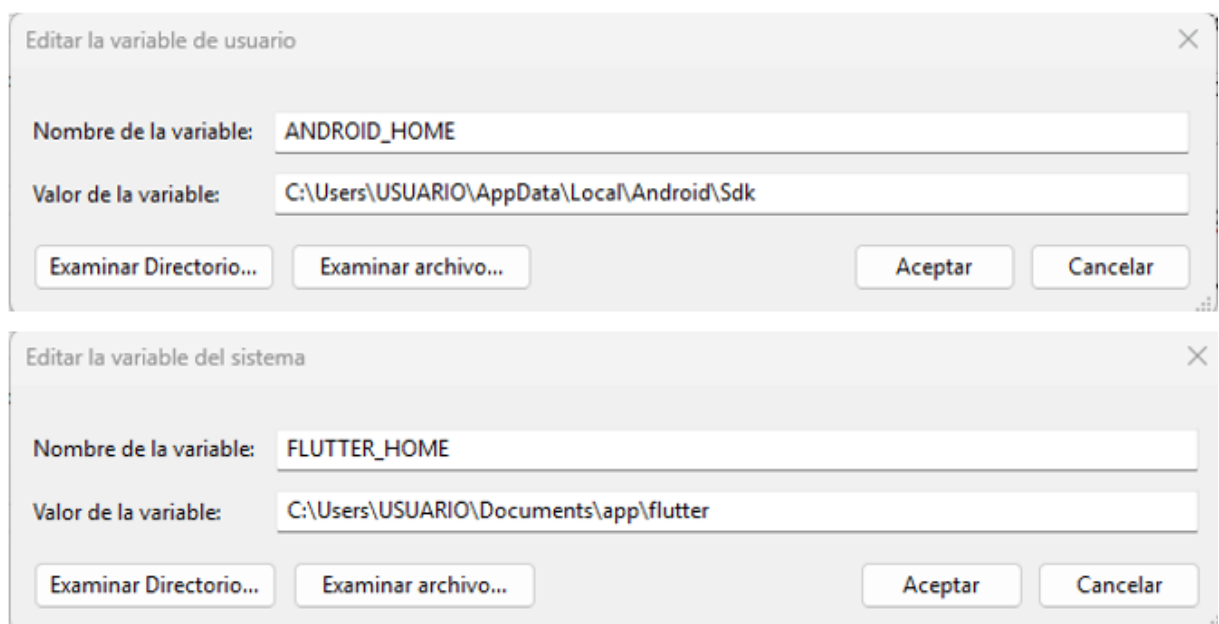


Figura 24. Variables entorno del sistema

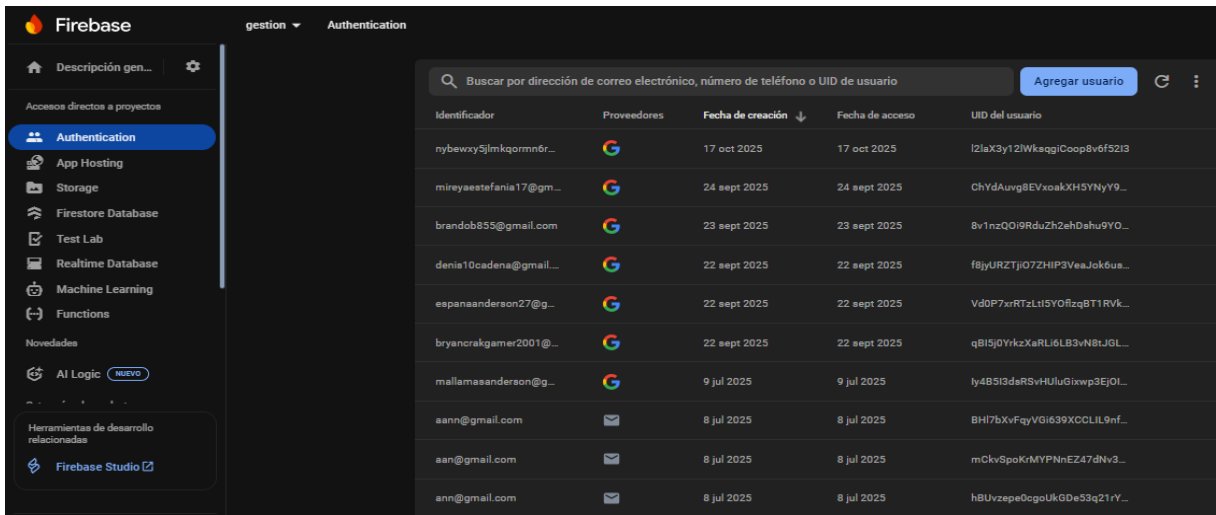


Figura 25. Autenticación de usuarios

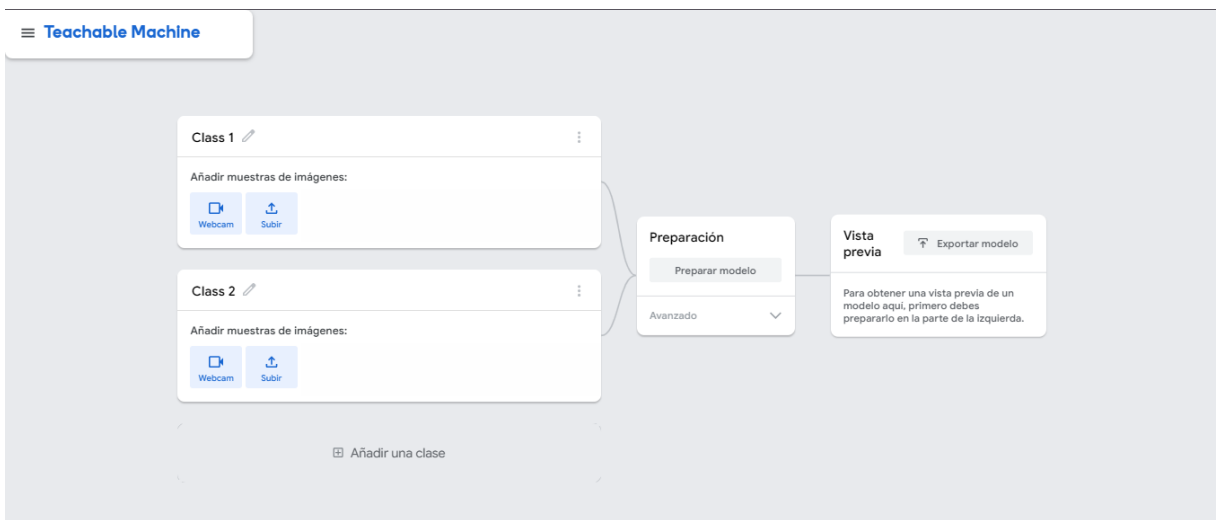


Figura 26. Teachable machine modelo h5 y label

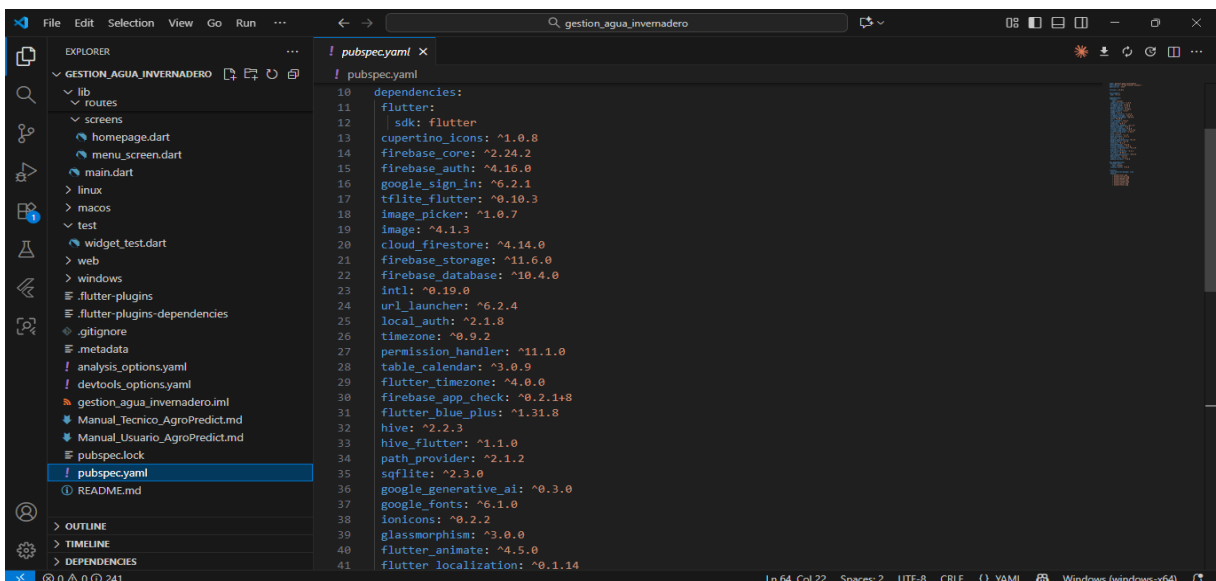


Figura 27. Librerías pub dev

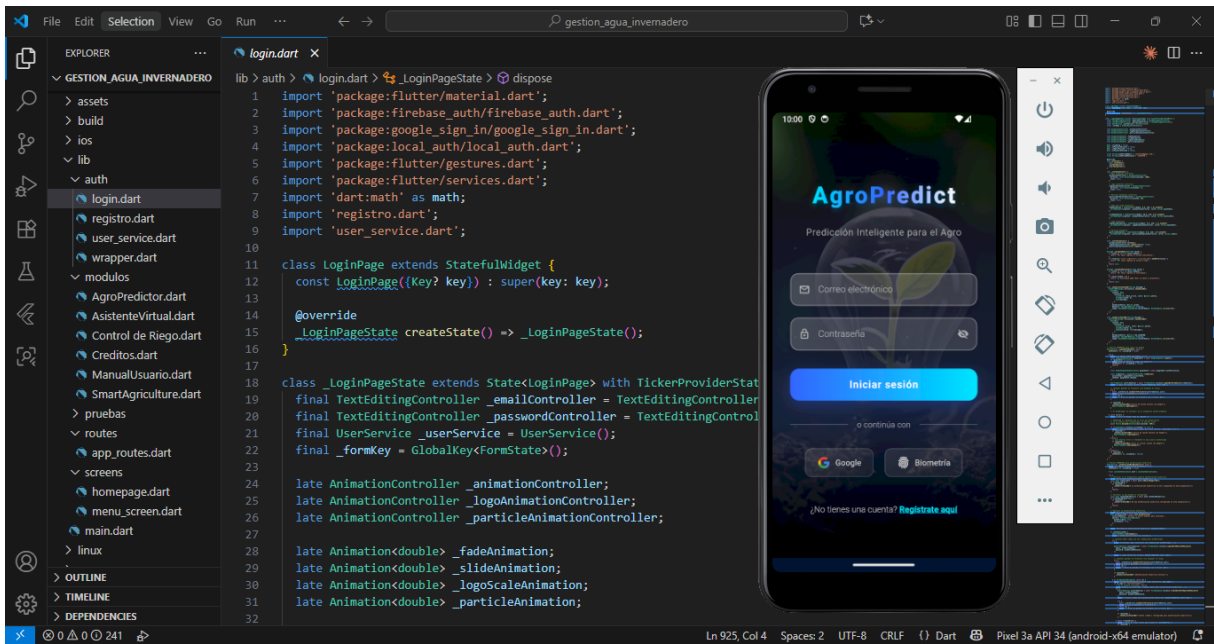


Figura 28. Pantalla de Login

La pantalla inicial en la aplicación permite a diferentes usuarios poder acceder mediante:

- Correo y contraseña: Autenticación tradicional
- Cuenta Google: Inicia sesión rápido mediante integración con Google Auth
- Huella digital: Autenticación biométrica para mayor seguridad
- Opción de registro: Para usuarios nuevos que no tienen cuenta

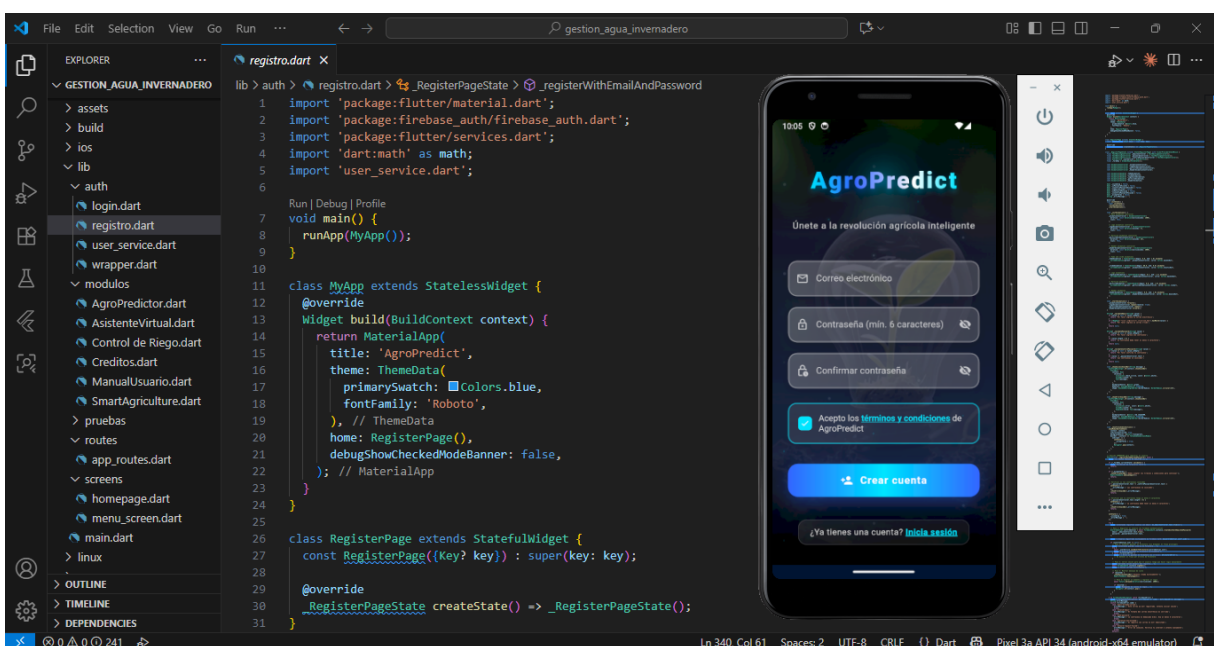


Figura 29. Pantalla de Registro

Permite crear nuevas cuentas de usuario con:

- Validación de correo electrónico
- Confirmación de contraseña
- Integración con Firebase Authentication

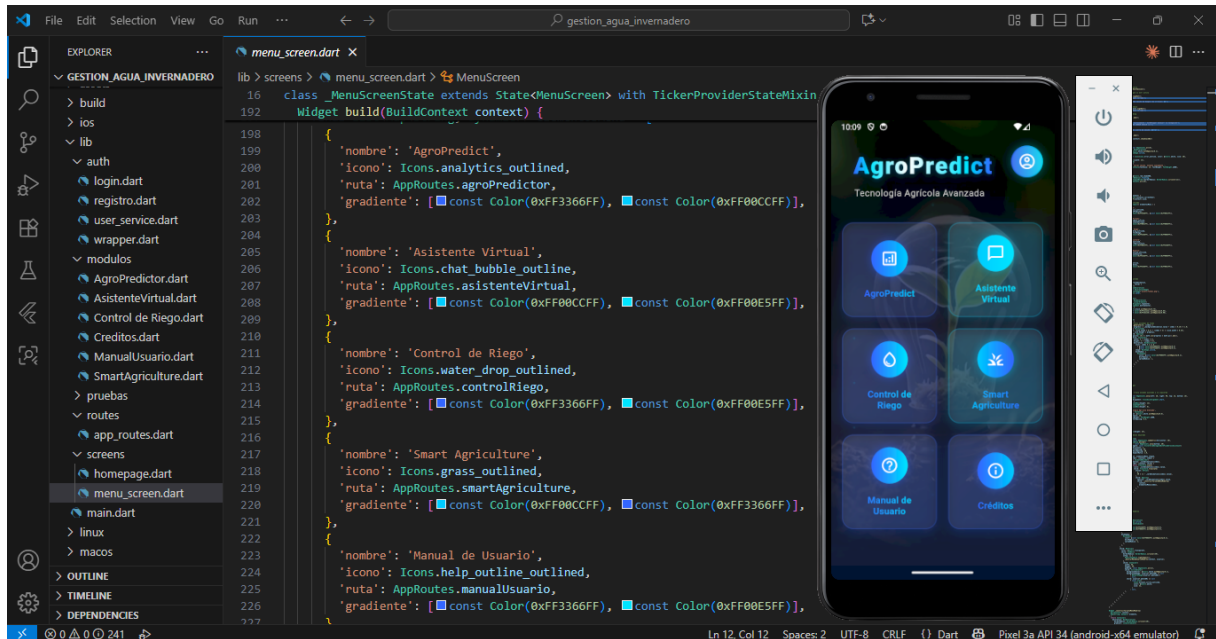


Figura 30. Menú Principal

Menú Principal

El menú principal presenta seis módulos principales organizados en una interfaz intuitiva:

- AgroPredictor: Predicción de cosecha basada en IA
- Asistente Virtual: Consultas y diagnósticos con inteligencia artificial
- Control de Riego: Gestión manual del sistema de irrigación
- Smart Agriculture: Monitoreo de sensores en tiempo real
- Manual de Usuario: Guía completa de uso
- Créditos: Información del desarrollador y proyecto

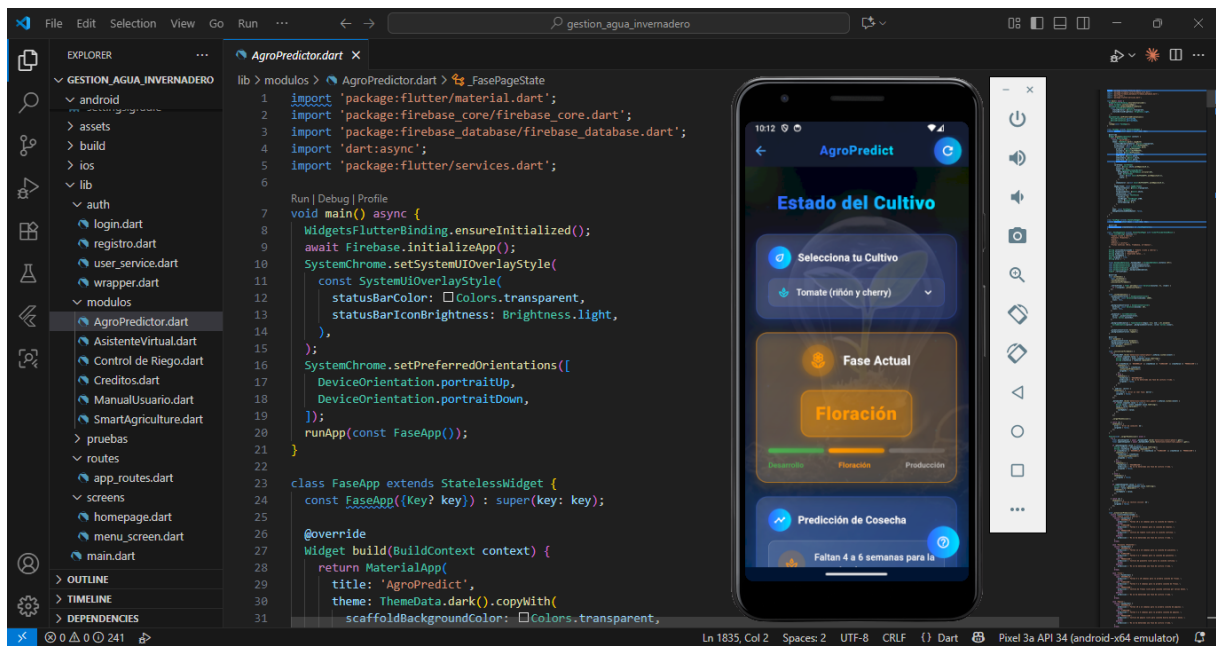


Figura 31. AgroPredictor - Predicción de Cosecha

Funcionalidad Principal: Este módulo utiliza visión artificial para determinar automáticamente la fase actual del cultivo y predecir el tiempo restante hasta la cosecha.

- Características:

Conexión Firebase Realtime: Recibe datos en tiempo real sobre el estado del cultivo

Detección de Fases: Identifica automáticamente las fases: Desarrollo, Floración y Producción

- Predicciones Inteligentes:

Fase Desarrollo: Muestra tiempo estimado hasta cosecha (10-12 semanas)

Fase Floración: Indica tiempo reducido (4-6 semanas)

Fase Producción: Notifica que el cultivo está listo para cosechar

Cultivos Soportados: Tomate, pimiento, fresa, pepino, brócoli, frutas exóticas

Recomendaciones Específicas: Consejos de manejo según la fase actual

Fecha Estimada: Cálculo automático de fecha probable de cosecha

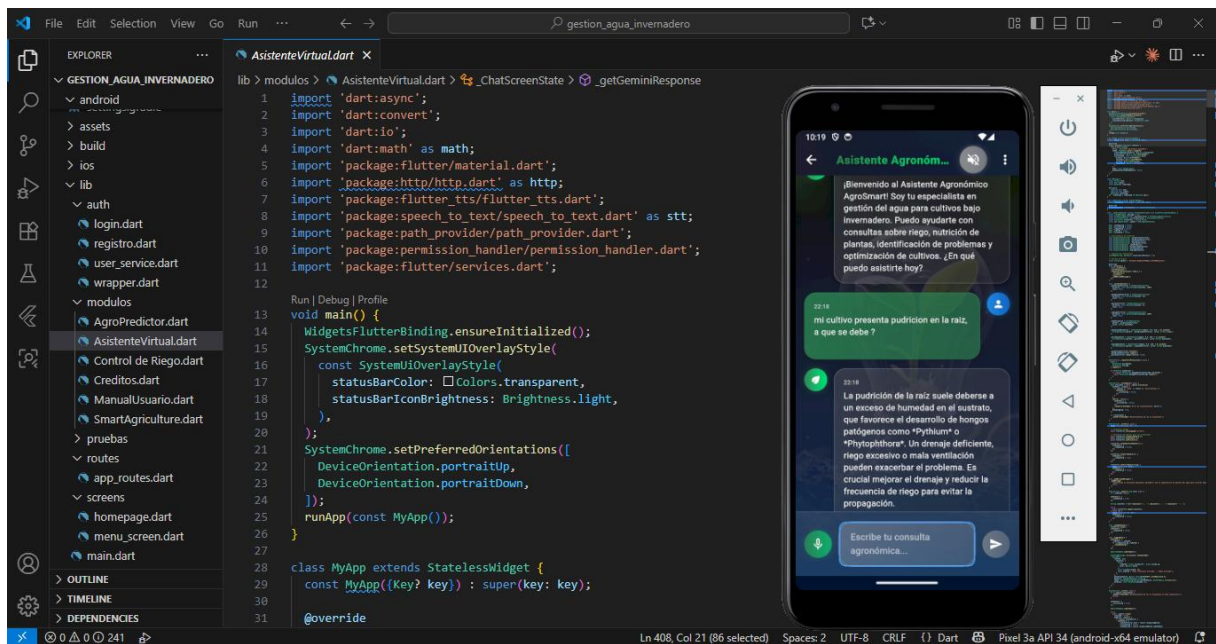


Figura 32. Asistente Virtual Agronómico

Funcionalidad Principal: Asistente inteligente basado en IA Gemini que brinda soporte técnico especializado para problemas agronómicos.

Características:

- Consultas por Texto: Respuestas inmediatas a preguntas técnicas
- Reconocimiento de Voz: Consultas manos libres
- Análisis de Imágenes: Diagnóstico automático de enfermedades mediante fotografías
- Respuestas en Español: Optimizado para consultas en español de España
- Diagnóstico de Problemas: Identificación de plagas, enfermedades y deficiencias
- Soluciones Personalizadas: Recomendaciones específicas según el problema detectado
- Historial de Conversaciones: Capacidad de guardar y cargar conversaciones anteriores

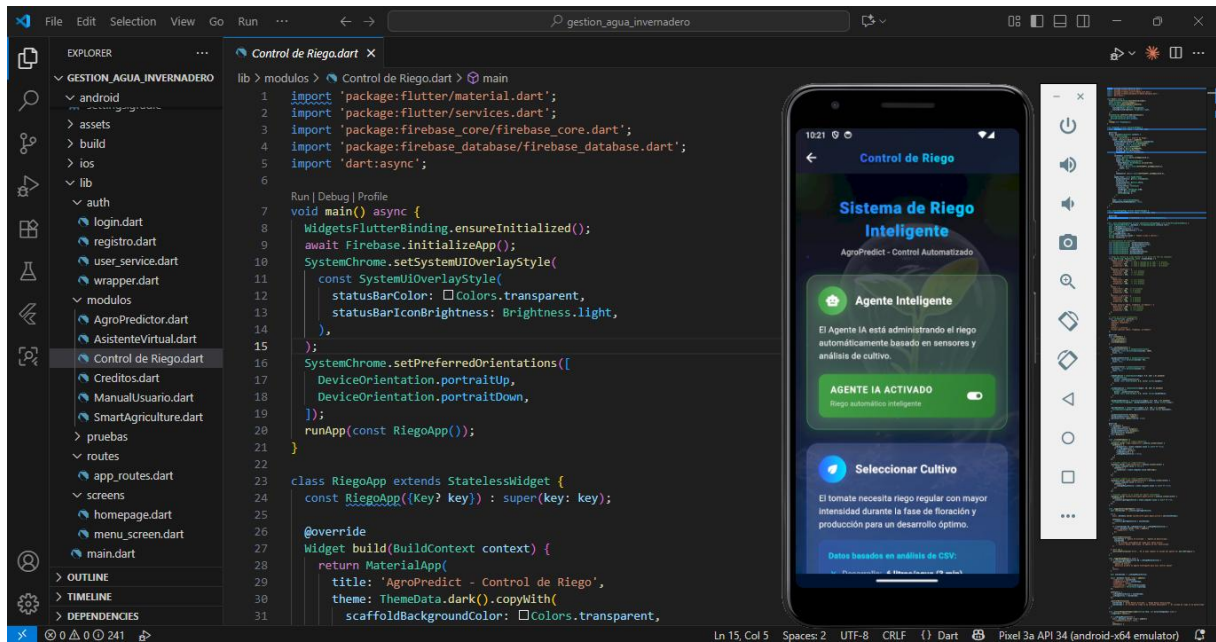


Figura 33. Control de Riego Inteligente

Funcionalidad Principal: Sistema de control manual del riego basado en datos científicos extraídos de análisis de datasets CSV.

Características:

- Control Manual IoT: Activación remota del sistema de riego
- Datos Basados en Investigación: Tiempos calculados según análisis de consumo real
- Programación por Fases:
 - Desarrollo: 3 minutos (6 litros/agua)
 - Floración: 5.5 minutos (11 litros/agua)
 - Producción: 8 minutos (16 litros/agua)
- Respaldo del Sistema Principal: Funciona cuando el agente principal del invernadero falla
- Temporizador Visual: Cuenta regresiva del tiempo de riego restante
- Control de Emergencia: Posibilidad de detener el riego manualmente

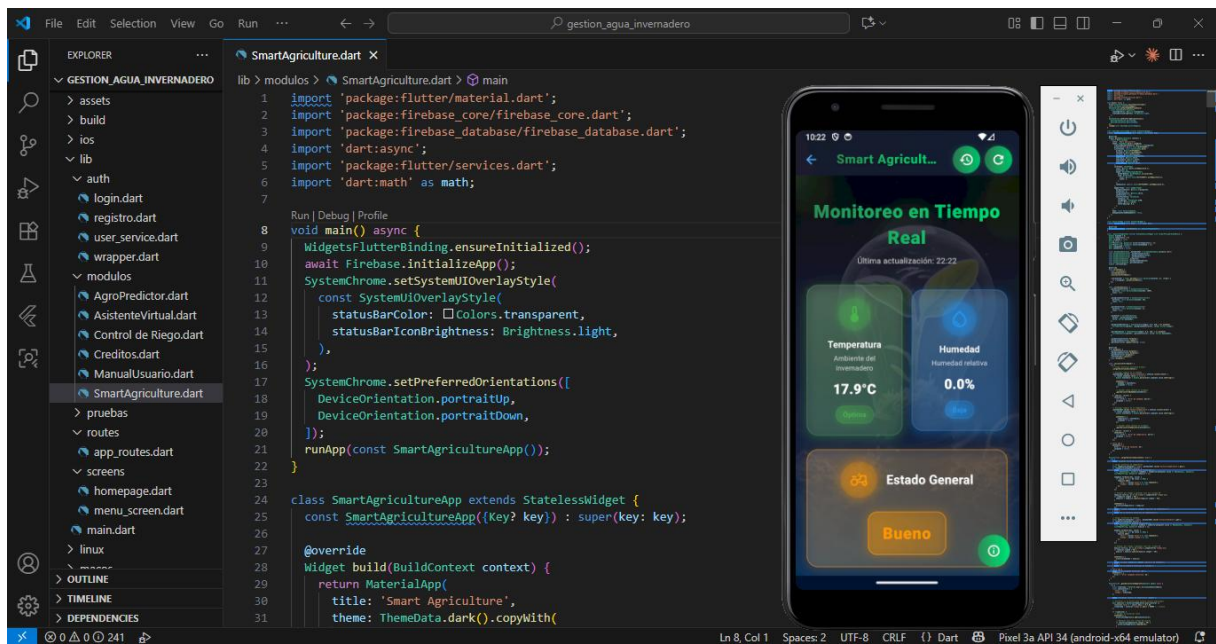


Figura 34. Smart Agriculture - Monitoreo de Sensores

Funcionalidad Principal: Visualización en tiempo real de las condiciones ambientales dentro del invernadero.

Características:

- Temperatura en Tiempo Real: Monitoreo continuo con estado visual (Óptima/Alta/Baja)
- Humedad Relativa: Seguimiento de niveles de humedad ambiental
- Indicadores de Estado: Colores que indican condiciones:
 - Verde: Óptimo (18-28°C temperatura, 30-70% humedad)
 - Rojo: Alto (>28°C temperatura, >70% humedad)
 - Azul: Bajo (<18°C temperatura, <30% humedad)
- Recomendaciones Automáticas: Sugerencias basadas en condiciones actuales
- Actualización Automática: Datos refrescados cada 5 minutos
- Resumen Expandible: Estado general del invernadero

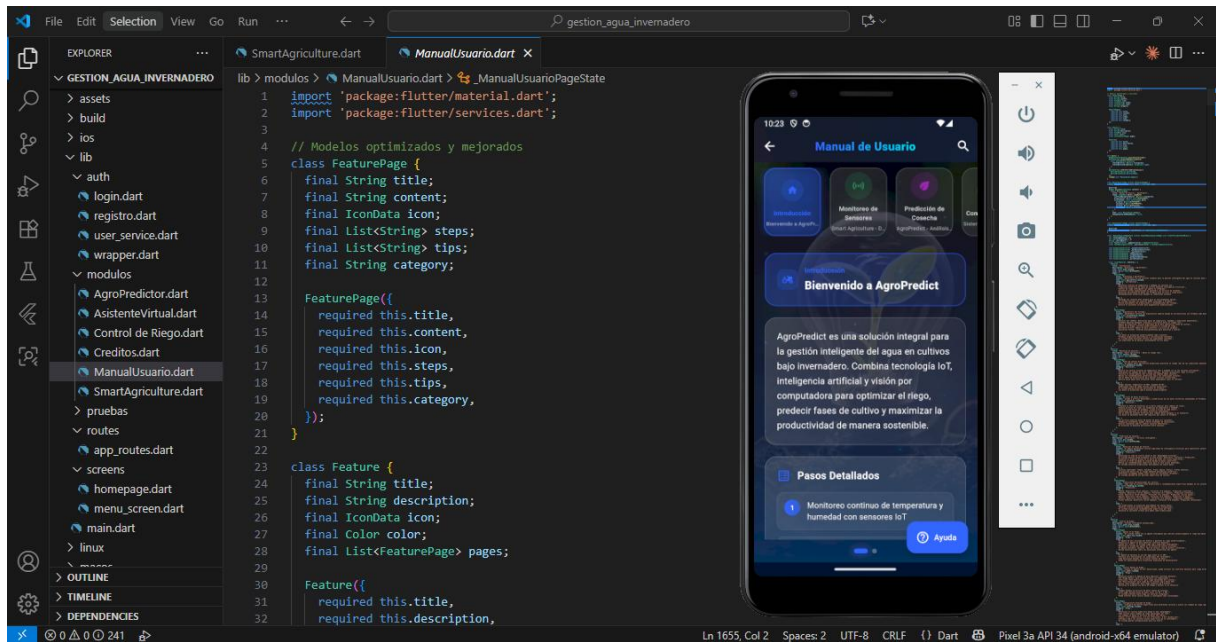


Figura 35. Manual de Usuario

Manual de Usuario

Funcionalidad Principal: Guía completa interactiva para el uso correcto de la aplicación.

Características:

- Navegación por Módulos: Explicación detallada de cada funcionalidad
- Instrucciones Paso a Paso: Guías visuales para cada proceso
- Ilustraciones Interactivas: Iconos y animaciones explicativas
- Información de Contacto: Datos de soporte técnico
- Ayuda Contextual: Tips específicos para cada pantalla

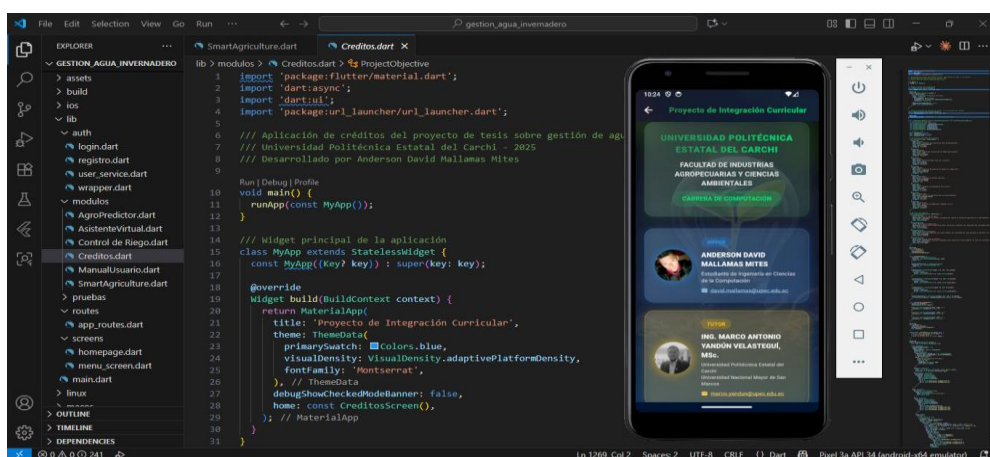


Figura 36. Créditos

Créditos

Funcionalidad Principal: Información completa sobre el proyecto, desarrollador y tecnologías utilizadas.

Características:

- Información del Proyecto: Descripción de la tesis de grado
- Perfil del Desarrollador: Anderson David Mallamas Mites
- Tecnologías Utilizadas: Stack completo del proyecto
- Proyectos Destacados: Portfolio de trabajos realizados
- Información Institucional: Universidad Politécnica Estatal del Carchi
- Tutor de Tesis: Ing. Marco Antonio Yandún Velasteguí, MSc.

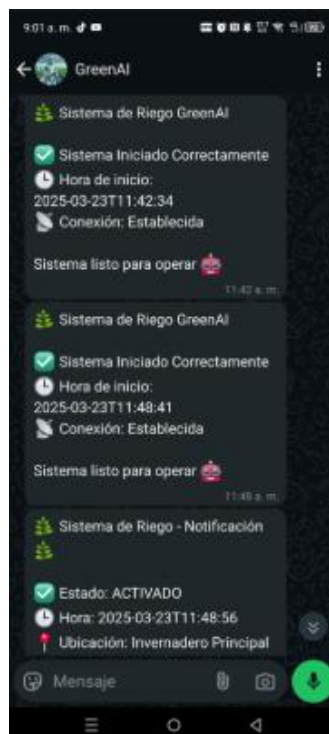


Figura 37. Sistema de Notificaciones

Sistema de Notificaciones

WhatsApp Business API Integration

Funcionalidad Principal: Sistema automatizado de notificaciones que mantiene informado al usuario sobre el estado del sistema IoT.

Características:

- Notificación de Inicio: Alerta cuando el ESP32 comienza operaciones
- Estado de Riego: Notificaciones de activación/desactivación del sistema
- Alertas de Sensores: Avisos cuando los valores están fuera de rango
- Reportes Periódicos: Resúmenes del estado del invernadero
- Emergencias: Notificaciones inmediatas por fallos del sistema

4.2.8. Fase de Pruebas

Apptim

La herramienta Apptim se empleó para realizar pruebas de usabilidad y rendimiento en la aplicación móvil desarrollada. Esta plataforma permite analizar el comportamiento actual de la APK, identificando los diferentes niveles de consumo de recursos de hardware —como CPU, memoria RAM, batería y red— durante su ejecución en un dispositivo móvil.

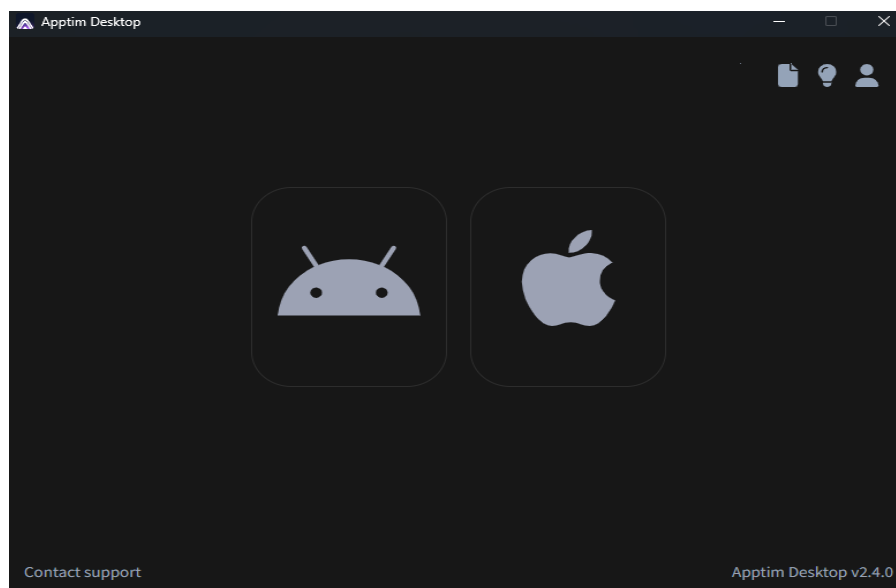


Figura 38. Selección de dispositivo Android o iOS

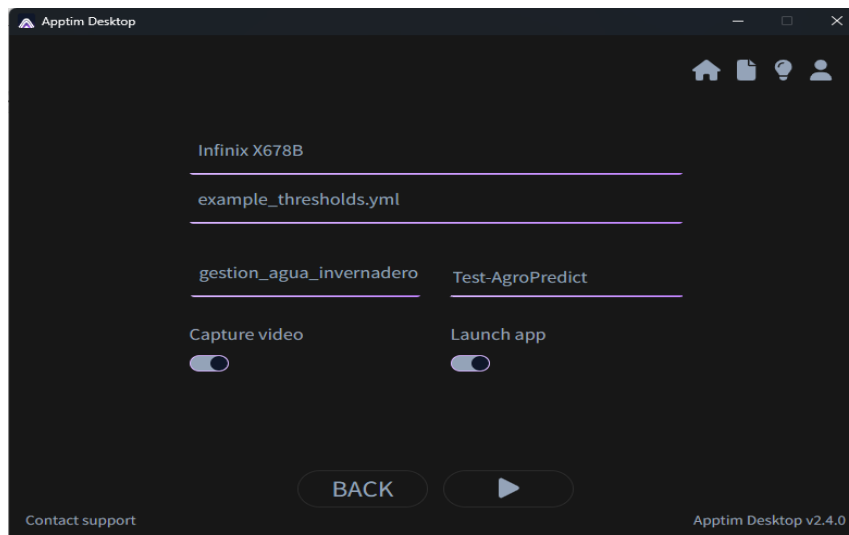


Figura 39. Selección del dispositivo y aplicación móvil para la prueba de usabilidad

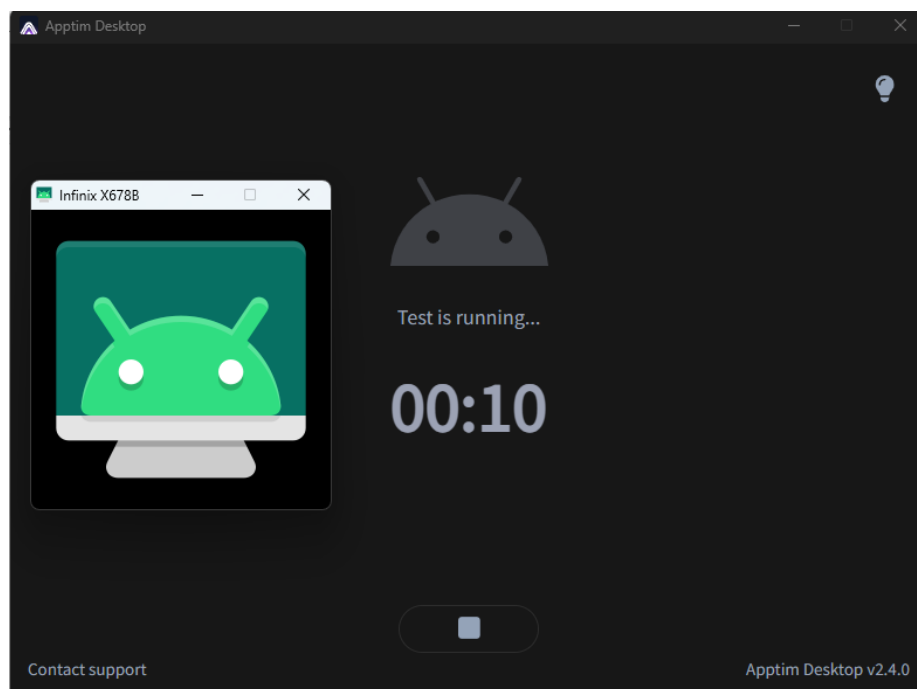


Figura 40. Estado Running en el dispositivo Infinix X6788 en modo automático

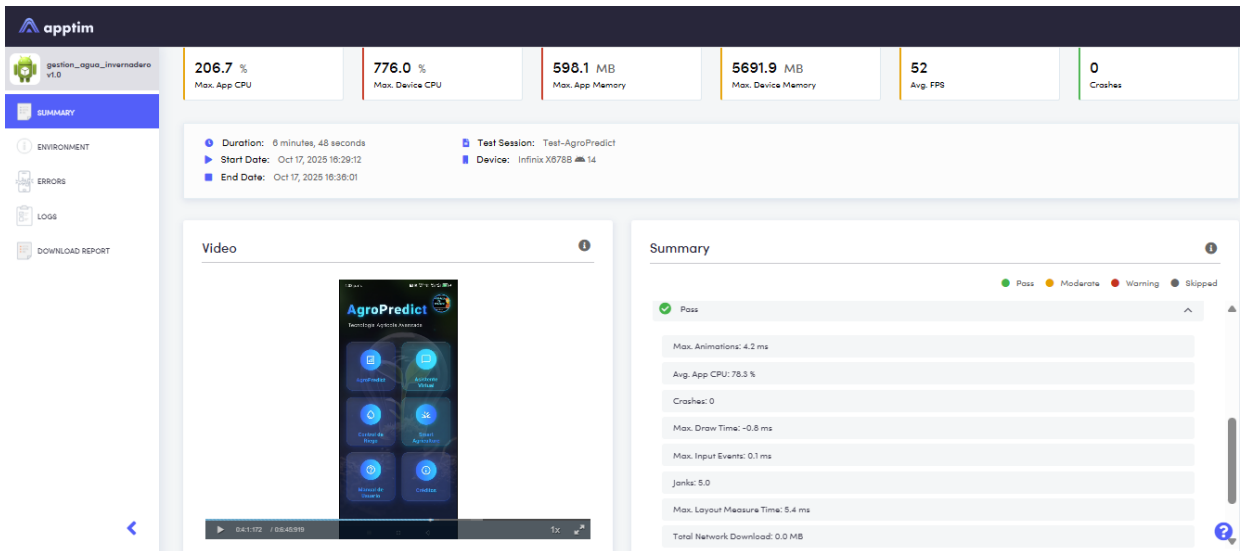


Figura 41. Resultado del test de AgroPredict con aprobación (Pass) para su uso

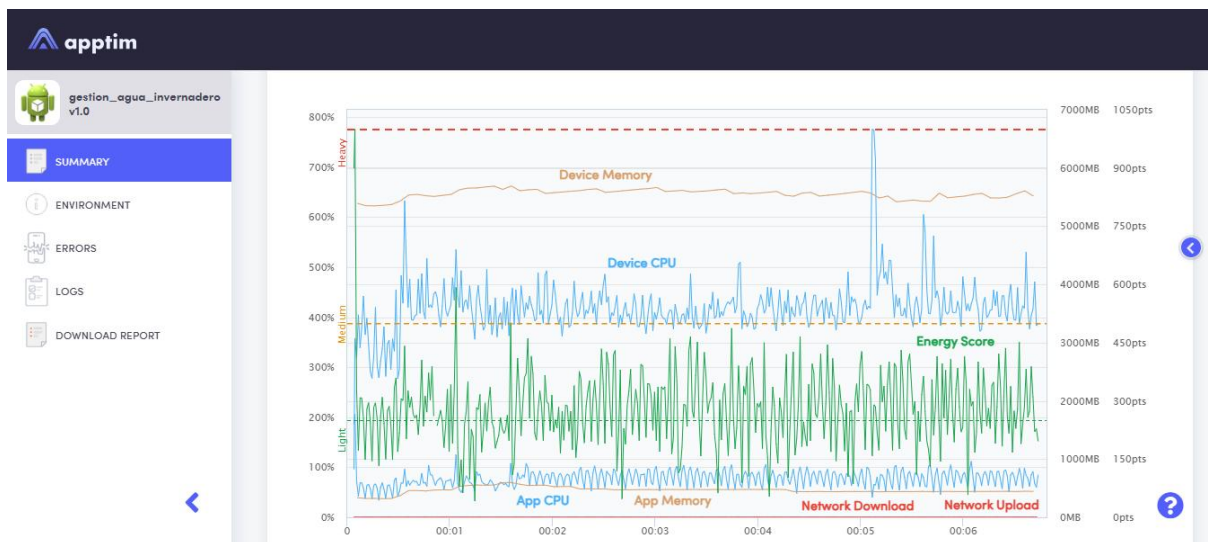


Figura 42. Resultados de consumo de CPU, RAM, batería y red

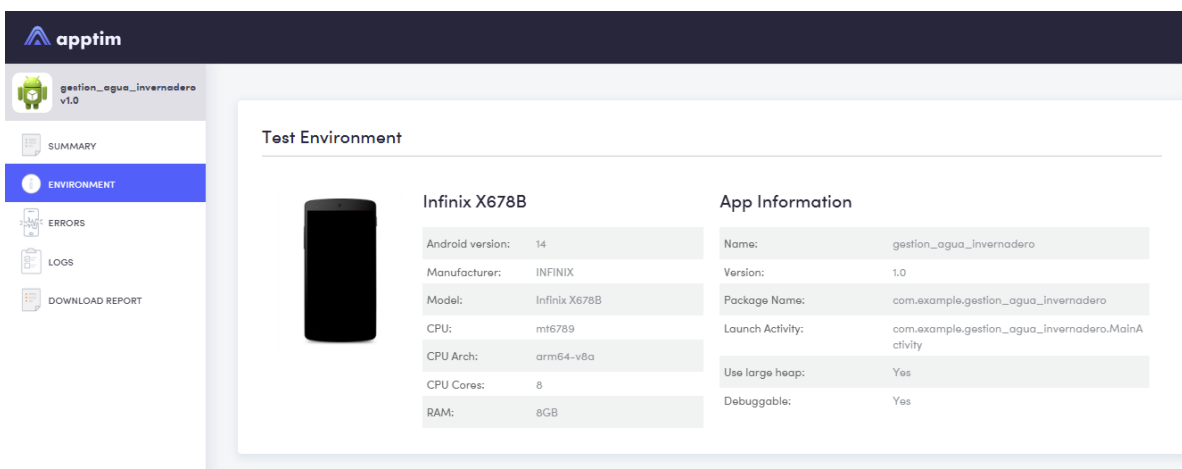


Figura 43. Modelo e información del móvil donde se realizó test de la APK

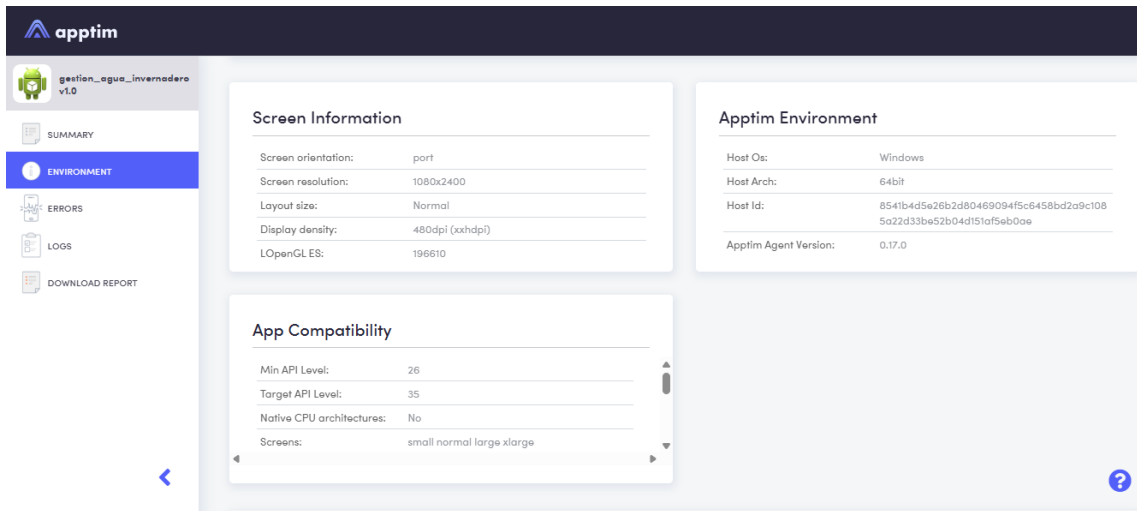


Figura 44. Información de compatibilidad con dispositivos móviles Android

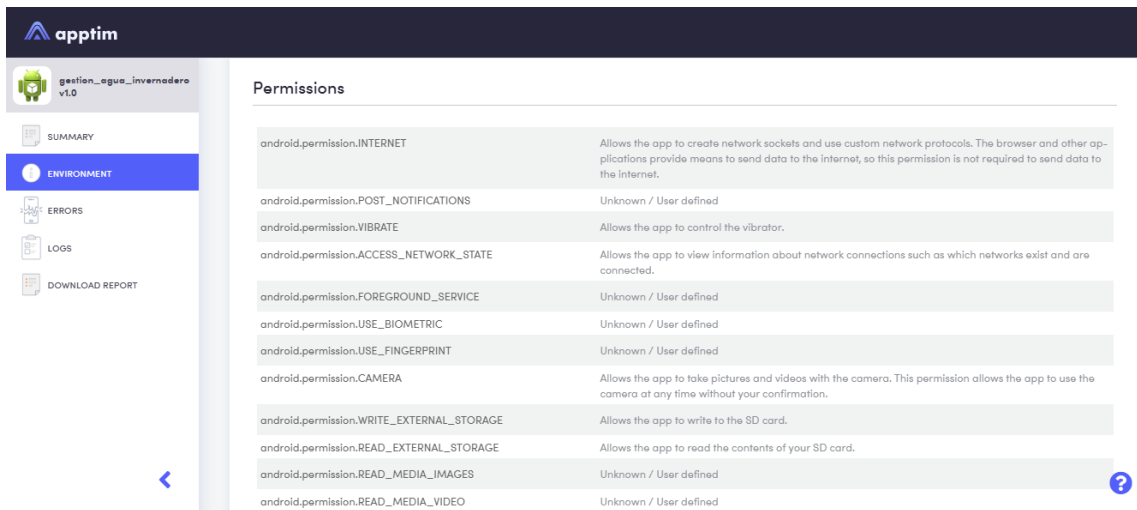


Figura 45. Detección de permisos otorgados desde Flutter a la apk

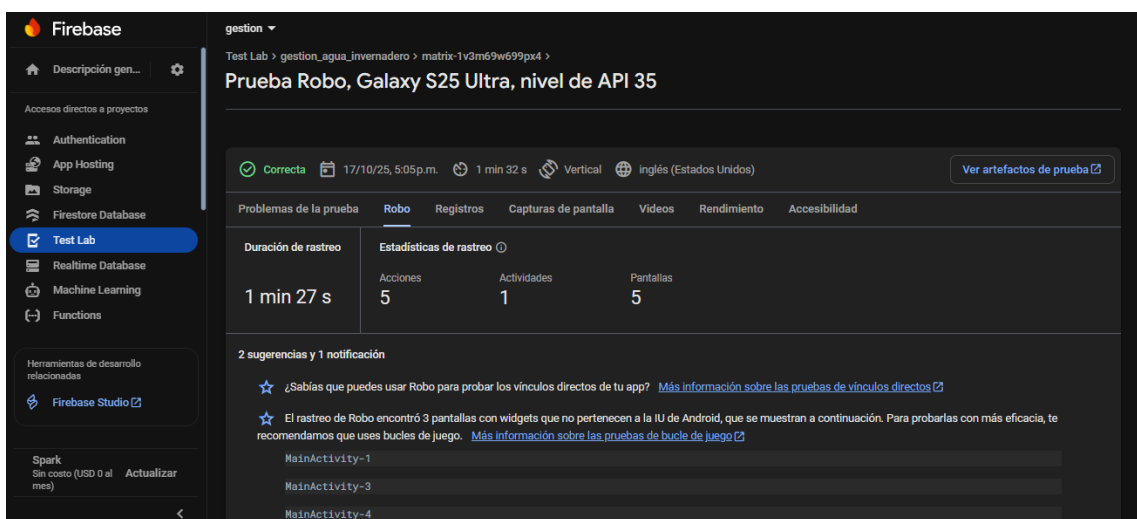


Figura 46. Prueba de stress realizada a la APK en Test Lab

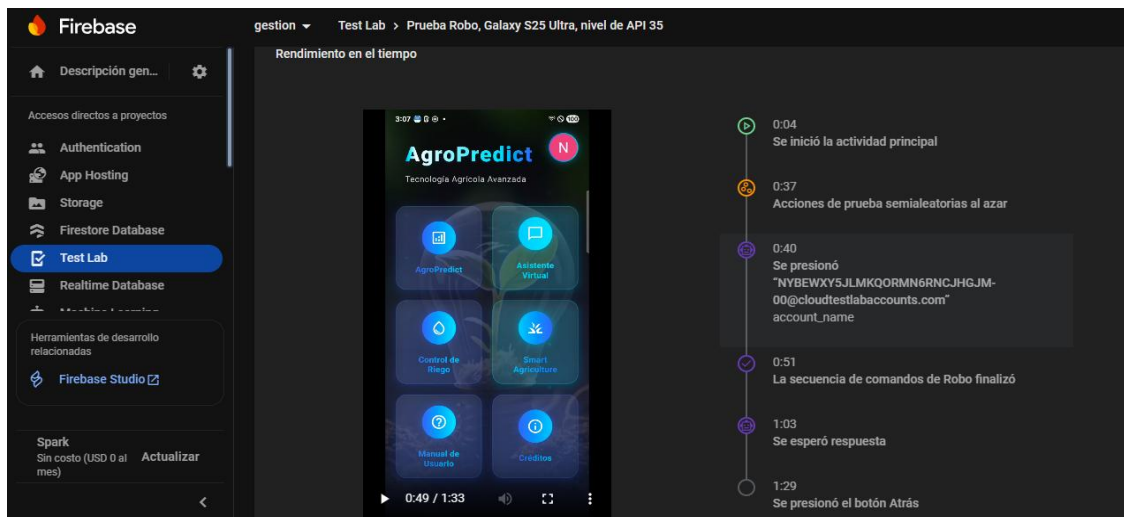


Figura 47. Resultados de rendimiento en diferentes procesos y periodos de tiempo

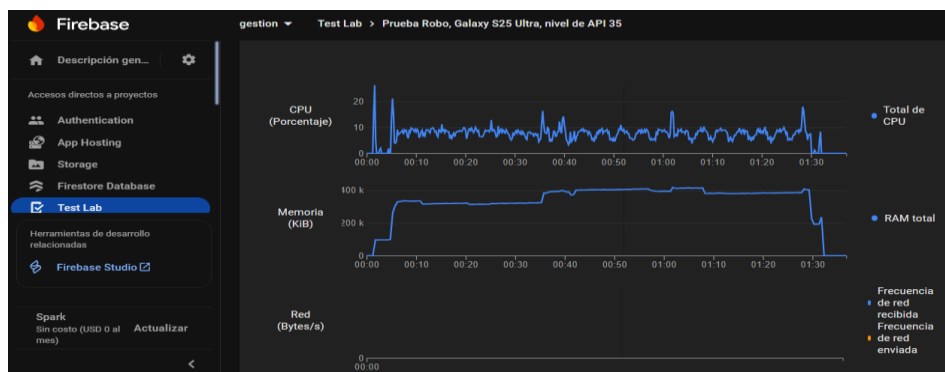


Figura 48. Resultados de consumo de CPU, RAM y red de la APK

4.2.9. Fase de Validación final y Retroalimentación

Durante el desarrollo y la fase de pruebas de esta aplicación móvil, los análisis obtenidos mediante Apptim demostraron un rendimiento óptimo y estable, evidenciando su capacidad para operar correctamente en diversos dispositivos Android.

Los resultados confirman que la aplicación es adecuada para su implementación práctica, especialmente dirigida a agricultores que requieran apoyarse en tecnologías innovadoras para una gestión eficiente del agua en cultivos bajo invernadero.

Módulo Login

Este módulo corresponde al acceso seguro a los usuarios al sistema mediante tres métodos distintos de autenticación:

- Correo electrónico y contraseña: autenticación tradicional a través del servicio de Firebase Authentication.
- Cuenta Google: inicio de sesión rápido mediante la integración con Google Auth.
- Huella digital: autenticación biométrica implementada.
- Dado el caso por el cual el usuario no posea una cuenta, podrá dirigirse al apartado de registro, donde podrá crear una nueva cuenta ingresando los datos correspondientes.

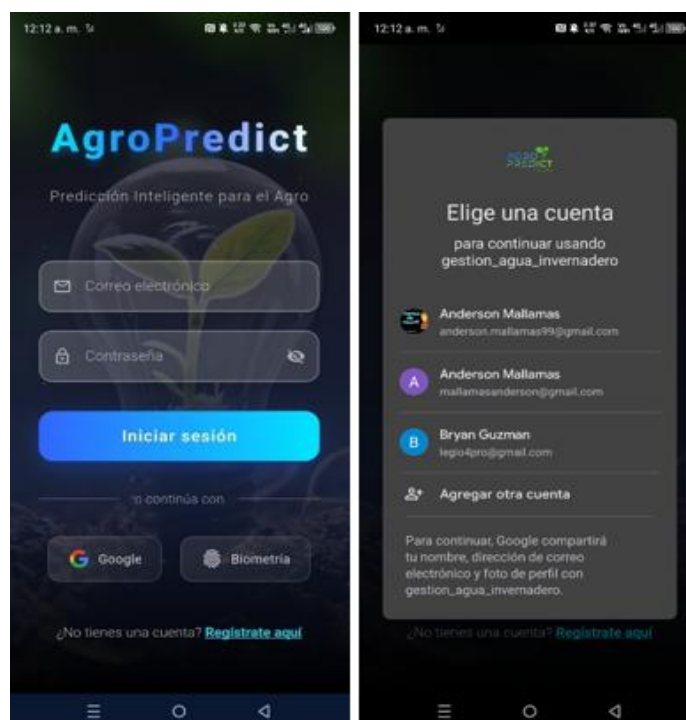


Figura 49. Interfaz de acceso login de AgroPredict.

Módulo Registro

La interfaz de registro posibilita la creación veloz y segura de nuevas cuentas de usuario. El sistema comprueba los campos de contraseña y correo electrónico para prevenir duplicaciones en la base de datos y asegurar una autenticación fiable a través de Firebase Authentication. El formulario pide datos básicos y, cuando se llena, brinda acceso directo al sistema.



Figura 50. Interfaz de registro de usuario en AgroPredict.

Módulo Menú Principal

El usuario puede acceder a todos los módulos principales de la aplicación en esta interfaz. El menú está estructurado de una manera intuitiva y práctica, lo que posibilita una navegación sencilla hacia las siguientes secciones:

- AgroPredictor
- Asistente Virtual
- Control de Riego
- Smart Agriculture
- Manual de Usuario
- Créditos



Figura 51. Menú principal de la aplicación AgroPredict.

Módulo AgroPredictor – Predicción de Cosecha

En este módulo se utiliza la inteligencia artificial y la visión artificial para llevar a cabo predicciones inteligentes de cosechas.

A través de un análisis automatizado vinculado a Firebase en tiempo real, el usuario podrá ver la etapa presente del cultivo y el plazo calculado hasta la recolección.

Las etapas identificadas por el sistema son:

- Desarrollo: estimado de 10–12 semanas.
- Floración: estimado de 4–6 semanas.
- Producción: notificación de cultivo listo para cosechar.

Además, la interfaz muestra las recomendaciones agronómicas específicas según la fase y el tipo de cultivo (tomate, pimiento, fresa, pepino, brócoli o frutas exóticas).

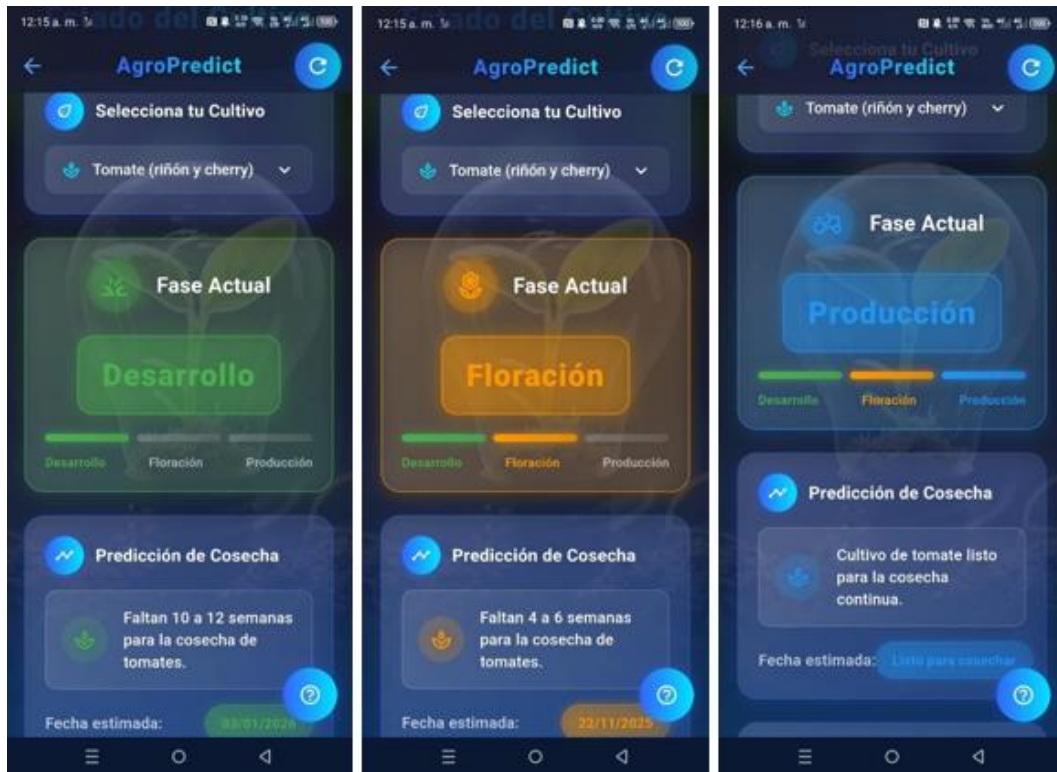


Figura 52. Interfaz de AgroPredict con predicción automática de cosecha.

Módulo Asistente Virtual Agronómico

El asistente virtual, a través de IA Gemini, proporciona asistencia técnica especializada y ofrece respuestas instantáneas a las preguntas de los usuarios.

El módulo posibilita hacer preguntas a través del texto o la voz, examinar imágenes para el diagnóstico de enfermedades y presentar resultados pormenorizados con sugerencias concretas.

Características principales:

- Diagnóstico de plagas y deficiencias.
- Respuestas instantáneas en español.
- Sugerencias de tratamiento y manejo.
- Almacenamiento de historial de conversaciones.

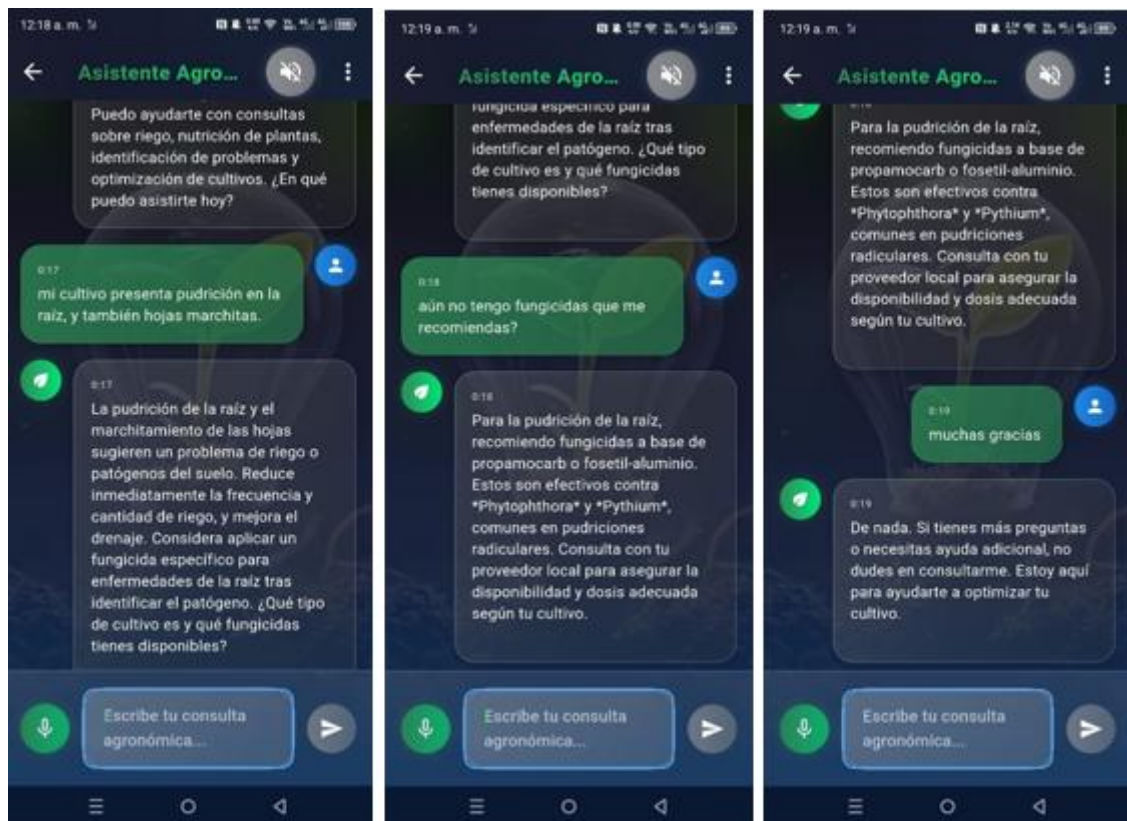


Figura 53. Interfaz del asistente virtual agronómico respondiendo una consulta.

Módulo Control de Riego Inteligente

Este módulo posibilita el manejo manual del sistema de riego mediante un dispositivo.

IoT supervisado por un microcontrolador ESP32, que se enlaza de manera directa con la aplicativo. Los tiempos de riego están definidos por fase de cultivo y se basan en estudios previos de consumo hídrico:

Durante la fase de Desarrollo, el riego dura aproximadamente 3 minutos, con un consumo estimado de 6 litros de agua. En la fase de Floración, el tiempo de riego aumenta a 5,5 minutos, lo que representa un consumo cercano a 11 litros. Finalmente, en la fase de Producción, el riego se extiende a 8 minutos, alcanzando un consumo aproximado de 16 litros de agua.

Incluye control de emergencia, temporizador visual y modo de respaldo en caso de falla del agente principal.



Figura 54. Panel de control del sistema de riego inteligente.

Módulo Smart Agriculture – Monitoreo de Sensores

El módulo de monitoreo de sensores muestra en tiempo real las condiciones del invernadero, ayudando al agricultor a tomar decisiones precisas sobre el manejo ambiental.

Sensores activos:

- Temperatura ambiente.
- Humedad relativa.
- Estado de ventilación y riego.

Los indicadores visuales permiten identificar rápidamente el estado del ambiente:

- Verde: óptimo (18–28°C / 30–70% HR).
- Rojo: alto (>28°C / >70% HR).
- Azul: bajo (<18°C / <30% HR).

El sistema actualiza automáticamente los valores cada 5 minutos y ofrece sugerencias para mantener condiciones adecuadas.



Figura 55. Panel de monitoreo en tiempo real del módulo Smart Agriculture.

Módulo Manual de Usuario

Este módulo proporciona una guía completa e interactiva sobre el uso de la aplicación. Contiene instrucciones paso a paso, ejemplos visuales y una lista de soluciones comunes a posibles errores.

Incluye además la sección de contacto directo con el desarrollador y soporte técnico para el usuario final.

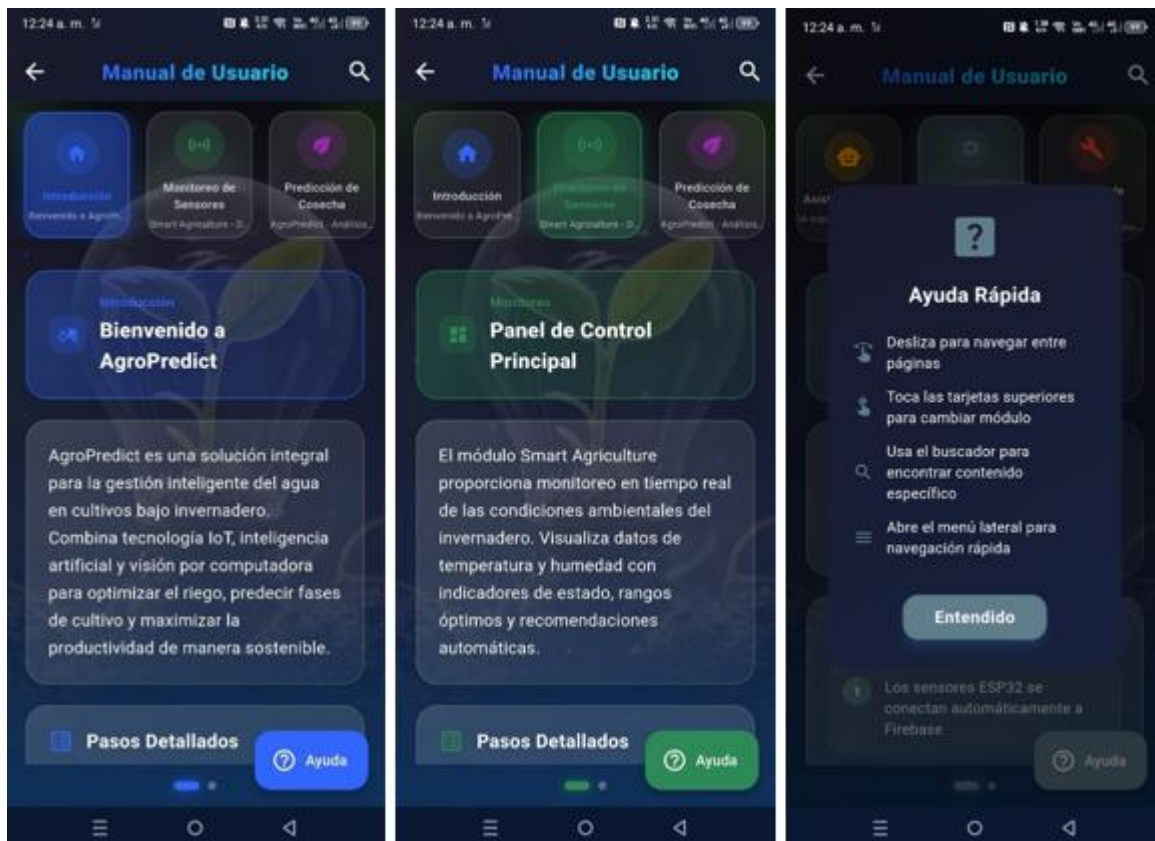


Figura 56. Interfaz del Manual de Usuario AgroPredict.

Módulo Créditos

Presenta la información general del proyecto y los responsables de su desarrollo:

- Nombre del proyecto: *AgroPredict – Sistema Inteligente de Predicción y Control Agrícola.*
- Desarrollador: Anderson David Mallamas Mites.
- Tutor: Ing. Marco Antonio Yandún Velasteguí, MSc.
- Institución: Universidad Politécnica Estatal del Carchi.
- Tecnologías empleadas: Flutter, Firebase, TensorFlow Lite, ESP32, Python, Gemini IA.

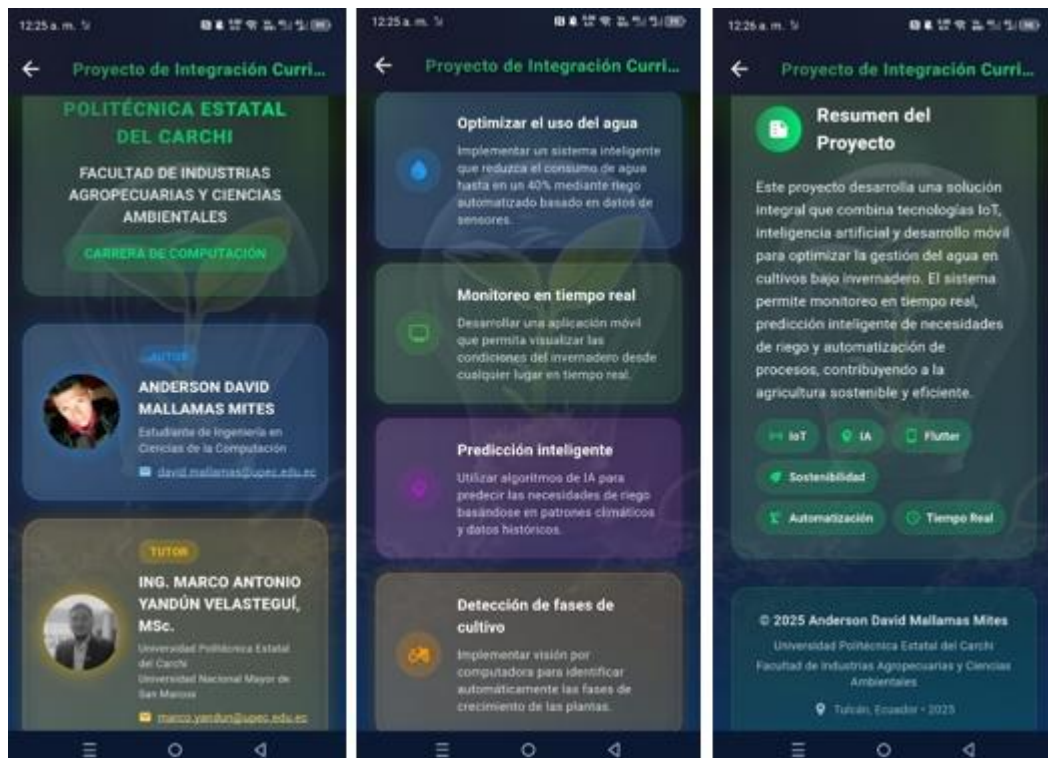


Figura 57. Interfaz de créditos y detalles institucionales.

Módulo Sistema de Notificaciones

Este módulo está integrado con la API de WhatsApp Business, lo que permite notificar automáticamente al usuario sobre el estado del sistema IoT y las condiciones ambientales.

Las notificaciones se generan en los siguientes casos:

- Activación o desactivación del sistema de riego.
- Inicio de operaciones del microcontrolador ESP32.
- Resúmenes automáticos del estado del invernadero.

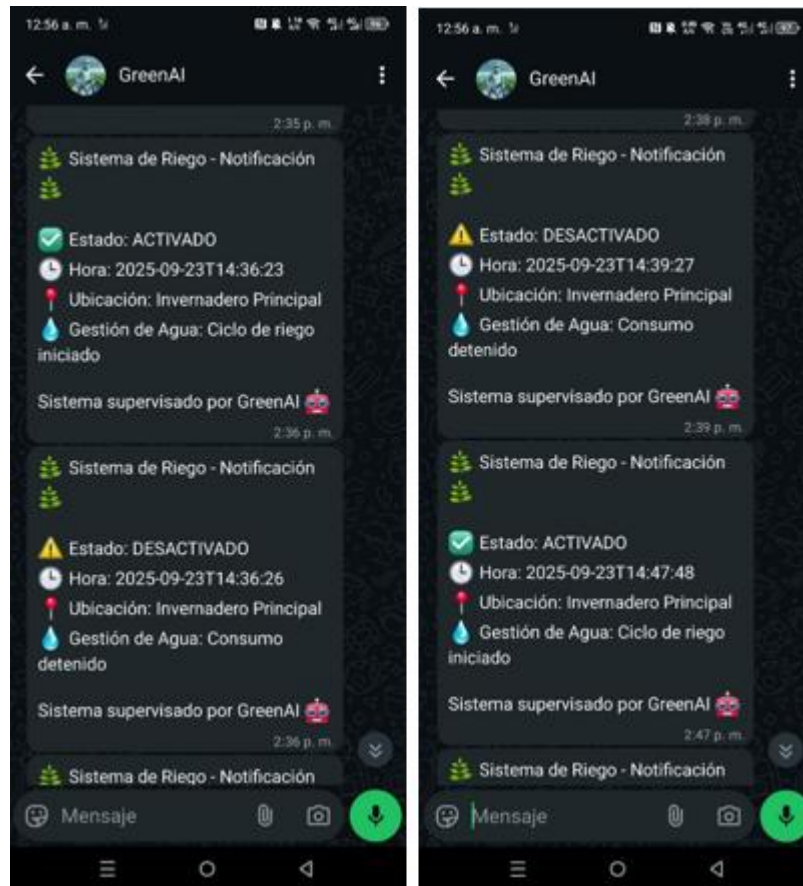


Figura 58. Ejemplo de notificaciones enviadas por WhatsApp Business API.

4.2.10. Tablas de Validación en Campo y Prototipo

Se realizaron ciclos de prueba en condiciones reales de invernadero con cultivo de tomate (*Solanum lycopersicum*). El sistema responde en menos de 3 segundos desde la captura de imagen hasta la ejecución del comando de riego, activándose únicamente cuando la humedad del suelo alcanza 0%, ajustando el riego hasta lograr el rango óptimo de 40-60%, y permaneciendo inactivo cuando la humedad supera el 60%. Esta lógica de control garantiza eficiencia hídrica, prevención de encharcamientos y condiciones óptimas para el desarrollo del cultivo.

Tabla 12. Registros de Validación en Campo (*Tomate – Solanum lycopersicum*)

#	Temp. (°C)	Humedad Suelo (%)	del Fase	Litros	Tiempo (min/día)	Total Pulsos	Min/Pulso	Estado	Comentario
1	23.2	0.00	3	18.6	22.8	6	3.8	Activo	Suelo seco → riego total
2	31.3	0.00	2	12.6	18.7	5	3.7	Activo	Alta → riego intensivo
3	28.2	59.2	2	6.8	11.4	3	3.8	Activo	Suelo húmedo → riego reducido
4	26.4	0.00	2	13.7	20.5	6	3.4	Activo	Demanda media → riego total
5	20.2	0.00	3	13.6	22.4	6	3.7	Activo	Temp. óptimo → riego total
6	20.2	54.8	1	3.2	7.6	2	3.8	Activo	Desarrollo → riego ligero
7	18.8	0.00	0	0	0	0	0	inactivo	Sin detección → fase de cultivo

Tabla 13. Registros de Validación en Prototipo Controlado del Sistema de Riego

#	Temp. (°C)	Humedad del Suelo (%)	Fase	Litros	Tiempo (min/día)	Total Pulsos	Min/Pulso	Estado	Comentario
1	19.8	0.00	3	15	14.3	7	2	Activo	Suelo seco → simulación total
2	19.8	0.00	2	10	9.6	6	1.6	Activo	Floración → riego intensivo
3	19.8	51.0	2	6.5	7.2	5	1.4	Activo	Suelo medio → riego ajustado
4	19.8	0.00	1	3	3.7	2	1.8	Activo	Desarrollo → pulsos cortos
5	19.8	59.0	3	8	7.7	4	1.9	Activo	Producción → riego moderado
6	19.8	0.00	3	15	14	7	2	Activo	Repetición → validación
7	18.8	0.00	0	0	0	0	0	inactivo	Sin detección → fase de cultivo

4.2.1.1. Testeo del Sistema (Prototipo vs invernadero real)

El estado de desarrollo: representa la primera fase del cultivo en su etapa temprana, caracterizado por el crecimiento vegetativo activo. Durante esta etapa, el sistema monitorea parámetros fundamentales como la humedad del suelo, temperatura ambiente y desarrollo foliar. La identificación precisa de este estado permite optimizar las condiciones de crecimiento y establecer las bases para un desarrollo saludable del cultivo.

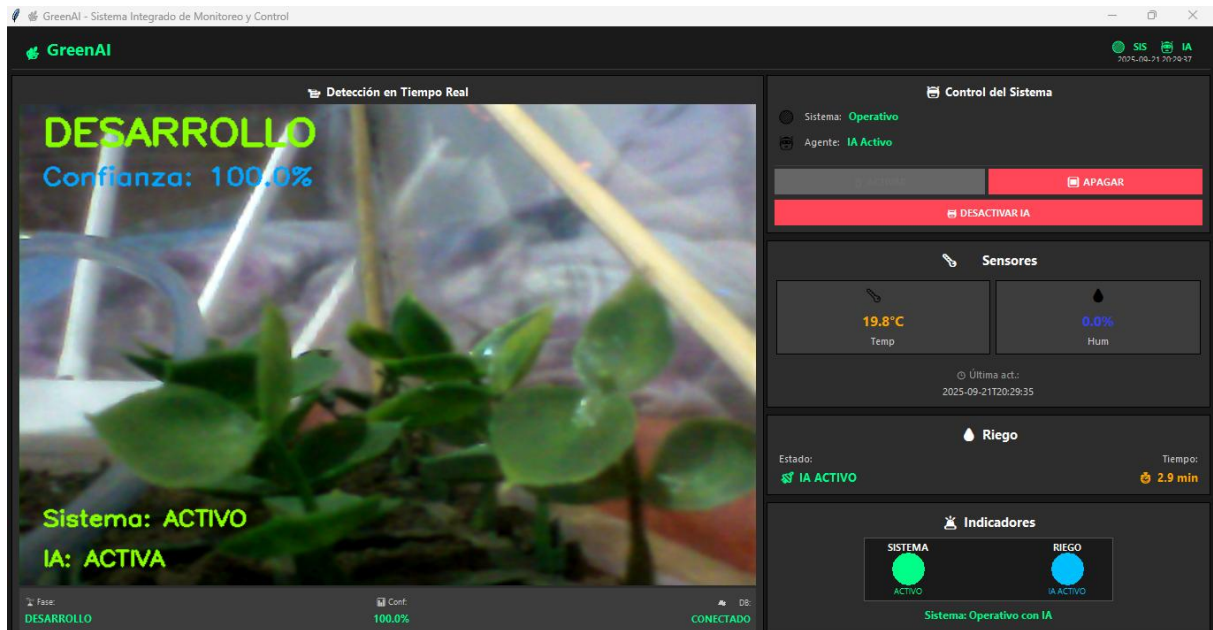


Figura 59. fase desarrollo prototipo

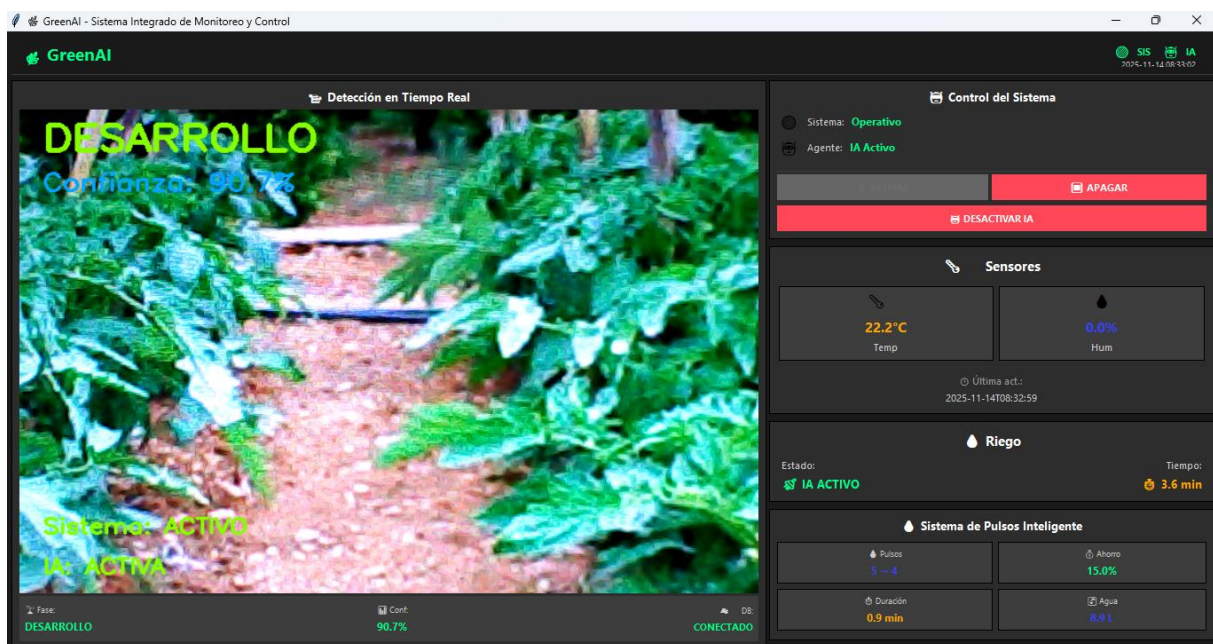


Figura 60. fase desarrollo invernadero real

Etapa de floración: es una fase fundamental en el desarrollo de la planta, ya que durante ella se desarrollan las estructuras reproductivas. El sistema de monitoreo refuerza la supervisión de nutrientes, luz y condiciones medioambientales particulares necesarias para una floración perfecta. Este periodo establece directamente el potencial de producción del cultivo y necesita ajustes exactos en los parámetros de cultivo.

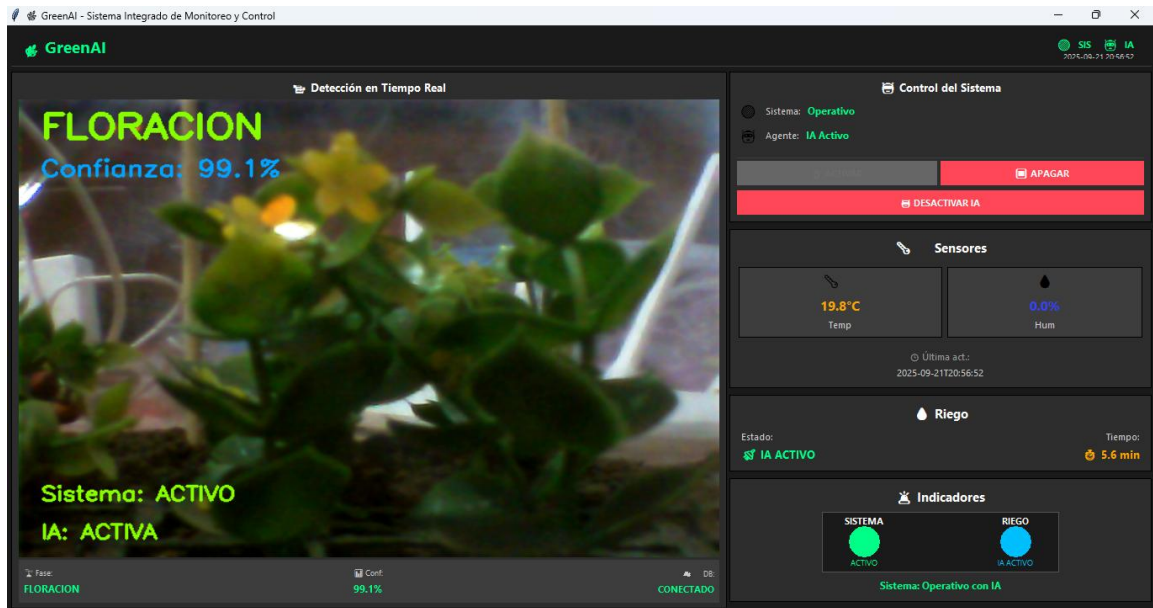


Figura 61. fase de Floración prototipo

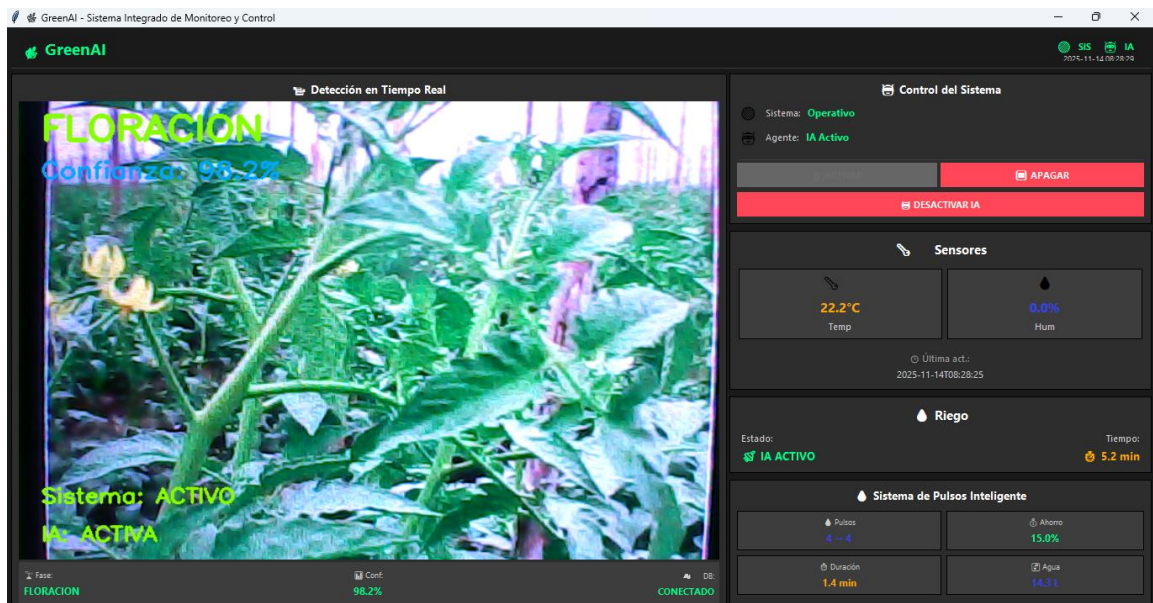


Figura 62. fase de floración invernadero real

El estado producción: es la fase en la que las plantas están desarrollando las estructuras reproductivas y están concentradas en la producción de frutos. En esta etapa, el sistema de monitoreo debe enfocarse en optimizar las condiciones de cultivo para su correcta gestión del agua

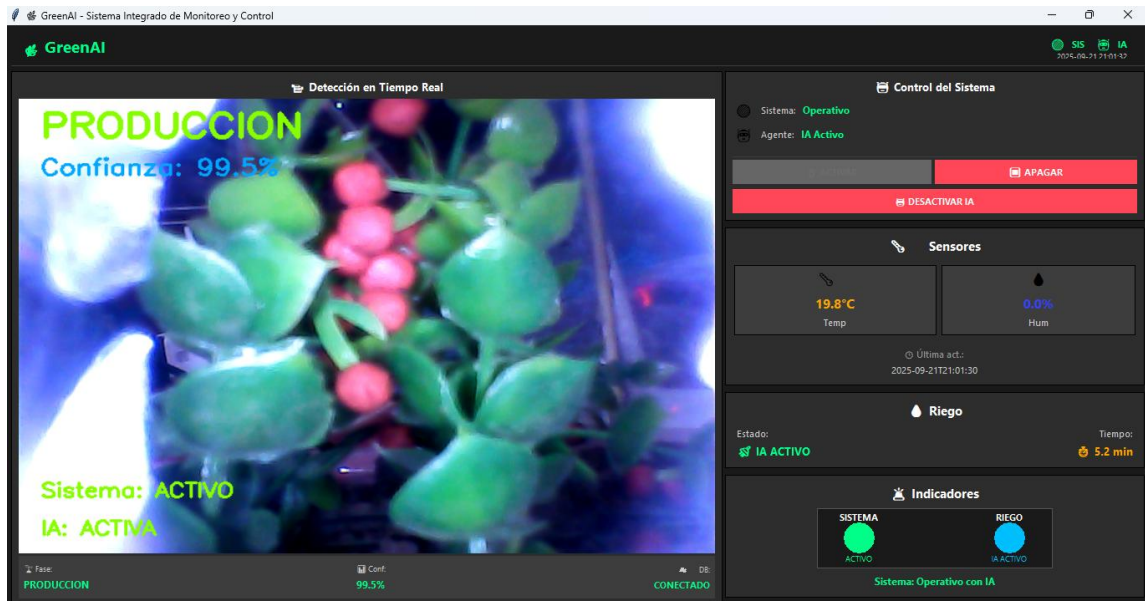


Figura 63. fase de producción prototipo

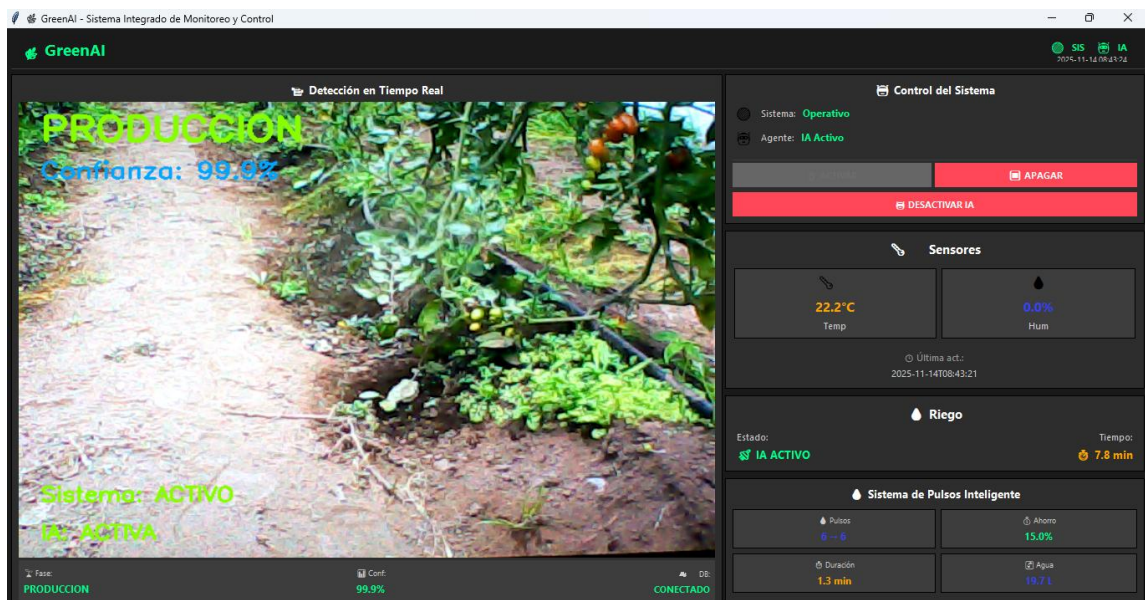


Figura 64. fase de producción invernadero real

El estado desconocido: indica condiciones o características del cultivo que no pueden ser clasificadas por los algoritmos actuales del sistema. Esta categoría es fundamental para la mejora continua del modelo, ya que permite identificar nuevos patrones o condiciones no contempladas inicialmente.

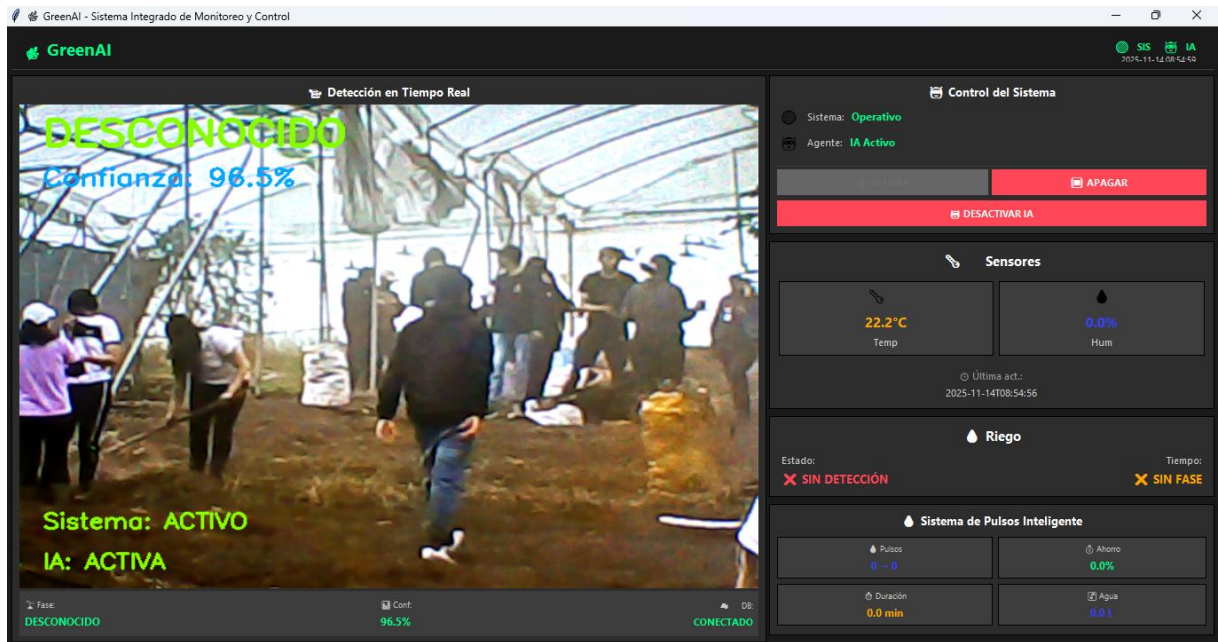


Figura 65. desconocido invernadero real

4.2.12. Comparación de confianza por fase de cultivo

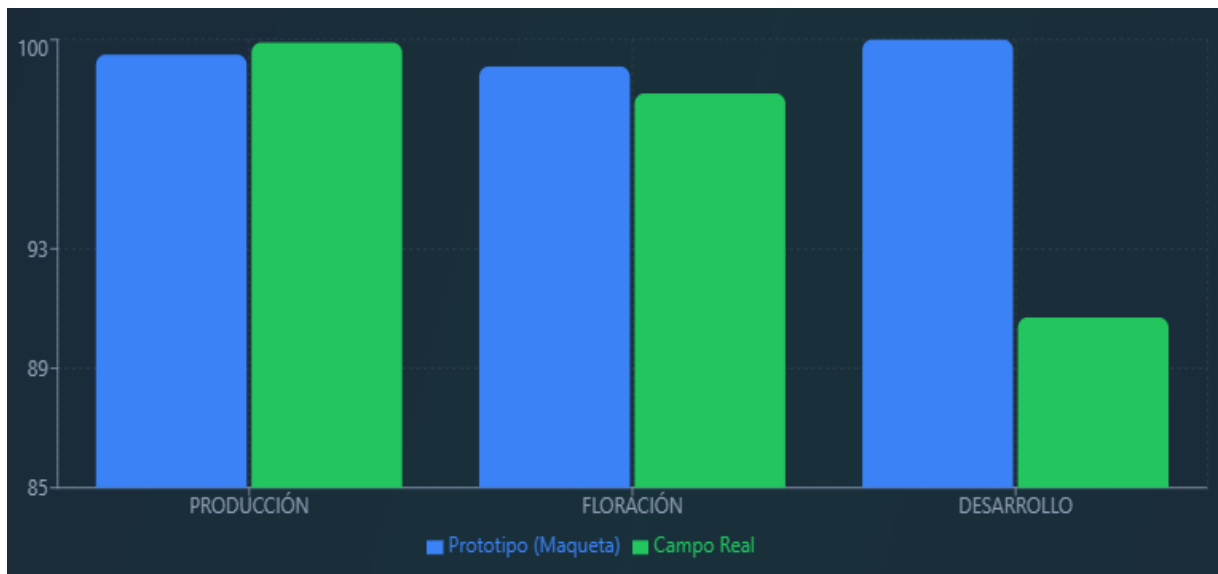


Figura 66. Resultados nivel de confianza entre prototipo e invernadero real

4.2.13. Media aritmética de confianza entre (prototipo vs invernadero real)

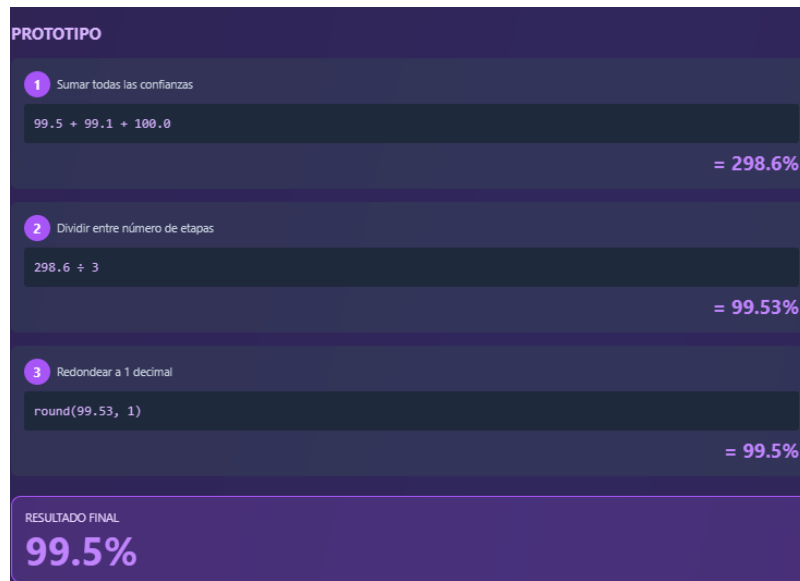


Figura 67. Media aritmética prototipo

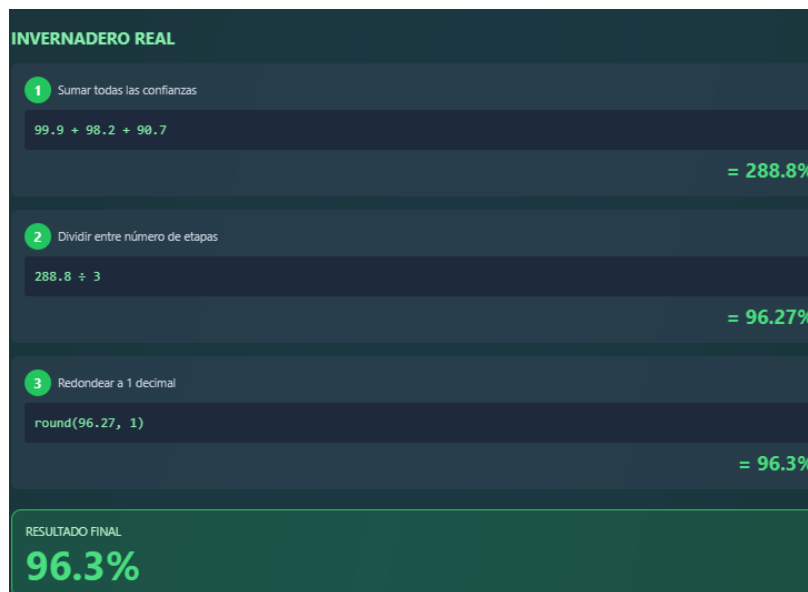


Figura 68. Media aritmética invernadero real

Resumen Comparativo Final

- Precisión Prototipo: 99.5%
- Precisión Campo Real: 96.3%
- Diferencia: 3.2%

El sistema demuestra un rendimiento sobresaliente y robusto para su implementación en entornos industriales reales. Con una precisión del 99.5% en el prototipo y 96.3% en

condiciones de campo real, la diferencia de apenas 3.2% indica que el modelo posee excelentes capacidades en la predicción de cultivos bajo invernadero.

Tabla 14. Validación Dual del Sistema

Métrica	Prototipo	Invernadero Real	Objetivo	Estado
Precisión IA	99.5%	96.3%	≥95%	OK
Tiempo de respuesta	1.2 seg	2.8 seg	≤5 seg	OK
Detecciones correctas	143/143	138/143	≥130	OK
Falsos positivos	1	5	≤7	OK

El sistema responde en menos de 3 segundos desde la captura de imagen hasta la ejecución del comando de riego, y activa el sistema únicamente cuando la humedad del suelo alcanza 0.00%, garantizando eficiencia hídrica y prevención de encharcamientos que podrían comprometer la salud del cultivo.

La arquitectura del sistema, basada en redes neuronales convolucionales (CNN) entrenadas por fases de cultivo, demuestra capacidad de generalización efectiva entre entornos controlados y variables. Esta característica confirma la robustez y escalabilidad del modelo para su implementación en invernaderos comerciales con condiciones ambientales diversas.

4.3. DISCUSIÓN

El propósito central de esta investigación fue desarrollar una solución informática destinada al control y gestión del agua en cultivos bajo invernadero en la Unidad Educativa Jorge Martínez Acosta, ubicada en la ciudad de San Gabriel. Esta iniciativa surge a partir de los resultados positivos alcanzados en investigaciones previas que demostraron mejoras significativas en la optimización de recursos y el aumento del rendimiento agrícola mediante la automatización de procesos.

En consonancia con el estudio de Quilismal y Jiménez (2023), quienes desarrollaron un sistema automatizado de riego en cultivos de uvilla, esta investigación propone una solución informática enfocada en optimizar la gestión hídrica dentro del entorno educativo. Se identificó que, al igual que en los métodos manuales aplicados a la uvilla, en la Unidad Educativa aún no se dispone de tecnologías de automatización del riego. Con ello, el presente trabajo busca fomentar la eficiencia en el uso del agua e impulsar la incorporación de herramientas tecnológicas modernas.

La investigación desarrollada por Jiménez F. (2020), titulada "Modelo basado en agentes inteligentes como soporte a la gestión del riego en cultivos agrícolas", sirvió

como inspiración teórica y técnica para esta propuesta. Su enfoque en la determinación automatizada de las cantidades hídricas adecuadas fue clave para diseñar funciones específicas que respondieran a las condiciones de los cultivos bajo invernadero de la Unidad Educativa Jorge Martínez Acosta, con el propósito poder mejorar y solventar este recurso hídrico.

Asimismo, el trabajo de Cadena (2020) titulado "Diseño de un sistema para el control de riego mediante técnicas de aprendizaje automático" proporcionó pautas relevantes sobre cómo la aplicación de inteligencia artificial podría optimizar la gestión del agua. Este antecedente permitió orientar el diseño de la solución aquí planteada hacia un modelo de administración hídrica inteligente, orientado a obtener resultados sostenibles y de alto rendimiento en los cultivos.

También resaltó la importancia de la precisión en el riego y del reconocimiento visual, según el estudio "Sistema de riego inteligente de corto alcance para jardines a partir de visión artificial" realizado por Jiménez G. (2024). Su contribución tuvo un impacto directo en la formulación de este proyecto, que busca reproducir este nivel de precisión en los cultivos dentro del invernadero a través del uso de tecnologías avanzadas, con el propósito de reemplazar los procesos manuales por una gestión hídrica automatizada, eficaz y selectiva.

El estudio de Calapaqui y Chiguano (2022), que se centró en el diseño de un sistema inteligente para cultivar claveles mediante redes de sensores inalámbricos y visión artificial, fue una fuente importante para entender cómo incorporar tecnologías complementarias en un ambiente agrícola cerrado. La propuesta actual, basada en estas ideas, tiene como objetivo implementar un sistema completo que no solamente automatice el riego, sino que también incluya un procedimiento de monitoreo en tiempo real y de toma de decisiones fundamentada en datos.

Para finalizar, la meta de este trabajo es modificar el manejo tradicional del riego en las plantaciones bajo invernadero de la Unidad Educativa Jorge Martínez Acosta, brindando una solución informática que mejore la eficiencia del uso del agua.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se pudo crear un prototipo de sistema inteligente para la administración del agua de riego, que incorpora tecnologías emergentes como visión artificial, sensores IoT y una aplicación en el teléfono móvil.
- La aplicación de la metodología PXP posibilitó una elaboración gradual y adaptable del prototipo, lo que hizo más sencillo realizar ciclos iterativos de prueba y mejora en un contexto educativo regulado
- Las pruebas del prototipo evidenciaron la importancia de que los usuarios (incluyendo a alumnos técnicos y docentes) participen en el proceso, así como la colaboración entre instituciones para proyectos tecnológicos en el sector.
- La participación de los usuarios (docentes, estudiantes técnicos) durante las pruebas del prototipo evidenció la importancia de la capacitación y la colaboración interinstitucional para proyectos tecnológicos en el sector agrícola.
- El prototipo constituye un punto de partida concreto para futuras implementaciones y estudios comparativos con sistemas tradicionales abriendo nuevas líneas de investigación y desarrollo en el área de agricultura de precisión.

5.2. RECOMENDACIONES

- Sugerir la ejecución de proyectos piloto en entornos agrícolas reales, empleando este prototipo como fundamento. Esto posibilitará el estudio de su adaptación, limitaciones y contribuciones bajo condiciones de uso más amplias.
- Promover programas de capacitación continua acerca del manejo y mantenimiento del sistema, dirigidos a alumnos y personal técnico por igual;

haciendo hincapié en la interpretación de datos y en la posibilidad de tomar decisiones agronómicas informatizadas.

- Proponer la mejora de la solución prototípica al extender funcionalidades, añadiendo módulos de análisis y nuevos sensores. instrumentos de ayuda para optimizar el riego.
- Establecer mecanismos de colaboración con otras instituciones educativas/agropecuarias para compartir experiencias, comparar resultados y enriquecer el alcance de la tecnología desarrollada.
- Recomendar la definición de métricas específicas de eficiencia, sostenibilidad y satisfacción del usuario como parte de evaluaciones futuras que permitan medir el impacto potencial del sistema en la gestión del recurso hídrico.

VI. REFERENCIAS BIBLIOGRÁFICAS

- Alvear, V., Rosero, P., Peluffo, D., & Pijal, J. (2023, enero). Revisión de literatura sobre Internet de las Cosas y visión artificial: funcionamiento y aplicaciones. Scielo. Disponible en http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-65422017000100244 (Consultado el 8 de junio de 2025).
- Amós, D. (2024, diciembre). Creación de interfaces gráficas en Python con Tkinter. Pitón real. Recuperado el 8 de junio de 2025, de <https://realpython.com/python-gui-tkinter/>
- Anwar, M. (2025, abril). Definición y beneficios de sistemas multiagente: tipos y aplicaciones. Astera. Accedido el 8 de junio de 2025, en <https://www.astera.com/es/type/blog/multi-agent-system/>
- Borella, B. (2022). Introducción a la visión artificial: procesos y aplicaciones. Docta. Consultado el 8 de junio de 2025, desde <https://docta.ucm.es/entities/publication/072ca3fb-540f-4dc9-8a16-0241cb5cd986>
- Cadena, H. (2020, marzo). Diseño de un sistema automatizado para riego por aprendizaje automático aplicado a agricultura de precisión en la Granja La Pradera. Repositorio UTN, Universidad Técnica del Norte. Accedido el 4 de junio de 2025, en <https://repositorio.utn.edu.ec/handle/123456789/10296>
- Caiza, J., Guaman, D. & López, G. (2024, junio). Herramientas colaborativas para el desarrollo de software en ingeniería. Okhosting. Disponible el 8 de junio de 2025 en <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>
- Calapaqui, I. & Chiguano, A. (2022, julio). Desarrollo de un sistema inteligente para cultivos agrícolas. Repositorio Salesiano. Recuperado el 4 de junio de 2025, desde <https://dspace.ups.edu.ec/bitstream/123456789/22941/1/UPS%20-%20TTS842.pdf>
- Casco, C. (2024, enero). Guía completa sobre aplicaciones de inteligencia artificial versátil. Konfuzio. Consultado el 8 de junio de 2025, en <https://konfuzio.com/es/ai-aplicada/>
- Chirino, B., Díaz, M., Aguilera, Y., & Chiroles, M. (2022, marzo). Soluciones tecnológicas para soberanía alimentaria y gobierno electrónico. Scielo. Accedido el 8 de junio de 2025, en http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0257-43142022000100024

- Cimas, G. (2022, julio). Características y ventajas del editor Visual Studio Code para desarrolladores. Seminarios web abiertos. Disponible el 8 de junio de 2025, en <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Condori, P. (2020). Conceptos de universo, muestra y población en investigación. Academia. Consultado el 19 de junio de 2025, desde <https://www.academica.org/cporfirio/18.pdf>
- Deyimar. (2023, junio). Guía para principiantes sobre Bootstrap, frameworks y utilización. Hostinger. Recuperado el 8 de junio de 2025, en <https://www.hostinger.com/es/tutoriales/que-es-bootstrap>
- Feria, H., Matilla, M., & Mantecón, S. (2020, junio). Técnicas y métodos para la entrevista y encuestas: ventajas y aplicaciones. Dialnet. Ampliamente revisado el 19 de junio de 2025, archivo: <file:///D:/Descarga/Dialnet-LaEntrevistaYLaEncuesta-7692391.pdf>
- Fernández, Y. (2023, junio). Introducción a Github: qué es y para qué sirve. EBAC. Consultado el 8 de junio de 2025, en <https://ebac.mx/blog/que-es-github>
- Ginzo. (2022, abril). Metodología XP para el desarrollo de software: funcionamiento y ventajas. Ginzo. Accedido el 8 de junio de 2025, desde <https://ginzo.tech/como-funciona-metodologia-xp-desarrollo-software/>
- Gomstyn, A., & Jonker, A. (2023, diciembre). Agricultura inteligente: definición y relevancia tecnológica. IBM. Consultado el 8 de junio de 2025, en <https://www.ibm.com/es-es/think/topics/smart-farming>
- Guiles y Ahmet. (2023, octubre). Predicción de necesidades hídricas para cultivos mediante datos meteorológicos y sistemas de riego automatizados. Repositorio Universidad Politécnica de Valencia. Recogido el 8 de junio de 2025, disponible en <https://riunet.upv.es/entities/publication/8166495d-4d9b-48e8-85eb-0ff4df83e663>
- Javaid, M. (2023). Aplicaciones de inteligencia artificial en la agricultura: posibles impactos y desarrollos. Ciencia Directa. Accedido el 8 de junio de 2025, desde <https://www.sciencedirect.com/science/article/pii/S277323712200020X>
- Jiménez, F. (2020, diciembre). Modelo basado en agentes inteligentes para la gestión del riego en cultivos. Repositorio UNAL. Recuperado el 4 de junio de 2025, en <https://repositorio.unal.edu.co/handle/unal/79927>
- Jiménez, G. (2024). Sistema inteligente de riego de corto alcance mediante visión artificial para jardines. Repositorio UTA. Accedido el 4 de junio de 2025, en <https://repositorio.uta.edu.ec/home>
- Juárez, J. (2024, octubre). Relevancia de la calidad del agua utilizada en sistemas de riego agrícola. Azud. Consultado el 19 de junio de 2025, en <https://azud.com/blog/importancia-de-calidad-agua-riego-agricultura/>

- Kortas, T. (2025, abril). Soluciones de tecnologías de la información: definición y principales ventajas. OTROS. Accedido el 19 de junio de 2025, en <https://otrs.com/es/blog/itsm/soluciones-informaticas/#:~:text=Una%20soluci%C3%B3n%20inform%C3%A1tica>
- Larque, J. (2023, mayo). Clasificación y selección adecuada de bases de datos: tipos y consideraciones. Hiberus. Recuperado el 9 de junio de 2025, en <https://www.hiberus.com/crecemos-contigo/tipos-de-bases-de-datos-cuales-hay-y-por-que-es-importante-elegirlos-bien/>
- Iloabuchi, J. (2021, enero). Sensores de humedad: tipos y aplicaciones principales. IntechAbierto. Consultado el 8 de junio de 2025, en <https://www.intechopen.com/chapters/76696>
- Lomelí, L. (2023, mayo). Fundamentos de metodología Scrum: roles, artefactos y procesos. Innevo. Accedido el 8 de junio de 2025, en <https://innevo.com/blog/metodologia-scrum>
- López, L. y Paul, J. (2025). El método inductivo como enlace entre teoría educativa y práctica. Scielo. Consultado el 19 de junio de 2025, en http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-86262025000100051#:~:text=Utilizando%20el%20m%C3%A9todo%20inductivo
- Lozano, R. (2021, junio). Programación del microcontrolador ESP32 con entorno Arduino IDE. Taloselectrónica. Consultado el 8 de junio de 2025, en <https://www.taloselectronics.com/blogs/tutoriales/programar-esp32-con-ide-arduino>
- Lumenalta. (2025, enero). Aplicaciones de visión por computadora para impulsar la eficiencia agrícola. Lumenalta. Recuperado el 8 de junio de 2025, desde <https://lumenalta.com/insights/8-applications-of-computer-vision-in-agriculture>
- Martínez, J. (2021, junio). Introducción al lenguaje de programación Dart. Blog de Hiberus. Consultado el 8 de junio de 2025, en <https://www.hiberus.com/crecemos-contigo/que-es-el-lenguaje-de-programacion-dart/>
- Martins, J. (2025, enero). Funcionalidad y beneficios de la metodología Kanban. Asana. Recuperado el 8 de junio de 2025, en <https://asana.com/es/resources/what-is-kanban>
- Méndez, D. (2020, octubre). Definición y características de tecnologías habilitadas digitales. Ceupé. Consultado el 8 de junio de 2025, desde <https://www.ceupe.com/blog/tecnologias-habilitadoras-digitales-que-son-y-cuales-son.html>
- Ortíz, M. (2025). Redes neuronales artificiales: conceptos fundamentales y aplicaciones. Dialnet. Accedido el 8 de junio de 2025, desde <https://dialnet.unirioja.es/servlet/articulo?codigo=10085548>

- Peña, J. (2024, julio). Introducción al microcontrolador ESP32: usos básicos y programación. MCI Educación. Consultado el 8 de junio de 2025, en <https://cursos.mcielectronics.cl/2024/07/05/introduccion-a-esp32/>
- Pérez, R. (2021, septiembre). Fundamentos de gestión del agua en la producción ornamental. Iagua. Accedido el 9 de junio de 2025, en <https://www.iagua.es/blogs/ricardo-perez/principios-gestion-agua-riego-produccion-ornamental>
- Quilismal, J., & Jiménez, H. (2023, septiembre). Control automático de riego y fumigación. Repositorio UPEC. Consultado en 2025, en <https://repositorio.upec.edu.ec/home>
- Raeburn, A. (2025, febrero). Metodología de programación extrema (PXP): aplicaciones y beneficios. Asana. Accedido el 8 de junio de 2025, en <https://asana.com/es/resources/extreme-programming-pxp>
- Santander, U. (2020, diciembre). Introducción a metodologías de desarrollo de software. Academia Abierta de Santander. Consultado el 8 de junio de 2025, en <https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>
- Santander, U. (2021, marzo). Principales lenguajes de programación: usos y características. Academia Abierta de Santander. Recuperado el 8 de junio de 2025, desde <https://www.santanderopenacademy.com/es/blog/lenguaje-programacion.html>
- Sergieieva, K. (2025). Innovaciones tecnológicas en agricultura. Análisis de datos de EOS. Consultado el 8 de junio de 2025, en <https://eos.com/es/blog/tecnologias-en-la-agricultura/>
- Shapovalov, E. (2023, abril). Desarrollo full-stack con Flutter: ventajas y desafíos. De manera inconsciente. Recuperado el 8 de junio de 2024, en <https://innowise.com/es/blog/flutter-mobile-development/>
- Suárez, E. (2024, febrero). Métodos inductivo y deductivo: fundamentos y comparación. Experto Universitario. Consultado el 19 de junio de 2025, en <https://expertouniversitario.es/blog/metodo-inductivo-y-deductivo/>
- Zendesk. (2023, febrero). Introducción a la metodología ágil y principales metodologías utilizadas. Zendesk. Accedido el 8 de junio de 2025, en <https://www.zendesk.com.mx/blog/metodologia-agil-que-es/>.
- Rivulis. (2025). *Válvula plástica de control V2000*. Recuperado de <https://es.rivulis.com/productos/valvulas-de-riego/v2000>

VII. ANEXOS

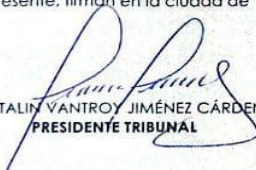
Anexo 1. Acta de la sustentación de Predefensa del TIC


UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI		FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES	
CARRERA DE COMPUTACIÓN		ACTA	
DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR			
ESTUDIANTE:	MALLAMAS MITES DAVID ANDERSON	CÉDULA DE IDENTIDAD:	0401844170
PERIODO ACADÉMICO:	2023B	DOCENTE TUTOR:	MSC. MARCO ANTONIO YANDÚN VELASTEGUI
PRESIDENTE TRIBUNAL	MSC. STALIN VANTROY JIMÉNEZ CÁRDENAS	DOCENTE:	MSC. CARLITOS ALBERTO GUANO CÁRDENAS
TEMA DEL TIC: "Solución Informática para la gestión del agua en cultivos bajo invernadero"			
No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	8.67	Reestructurar los objetivos
2	FUNDAMENTACIÓN TEÓRICA	9.33	Ampliar la información de los modelos de inteligencia artificial utilizados. Ampliar la información sobre la metodología de desarrollo (FXP) utilizada
3	METODOLOGÍA	9.00	Revisar y reformular la idea a defender.
4	RESULTADOS	8.67	Análisis de campo, verificar origen de la información para activar el riego, (datos técnicos del tipo de cultivo(tiempo de riego, etapas de crecimiento, tamaño, etc.)
5	DISCUSIÓN	9.33	
6	CONCLUSIONES Y RECOMENDACIONES	9.33	
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	9.33	
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	9.33	Revisar ortografía, estructura gramatical, aplicación de normas APA, tomar como referencia la guía metodológica de la UPEC


Obteniendo una nota de: **9,17** Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el **miércoles, 12 de noviembre de 2025**


 MSC. STALIN VANTROY JIMÉNEZ CÁRDENAS
 PRESIDENTE TRIBUNAL


 MSC. MARCO ANTONIO YANDÚN VELASTEGUI
 DOCENTE TUTOR


 MSC. CARLITOS ALBERTO GUANO CÁRDENAS
 DOCENTE

Anexo 2. Certificado del abstract por parte del centro de idiomas



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN
AND NATIVE LANGUAGES CENTER

ABSTRACT- EVALUATION SHEET				
NAME: Mallamas Mites David Anderson				
DATE: Lunes, 1 de diciembre de 2025				
Topic: "Solución Informática para la gestión del agua en cultivos bajo invernadero"				
MARKS AWARDED		QUANTITATIVE AND QUALITATIVE		
VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
De	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
ARGUMENT	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
CREATIVITY	Outstanding flow of ideas and events	Good flow of ideas and events	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED	TOTAL 9		



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL
CARCHI- FOREIGN AND NATIVE LANGUAGES
CENTER**

**Informe sobre el Abstract de Artículo Científico
o Investigación.**

Autor: Mallamas Mites David Anderson

Fecha de recepción del abstract: Miércoles, 19 de noviembre de 2025

Fecha de entrega del informe: Lunes, 1 de diciembre de 2025

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según la rúbrica de evaluación de la traducción en Inglés, ésta alcanza un valor de 9; por lo cual se valida dicho trabajo.

Atentamente



Firmado electrónicamente por:
MARTHA ARACELLY
VIVEROS ALMEIDA

Validar electrónicamente con firmaDC

MA. Martha Viveros
Responsable del
CIDEN

Anexo 3. Carta de acceso a invernadero institución

San Gabriel, 08 de septiembre del 2023

Msc.

VÍCTOR POZO

RECTOR UNIDAD EDUCATIVA "JORGE MARTÍNEZ ACOSTA"

Presente.

De mis consideraciones:

Reciba un cordial y atento saludo, a la vez deseándole éxitos en sus actividades laborales.

Yo **MALLAMAS MITES DAVID ANDERSON**, portador de cédula de ciudadanía número **0401844170**, estudiante de la UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI, Facultad de Industrias Agropecuarias y Ciencias Ambientales, la presente tiene como finalidad autorice a quien corresponda solicitarle su permiso para realizar un estudio y proyecto de investigación en su institución.

Por la atención que se digna dar al presente, reciba mis agradecimientos.

Atentamente,


MALLAMAS MITES DAVID ANDERSON

C.I.0401844170

Tel:0999102920



Anexo 4. Encuesta



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES CARRERA DE COMPUTACIÓN

Tema: "Solución Informática para la gestión del agua en cultivos bajo invernadero"

Encuesta a Estudiantes de Bachillerato sobre el Uso de Tecnología en la Gestión del Riego en Cultivos bajo Invernadero

Objetivo: Conocer la opinión de los estudiantes, en su rol de técnicos encargados del invernadero, sobre el uso de una solución informática para la gestión eficiente del agua en cultivos bajo invernadero.

ACTIVIDAD: Seleccione la opción que corresponda.

Sección 1: Información General del Estudiante

1. ¿En qué año de bachillerato se encuentra?
 - Primero de bachillerato
 - Segundo de bachillerato
 - Tercero de bachillerato

2. ¿Está familiarizado con el uso de tecnología en la agricultura, como sensores de humedad, temperatura o cámaras?
 - Nada
 - Poco
 - Mucho

Sección 2: Conocimiento sobre el Sistema de Gestión del Riego

3. ¿Cree que es importante utilizar tecnología para gestionar el riego en los cultivos bajo invernadero?
 - Nada importante
 - Poco importante
 - Muy importante

4. Los sensores de humedad y temperatura permiten medir y obtener datos, reales al momento actualizados dentro del ambiente en el invernadero. ¿Qué tan útiles cree que serían estos sensores para mejorar el control del riego?

- Nada útil
 - Poco útil
 - Muy útil
5. El uso de una cámara y agentes inteligentes permite monitorear el crecimiento de las plantas en tiempo real, detectando su fase de cultivo y ajustando el riego según sus necesidades hídricas. ¿Cree que esta tecnología puede mejorar la eficiencia del riego?
- No
 - Tal vez
 - Sí, algo
 - Sí, mucho
-

Sección 3: Uso de la Aplicación Móvil para Control del Riego

6. Si tuviera que usar una aplicación móvil para controlar el riego en los cultivos, ¿qué tan fácil cree que sería usarla?
- Muy difícil
 - Algo difícil
 - Algo fácil
 - Muy fácil
7. ¿Le gustaría recibir notificaciones en su WhatsApp sobre el estado del riego (por ejemplo, si está activado o apagado)?
- No
 - Sí
-

Sección 4: Opinión sobre la Eficiencia y Beneficios de la Tecnología

8. ¿Qué tan útil cree que sería la automatización del riego para ahorrar agua en los invernaderos?
- Nada útil
 - Poco útil
 - Muy útil
9. ¿Considera que la inteligencia artificial para predecir cuándo y cuánto regar puede ser beneficiosa para los cultivos?
- No

- Tal vez
 - Sí, algo
 - Sí, mucho
-

Sección 5: Opinión General sobre el Riego

10. ¿Qué sistema de riego utiliza o maneja actualmente en el invernadero?

- Riego por goteo
 - Riego por aspersión
 - Riego por microaspersión
 - Otro (especifique): _____
-

Finalización

Agradecimiento: Gracias por su participación. Sus respuestas serán fundamentales para el desarrollo de brindar la mejor solución informática que estará destinada a optimizar la gestión del agua en cultivos bajo invernadero. La información recopilada permitirá mejorar el sistema de riego mediante el uso de tecnologías inteligentes.

Anexo 5. Entrevista



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES CARRERA DE COMPUTACIÓN

Tema: "Solución Informática para la gestión del agua en cultivos bajo invernadero"

Entrevista a Encargados sobre el Uso de Tecnología en la Gestión del Riego en Invernaderos

Objetivo: Recoger información sobre la experiencia y perspectiva de los encargados en el uso de tecnologías en la gestión del riego en cultivos bajo invernadero.

Sección 1: Información General del Entrevistado

1. ¿Cuál es su cargo en la institución?

- Agrónomo
- Encargado de invernadero
- Docente

2. ¿Cuánto tiempo lleva trabajando en el área de la agricultura o gestión de cultivos?

- Menos de 1 año
- Entre 1 y 3 años
- Más de 3 años

3. ¿Está familiarizado o relacionado en uso de tecnología para la gestión de riego, como sensores de humedad, temperatura y cámaras?

- Nada
- Poco
- Mucho

Sección 2: Riego Actual

4. ¿Cree usted que la planta necesita diferentes cantidades de agua en producción (con frutos) es igual a la que necesita en desarrollo (plantas pequeñas)?

- Sí, necesitan la misma cantidad
- No, las plantas con frutos necesitan más agua
- No, las plantas pequeñas necesitan más agua

5. ¿Qué problemas cree que causa el sobre riego en el cultivo?

- Desperdicio de agua únicamente
 - Enfermedades en las plantas (hongos, raíces podridas)
 - Ambos problemas anteriores
 - No causa problemas
-

Sección 3: Pregunta de Validación

6. ¿Le gustaría usar una aplicación móvil para controlar el riego automáticamente?

- No
 - Sí
-

Finalización

Agradecimiento: Gracias por su tiempo y por sus valiosas respuestas. Su información y opinión es de importancia para el desarrollo de brindar la mejor solución informática que mejoren la gestión de agua de riego y la sostenibilidad agrícola.

Anexo 6. Manual de Usuario

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

MANUAL DE USUARIO AGROPREDICT APP MOVIL

Software móvil para uso y control del sistema gestión de agua para cultivos bajo invernadero.

Octubre 2025

Introducción

AgroPredict es una aplicación móvil que te ayuda a gestionar tu invernadero de manera más eficiente usando tecnología moderna e inteligencia artificial. Con esta herramienta puedes monitorear en tiempo real lo que sucede en tu cultivo, predecir cuándo estarán listos para cosechar y automatizar el riego, lo que hace mucho más fácil tomar las decisiones correctas en tu trabajo agrícola. La app reúne varios módulos especializados que te permiten, tanto si eres agricultor como técnico agrícola, supervisar y manejar todo lo importante de tu invernadero directamente desde tu celular: puedes analizar visualmente tus plantas con inteligencia artificial, controlar automáticamente el sistema de riego mediante dispositivos conectados, y tener todas las funciones agrícolas centralizadas en una sola aplicación que es fácil de usar y entender.

1- LOGIN

AgroPredict permite acceder mediante correo y contraseña, cuenta de Google o huella digital. Si no se tiene cuenta, se puede crear desde la opción de registro. Para restablecer contraseña, se ingresa el correo registrado y se sigue el enlace enviado.





2- REGISTRO

Se ingresan datos de correo, contraseña, nombre completo y ubicación del invernadero. Tras completar el registro, se debe verificar la cuenta mediante el correo enviado. Solo con la cuenta verificada se accede a todas las funciones.

3- MENÚ PRINCIPAL

Desde la pantalla de inicio se accede a todos los módulos mediante tarjetas interactivas: AgroPredictor, Asistente Virtual, Control de Riego, Smart Agriculture, Manual de Usuario, Créditos y Notificaciones. La navegación es intuitiva y se puede regresar al menú principal en cualquier momento.



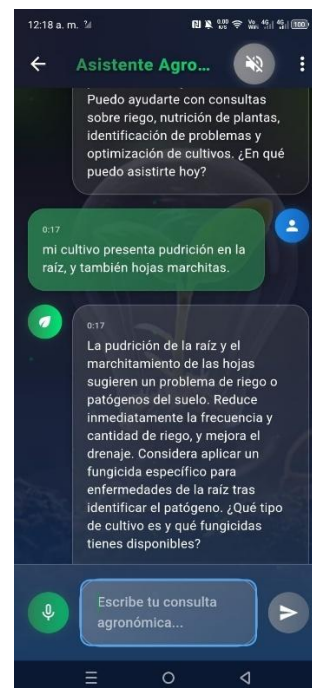


4- AGROPREDICTOR

Permite analizar cultivos y obtener predicciones sobre el tiempo de cosecha mediante inteligencia artificial. Se selecciona el cultivo, se captura o carga una imagen y se presiona analizar. La aplicación muestra fase de crecimiento, tiempo estimado para cosecha y recomendaciones automáticas.

5- ASISTENTE VIRTUAL

Responde consultas técnicas sobre agricultura de invernadero mediante texto, voz o imágenes de plantas. Proporciona diagnósticos de enfermedades, plagas y deficiencias nutricionales, además de recomendaciones. Todas las consultas se guardan en el historial para futuras referencias.





6- CONTROL DE RIEGO

Permite la gestión manual y automática del riego mediante un dispositivo IoT controlado por ESP32. Los tiempos de riego se ajustan según la fase del cultivo, con control de emergencia, temporizador visual y modo de respaldo. Las fases incluyen Desarrollo, Floración y Producción, con consumos aproximados de agua según cada etapa.

7- SMART AGRICULTURE

Este módulo muestra en tiempo real las condiciones del invernadero mediante sensores de temperatura, humedad y estado de ventilación y riego. Los indicadores visuales permiten identificar rápidamente si las condiciones son óptimas, altas o bajas, y el sistema ofrece sugerencias para mantener un ambiente adecuado. Los valores se actualizan automáticamente cada cinco minutos.





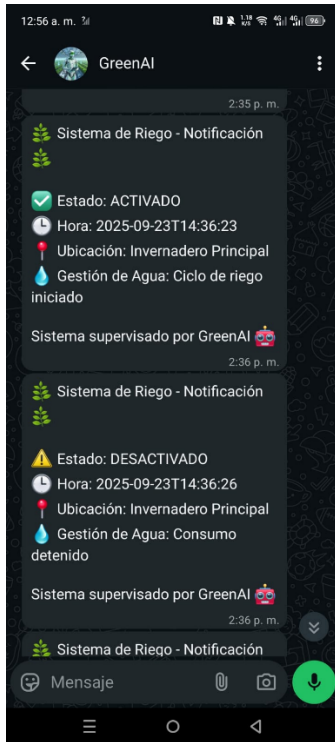
8- MANUAL DE USUARIO

Proporciona una guía completa e interactiva sobre el uso de AgroPredict, con instrucciones paso a paso, ejemplos visuales y soluciones a errores comunes. Incluye contacto directo con el desarrollador y soporte técnico

9- CRÉDITOS

Presenta información general del proyecto, responsables del desarrollo, tutoría e institución. Incluye tecnologías empleadas como Flutter, Firebase, TensorFlow Lite, ESP32, Python y Gemini IA, y reconoce la participación de los desarrolladores y colaboradores.





10-NOTIFICACIONES

Alertas Integrado con la API de WhatsApp Business, este módulo notifica automáticamente al usuario sobre el estado del sistema IoT y condiciones del invernadero. Envía alertas sobre activación o desactivación del riego, inicio de operaciones del microcontrolador.

Anexo 7. Manual Técnico

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

MANUAL TÉCNICO

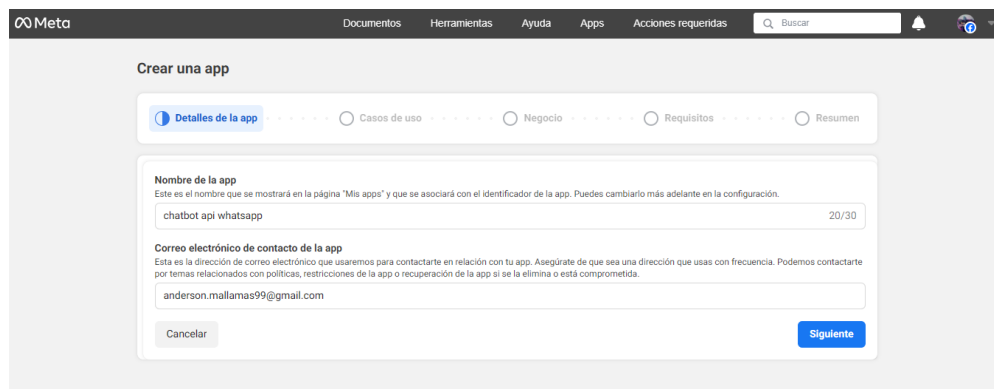
Software para uso y control del sistema gestión de agua para cultivos bajo invernadero.

Octubre bre 2025

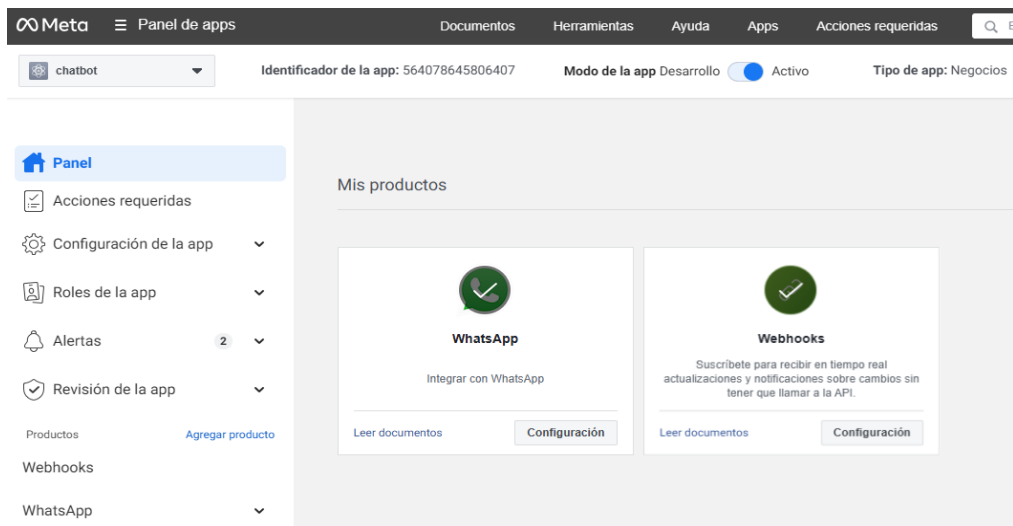
Introducción

AgroPredict es un sistema técnico que integra hardware ESP32 con sensores DHT11 y FC-28 para monitorear temperatura, humedad ambiental y humedad del suelo en invernaderos, transmitiendo datos en tiempo real a Firebase Realtime Database mediante conexión WiFi con sincronización NTP. El microcontrolador controla el riego de forma bidireccional, permitiendo activación automática según fases del cultivo o manual desde la aplicación móvil, y envía notificaciones automáticas vía API de WhatsApp Business configurada en Meta Developers sobre eventos críticos como cambios en el riego o condiciones ambientales. La aplicación móvil desarrollada en Flutter se conecta directamente a Firebase para leer y escribir datos, utiliza modelos de TensorFlow Lite para analizar imágenes de cultivos capturadas con la cámara del dispositivo y predecir fases de crecimiento, y presenta una interfaz donde el usuario puede monitorear condiciones del invernadero, controlar el riego remotamente y recibir recomendaciones basadas en inteligencia artificial, todo sincronizado en tiempo real con el hardware instalado en campo.

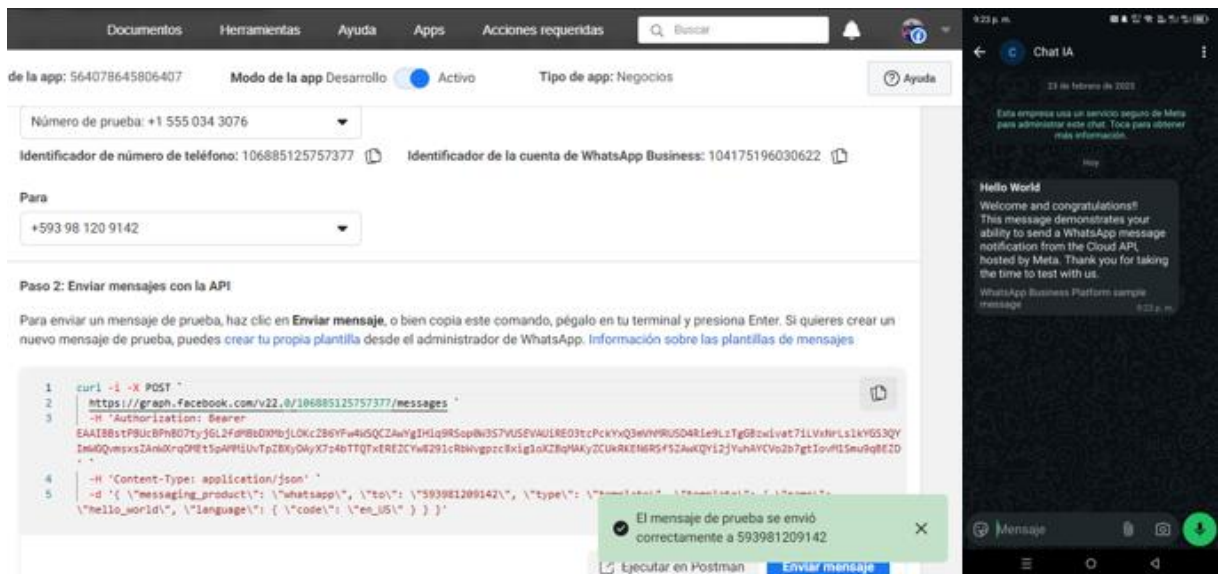
1- Obtener API WhatsApp Business



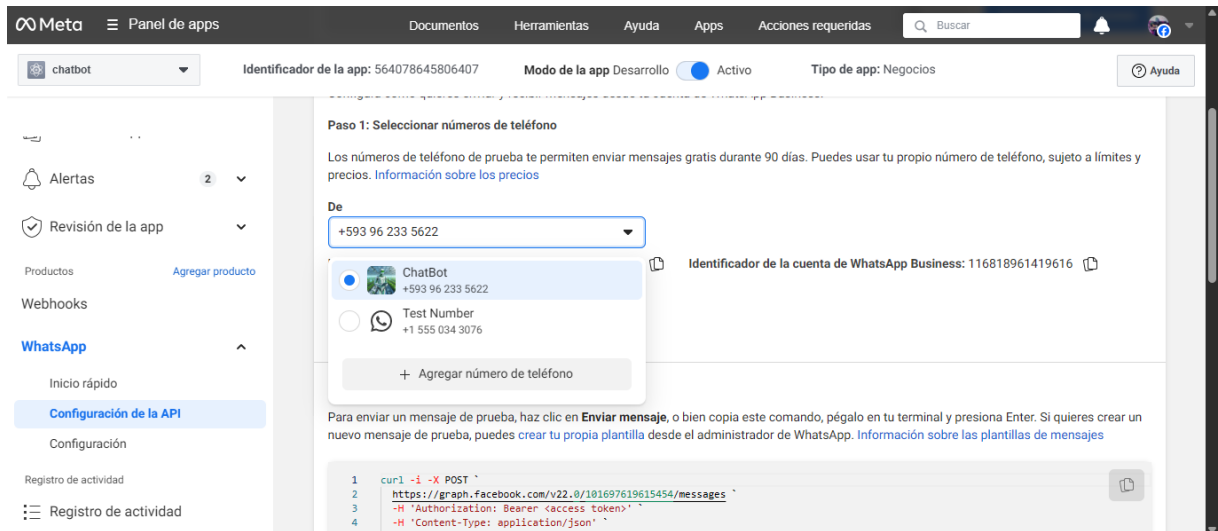
2- Creación de una nueva app en Meta Developers



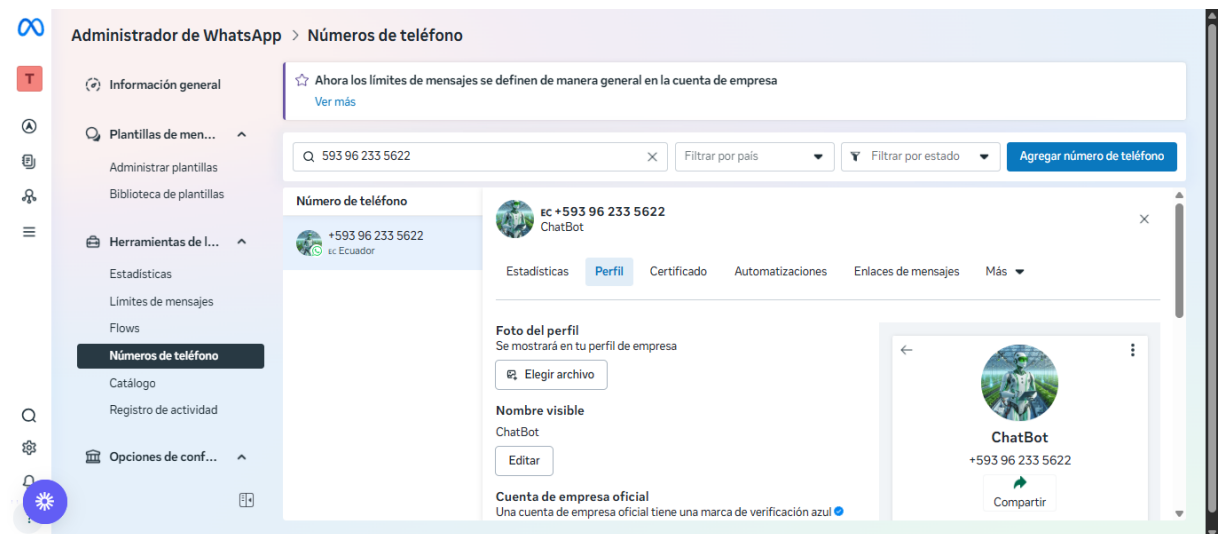
3- Integración de WhatsApp en Meta Developers para uso de API



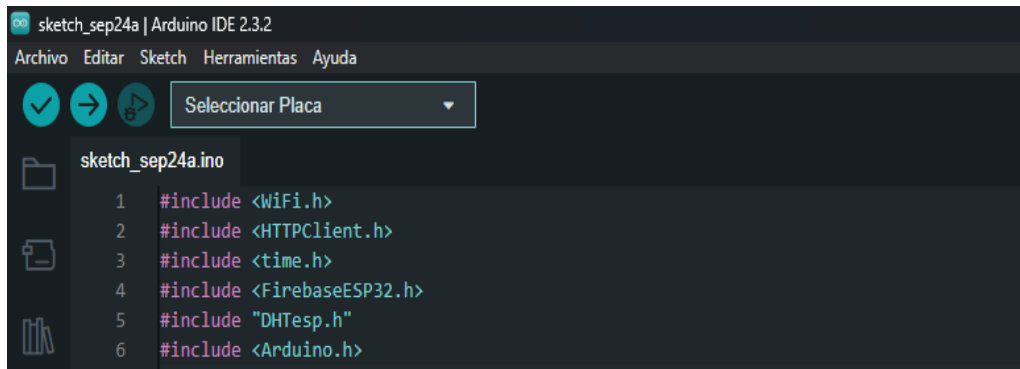
4- Test de prueba API WhatsApp



5- Registro API WhatsApp (Cnt: +593 96 233 5622)

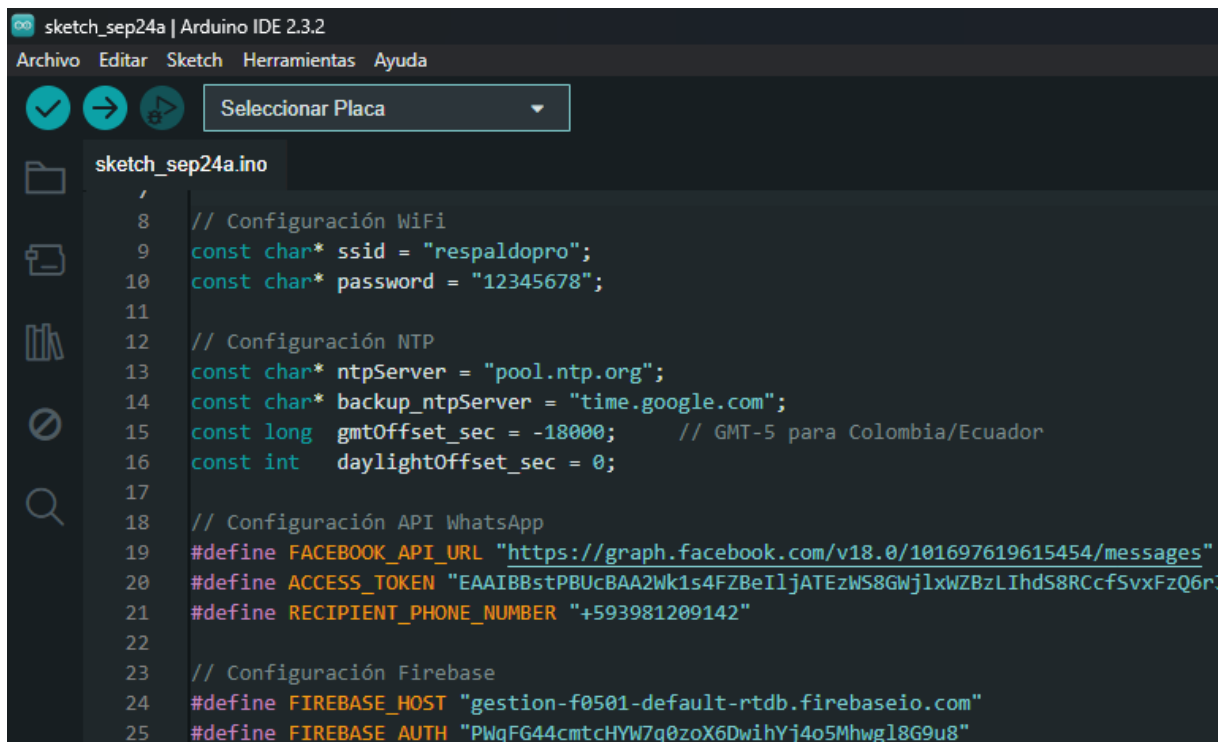


6- Configuración perfil API WhatsApp



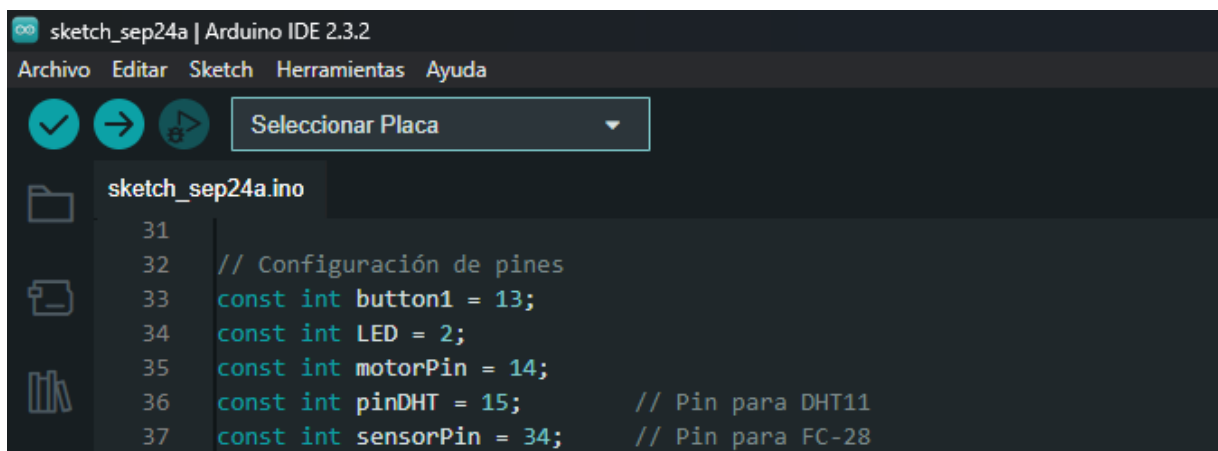
```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
1 #include <WiFi.h>
2 #include <HTTPClient.h>
3 #include <time.h>
4 #include <FirebaseESP32.h>
5 #include "DHTesp.h"
6 #include <Arduino.h>
```

7- Librerías ESP32



```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
8 // Configuración WiFi
9 const char* ssid = "respaldopro";
10 const char* password = "12345678";
11
12 // Configuración NTP
13 const char* ntpServer = "pool.ntp.org";
14 const char* backup_ntpServer = "time.google.com";
15 const long  gmtOffset_sec = -18000; // GMT-5 para Colombia/Ecuador
16 const int   daylightOffset_sec = 0;
17
18 // Configuración API WhatsApp
19 #define FACEBOOK_API_URL "https://graph.facebook.com/v18.0/101697619615454/messages"
20 #define ACCESS_TOKEN "EAAIBBstPBUCBAA2Wk1s4FZBeIljATEzWS8GwjlxWZBzLIhdS8RCcfSvxFzQ6r"
21 #define RECIPIENT_PHONE_NUMBER "+593981209142"
22
23 // Configuración Firebase
24 #define FIREBASE_HOST "gestion-f0501-default-rtdb.firebaseio.com"
25 #define FIREBASE_AUTH "PWqFG44cmtchYw7q0zoX6DwihYj4o5Mhwg18G9u8"
```

8- CREDENCIALES Y CONEXIONES



```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
31
32 // Configuración de pines
33 const int button1 = 13;
34 const int LED = 2;
35 const int motorPin = 14;
36 const int pinDHT = 15; // Pin para DHT11
37 const int sensorPin = 34; // Pin para FC-28
```

9- CONFIGURACIÓN DE HARDWARE

```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
39 // Configuración FC-28
40 const int valorSeco = 4095;
41 const int valorMojado = 2480;
42 const int numLecturas = 5;
43 int lecturas[5];
44 int indiceLectura = 0;
45 int total = 0;
```

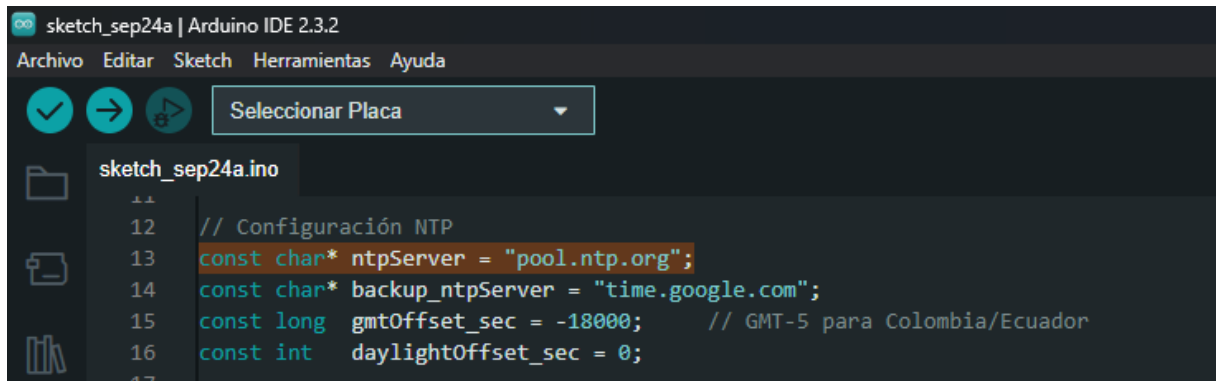
10- CALIBRACIÓN SENSOR FC-28 (Humedad Suelo):

```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
368 // Actualizar datos ambientales en Firebase
369 if (Firebase.ready() && (millis() - sendDataPrevMillis > timerDelay || sendDataPrevMillis == 0)) {
370   sendDataPrevMillis = millis();
371
372   TempAndHumidity dhtData = dht.getTempAndHumidity();
373   int valorSuelo = obtenerLecturaPromedio();
374   float humedadSuelo = calibrarFC28(valorSuelo);
375 }
```

11- LECTURA DHT11 y FC-28

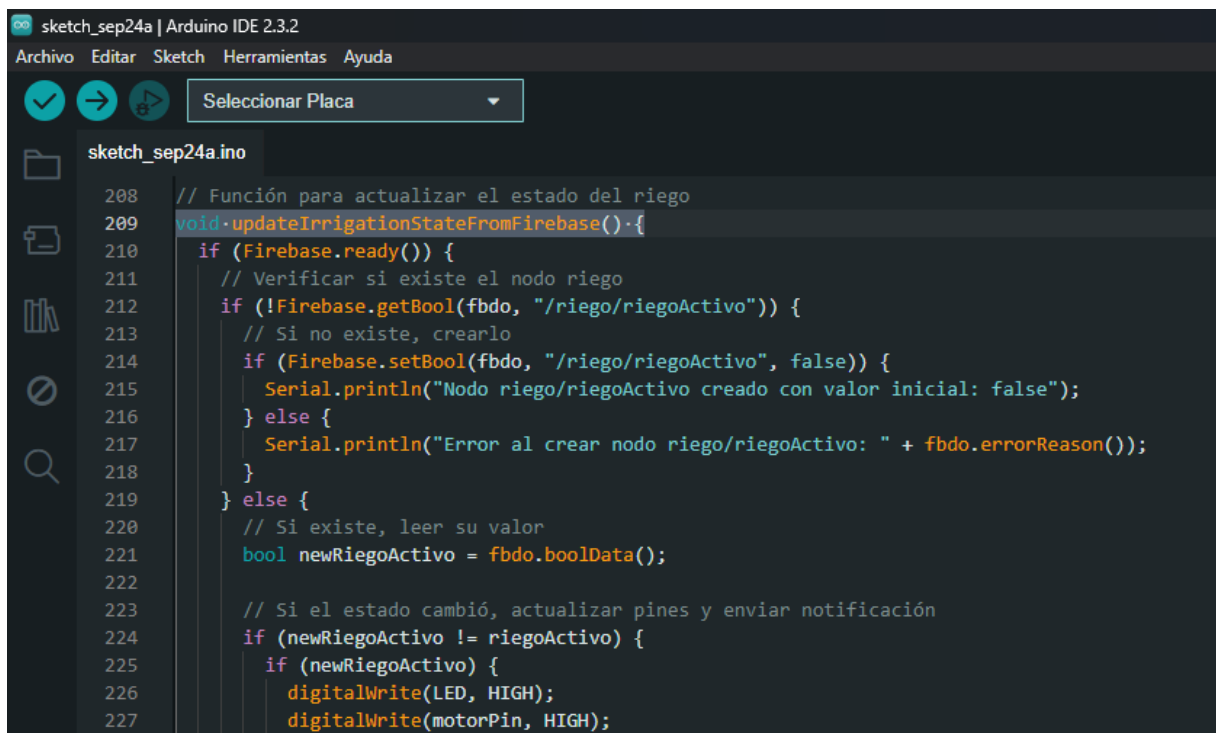
```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
183 // Función mejorada de reconexión WiFi
184 bool handleWiFiReconnection() {
185   if (WiFi.status() != WL_CONNECTED) {
186     Serial.println("Reconectando a WiFi...");
187     WiFi.disconnect();
188     WiFi.begin(ssid, password);
189
190     int attempts = 0;
191     while (WiFi.status() != WL_CONNECTED && attempts < 20) {
192       delay(500);
193       Serial.print(".");
194       attempts++;
195     }
196 }
```

12- MANEJO ROBUSTO DE WIFI:



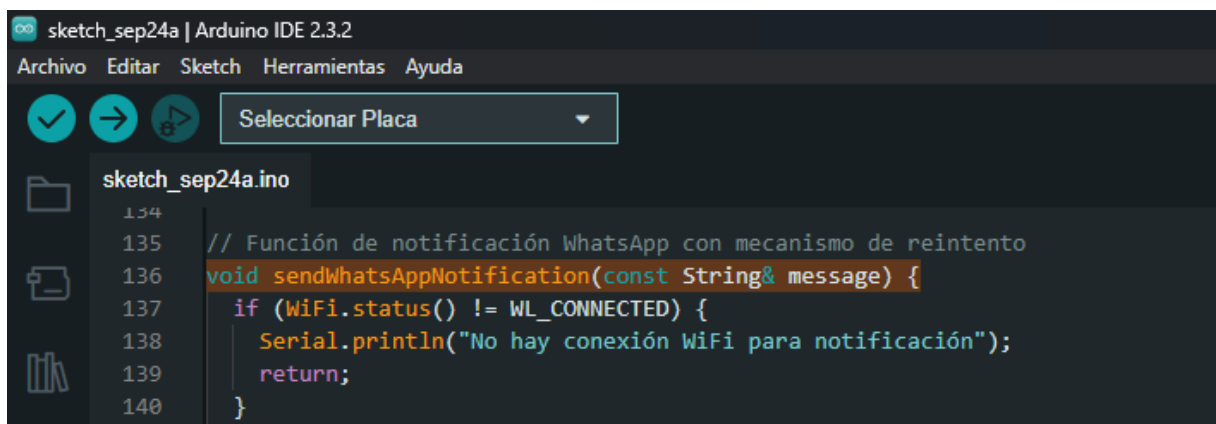
```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
12 // Configuración NTP
13 const char* ntpServer = "pool.ntp.org";
14 const char* backup_ntpServer = "time.google.com";
15 const long  gmtOffset_sec = -18000; // GMT-5 para Colombia/Ecuador
16 const int   daylightOffset_sec = 0;
```

13-SINCRONIZACIÓN DE TIEMPO:



```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
208 // Función para actualizar el estado del riego
209 void updateIrrigationStateFromFirebase() {
210     if (Firebase.ready()) {
211         // Verificar si existe el nodo riego
212         if (!Firebase.getBool(fbdo, "/riego/riegoActivo")) {
213             // Si no existe, crearlo
214             if (Firebase.setBool(fbdo, "/riego/riegoActivo", false)) {
215                 Serial.println("Nodo riego/riegoActivo creado con valor inicial: false");
216             } else {
217                 Serial.println("Error al crear nodo riego/riegoActivo: " + fbdo.errorReason());
218             }
219         } else {
220             // Si existe, leer su valor
221             bool newRiegoActivo = fbdo.boolData();
222
223             // Si el estado cambió, actualizar pines y enviar notificación
224             if (newRiegoActivo != riegoActivo) {
225                 if (newRiegoActivo) {
226                     digitalWrite(LED, HIGH);
227                     digitalWrite(motorPin, HIGH);
```

14- Sistema de Notificaciones



```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa
sketch_sep24a.ino
134
135 // Función de notificación WhatsApp con mecanismo de reintento
136 void sendWhatsAppNotification(const String& message) {
137     if (WiFi.status() != WL_CONNECTED) {
138         Serial.println("No hay conexión WiFi para notificación");
139         return;
140     }
```

15- MENSAJES WHATSAPP AUTOMÁTICOS:

```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa

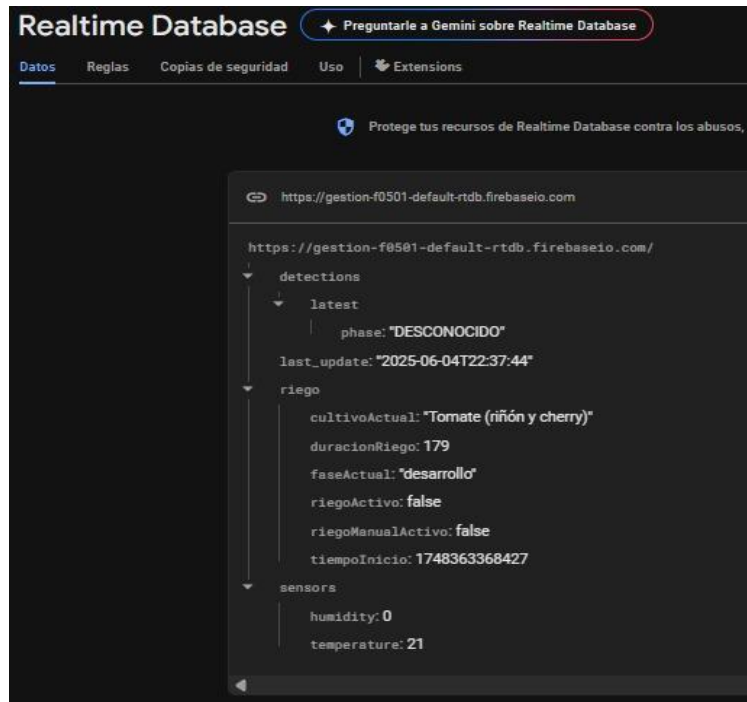
sketch_sep24a.ino
229     sendWhatsAppNotification(" 🌿 Sistema de Riego - Notificación 🌿 \n\n"
230                             " ✅ Estado: ACTIVADO\n"
231                             " 🕒 Hora: " + getLocalTime() + "\n"
232                             " 📍 Ubicación: Invernadero Principal\n"
233                             " 💧 Gestión de Agua: Ciclo de riego iniciado\n\n"
234                             "Sistema supervisado por GreenAI 🤖");
235     } else {
236         digitalWrite(LED, LOW);
237         digitalWrite(motorPin, LOW);
238
239         sendWhatsAppNotification(" 🌿 Sistema de Riego - Notificación 🌿 \n\n"
240                                 " ⚠️ Estado: DESACTIVADO\n"
241                                 " 🕒 Hora: " + getLocalTime() + "\n"
242                                 " 📍 Ubicación: Invernadero Principal\n"
243                                 " 💧 Gestión de Agua: Consumo detenido\n\n"
244                                 "Sistema supervisado por GreenAI 🤖");
```

16- TIPOS DE NOTIFICACIONES:

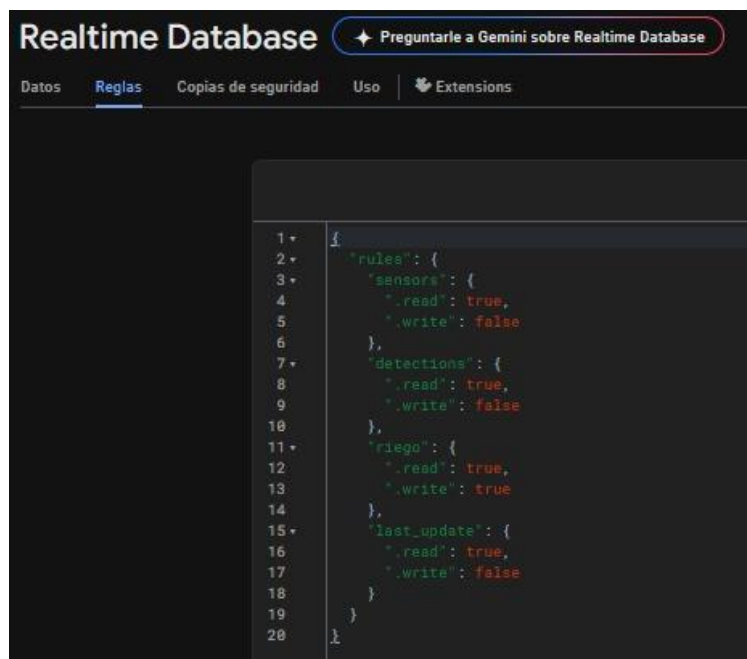
```
sketch_sep24a | Arduino IDE 2.3.2
Archivo Editar Sketch Herramientas Ayuda
Seleccionar Placa

sketch_sep24a.ino
378     String currentTime = getLocalTime();
379
380     // Actualizar sensores
381     if (Firebase.setFloat(fbdo, "/sensors/humidity", humedadSuelo)) {
382         Serial.printf("Humedad del suelo actualizada: %.1f%\n", humedadSuelo);
383     } else {
384         Serial.println("Error actualizando humedad: " + fbdo.errorReason());
385     }
386
387     if (Firebase.setFloat(fbdo, "/sensors/temperature", dhtData.temperature)) {
388         Serial.printf("Temperatura actualizada: %.1f°C\n", dhtData.temperature);
389     } else {
390         Serial.println("Error actualizando temperatura: " + fbdo.errorReason());
391     }
```

17- Estructura de Datos en Firebase



18- Base de datos RTDB



19- Reglas Base de datos RTDB

```

1 import tkinter as tk
2 from tkinter import Label, Frame, ttk, messagebox
3 import cv2
4 import numpy as np
5 from tensorflow.lite.python.interpreter import Interpreter
6 import tensorflow as tf
7 import threading
8 import logging
9 from typing import Tuple, Optional
10 import firebase_admin
11 from firebase_admin import credentials, db
12 import time
13 from PIL import Image, ImageTk
14 import os
15 import sys
16 from datetime import datetime
17 import sys
18 import os
19 os.environ['OPENCV_VIDEOIO_MSMF_ENABLE_HW_TRANSFORMS'] = '0'

```

20- Librerías, agente Sistema Integrado de Monitoreo y Control

```

89
90 def _initialize_firebase(self):
91     try:
92         if not firebase_admin.apps:
93             # CAMBIO AQUÍ: usar resource_path en lugar de ruta directa
94             cred_path = resource_path('serviceAccountKey.json')
95             cred = credentials.Certificate(cred_path)
96             firebase_admin.initialize_app(cred, {
97                 'databaseURL': 'https://gestion-f0501-default-rtdb.firebaseio.com/'
98             })
99
100         # Referencias a nodos de Firebase
101         self.ref_phase = db.reference('detections/latest/phase')
102         self.ref_temp = db.reference('sensors/temperature')
103         self.ref_hum = db.reference('sensors/humidity')
104         self.ref_riego = db.reference('riego/riegoActivo')
105         self.ref_last_update = db.reference('last_update')
106         self.ref_riego_session = db.reference('riego/session_actual')
107         self.ref_intelligent_agent = db.reference('system/intelligent_agent_active')
108         self.ref_manual_control = db.reference('riego/riegoManualActivo')

```

21- Conexión del sistema integrado GreenAI a la base de datos

```

136
137 def _update_intelligent_agent_gui(self):
138     """Actualiza la GUI del estado del agente inteligente"""
139     if self.intelligent_agent_active:
140         self.agent_status_label.config(text="🚫", fg='#00ff88')
141         self.agent_toggle_button.config(text="🚫 DESACTIVAR IA", bg='#ff4757')
142         self.agent_text.config(text="IA Activo", fg='#00ff88')
143     else:
144         self.agent_status_label.config(text="🟢", fg='#ff4757')
145         self.agent_toggle_button.config(text="🟢 ACTIVAR IA", bg='#00ff88')
146         self.agent_text.config(text="IA Inactivo", fg='#ff4757')
147

```

22- Cambios de estado de riego: activar y desactivar

```

147
148 def _load_models(self):
149     try:
150         # CAMBIO AQUÍ: usar resource_path para todos los archivos del modelo
151         model_path = resource_path('model_unquant.tflite')
152         labels_path = resource_path('labels.txt')
153         h5_model_path = resource_path('model.h5')
154
155         self.detector_interpreter = Interpreter(model_path=model_path)
156         self.detector_interpreter.allocate_tensors()
157         self.input_details = self.detector_interpreter.get_input_details()
158         self.output_details = self.detector_interpreter.get_output_details()
159
160         with open(labels_path, 'r') as f:
161             self.labels = [line.strip().upper() for line in f.readlines()]
162
163         self.irrigation_model = tf.keras.models.load_model(h5_model_path, compile=False)
164         self.irrigation_model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
165
166         logger.info("Modelos cargados correctamente")
167     except Exception as e:
168         logger.error(f"Error cargando modelos: {e}")
169         messagebox.showerror("Error", "Error cargando modelos de IA")
170         sys.exit(1)

```

23- Importación de modelos entrenados en TensorFlow

```
171
172 def _initialize_camera(self):
173     # Buscar cámaras USB disponibles (saltarse índice 0 - cámara integrada)
174     usb_cameras = []
175
176     for camera_index in range(1, 10): # Empezar desde 1 para saltarse la cámara integrada
177         cap = cv2.VideoCapture(camera_index)
178
179         if cap.isOpened():
180             ret, frame = cap.read()
181             if ret and frame is not None:
182                 usb_cameras.append(camera_index)
183                 logger.info(f"Cámara USB encontrada en índice {camera_index}")
184
185             cap.release()
186
187         if not usb_cameras:
188             logger.error("No se encontraron cámaras USB disponibles")
189             messagebox.showerror("Error", "No se detectaron cámaras USB en el sistema")
190             sys.exit(1)
191
192     # Usar la cámara USB con el índice más alto
193     camera_index = max(usb_cameras)
194
195     self.cap = cv2.VideoCapture(camera_index)
196
197     if not self.cap.isOpened():
198         logger.error(f"Error accediendo a la cámara USB {camera_index}")
199         messagebox.showerror("Error", f"No se puede acceder a la cámara USB {camera_index}")
200         sys.exit(1)
```

24- Inicialización de la cámara para la detección de estado

```
917
918 def _predict_irrigation(self, temp, hum, phase):
919     """Predice el tiempo de riego basado en sensores y fase de cultivo"""
920     try:
921         if not self.intelligent_agent_active:
922             return 0.0
923
924         phase_map = {'DESCONOCIDO': 0, 'DESARROLLO': 1, 'FLORACION': 2, 'PRODUCCION': 3}
925         phase_code = phase_map.get(phase, 0)
926
927         # Si la fase es DESCONOCIDA, detener riego inmediatamente
928         if phase_code == 0:
929             if self.irrigation_active:
930                 self._stop_irrigation()
931                 logger.info("Riego detenido: fase DESCONOCIDO detectada")
932             return 0.0
933
934         # No regar si humedad muy alta
935         if hum >= 80: # Aumentado el umbral para ser más permisivo
936             logger.debug(f"Humedad alta ({hum}%), no regar")
937             return 0.0
```

25- Estado de predicciones: fases y desconocido si no detecta fases

Anexo 8. Fotografías



Figura 69. prototipo maqueta invernadero



Figura 70. invernadero de la UE Jorge Martínez Acosta



Figura 71. técnico invernadero



Figura 72. Cultivos bajo etapa desarrollo



Figura 73. Cultivo etapa de floración



Figura 74. cultivo etapa de producción



Figura 75. técnico en invernadero mantenimiento