



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE SOFTWARE

Tema:

**DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL DE
CONTROL Y MONITOREO PARA LA PROTECCIÓN DE LOS CULTIVOS
DE PAPAS CONTRA LAS HELADAS UTILIZANDO ENERGÍAS
RENOVABLES EN LA ZONA ALTA DEL CANTÓN QUERO**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniera de Software

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Silvia Esthefania Villacrés Chico

TUTOR: Ing. Oscar Fernando Ibarra Torres, Mg.

Ambato - Ecuador

agosto – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL DE CONTROL Y MONITOREO PARA LA PROTECCIÓN DE LOS CULTIVOS DE PAPAS CONTRA LAS HELADAS UTILIZANDO ENERGÍAS RENOVABLES EN LA ZONA ALTA DEL CANTÓN QUERO desarrollado bajo la modalidad Proyecto de Investigación por la señorita Silvia Esthefania Villacrés Chico estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2024.

Ing. Oscar Fernando Ibarra Torres, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL DE CONTROL Y MONITOREO PARA LA PROTECCIÓN DE LOS CULTIVOS DE PAPAS CONTRA LAS HELADAS UTILIZANDO ENERGÍAS RENOVABLES EN LA ZONA ALTA DEL CANTÓN QUERO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2024.



Silvia Esthefania Villacrés Chico

C.C. 1804816104

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2024.



Silvia Esthefania Villacrés Chico

C.C. 1804816104

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Silvia Esthefania Villacrés Chico estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL DE CONTROL Y MONITOREO PARA LA PROTECCIÓN DE LOS CULTIVOS DE PAPAS CONTRA LAS HELADAS UTILIZANDO ENERGÍAS RENOVABLES EN LA ZONA ALTA DEL CANTÓN QUERO nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2024.

Ing. Elsa Pilar Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Santiago David Jara Moya, Mg.
PROFESOR CALIFICADOR

Ing. Paulo César Torres Abril, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Dedico este proyecto a mis padres, Leonidas y Geny quienes me apoyaron incondicionalmente en este largo camino, su cariño y comprensión me ayudaron a seguir adelante a pesar de los obstáculos que se presentaron.

A mi hermano Christian que siempre me dio fuerzas para continuar, nunca faltó un estoy orgulloso de ti y sus consejos me guiaron a ser mejor cada día.

A mi sobrino Josué que a pesar de no entender que hacía siempre me quería ayudar y me sacaba sonrisas con sus ocurrencias.

Silvia Esthefania Villacrés Chico

AGRADECIMIENTO

Quiero agradecer primero a Dios por brindarme salud y sabiduría en esta investigación.

Agradezco a mis padres por siempre guiarme en el camino del bien. A mi hermano por estar a mi lado en los momentos más difíciles.

A mi amiga Shirley por escucharme y apoyarme cuando más lo necesitaba.

A mi amiga Madelyn por brindarme sus consejos que me ayudan a ser mejor persona.

A mi prima Nayely gracias por estar siempre al pendiente a pesar de la distancia.

A mis amigos que conocí en la carrera Jadira, Jessica, Diana, Edison, Abraham, Diego gracias por brindarme su ayuda no solo académicamente sino emocionalmente.

A mi tutor académico, Ing. Fernando Ibarra le agradezco por su paciencia y orientación para el desarrollo de este trabajo.

Al Ing. Rubén Nogales por guiarme en la selección y delimitación de mi proyecto.

A todos aquellos que estuvieron presentes a lo largo de este proceso este logro no hubiera sido posible sin su apoyo, muchas gracias.

Silvia Esthefania Villacrés Chico

ÍNDICE GENERAL DE CONTENIDOS

PURTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xv
ÍNDICE DE ANEXOS	xxi
RESUMEN EJECUTIVO	xxii
ABSTRACT	xxiii
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	10

1.3.1 Placas de recolección de datos	11
1.3.2 Sensor de temperatura	11
1.3.3 Sensor de humedad	12
1.3.4 Sensor de velocidad del viento.....	12
1.3.5 Tipo de comunicación	12
1.3.6 Almacenamiento	13
1.3.7 Tipo de riego	13
1.3.8 Lenguajes de programación	13
1.3.9 Interfaz de Programación de Aplicaciones (API)	14
1.3.10 Plataformas de la nube	14
1.3.11 Metodologías de desarrollo	15
1.3.12 Cultivo de papa	15
1.3.13 Heladas	16
1.4 Objetivos	21
1.4.1 Objetivo general	21
1.4.2 Objetivos específicos	21
CAPÍTULO II. METODOLOGÍA.....	22
2.1 Materiales	22
2.2 Métodos.....	23

2.2.1 Modalidad de la investigación	23
2.2.2 Población y muestra	23
2.2.3 Recolección de información.....	25
2.2.4 Procesamiento y análisis de datos	40
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	42
3.1 Análisis y discusión de resultados.....	42
3.2 Desarrollo de la propuesta.....	65
3.2.1 Análisis de las condiciones climáticas	65
3.2.2 Fase I: Recopilación de requisitos.....	66
3.2.3 Fase II: Diseño del prototipo.....	77
3.2.4 Fase III: Desarrollo del prototipo.....	85
3.2.5 Fase IV: Pruebas del prototipo.....	113
3.2.6 Fase V: Implantación	118
3.3 Resultados	125
3.3.1 Modelo de aceptación de tecnología TAM	125
3.3.2 Cuestionario para el modelo TAM.....	126
3.3.3 Resultados del modelo TAM	127
3.4 Presupuesto	129
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	130
4.1 Conclusiones	130
4.2 Recomendaciones.....	131

REFERENCIAS BIBLIOGRÁFICAS	132
ANEXOS	137

ÍNDICE DE TABLAS

Tabla 1. Temperatura soportada por la papa[35].	16
Tabla 2. Factores de inciden en una helada[37].....	17
Tabla 3. Tipos de heladas [38].	18
Tabla 4. Métodos para mitigar las heladas[39].	19
Tabla 5. Efectos de las heladas[39].....	19
Tabla 6. Resultados de la encuesta-Pregunta 1	25
Tabla 7. Resultados de la encuesta-Pregunta 2	27
Tabla 8. Resultados de la encuesta-Pregunta 3	28
Tabla 9. Resultados de la encuesta-Pregunta 4	29
Tabla 10. Resultados de la encuesta-Pregunta 5	30
Tabla 11. Resultados de la encuesta-Pregunta 6	31
Tabla 12. Resultados de la encuesta-Pregunta 7	32
Tabla 13. Resultados de la encuesta-Pregunta 8	33
Tabla 14. Resultados de la encuesta-Pregunta 9	34
Tabla 15. Resultados de la encuesta-Pregunta 10	35
Tabla 16. Resultados de la encuesta-Pregunta 11	36
Tabla 17. Metodologías ágiles[41].....	45
Tabla 18. Herramientas de gestión de proyectos	47
Tabla 19. Herramientas de control de versiones	49
Tabla 20. Herramientas de prototipado.....	50

Tabla 21. Bases de datos	51
Tabla 22. Framework de desarrollo	52
Tabla 23. Framework JavaScript.....	53
Tabla 24. Lenguajes backend	54
Tabla 25. Framework backend	55
Tabla 26.Comparación de los microprocesadores	56
Tabla 27.Sensores de temperatura.....	57
Tabla 28. Comparación sensores de humedad del suelo.....	58
Tabla 29.Comparación sensores de velocidad del viento.	59
Tabla 30.Comparación del riego	60
Tabla 31. Comparación de tipos de aspersores.	61
Tabla 32 . Bomba de agua.....	63
Tabla 33. Tecnologías de comunicación	64
Tabla 34. Energías renovables	64
Tabla 35. Plantillas de Historias de Usuario[41]	67
Tabla 36. Historia de usuario- Conexión de Sensores	70
Tabla 37. Historia de usuario- Conexión inalámbrica	70
Tabla 38. Historia de usuario- Activar el riego automáticamente	71
Tabla 39. Historia de usuario- Mostrar datos de los sensores.....	71
Tabla 40. Historia de usuario- Control remoto del riego	72
Tabla 41. Historia de usuario- Listado de actividades del riego	72

Tabla 42. Historia de usuario- Notificar al usuario el estado del riego.....	73
Tabla 43. Historia de usuario- Registrar usuario	73
Tabla 44. Historia de usuario- Ingreso al Sistema	74
Tabla 45. Historia de usuario- Actualizar usuario	74
Tabla 46. Historia de usuario- Recuperar contraseña	75
Tabla 47. Historia de usuario- Implementar Energía Renovable.....	75
Tabla 48. Estimación de historias de usuario método tres puntos	76
Tabla 49. Plan de entrega.....	76
Tabla 50. Pruebas de aceptación-Iteración 1.....	113
Tabla 51. Pruebas de aceptación-Iteración II.....	114
Tabla 52, Pruebas de aceptación-Iteración III.....	115
Tabla 53. Resultado de las pruebas de usabilidad.....	117
Tabla 54.Pruebas del alcance de comunicación Lora	118
Tabla 55. Cuestionario de evolución del modelo TAM	126
Tabla 56. Resultados modelo TAM-Utilidad percibida.....	127
Tabla 57. Resultados modelo TAM-Utilidad percibida.....	128
Tabla 58. Presupuesto del prototipo.....	129

ÍNDICE DE FIGURAS

Figura 1. Helada en el cultivo de papa- Imagen generada con Chat GPT 20 de marzo de 2024	17
Figura 2. Consecuencias de las heladas en el cultivo de papas- Imagen generada con Chat GPT 20 de marzo de 2024	20
Figura 3. Resultados de la encuesta-Pregunta 1	26
Figura 4. Resultados de la encuesta-Pregunta 2	27
Figura 5. Resultados de la encuesta-Pregunta 3	28
Figura 6. Resultados de la encuesta-Pregunta 4	29
Figura 7. Resultados de la encuesta-Pregunta 5	30
Figura 8.Resultados de la encuesta-Pregunta 6.....	31
Figura 9. Resultados de la encuesta-Pregunta 7	32
Figura 10. Resultados de la encuesta-Pregunta 8.....	33
Figura 11. Resultados de la encuesta-Pregunta 9	34
Figura 12. Resultados de la encuesta-Pregunta 10.....	35
Figura 13. Resultados de la encuesta-Pregunta 11.....	36
Figura 14. Diagrama del proceso actual de los agricultores con el riego.	43
Figura 15. Diagrama del proceso con la aplicación para el riego automático	44
Figura 16. Diagrama del proceso con la intervención del usuario	45
Figura 17. Fases RAD	47
Figura 18. Diagrama de base de datos	77

Figura 19. Esquema del proyecto	78
Figura 20. Diagrama de la alimentación del sistema(solar)	78
Figura 21. Diagrama de sistema electrónico (Emisor).....	79
Figura 22. Diagrama del receptor	80
Figura 23. Arquitectura del sistema	81
Figura 24. Prototipo para a inicio de sesión de la aplicación móvil	81
Figura 25. Recuperar Contraseña.....	82
Figura 26. Prototipo del registro de la aplicación móvil.....	82
Figura 27. Prototipo de la visualización de datos de los sensores	83
Figura 28. Prototipo del historial de los datos de los sensores	83
Figura 29. Prototipo de las alertas emitidas por la aplicación móvil.	84
Figura 30. Prototipo del perfil de usuario de la aplicación móvil.....	84
Figura 31.Tareas en Jira	85
Figura 32. Repositorio GitHub.....	85
Figura 33. Comando para crear el proyecto Ionic de tipo React.....	86
Figura 34. Comando para añadir plataforma Android	86
Figura 35. Comando para notificaciones push.....	86
Figura 36. Comando para toast	87
Figura 37.Importación de librerías	87
Figura 38. Configuración de sensores	88
Figura 39. Configuración parámetros Lora	88

Figura 40. Variables de control	88
Figura 41. Configuración e Inicialización en el setup()	89
Figura 42. Bucle Principal (loop())	90
Figura 43. Definición de Constantes y Variables.....	91
Figura 44. Configuración e Inicialización.....	91
Figura 45. Bucle Principal.....	92
Figura 46. Función setupWiFi()	92
Figura 47. Función setupMQTT()	92
Figura 48. Función reconnectMQTT().....	93
Figura 49.Funció mqttCallback()	93
Figura 50. Función setupRadio().....	94
Figura 51. Función processReceivedData()	94
Figura 52. Función sendActivationCommand().....	95
Figura 53. Inserción de Datos en la Base de Datos	95
Figura 54. Verificación y Actualización del Estado del Riego.....	96
Figura 55. Control del Sistema de Riego Automático	97
Figura 56. Función insert_irrigation_history_on_manual.....	98
Figura 57. Función control_irrigation_manual	98
Figura 58. Inicialización de Firebase y Envío de Notificaciones.....	99
Figura 59. Estructura de carpetas del backend web	100
Figura 60. Endpoint creación de usuario.....	100

Figura 61. Endpoint recuperar contraseña	101
Figura 62. Endpoint ingresar al sistema.....	101
Figura 63. Endpoint consulta el ultimo registro de los sensores.....	102
Figura 64. Endpoint estado del riego	102
Figura 65. Endpoint Recupera la humedad del suelo.....	103
Figura 66. Endpoint Historial del riego	103
Figura 67. Endpoint Activación del riego manualmente	104
Figura 68. Endpoint Actualizar usuario	104
Figura 69. Endpoint Actualizar contraseña.....	105
Figura 70. Estructura de las carpetas del sistema.....	106
Figura 71. Formulario para registrar usuario	106
Figura 72. Función para registrar usuario- Frontend	106
Figura 73. Recuperar contraseña.....	107
Figura 74. Función recuperar contraseña- Frontend	107
Figura 75. Ingreso al sistema	108
Figura 76. Función ingresar usuario- Frontend.....	108
Figura 77. Pantalla principal datos sensores	109
Figura 78. Función obtener los datos de los sensores - Frontend	109
Figura 79. Función obtener el ultimo estado del riego- Frontend.....	109
Figura 80. Función obtener datos de sensores del suelo- Frontend	110
Figura 81. Actividades del riego	110

Figura 82. Función registrar el historial del riego- Frontend	111
Figura 83. Actualizar usuario	111
Figura 84. Función editar usuario- Frontend	112
Figura 85. Función actualizar usuario- Frontend	112
Figura 86. Construcción proyecto Ionic	119
Figura 87. Capacitor Android	119
Figura 88. Sincronización del proyecto	119
Figura 89. Abrir Android Studio	120
Figura 90. Generación de apk	120
Figura 91. Apk instalable Android.....	121
Figura 92. Terrero de pruebas del prototipo	121
Figura 93. Panel Solar	122
Figura 94. Controlador de carga.....	122
Figura 95. Batería.....	122
Figura 96. Emisor de datos	123
Figura 97. Sensor de velocidad del viento	123
Figura 98. Sensor de humedad del suelo.....	123
Figura 99. Sensor de temperatura y humedad del ambiente	124
Figura 100. Receptor de datos.....	124
Figura 101. Bomba de agua	125
Figura 102. Micro aspersores	125

Figura 103. Resultados modelo TAM-Utilidad percibida..... 127

Figura 104. Resultados modelo TAM-factibilidad de uso percibida 128

ÍNDICE DE ANEXOS

Anexo A. Cuestionario para la encuesta de los agricultores	137
Anexo B. Solicitudes de validación de la encuesta.....	139
Anexo C. Manual de usuario.....	142
Anexo D. Pruebas de distancia módulos Lora	153

RESUMEN EJECUTIVO

Los agricultores en la zona alta del cantón Quero enfrentan el desafío de las heladas, bajas temperaturas que afectan negativamente el desarrollo de los cultivos. Las técnicas empleadas carecen de efectividad y de una base científica sólida para mitigar estos efectos adversos.

Este proyecto se centra en el desarrollo de un prototipo de aplicación móvil para el control y monitoreo de los cultivos de papa durante eventos de heladas. La aplicación se integra con sensores que miden la temperatura, humedad y velocidad del viento, permitiendo un monitoreo continuo de las condiciones climáticas. Además, envía alertas a los agricultores sobre el estado del riego. Cabe destacar que el sistema de riego se activa y desactiva automáticamente al cumplir con los umbrales establecidos, aunque también ofrece la posibilidad de que el usuario decida manualmente su funcionamiento. La sostenibilidad del sistema está garantizada mediante el uso de paneles solares como fuente de energía, y el riego por aspersión se emplea como la principal técnica de protección contra las heladas.

En la fase de recolección de datos, se realizaron encuestas a los agricultores y se consultó a tres expertos en agronomía para definir los umbrales de los sensores. Para la construcción del prototipo, se empleó la metodología RAD (Rapid Application Development), que permite una entrega rápida del producto. La aplicación móvil fue desarrollada utilizando Ionic con React, mientras que el sistema de emisor y receptor se construyó con dos placas 915MHz ESP32 LORA V3 que utilizan la comunicación Lora.

Palabras clave: Aplicación móvil, sensores meteorológicos, Lora, heladas, cultivo de papa.

ABSTRACT

Farmers in the highlands of the Quero canton face the challenge of frosts, low temperatures that negatively affect crop development. The techniques currently employed lack effectiveness and a solid scientific basis to mitigate these adverse effects.

This project focuses on developing a prototype mobile application for the control and monitoring of potato crops during frost events. The application integrates sensors that measure temperature, humidity, and wind speed, allowing continuous monitoring of climatic conditions. Additionally, it sends alerts to farmers about the irrigation status. It is noteworthy that the irrigation system activates and deactivates automatically upon reaching the established thresholds, although it also provides the option for the user to decide its operation manually. The sustainability of the system is ensured using solar panels as an energy source, and sprinkler irrigation is employed as the main technique for frost protection.

During the data collection phase, surveys were conducted with farmers and three agronomy experts were consulted to define the sensor thresholds. For the construction of the prototype, the RAD (Rapid Application Development) methodology was used, which allows for the rapid delivery of the product. The mobile application was developed using Ionic with React, while the transmitter and receiver system was built with two 915MHz ESP32 LORA V3 boards that use LoRa communication.

Keywords: Mobile application, weather sensors, LoRa, frosts, potato cultivation.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

Desarrollo de un prototipo de aplicación móvil de control y monitoreo para la protección de los cultivos de papas contra las heladas utilizando energías renovables en la zona alta del cantón Quero

1.1.1 Planteamiento del problema

En [1] manifiesta que las heladas son un fenómeno atmosférico que se da por la disminución de nubosidad de los Andes produciendo el descenso de la temperatura inferior a 0 °C, a una altura de 1,25m y 2m sobre el nivel de suelo. Los daños pueden ser directo o indirecto en las plantas, en [2] el daño directo se produce cuando existe formación de cristales de hielo dentro de las células de la planta. Mientras que el daño indirecto se forma hielo dentro de la planta, pero fuera de las células denominado congelación extracelular. Ecuador principalmente en la región Sierra poseen suelos muy fértiles para los cultivos de tubérculos, hortalizas y legumbres; pero debido a ubicación geográfica las heladas son un fenómeno climático que limita la producción de pequeños y grandes agricultores especialmente en los meses de enero, marzo, julio, agosto y diciembre.

En [3] las consecuencias de las heladas en Tungurahua se han extendido a diversos cultivos principalmente en la producción de papa, fresa y maíz generando preocupación por los habitantes. Se estima que al menos 510 hectáreas de tierras cultivables en las localidades de Ambato, Baños de Agua Santa, Patate, Pelileo, Mocha, Quero y Tisaleo han sido afectadas por eventos climáticos desde enero hasta septiembre de 2023. Según el Instituto Nacional de Meteorología e Hidrología (Inamhi) indica un descenso de las temperaturas en el callejón interandino oscilando entre 1 °C y 6 °C en las zonas elevadas superiores a los 2700 metros sobre el nivel del mar. Este fenómeno climático ha intensificado la vulnerabilidad de los cultivos, los efectos negativos de las heladas y afectando directamente la productividad agrícola. Según [4] el Ministerio de Agricultura y Ganadería (MAG) en el año 2023

Tungurahua ocupa el quinto lugar a nivel provincial en términos de la producción de papas en un área de 2.557 hectáreas. En 2019 sembrar una hectárea tenía un costo de \$5,000, pero en 2023 se estima que la inversión oscila entre \$6,500 y \$7,500.

Según los autores de [5] se registró temperaturas inferiores a 1,8 grados centígrados en Quero, generando preocupación entre los agricultores locales por el impacto negativo en los cultivos. Ante esta situación, los productores han implementado diversas medidas para mitigar los efectos del frío. Entre estas estrategias se encuentra la quema de residuos vegetales que ayuda generando calor, pero tiene consecuencias no siempre positivas para las plantas. Además, se ha observado el riego manual de los campos como un intento de combatir las heladas, aunque su eficacia puede ser limitada en condiciones extremas. Otra táctica adoptada es la colocación estratégica de recipientes con agua a lo largo de los terrenos afectados. Esta medida busca aprovechar la capacidad del agua para retener y liberar calor gradualmente, actuando como un regulador térmico. Sin embargo, su efectividad puede variar y no siempre cuenta con respaldo científico.

Quero experimenta bajas temperaturas durante ciertas estaciones del año, lo cual ha tenido un impacto significativo en los cultivos, especialmente en el caso de la papa que representa la mayor inversión. Además, las heladas amenazan la viabilidad de los cultivos. Cabe destacar que este fenómeno afecta extensas áreas de terreno cultivadas. Sin embargo, los productores no cuentan con medidas eficientes que ayuden a mitigar los daños causados por este fenómeno climático.

1.2 Antecedentes investigativos

El proyecto que menciona en [6] consiste en un sistema electrónico para monitorear, controlar la helada y gestionar el riego en huertos de fresas utilizando redes de sensores inalámbricos con hardware libre. La metodología empleada fue analítica e incluyó actividades como investigación teórica, diseño e implementación del hardware y software, así como análisis y resultados del sistema implementado. El sistema se basa en un Arduino Mega con comunicación por radiofrecuencia y sensores de humedad (Functional Category) FC-28 y Sensor de Humedad y Temperatura Ambiente (Digital Humidity and Temperature) DHT 21. La prevención

de heladas se realiza mediante aspersores de 2.5 mm/h alimentados por una batería de 5V. La comunicación se establece a través de alertas móviles y web, con una interfaz de operación manual a través de una pantalla LCD (Liquid Crystal Display). El sistema opera eficientemente con una respuesta de 15 segundos, una transmisión de radiofrecuencia de 60 metros y antenas de 17 cm. Sin embargo, desventaja principal es la operación exclusivamente manual y la interfaz limitada a LCD.

En [7] se llevó a cabo el desarrollo de un prototipo electrónico destinado al control de heladas en cultivos de arándanos. Esta investigación aplicada se basó en un enfoque metodológico analítico y experimental, incorporando observaciones detalladas. El rango de temperaturas considerado para la prevención de heladas se dividió en categorías, desde alta (-6 °C a -3°C) hasta muy baja (3°C a 6°C). El sistema se construyó utilizando plataformas como Arduino, Raspberry, Waspmove y BeagleBone Black, con la placa Arduino como base. La comunicación se establece mediante radiofrecuencia, y los sensores utilizados incluyen DHT21 para temperatura y humedad, Radiación UVGUVA-S12SD y Humedad de Suelo FC-28. Aunque el tiempo de respuesta es de 30 segundos con una cobertura de 150 metros, la desventaja principal radica en que la interfaz solo detecta heladas y no ofrece protección activa.

En [8] se creó un sistema electrónico de monitoreo climático y detección de heladas destinado al Gobierno Autónomo Descentralizado (GAD) del cantón Pujilí. Este proyecto, enmarcado en una investigación aplicada con enfoque bibliográfico, se basó en la plataforma LILYGO TTGO LoRa32, empleando comunicación de largo alcance (Lora). Los sensores utilizados incluyen DHT22 para la medición de humedad y temperatura, así como el ADS1115 para la velocidad del viento. El sistema, que abarca un área de 1183.14 m². Cuenta con una pantalla LCD, aunque el tiempo de respuesta es eficiente, con 30 segundos y una cobertura de 966 metros, es importante destacar que el sistema no proporciona una capa de protección activa contra heladas.

Según los autores de [9] se desarrolló un software de control meteorológico con el propósito de detectar heladas y emitir alertas tempranas, integrándolo en el Geoportal del Honorable Gobierno Provincial de Tungurahua. Esta investigación, que abarcó

métodos bibliográficos, de campo y experimentales, se destacó por su enfoque integral. El software, basado en PHP (Personal Home Page), MongoDB y CodeIgniter, permite la emisión de alertas mediante mensajes SMS (Short Message Service). La detección se realiza a través de datos recopilados de estaciones meteorológicas, proporcionando una comparación detallada de la información registrada por los sensores. Aunque el sistema ofrece una detección anticipada de heladas, su enfoque se limita exclusivamente a la identificación de este fenómeno meteorológico.

En [10] el desarrollo de un sistema electrónico avanzado destinado al monitoreo y control para la prevención de heladas en cultivos de mora, implementando la Arquitectura de Internet de las Cosas (IOT). Este sistema incorpora tarjetas electrónicas como ESP32 y Arduino UNO, junto con un módulo de relés, para posibilitar la comunicación a través de diversas tecnologías como WI-FI, Lora y Zigbee. Los sensores utilizados incluyen DTH11 para temperatura y humedad, YL-69 para humedad del suelo y GUVA-S12SD para radiación solar UV. La prevención de heladas se lleva a cabo mediante el sistema de aspersión ASP-8, alimentado por una batería de 12V. La comunicación de alertas se realiza a través de WhatsApp. La interfaz de usuario se ha desarrollado utilizando HTML (Hypertext Markup Language), PHP y CSS (Cascading Style Sheets), con opciones de visualización tanto en dispositivos móviles como en la web. Este sistema, enfocado en cultivos de mora en una extensión de 200 m². La investigación se centra en la recopilación de datos de sensores y en la comparación de soluciones para la protección contra heladas.

En [11] presenta una metodología descriptiva, incorporando elementos cuantitativos y cualitativos. La comunicación se logra a través del módulo Lora: RA02 433 MHz, y la placa utilizada es el LoRA02. Además, se emplean diversos sensores, como el DHT11 para medir temperatura y humedad relativa, un sensor de lluvia YC-69, un sensor de intensidad lumínica y un sensor de presión barométrica BMP280. El sistema permite la comunicación mediante alertas a través de dispositivos móviles y web. El enfoque del dispositivo se centra en el monitoreo de cultivos de aguacate, abarcando un área de 194.69 m². A pesar de sus resultados, el principal inconveniente reside en su tiempo de respuesta, que es de 10 minutos y cubre una

distancia de 350 metros. Esta demora podría limitar su eficacia en situaciones críticas, constituyendo una desventaja significativa para el monitoreo climatológico en tiempo real.

En [12] el sistema implementado a través de la plataforma Arduino NANO y conectividad bluetooth, se diseñó para proteger los cultivos contra daños causados por bajas temperaturas. El sistema incluye sensores como el DHT11 para medir la humedad y el LM35 para la temperatura. Para la aplicación del riego preventivo, se utiliza un aspersor plástico PRETUL (162). La autonomía del sistema se garantiza mediante una batería MT12180HR. La cobertura del sistema es extensa, abarcando 7.5 hectáreas divididas en 6 módulos de 1.25 hectáreas cada uno. El costo de implementación del sistema se estima en \$5,751,675, y su funcionamiento se basa en una alimentación eléctrica constante. Los datos recopilados por los sensores son almacenados en bancos de datos, incluyendo el IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales) para referencia. Aunque el sistema presenta una eficiente distancia de comunicación de 10 metros, es importante destacar que esta limitación podría afectar su aplicabilidad en áreas más extensas. La distancia de comunicación podría ser una desventaja en situaciones donde la cobertura requerida es mayor, y se debería considerar al evaluar su implementación en distintos entornos agrícolas.

En [13] el sistema utiliza el microcontrolador Arduino MEGA 2560 como plataforma principal. La implementación de sensores, incluyendo el sensor de temperatura y humedad DHT11 y el sensor de lluvia FC-37, permite la monitorización de condiciones climáticas específicas. Este sistema automatizado no cuenta con una función de operación automática. En cambio, su operación se basa en un enfoque manual, lo que sugiere que la protección de los cerezos debe ser activada por el usuario según sea necesario. Los datos recopilados por los sensores son utilizados para informar al usuario sobre las condiciones climáticas, permitiéndole tomar decisiones informadas respecto a la activación del sistema de protección. Aunque el sistema ofrece una solución para la protección de los cerezos, la limitación en la operación automática podría considerarse una desventaja, ya que podría depender en gran medida de la prontitud y atención del usuario ante cambios climáticos inesperados.

En [14] este sistema utiliza la plataforma Raspberry Pi 3 con conexión WI-FI para realizar el monitoreo y la detección de condiciones de heladas blancas en áreas rurales. El prototipo incorpora sensores de temperatura y humedad, específicamente el DHT22, para recopilar datos relevantes. La capacidad de operar bajo el entorno de MicroPython destaca la flexibilidad de programación del sistema. Además, se hace uso de la conectividad WI-FI para transmitir datos y recibir alarmas, proporcionando una respuesta rápida ante condiciones adversas. A pesar de su funcionalidad efectiva en la detección de heladas blancas, la desventaja notoria es la falta de protección mencionada. Este aspecto podría limitar la durabilidad del prototipo frente a condiciones climáticas extremas o situaciones adversas, lo que se debe tener en cuenta al considerar su implementación en entornos rurales específicos.

En [15] el sistema integra las plataformas Arduino Uno y Raspberry Pi 3 modelo B para controlar de manera automática la protección contra heladas en huertos de paltos. La conexión se realiza a través de WI-FI. El sistema incorpora un sensor de temperatura ambiente y humedad AM2301 para monitorear las condiciones climáticas. Para prevenir daños por heladas, se utiliza un ventilador modelo WSG1068 para la ventilación. La alimentación del sistema es mediante energía eléctrica convencional. Aunque el sistema cuenta con la capacidad de operar de forma automática, permitiendo la activación del ventilador según las condiciones climáticas, destaca que también es posible realizar operaciones manuales. Esta versatilidad permite a los usuarios intervenir directamente en el sistema según sea necesario. A pesar de su eficacia en la protección contra heladas, una posible desventaja podría surgir de la dependencia de energía eléctrica convencional, lo que podría limitar su funcionalidad en situaciones de cortes de energía o en lugares remotos sin fácil acceso a suministros eléctricos estables.

En [16] el sistema utiliza el microcontrolador Arduino ATmega y se conecta mediante WI-FI (Wireless Fidelity). La infraestructura del sistema incorpora sensores DHT22 y YL-69, diseñados para monitorear parámetros clave relacionados con la sequía. El sistema se alimenta mediante energía solar, respaldado por una batería de 12V para garantizar su operación continua. Además, se implementa comunicación a través de GSM para facilitar el control remoto y la supervisión del tubo de pozo. Una de las características destacadas es la capacidad de riego

automático basado en los datos recopilados por los sensores. Sin embargo, es importante señalar que la operación manual también es posible a través de una interfaz LCD. Aunque este enfoque brinda flexibilidad, la desventaja potencial podría radicar en la complejidad de la interfaz y en la dependencia de la disponibilidad de luz solar para garantizar un suministro de energía constante. Además, la operación manual a través de una interfaz LCD podría limitar la accesibilidad para aquellos usuarios menos familiarizados con tecnologías avanzadas.

En [17] se destaca por el despliegue integral de sensores, la automatización del riego, la utilización de drones y el aprendizaje automático para la detección de enfermedades, junto con la aplicación de redes neuronales artificiales para prever el rendimiento de los cultivos. El sistema, que utiliza la plataforma Arduino Uno, se conecta a través de WI-FI. Los sensores implementados incluyen el Sensor de Temperatura y Humedad (DHT11), que permite un monitoreo preciso de las condiciones ambientales. Este enfoque se distingue por la implementación de riego mediante drones, con comunicación a través de GSM para optimizar la eficacia en áreas agrícolas extensas. El proyecto también enfatiza la importancia de la toma de decisiones automatizada y la optimización de parámetros agrícolas para maximizar la eficiencia del cultivo. Sin embargo, es necesario considerar que la desventaja potencial de este sistema podría residir en la complejidad y el costo asociado con la integración de múltiples tecnologías avanzadas, lo que podría limitar su accesibilidad para agricultores con recursos limitados o en áreas con infraestructura limitada.

En [18] el estudio presenta tres tipos de sensores diferentes: el Type-I de sonda única, el Type-II de doble sonda y el Type-III de doble sonda. Estos sensores, diseñados para medir la humedad del suelo en diversas profundidades, son construidos con materiales específicos como cobre, bronce, acero y acero galvanizado. Cada tipo de sensor tiene longitudes distintas, permitiendo un monitoreo detallado de las condiciones del suelo en una plantación agrícola. La elección de Arduino Mega y Raspberry PI3 como plataformas de control proporciona una conectividad a través de WI-FI, facilitando la recolección y análisis de datos de manera remota. Aunque este enfoque de sensores ofrece una mayor adaptabilidad a las condiciones específicas del suelo, es importante destacar que la diversidad de

materiales y longitudes podría generar una desventaja en términos de estandarización y mantenimiento, requiriendo una gestión cuidadosa para garantizar la precisión y eficacia continua del sistema.

En [19] el sistema tiene como objetivo mejorar la eficiencia del riego por aspersión al incorporar sensores clave en el proceso. La plataforma central empleada es Arduino Mega, con conectividad a través de WI-FI. El sistema integra sensores tales como el DHT11 para medir la humedad y el LM35 para la temperatura, optimizando así el control del riego por aspersión. Destaca por su capacidad de operar sin necesidad de intervención manual, proporcionando una solución automatizada para la gestión del riego. Además, permite la conectividad a través de WI-FI, facilitando el monitoreo remoto y la configuración del sistema.

En [20] la investigación se basó en un sistema automatizado de rociadores que ajustaba dinámicamente la tasa de aplicación de agua según las condiciones atmosféricas. Se implementó un modelo modificado para calcular esta tasa, considerando variables como la velocidad del flujo de aire, la temperatura y humedad del aire, y la temperatura del agua de pulverización. Los resultados de los experimentos de campo revelaron que una tasa variable de aplicación de agua resultó más efectiva para prevenir daños por heladas, y el control intermitente modificado regulaba automáticamente dicha tasa. Durante noches de heladas pesadas, el área irrigada mantuvo temperaturas significativamente superiores a las áreas no irrigadas, siendo hasta 2.8°C más alta. Además, se observó que el área irrigada con el modelo modificado experimentó un aumento más gradual de temperatura después del amanecer, proporcionando un beneficio adicional para la protección contra heladas. Estos resultados sugieren que la estrategia de control de rociadores implementada es un método efectivo que podría aplicarse en campos de té, demostrando su potencial para mejorar la resiliencia de los cultivos frente a condiciones climáticas adversas en la región del río Yangtze.

En [21] se recopilaron datos multidimensionales y no lineales, incluyendo parámetros como temperatura, humedad, radiación y velocidad del viento, durante un periodo extenso. La investigación utilizó SVM (Support Vector Machine) con tres tipos de funciones de núcleo, destacando el modelo de núcleo de función de base

radial (SVM-BRF) con los 32 parámetros micro meteorológicos. La validación cruzada de cinco veces se empleó para mejorar la confiabilidad de los resultados de predicción. Los hallazgos revelaron que el SVM-BRF con todos los datos micro meteorológicos obtuvo una alta precisión en los conjuntos de entrenamiento y prueba. Además, se observó que la inclusión de datos multidimensionales y no lineales resultó crucial para mejorar la precisión de la predicción. Considerando la complejidad del modelo y su precisión, se recomendó el uso de datos micro meteorológicos a la altura del dosel con el modelo SVM-BRF para la predicción de heladas por radiación en el Delta del río Yangtze, sugiriendo su aplicabilidad en regiones similares con terrenos y microclimas comparables.

En [22] los eventos de congelación entre enero y abril representan una amenaza significativa para los productores de arándanos en Florida y Georgia, y la encuesta a los productores indicó diversas opiniones sobre los criterios de decisión para la protección contra heladas. Casi el 99% de los encuestados que utilizaron al menos un método de protección activa contra las heladas informaron el uso de riego como medida preventiva. La investigación encontró que el tamaño de la granja, en términos de la superficie de arándanos, no influyó en las decisiones de implementar medidas activas de protección contra heladas. Además, se destacó que los productores estaban dispuestos a tolerar pérdidas considerablemente altas (30% al 39%) y aún producir un cultivo comercializable. Sin embargo, se observó que algunos productores podrían haber sido demasiado cautelosos, aplicando protección durante etapas tempranas del brote 40% y etapas del clúster apretado 55%. Sin embargo, la extensión del terreno hace que el factor económico sea un factor limitante.

En [23] dada la vulnerabilidad de los cultivos hortícolas a la escarcha y las consecuentes pérdidas económicas, la necesidad de tecnologías efectivas de protección contra heladas es crucial. El estudio revisó el desarrollo de estas tecnologías, centrándose en investigaciones mecanizadas recientes. Ante el limitado éxito de algunas tecnologías individuales, se propuso la tecnología combinada, específicamente la protección contra heladas que involucra tanto la ventilación, así como el riego por aspersión. Los resultados y análisis proporcionados en el estudio respaldan la efectividad de esta combinación, sugiriendo un enfoque prometedor para

mitigar los impactos negativos de la escarcha en los cultivos hortícolas. Pero el costo del sistema fue muy alto en comparación a los anteriores.

En [24] se destaca avances recientes en la investigación que aborda la prevención de daños por heladas, centrándose en mecanismos adaptativos como la desaclimatación y aclimatación. Además, se exploraron enfoques distintos que revelaron integralmente la fisiología y los mecanismos moleculares del melocotón. La revisión también abordó el uso de productos químicos y la comprensión de los mecanismos de las heladas primaverales mediante la estimulación de la temperatura en cámaras ambientales y estudios de termografía infrarroja. Sin embargo, este proyecto necesita la intervención de agrónomos para su implementación.

En [25] se emplea una conexión de Raspberry a Arduino, y la comunicación se realiza mediante la tecnología LORA WAN (Wide Área Network). Los sensores incorporados incluyen LM35, DHT11 y BMP180, diseñados para monitorear variables clave como la temperatura, humedad y presión atmosférica. Este sistema se enfoca en la automatización del riego por aspersión como medida preventiva contra los efectos de las heladas. A diferencia de otros, no requiere la intervención manual en el proceso de riego. Además, destaca por su capacidad de aprendizaje automático, lo que sugiere una adaptabilidad a las condiciones cambiantes del entorno. El sistema también incorpora funciones de alimentación automatizada para facilitar su operación continua. Pero su desventaja podría surgir en el costo y complejidad asociados con la implementación de la tecnología de aprendizaje automático, lo que podría limitar su accesibilidad para ciertos usuarios o agricultores con recursos limitados.

1.3 Fundamentación teórica

Es la base conceptual que respalda un proyecto, investigación o trabajo académico. En el contexto de la protección contra las heladas, implica revisar y comprender investigaciones previas y prácticas relacionadas con estrategias de mitigación.

1.3.1 Placas de recolección de datos

Las tarjetas electrónicas son circuitos que están compuestos por un microcontrolador, resistencias, condensadores, transistores y conectores. En los artículos [6],[13],[16],[18] se usó Arduino Mega debido a su mayor cantidad de pines de E/S (Entrada/Salida), memoria y capacidad de procesamiento. Esto permite la conexión de más sensores y actuadores, así como el manejo de tareas más complejas. Sin embargo, Arduino Mega utiliza un voltaje de alimentación de 7 voltios, siendo un factor crítico donde la eficiencia energética y la gestión de la potencia no son suficientes. En [10],[15],[17] los autores hacen uso de Arduino Uno porque es más compacto, además su precio es mucho menor al de Arduino Mega. Al ser más antiguo tiene mayor disponibilidad de placas de expansión lo que permite la integración de los sensores para la recolección de datos. Su voltaje de alimentación de 5 voltios por lo optimiza el consumo de energía. Por otro lado, en [8], [11] priorizan la comunicación de los datos a largo alcance por lo cual las placas LORA al utilizar radiofrecuencia llegan hasta 10km. Esto importante debido a la extensión de los terrenos que se pretenden monitorear y controlar en relación con las heladas.

1.3.2 Sensor de temperatura

Un sensor de temperatura es un dispositivo que detecta y mide la temperatura del entorno, convirtiendo esta información en una señal eléctrica o digital. En [[11], [12], [13]] el sensor DTH11 es más económico y conocido. Presenta un rango de medición de temperatura de 0 a 50 grados Celsius con una precisión moderada, generalmente dentro de un rango de ± 2 grados Celsius. Esto es un limitante para mediciones precisas en condiciones agrícolas críticas como las heladas. Además de medir la temperatura, también puede medir la humedad relativa en un rango del 20\% al 80\%. Por otro lado el sensor LM35 empleado por los autores en [13], [25] se destaca por su alta precisión y su salida lineal con respecto a la temperatura. Ofrece un rango de medición más amplio en comparación con el DHT11, normalmente entre -55 y 150 grados Celsius. Su precisión es de ± 0.5 grados Celsius. La alta presión y el rango de medición hace que sea una opción para la medición de heladas. En [14], [16] utilizan el sensor DHT22, también conocido como AM2302, comparte similitudes con el

DHT11 al medir temperatura y humedad relativa. Sin embargo, se distingue por su mayor precisión y un rango de medición de temperatura más amplio, generalmente de -40 a 80 grados Celsius, con una precisión típica de ± 0.5 grados Celsius. También mide la humedad relativa en un rango más amplio (0% a 100%), es relevante en entornos agrícolas cambiantes.

1.3.3 Sensor de humedad

Un sensor de humedad es un dispositivo que mide la cantidad de vapor de agua presente en el aire o suelo, generando una señal eléctrica proporcional a la humedad relativa. En [7],[6] con el uso de el sensor FC-28 destacado por su diseño simple y su bajo costo, proporcionando una salida analógica. En cambio, en los artículos [10], [15]el sensor YL-69 es similar al FC-28 en términos de funcionamiento, pero suele ofrecer una respuesta más rápida y una mayor sensibilidad. Ambos sensores son compatibles con placas de desarrollo como Arduino y se utilizan para evaluar la humedad presente en el ambiente.

1.3.4 Sensor de velocidad del viento

Un sensor de velocidad del viento cuantifica la velocidad de movimiento del aire, utilizando la rotación de hélices y proporciona una salida eléctrica proporcional a dicha velocidad. Los artículos [7], [11] mencionan que el sensor TERRIFIC está diseñado específicamente para aplicaciones agrícolas y combina la medición de humedad del suelo con la temperatura y la conductividad eléctrica, ofreciendo una evaluación más completa del estado del suelo. Por otro lado, el sensor ADS1115 utilizado en [8] es un Convertidor Analógico a Digital (ADC, en inglés) que se puede utilizar para mejorar la precisión y la resolución de la lectura analógica de sensores de humedad.

1.3.5 Tipo de comunicación

La conectividad inalámbrica es la capacidad de los dispositivos o sistemas para intercambiar información entre sí sin la necesidad de cables físicos. En [11], [25] Lora, conocido por su largo alcance y eficiencia energética, se destaca en

aplicaciones que requieren comunicación a larga distancia y consumo de energía optimizado, siendo una elección frecuente para sistemas de monitoreo remoto en cultivos. Por otro lado, en [15],[16], [18], [19] Wifi ofrece una conexión de alta velocidad y ancho de banda, siendo más adecuado para entornos donde la velocidad de transmisión de datos es esencial, como en sistemas de monitoreo en tiempo real en áreas urbanas o zonas donde se tenga la disponibilidad de internet.

1.3.6 Almacenamiento

El almacenamiento de datos se refiere al proceso de guardar información de manera temporal o permanente en dispositivos físicos o en la nube para su posterior acceso y utilización. En [8],[9] la nube ofrece la capacidad de almacenar y acceder a datos desde cualquier lugar con conexión a internet. Proporciona flexibilidad, colaboración en tiempo real y la posibilidad de escalar el espacio de almacenamiento según las necesidades. Por otro lado, en [7], [6], [15] las tarjetas microSD son soluciones de almacenamiento local que se insertan en dispositivos como cámaras, teléfonos o sistemas embebidos. Ofrecen una opción para almacenar datos de manera segura y accesible, incluso en entornos sin conectividad constante a la red.

1.3.7 Tipo de riego

El riego es el proceso de suministrar agua artificialmente a plantas, cultivos o terrenos agrícolas para garantizar su crecimiento y desarrollo adecuados. Además, ayuda a mitigar daños causados por las bajas temperaturas. En [6],[10],[12] el riego por aspersión al rociar agua sobre las plantas ayuda a formar una capa de hielo protectora que actúa como aislante térmico, reduciendo el daño potencial causado por las heladas. En contraste en [16], el riego por goteo al aplicar agua de manera más controlada y dirigida a las raíces puede minimizar la formación de hielo en la superficie de las plantas, reduciendo así el riesgo de daño por heladas.

1.3.8 Lenguajes de programación

Un lenguaje de programación es un conjunto formal de reglas sintácticas y semánticas que define la estructura y el significado de instrucciones destinadas a ser

ejecutadas por un dispositivo computacional. En [26] se basa en el Starter Kit Javelin Stamp, programado con el lenguaje Java de Sun Microsystems. Este enfoque sugiere una elección específica y dedicada al desarrollo en Java, proporcionando un control inteligente para el dispositivo de temperatura DS1620. En contraste, en [27] presenta una perspectiva centrada en el desarrollo de aplicaciones móviles que publican y alertan sobre datos meteorológicos e índices nocivos. Utilizan herramientas como Xamarin y .NET Core, junto con el sistema de chip ESP 32. Este enfoque implica una combinación de tecnologías modernas y multiplataforma, permitiendo una implementación más versátil y conectada.

1.3.9 Interfaz de Programación de Aplicaciones (API)

API (Application Programming Interface) es un conjunto de reglas y herramientas que permite a diferentes tipos de software comunicarse entre sí, facilitando la integración de funcionalidades y datos de una aplicación a otra. Firebase y Back4App son dos plataformas de desarrollo de aplicaciones móviles que ofrecen servicios de backend. En [28] Firebase respaldado por Google utiliza API REST (Application Programming Interface based on Representational State Transfer) para integrar características y funcionalidades en las aplicaciones, y se distingue al permitir la incorporación de Aprendizaje Automático (Machine Learning) para diversas tareas. En cuanto al desarrollo de aplicaciones, Firebase simplifica el proceso mediante la utilización de API, facilitando la implementación de características como bases de datos, aprendizaje automático, notificaciones, análisis y autenticación. Por otro lado, en [29] Back4App es reconocido como un proveedor de servicios backend de código abierto. Se destaca por características como la escalabilidad de recursos, desarrollo fácil, rapidez en el desarrollo y despliegue veloz de aplicaciones. A través de API REST y GraphQL, permite funciones como almacenamiento de datos, API, notificaciones y almacenamiento.

1.3.10 Plataformas de la nube

Una plataforma de la nube es un entorno en línea que ofrece servicios y recursos computacionales, de almacenamiento y de red, permitiendo a los usuarios desarrollar,

implementar y gestionar aplicaciones sin depender de infraestructuras físicas. En [30] AWS (Amazon Web Services) proporciona una extensa variedad de servicios, incluyendo AWS IOT Core, que facilita la conectividad y administración de dispositivos IOT, como sensores meteorológicos. Además, su escalabilidad y robustez son notables, permitiendo gestionar grandes volúmenes de datos generados por sensores. Por otro lado, en [31] Azure ofrece Azure IOT Hub y Azure Functions, brindando una integración fluida con dispositivos IOT y permitiendo la implementación de funciones sin servidor para el procesamiento de datos meteorológicos en tiempo real. Ambas plataformas ofrecen sólidas características de seguridad y cumplimiento normativo, aunque Azure destaca por su estrecha integración con tecnologías Microsoft.

1.3.11 Metodologías de desarrollo

Las metodologías agiles mejoran la flexibilidad y la capacidad de respuesta en el desarrollo de software. En [32] Scrum está centrado en la gestión del proyecto, divide el trabajo en Sprint y se basa en roles claramente definidos, como el Product Owner, el Scrum Máster y el Equipo de Desarrollo. Prioriza la entrega de incrementos de producto al final de cada sprint, fomentando la adaptación continua a través de reuniones regulares y revisiones. En contraste, XP (Extreme Programming) se centra en la mejora continua del código y la colaboración estrecha. En [33] permite que las pruebas unitarias sean continuas y el diseño simple, promoviendo la calidad del código y la velocidad de respuesta a cambios. Aunque Scrum proporciona un marco general para la gestión de proyectos, XP se sumerge más en las prácticas técnicas específicas que buscan maximizar la productividad y la calidad del software.

1.3.12 Cultivo de papa

El cultivo de papas es una práctica agrícola de gran importancia a nivel mundial. En [34] los autores expresan que se trata de un tubérculo perteneciente a la familia de las solanáceas, científicamente conocido como *Solanum tuberosum*, y su cultivo abarca una amplia variedad de climas y suelos. Estas plantas desarrollan tubérculos subterráneos, que son la parte comestible y se utilizan para la producción de

alimentos como papas fritas, puré de papas. La siembra de papas generalmente se realiza en primavera, y la cosecha ocurre cuando las plantas han completado su ciclo de crecimiento y los tubérculos están listos para su recolección.

En la Tabla 1 se muestra la clasificación de las especies más conocidas de papa, así como su altura.

Tabla 1. Temperatura soportada por la papa[35].

Nombre comercial	Altura máxima (cm)	Temperatura soportada
Papa Única	30-50	15-20 °C
Papa Chola	40-60	10-15 °C
Papa Pera	50-70	0-15 °C
Papa Ubilla	60-80	0-10 °C

1.3.13 Heladas

Es un fenómeno meteorológico caracterizado por una disminución de la temperatura ambiente a niveles inferiores al punto de congelación del agua. Este descenso en la temperatura conlleva a que el agua o el vapor de agua presentes en el aire se solidifiquen, depositándose en forma de hielo en las superficies. Según la Organización Meteorológica Mundial define la helada como la formación de diversos tipos de cobertura de hielo sobre el suelo, resultado de la acumulación directa del vapor de agua[36].



Figura 1. Helada en el cultivo de papa- Imagen generada con Chat GPT 20 de marzo de 2024

a. Factores que producen una helada

En la Tabla 2 se muestra los tipos de heladas con su respectiva descripción.

Tabla 2. Factores de inciden en una helada[37].

Factor	Descripción
Nubosidad	Las nubes actúan como una barrera que evita las pérdidas de energía. Absorben y emiten radiación, reduciendo las diferencias térmicas.
Velocidad del Viento	El viento mezcla las capas de aire, igualando las temperaturas. La falta de viento aumenta el riesgo de helada.
Humedad del Aire	El aire húmedo puede saturarse y liberar calor al condensarse o congelarse, frenando el enfriamiento. El aire seco aumenta el riesgo de helada (helada negra).
Laboreo del Suelo	Un suelo trabajado es más poroso y tiene menor conductividad térmica. La vegetación uniforme actúa como aislante térmico.
Relieve del Terreno	La inclinación del terreno afecta la intensidad de la helada. Las áreas más bajas acumulan aire frío, mientras que en las cimas de las colinas la temperatura puede ser más alta.

Factor	Descripción
Altura sobre el Suelo	La temperatura disminuye con la altura durante la noche, ya que la superficie del suelo se enfriá primero.

b. Clasificación de heladas

En la Tabla 3 se muestra los tipos de heladas con su respectiva descripción.

Tabla 3. Tipos de heladas [38].

Tipo de helada	Descripción
Heladas blancas	Ocurren en condiciones de mucho frío y humedad en el ambiente, formando escarcha blanca sobre las plantaciones agrícolas. Temperatura: menores a 2°C. Humedad: 80%-100% Velocidad del viento: inferior a 1.3 m/s
Heladas Negras	Son especialmente perjudiciales, con baja humedad del aire que impide la formación de escarcha, permitiendo que el frío dañe directamente las plantas. Temperatura: menores a -2°C. Humedad: 70% Velocidad del viento: menor a 2.7m/s
Heladas por advección	Se producen cuando grandes masas de aire frío se desplazan, generando vientos fuertes y destruyendo los cultivos Temperatura: menores a -5°C. Humedad: 70%-90% Velocidad del viento: 2.7m/s – 4.1m/s
Heladas por Radiación	Originadas por el enfriamiento del suelo durante la noche debido a la pérdida de calor hacia la atmósfera, pueden ser más frecuentes en valles y requieren condiciones específicas para su formación. Temperatura: menores a -3°C. Humedad: 70%-80% Velocidad del viento: inferiores a 1.3m/s
Heladas por Evaporación	Se producen cuando la evaporación del rocío de la mañana lleva a una notable disminución de la temperatura de las plantas. Temperatura: 0°C a -2°C Humedad: 70%-75% Velocidad del viento: inferior a 2.7m/s

c. Mitigación de heladas

En la Tabla 4 se muestra los métodos para mitigar las heladas con su respectiva descripción.

Tabla 4. Métodos para mitigar las heladas[39].

Método de Mitigación	Descripción
Riego por aspersión	Consiste en aplicar agua sobre las plantas mediante aspersores, lo que libera calor latente durante la fase de congelación, manteniendo así la temperatura de las plantas por encima del punto de congelación, lo que ayuda a protegerlas de las heladas.
Uso de coberturas	Implica el empleo de coberturas como mantas térmicas, plásticos o túneles de cultivo para proporcionar una barrera física que proteja las plantas de las bajas temperaturas y reduzca la pérdida de calor radiante, contribuyendo así a su protección durante las heladas.
Aplicación de agroquímicos	Se refiere al uso de agroquímicos como anticongelantes y protectores de tejidos vegetales, los cuales pueden ayudar a reducir los daños causados por las heladas al mejorar la resistencia de las plantas al estrés térmico, protegiéndolas durante eventos de frío extremo.
Fuentes de calor	Elevar la temperatura utilizando pequeñas fogatas dentro de toneles, con un enfoque centrado en el humo en lugar del fuego.

d. Consecuencias de las heladas

En la Tabla 5 se muestra los efectos de las heladas con su respectiva descripción.

Tabla 5. Efectos de las heladas[39].

Efectos	Descripción
Internos	Las heladas pueden causar daños internos en las plantas, como la ruptura de membranas celulares debido a la formación de hielo, lo que provoca deshidratación.
Externos	Externamente, las heladas pueden resultar en la muerte de hojas, tallos, flores y frutos pequeños,

Efectos	Descripción
	e incluso la muerte total de la planta. La resistencia de la planta varía según su etapa de desarrollo.
Inmediatos	Las heladas pueden tener efectos inmediatos como deshidratación y daños en las membranas celulares.
Acumulativos	A largo plazo, las heladas repetidas pueden provocar deshidratación prolongada y, en algunos casos, intoxicación por sales minerales en las plantas.

En la Figura 2 se muestra las consecuencias que tienen una helada en las plantas de papas ocasionando graves pérdidas económicas para los agricultores.



Figura 2. Consecuencias de las heladas en el cultivo de papas- Imagen generada con Chat GPT 20 de marzo de 2024

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un prototipo de aplicación móvil para control y monitoreo de temperatura, humedad y velocidad del viento para la protección de los cultivos de papas contra las heladas, mediante el uso de tecnologías renovables en la zona alta del cantón Quero.

1.4.2 Objetivos específicos

- Determinar las principales características de las tecnologías inalámbricas de transmisión de datos de sensores climáticos usadas en condiciones de heladas en los cultivos de papa.
- Elaborar un prototipo electrónico de monitoreo y control para medición de temperatura, humedad y velocidad del viento que afectan a los cultivos de papa en las heladas.
- Diseñar una aplicación móvil para gestión de información del prototipo electrónico de monitoreo y control de temperatura, humedad y velocidad del viento que afectan a los cultivos de papas en las heladas.

CAPÍTULO II. METODOLOGÍA

Proporciona el marco estructural necesario para abordar y resolver preguntas de investigación específicas. Esta fase se presenta la planificación y ejecución del estudio, delineando los materiales, métodos y técnicas que serán empleados para recopilar y analizar datos de manera sistemática. Está compuesta por: *Materiales, Modalidad de investigación, Población y muestra y Recolección de información.*

2.1 Materiales

Los recursos materiales se refieren a elementos tangibles y físicos esenciales para la realización de la investigación o proyecto. Estos pueden comprender equipos, instrumentos, dispositivos tecnológicos, muestras, materiales de laboratorio, libros, documentos impresos, entre otros.

- Placas de desarrollo para microcontroladores.
- Sensores de temperatura.
- Sensores de humedad del ambiente.
- Sensores de velocidad del viento.
- Módulos de comunicación inalámbrica.
- Paneles solares.
- Componentes electrónicos.
- Carcasa o caja para proteger el hardware de las condiciones climáticas.
- Aspersores.
- Celular.
- Herramientas de desarrollo de software (IDE para programar microcontroladores).

- Lenguajes de programación.
- Plataformas de desarrollo de aplicaciones móviles (por ejemplo, Android Studio, Xcode).
- Base de datos para almacenar los datos recopilados por los sensores.
- Laptop.
- Internet.
- Materiales de oficina.

2.2 Métodos

2.2.1 Modalidad de la investigación

La investigación propuesta adoptará un enfoque cuantitativo. La naturaleza cuantitativa de la investigación se evidencia en la implementación de la Investigación Aplicada, que se centrará en el monitoreo y control de heladas para mitigar los daños causados en los cultivos de los pequeños productores de papas en la zona alta del cantón Quero. Este enfoque cuantitativo se refuerza con la Investigación de Campo que implica la recopilación de variables climatológicas tangibles como temperatura, velocidad del viento y humedad. Además, la Investigación Experimental que busca realizar pruebas del sistema de monitoreo para verificar la precisión de los datos recopilados, también se alinea con una metodología cuantitativa al buscar obtener resultados medibles y cuantificables.

2.2.2 Población y muestra

La fórmula para determinar el número mínimo de muestras (n) en una investigación está dada por:

$$n = \frac{N * Z^2 * p * q}{(N - 1) * e^2 + Z^2 * p * q} \quad (1)$$

Donde:

n: Tamaño de la muestra.

N: Población

Z: Nivel de confianza.

p: Probabilidad de éxito.

q: Probabilidad en contra, es decir, $q = 1 - p$.

e: Margen de error permitido en la estimación.

Nivel de Confianza (Z) de 1.65: Se ha elegido un nivel de confianza del 90% para que las estimaciones sean representativas de la población, evitando un tamaño de muestra excesivo y costoso, considerando que la población objetivo está compuesta por agricultores con poca familiaridad con la tecnología.

Probabilidad de Éxito (p) y Probabilidad de Fracaso (q) de 0.5: Este valor se elige para reflejar la incertidumbre respecto a la proporción de la población que podría estar abierta o reacia al uso de tecnología en sus prácticas agrícolas.

Error de Estimación Máximo Aceptado (e) de 0.1: Se estableció un margen de error del 10%, balanceando la precisión de las estimaciones con la factibilidad de obtener una muestra adecuada, considerando las limitaciones de tiempo y recursos en una población rural.

$$n = \frac{200 * 1.65^2 * 0.5 * (1 - 0.5)}{(200 - 1) * 0.1^2 + 1.65^2 * 0.5 * (1 - 0.5)} \quad (2)$$

$$n = 50.97$$

$$n = 51$$

Al resolver la ecuación 2 se obtiene como resultado 51 agricultores para realizar la encuesta. La población de 200 agricultores se obtuvo del censo 2020 de INEC[40].

Adicionalmente se realiza entrevistas a tres ingenieros agrónomos expertos en el tema de las heladas para la delimitación del proyecto.

2.2.3 Recolección de información

La recolección de información se realizó a través de encuestas a los agricultores dedicados a la siembra de papas la cual consta de 11 preguntas las cuales fueron validadas por tres ingenieros agrónomos que tiene experiencia en las heladas de los cultivos. Los resultados se detalla a continuación los resultados:

- ¿Qué indicadores o señales considera más importantes para determinar cuándo se va a producir una helada?

Tabla 6. Resultados de la encuesta-Pregunta 1

Indicador	Frecuencia(personas)	Porcentaje
Viento nulo, temperatura baja, ambiente húmedo.	35	68.6%
Pronósticos meteorológicos	16	31.4%

Indicador	Frecuencia(personas)	Porcentaje
Otros	0	0%
Subtotal	51	100%

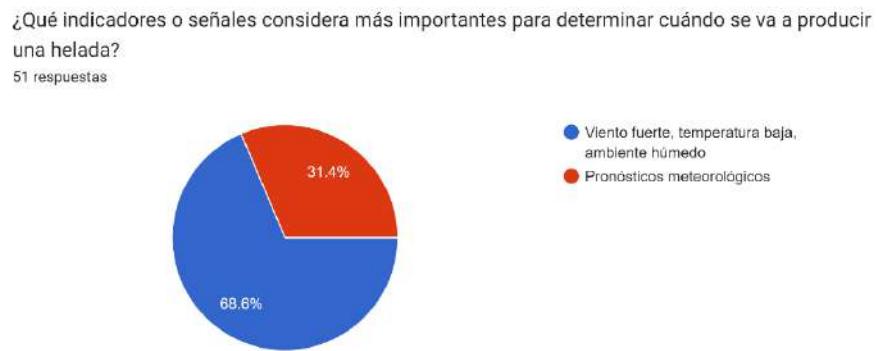


Figura 3. Resultados de la encuesta-Pregunta 1

Análisis e interpretación de los resultados

La Figura 3 muestra que el 68.6% de la población encuestada identifica los tres indicadores como son temperatura, humedad y velocidad del viento para poder predecir una helada, mientras que el 31.4 % se basa en los pronósticos meteorológicos.

En base a estos resultados, se puede proporcionar que existe una gran cantidad de agricultores que consideran importante tomar en cuenta datos del ambiente para detectar una helada.

- ¿Con qué frecuencia experimenta heladas en su área de cultivo de papas durante la temporada de crecimiento?

Tabla 7. Resultados de la encuesta-Pregunta 2

Indicador	Frecuencia(personas)	Porcentaje
Rara vez	23	45.1%
Frecuentemente	28	54.9%
Siempre	0	0%
Subtotal	51	100%

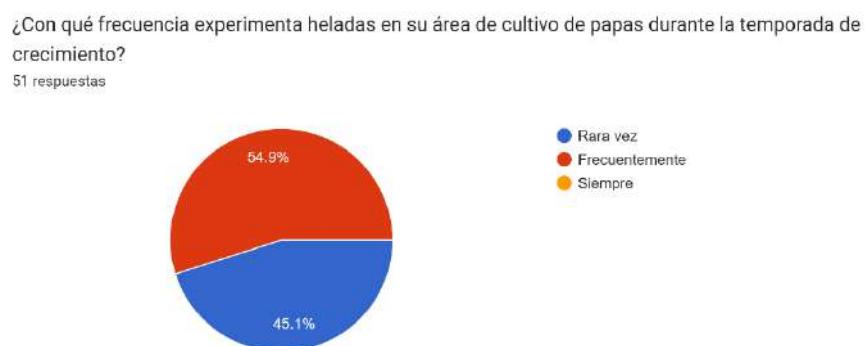


Figura 4. Resultados de la encuesta-Pregunta 2

Análisis e interpretación de los resultados

La Figura 4 muestra que 54.9 % de la población encuestada proporciona información sobre la frecuencia de las heladas siente este frecuentemente por los cambios climáticos constantes del planeta, por otro lado, el 45.1% menciona que rara vez se produce este fenómeno ambiental dado que no tienen un registro de cada acontecimiento.

En base a estos resultados, los agricultores mencionan que las heladas se producen con mayor frecuencia durante el año.

- ¿Qué tan significativas son las pérdidas económicas causadas por las heladas en su cultivo de papas?

Tabla 8. Resultados de la encuesta-Pregunta 3

Indicador	Frecuencia(personas)	Porcentaje
Mínimas	12	23.5%
Moderadas	20	39.2%
Considerables	12	23.5%

¿Qué tan significativas son las pérdidas económicas causadas por las heladas en su cultivo de papas?
51 respuestas

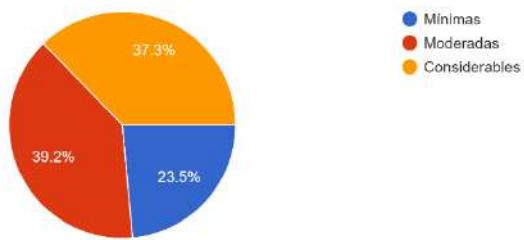


Figura 5. Resultados de la encuesta-Pregunta 3

Análisis e interpretación de los resultados

La Figura 5 muestra que 39.2% de la población encuestada ha tenido una pérdida moderada en la inversión de los cultivos de papas, es decir no se ha perdido con totalidad o la parcela no ha sido muy extensa que fue afectada. Mientras que el 37.3% menciona que sus pérdidas han sido considerables esto puede ocurrir cuando los daños han sido extremos e incluso totales y el 23.5% sus afectaciones han sido mínimas esto se debe al relieve del cultivo.

En base a estos resultados, se puede proporcionar que las afectaciones de las heladas son variadas y existen muchos factores que inciden o perjudican a los cultivos.

- ¿Qué métodos utilizan actualmente para proteger sus cultivos de papa contra las heladas?

Tabla 9. Resultados de la encuesta-Pregunta 4

Indicador	Frecuencia(personas)	Porcentaje
Ninguno	28	54.9%
Humo	17	33.3%
Recipientes de agua en los cultivos	6	11.8%
Subtotal	51	100%

¿Qué métodos utilizan actualmente para proteger sus cultivos de papa contra las heladas?
51 respuestas

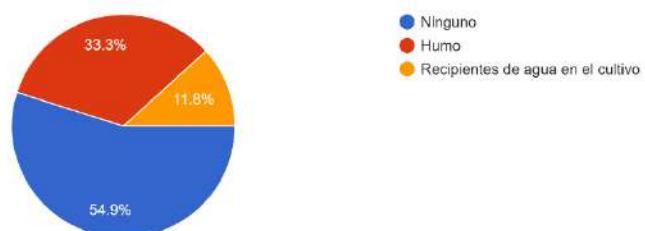


Figura 6. Resultados de la encuesta-Pregunta 4

Análisis e interpretación de los resultados

La Figura 6 muestra que 54.9 % de la población encuestada no utiliza ningún método de protección para sus cultivos, mientras que el 33.3% utiliza humo es decir quema de los residuos orgánicos alrededor de las parcelas y el 11.8% utiliza recipientes de agua que muchas veces solo es por creencias que ayudan a mitigar el daño de las heladas.

En base a estos resultados, las personas de la zona alta del cantón Quero tiene un gran déficit en cuanto a la mitigación de las heladas causando estragos muy graves.

- ¿Cómo calificaría la efectividad de los métodos actuales para proteger sus cultivos de papas contra las heladas?

Tabla 10. Resultados de la encuesta-Pregunta 5

Indicador	Frecuencia(personas)	Porcentaje
Muy poco efectivas	41	80.4%
Moderadamente efectivas	9	17.6%
Muy efectivas	1	2%
Subtotal	51	100%



Figura 7. Resultados de la encuesta-Pregunta 5

Análisis e interpretación de los resultados

La Figura 7 muestra que 80.4 % de la población encuestada califica como los métodos actuales son muy poco efectivos e incluso nulos y el 17.6% como moderadamente efectivas es decir si les ayuda en un 50% a proteger sus cultivos y 2% asegura que sus cultivos si tienen una protección total.

En base a estos resultados, los agricultores no tienen buenos métodos para sobrellevar las heladas y sus afectaciones a los cultivos.

- ¿En qué dispositivo le gustaría tener la aplicación?

Tabla 11. Resultados de la encuesta-Pregunta 6

Indicador	Frecuencia	Porcentaje
Computadora	12	23.5%
Celular	39	76.5%
Dispositivos portátiles(smartwatches)	0	0%
Subtotal	51	100%

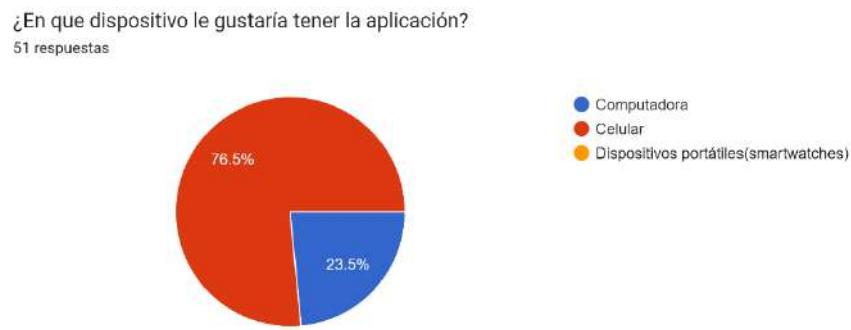


Figura 8.Resultados de la encuesta-Pregunta 6

Análisis e interpretación de los resultados

La Figura 8 muestra que 76.5 % de la población encuestada prefiere que la aplicación sea móvil por su facilidad de implementación en un celular, mientras que el 23.5% prefiere que sea en una computadora.

En base a estos resultados, una aplicación móvil para el monitoreo de las heladas es una gran opción ya que los agricultores pueden alojarla en sus celulares inteligentes.

- Para proteger sus cultivos de papas contra las heladas, ¿Cuáles de las siguientes opciones considera esenciales o útiles?

Tabla 12. Resultados de la encuesta-Pregunta 7

Indicador	Frecuencia(personas)	Porcentaje
Empleo de fuentes de calor adicionales (calentadores, fogatas, etc.).	8	15.7%
Uso de aspersores para rociar agua sobre los cultivos.	35	68.6%
Aplicación de productos químicos protectores.	8	15.7%
Subtotal	51	100%

Para proteger sus cultivos de papas contra las heladas, ¿Cuáles de las siguientes opciones considera esenciales o útiles?

51 respuestas

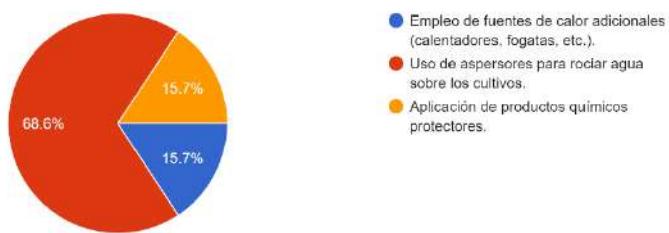


Figura 9. Resultados de la encuesta-Pregunta 7

Análisis e interpretación de los resultados

La Figura 9 muestra que 68.6% de la población encuestada menciona que el uso de aspersores de riego es una gran alternativa para mitigar el daño de las heladas. Y el 15.7% está a favor que la aplicación de productos químicos si ayudara con este fenómeno ambiental, finalmente el 15.7% asegura que con fuentes de calor protegería sus cultivos.

En base a estos resultados, el riego por aspersión tiene una gran ventaja en comparación con las otras opciones ya que ayudará a que los cristales que están formados de la planta se deslicen y no se congelen en las hojas.

- ¿Qué barreras o limitaciones podrían enfrentar al utilizar esta aplicación móvil?

Tabla 13. Resultados de la encuesta-Pregunta 8

Indicador	Frecuencia(personas)	Porcentaje
Costo	33	64.7%
Acceso a internet	9	17.6%
Acceso tecnológico	9	17.6%
Subtotal	51	100%

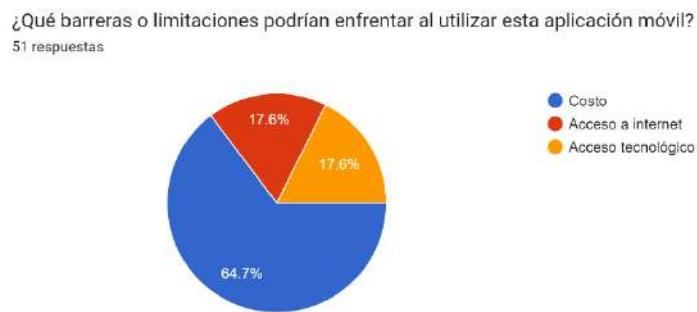


Figura 10. Resultados de la encuesta-Pregunta 8

Análisis e interpretación de los resultados

La Figura 10 muestra que 64.7% de la población encuestada tiene como limitante el costo del sistema, el 17.6% no tiene acceso a internet todo el tiempo y 17.6% no tiene acceso tecnológico.

En base a estos resultados, el costo económico resulta una gran desventaja a la hora de la implantación del sistema sin embargo el acceso a Internet y el tecnológico también juegan un gran rol en este proyecto.

- ¿Cuánto estaría dispuesto a invertir en una solución tecnológica que garantice la protección de sus cultivos de papas contra las heladas?

Tabla 14. Resultados de la encuesta-Pregunta 9

Indicador	Frecuencia(personas)	Porcentaje
\$500	29	56.9%
\$400	12	23.5%
\$300	10	19.6%
Subtotal	51	100%

¿Cuánto estaría dispuesto a invertir en una solución tecnológica que garantice la protección de sus cultivos de papas contra las heladas?
51 respuestas

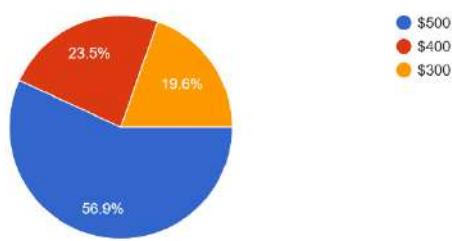


Figura 11. Resultados de la encuesta-Pregunta 9

Análisis e interpretación de los resultados

La Figura 11 muestra que 56.9 % de la población está dispuesta a invertir \$500 en la adquisición del sistema, el 23.5% invertirá \$400 y el 18.6% \$300 siendo este el valor mínimo.

En base a estos resultados, se puede mencionar que existe una gran cantidad de agricultores que está dispuesto a invertir \$500 para proteger sus cultivos de papas de las heladas.

- ¿Cree usted que el uso de energías renovables es importante en las soluciones tecnológicas para la protección contra heladas en sus cultivos de papas?

Tabla 15. Resultados de la encuesta-Pregunta 10

Indicador	Frecuencia(personas)	Porcentaje
No es importante	10	19.6%
Neutral	28	54.9%
Muy importante	13	25.5%
Subtotal	51	100%

¿Cree usted que el uso de energías renovables es importante en las soluciones tecnológicas para la protección contra heladas en sus cultivos de papas?
51 respuestas

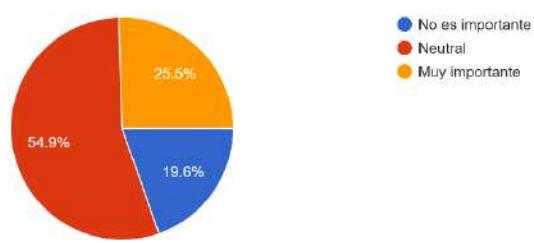


Figura 12. Resultados de la encuesta-Pregunta 10

Análisis e interpretación de los resultados

La Figura 12 muestra que 54.9% de la población encuestada identifica como neutral que el sistema utilice energía renovable, el 25.5% como muy importante que sea autosuficiente y no depende de energía eléctrica o fósil y el 19.6% como no importante.

En base a estos resultados, se puede mencionar que los agricultores están en una posición neutral en la implementación de energías renovables en el sistema.

- ¿Qué tipo de tecnologías renovables consideraría más prácticas o accesibles para implementar en sus campos agrícolas?

Tabla 16. Resultados de la encuesta-Pregunta 11

Indicador	Frecuencia(personas)	Porcentaje
Paneles solares	37	72.5%
Energía eólica (viento)	8	15.7%
Biogás (materiales orgánicos)	6	11.8%
Subtotal	51	100%

¿Qué tipo de tecnologías renovables consideraría más prácticas o accesibles para implementar en sus campos agrícolas?

51 respuestas

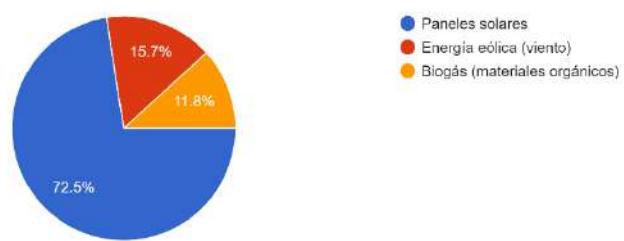


Figura 13. Resultados de la encuesta-Pregunta 11

Análisis e interpretación de los resultados

La Figura 13 muestra que 72.5% de la población encuestada prefiere utilizar paneles solares en la implementación del sistema y 15.7% energía eólica además que el 11.8% biogás.

En base a estos resultados, la presencia de paneles solares es una gran opción para que el sistema sea autosuficiente en cuanto a alimentación energética.

A continuación, las entrevistas realizadas a los expertos en el área de la agricultura.

Entrevista realizada al Ing. Ángel Beltrán

Pregunta	Respuesta
¿Cuáles son los factores determinantes de una helada en la agricultura?	El principal factor es la temperatura ya que con su descenso se congela el rocío y hace que sufra daños en las plantas, además influye mucho la humedad, el viento y la sanidad de la planta.
¿Cuáles son los umbrales de temperatura, humedad y velocidad del viento que indican la posibilidad de una helada?	En la temperatura del ambiente inferiores a los 2 °C, humedad 70%-75% y velocidad del viento generalmente fuertes mayores a 15 km/h (4.1 m/s).
¿Qué métodos existen para proteger los cultivos de papas contra los daños causados por las heladas?	El más conocido es el por riego a aspersión, pero usando micro aspersores ya que ayudan a que el agua se disperse en forma de nubosidad dando una capa protectora a las hojas de la planta.
¿Cuál es la cantidad óptima de agua por metro cuadrado para el riego, considerando el ciclo de crecimiento de la planta y evitando enfermedades derivadas del exceso de agua?	Con la aplicación de micro aspersores la cantidad de agua es mínima lo que no causa problemas por el exceso de agua.
¿Cuál es la duración óptima del riego, ya sea periódico o continuo, para proteger las plantas durante las heladas?	El riego debe ser continuo donde los parámetros de los sensores estén en condiciones de heladas, generalmente estas ocurren entre las 3 am y las 5 am.
¿Es necesario calentar el agua que se va a regar para aumentar la temperatura?	No, ya que el agua del ambiente está a un promedio de 6 °C a 7°C y el agua del rocío de la planta está cercano a 0°C por lo cual es suficiente regar con el agua de un reservorio.

Entrevista realizada al Ing. Claudio Silva

Pregunta	Respuesta
¿Cuáles son los factores determinantes de una helada en la agricultura?	Los factores que inciden en las heladas son las bajas temperaturas, la falta de aeración es decir la ausencia de viento y la humedad alta.
¿Cuáles son los umbrales de temperatura, humedad y velocidad del viento que indican la posibilidad de una helada?	Temperatura: Bajo 0°C, la humedad 80% y el viento inferior a 15 km/h (4.1m/s).
¿Qué métodos existen para proteger los cultivos de papas contra los daños causados por las heladas?	Quema de residuos y generación de humo y riego por aspersión.
¿Cuál es la cantidad óptima de agua por metro cuadrado para el riego, considerando el ciclo de crecimiento de la planta y evitando enfermedades derivadas del exceso de agua?	En metro cuadrado el riego optimo sería un litro, pero existen aspersores que ayudan a minimizar el riego del agua en los cultivos.
¿Cuál es la duración óptima del riego, ya sea periódico o continuo, para proteger las plantas durante las heladas?	La hora que generalmente ocurren las heladas son de 3am a 5 am y el riego debe durar todo el periodo de la helada.
¿Es necesario calentar el agua que se va a regar para aumentar la temperatura?	No ya que puede ocasionar enfermedades a la planta.

Entrevista realizada al Ing. Robinson Moreta

Pregunta	Respuesta
¿Cuáles son los factores determinantes de una helada en la agricultura?	El viento es muy importante para que no se produzca las heladas ya que ayuda a que la humedad del ambiente no se congele en un lugar específico y si no existe se produce daños a las plantas además la temperatura baja juega un papel muy importante.
¿Cuáles son los umbrales de temperatura, humedad y velocidad del viento que indican la	En la temperatura inferiores a 2 °C, en la humedad superiores al 70% y los vientos casi

Pregunta	Respuesta
posibilidad de una helada?	nulos.
¿Qué métodos existen para proteger los cultivos de papas contra los daños causados por las heladas?	Producir humo, regar agua, ubicar botellas de agua a lo largo de todo el cultivo además de la aplicación de productos agroquímicos.
¿Cuál es la cantidad óptima de agua por metro cuadrado para el riego, considerando el ciclo de crecimiento de la planta y evitando enfermedades derivadas del exceso de agua?	Un aproximado de 600 ml durante todo el ciclo de cultivo ya que es una zona con alta humedad del suelo.
¿Cuál es la duración óptima del riego, ya sea periódico o continuo, para proteger las plantas durante las heladas?	Debe regarse durante todo el periodo de la helada ya que con el agua tiende a crearse un microclima que evita el descenso de la temperatura.
¿Es necesario calentar el agua que se va a regar para aumentar la temperatura?	No ya que el agua del ambiente está a más grados de la que cae en el rocío.

Análisis e interpretación de los resultados

Con la entrevista a los expertos se puede evidenciar los factores principales para la ocurrencia de heladas en la agricultura son la temperatura, humedad, el viento y la sanidad de la planta. Específicamente, una temperatura ambiente inferior a 2 °C, una humedad relativa superior al 70%, y velocidades de viento inferiores a 4.1 m/s son indicadores clave de posibles heladas. Para proteger cultivos como el de papas, se recomienda el uso de riego por microaspersión, el cual distribuye el agua en forma de nubosidad, creando una capa protectora en las hojas de las plantas. Este método utiliza una cantidad mínima de agua, reduciendo el riesgo de enfermedades por exceso de humedad. La aplicación del riego debe ser continua durante las condiciones de helada, típicamente entre las 3 am y las 5 am, sin necesidad de calentar el agua, ya que la temperatura del agua de los reservorios es suficiente para contrarrestar el efecto del rocío congelado en las plantas.

2.2.4 Procesamiento y análisis de datos

De acuerdo con las encuestas realizadas se concluyó que:

- Las heladas ejercen un impacto significativo tanto en la salud de los cultivos de papas como en la economía de los agricultores. La vulnerabilidad de los cultivos a las heladas resalta la importancia de implementar estrategias efectivas de protección para minimizar las pérdidas y preservar el sustento económico de los agricultores.
- La elección de una aplicación móvil se justifica por la creciente preferencia de los usuarios por la comodidad y accesibilidad que brinda el uso de dispositivos móviles. Al ser una herramienta que permite acceder a la información y funcionalidades desde cualquier lugar y en cualquier momento, una aplicación móvil facilitaría su adopción y uso por parte de los agricultores, quienes pueden gestionar sus cultivos de manera más conveniente a través de sus dispositivos celulares.
- El riego por aspersión es una opción favorable debido a su eficacia en la protección contra heladas. Su capacidad para distribuir el agua uniformemente sobre los cultivos y su relativo bajo costo de implementación lo convierten en una alternativa para mitigar los daños causados por las heladas en los campos de cultivo de papas.
- La elección de paneles solares como fuente de energía se sustenta en su viabilidad para proporcionar energía sostenible y renovable en áreas rurales donde los cultivos de papas pueden no estar cercanos a la infraestructura eléctrica convencional. Al ser una fuente de energía autónoma y amigable con el medio ambiente, los paneles solares ofrecen una solución energética eficiente y adaptable para alimentar al sistema y otras herramientas tecnológicas utilizadas en la protección de los cultivos contra las heladas.
- Como una idea inicial se quería implementar un sistema de calentamiento del agua para el riego, pero dado la entrevista con los ingenieros agrónomos no se optó por esa estrategia. Además, se concluye que el riego debe ser

continuo, es decir lo que dure la helada por lo cual el sistema debe estar periódicamente censando el cultivo.

En conclusión, la implementación de una aplicación móvil se presenta como una solución conveniente para los agricultores debido a su fácil acceso. El riego por microaspersión destaca como una opción económica y efectiva contra las heladas, mientras que la implementación de paneles solares ofrece una fuente energética sostenible. La inversión a largo plazo se justifica por su capacidad de proporcionar beneficios continuos, y la importancia de proteger los cultivos de papas frente a las heladas resalta su impacto en la economía agrícola.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de resultados

Al concluir las encuestas realizadas a los usuarios del sistema, la relevancia y utilidad en el ámbito agrícola. A través de las entrevistas, se recopila información sobre las estrategias y métodos que los agricultores emplean actualmente para mitigar los efectos de las heladas en sus cultivos de papa en la zona alta del cantón Quero. Esta retroalimentación directa de los agricultores proporciona una mejor comprensión de las necesidades y desafíos específicos que enfrentan en el campo, lo que orienta el desarrollo del sistema de monitoreo y control de heladas. Posteriormente, se elabora una tabla comparativa que resume todos los componentes esenciales requeridos para la implementación del proyecto, facilitando la evaluación y selección de las tecnologías más apropiadas para garantizar la eficacia del sistema. Además, en la sección de metodología de desarrollo se identifican los módulos del sistema y se describe su aplicación en un ambiente de pruebas que simula condiciones extremas de baja temperatura, asegurando así la funcionalidad del sistema en situaciones reales de campo en la zona alta del cantón Quero.

3.1.1 Proceso actual de la protección de los cultivos contra las heladas

Para analizar el proceso que los agricultores realizan en los cultivos de papa cuando se produce una helada, se optó por un diagrama de flujo que representa la secuencia de las acciones que se lleva a cabo.

1. El agricultor debe estar pendiente de acuerdo con sus conocimientos cuando se va a producir una helada.
2. En caso de que exista una helada el agricultor se dirige al cultivo para aplicar el método para mitigar la helada.

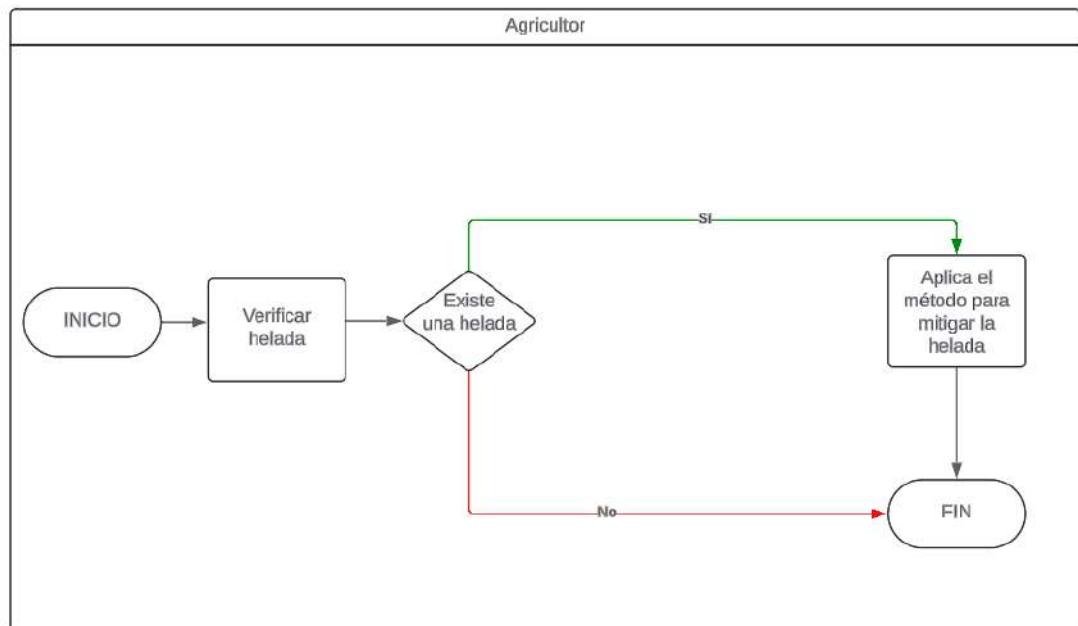


Figura 14. Diagrama del proceso actual de los agricultores con el riego.

2.1.2 Proceso con la aplicación móvil para la protección de los cultivos.

En la forma convencional presenta varios inconvenientes primero la detección de una helada está basada en los conocimientos del agricultor sin un análisis con datos reales de las condiciones climáticas, segundo el método de mitigación se hace de manera manual el agricultor acercándose al cultivo sin importar la hora. Por ello el uso del sistema existen dos modos el automático y el que requiere intervención del agricultor.

Automático

1. Captura los datos de los sensores.
2. Se verifica el estado del riego.
 - a. Si esta encendido se compara los datos de los sensores con los parámetros de apagado.
 - Verifica si cumple con los parámetros de apagado.
 - Si cumple se apaga el sistema de riego.
 - Si los datos no cumplen con los parámetros de apagado, se vuelve a capturar los datos de los sensores y el proceso se repite.

- b. Si el sistema de riego está apagado, se procede a comparar los datos de los sensores con los parámetros de encendido.
- Verifica si cumple con los parámetros de encendido.
 - Si cumple se enciende el sistema de riego y se vuelve a capturar los datos de los sensores y el proceso se repite.
 - Si los datos no cumplen con los parámetros de encendido, se vuelve a capturar los datos de los sensores y el proceso se repite.

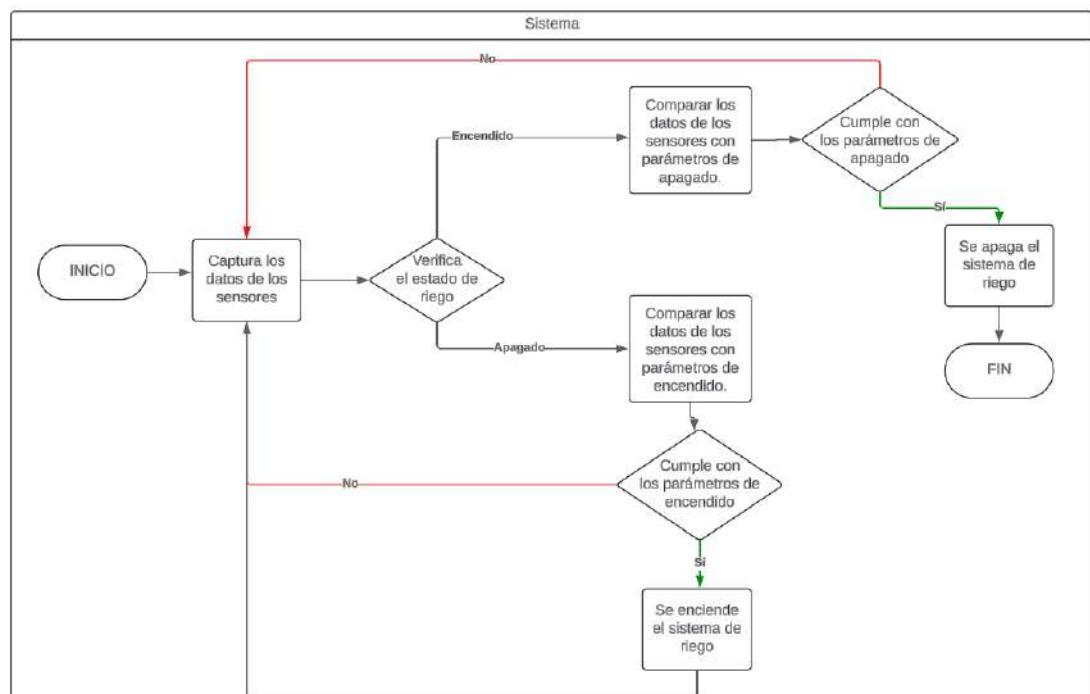


Figura 15. Diagrama del proceso con la aplicación para el riego automático

Manual

1. El usuario tiene el control sobre cuando activar el riego desde la pantalla principal de la aplicación.
2. Se establece un tiempo de riego en minutos en la aplicación.
3. Verifica si cumple el tiempo del riego programado.
 - a. Si cumple activa el apagado automáticamente el riego.
 - b. Caso contrario verifica si se activó el apagado manual.
 - Si se activó se apaga el sistema de riego.
 - Caso contrario se verifica el tiempo del riego.
4. Se apaga el sistema de riego.

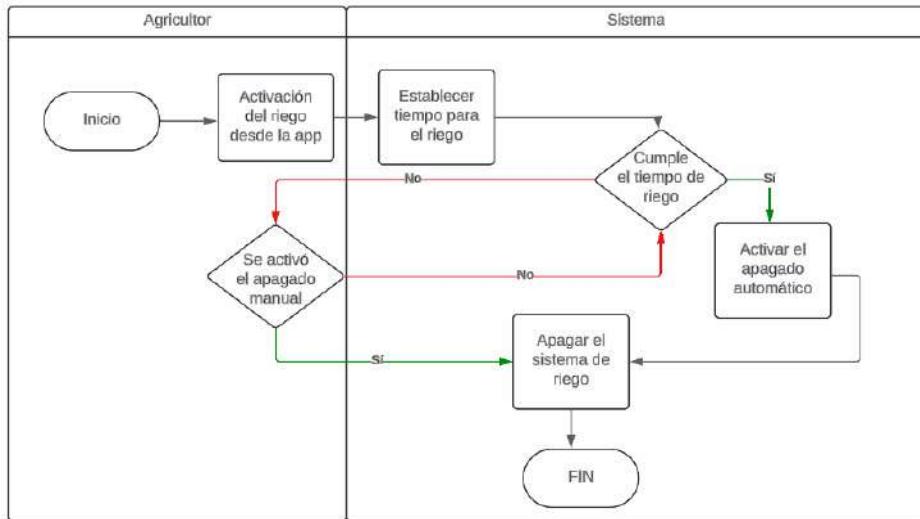


Figura 16. Diagrama del proceso con la intervención del usuario

2.1.3 Análisis y selección de la metodología de desarrollo de software

La metodología ágil ha revolucionado la forma en que se desarrolla el software, brindando un enfoque flexible y colaborativo que prioriza la entrega temprana y continua de valor al cliente. Su importancia radica en su capacidad para adaptarse rápidamente a los cambios, mitigar riesgos y maximizar la satisfacción del cliente. Al promover la iteración continua, la retroalimentación constante y la colaboración estrecha entre los miembros del equipo, las metodologías ágiles permiten a los equipos de desarrollo enfrentar los desafíos en un entorno altamente dinámico y competitivo.

Tabla 17. Metodologías ágiles[41]

Característica	KANBAN	RAD (Rapid Application Development)	XP (Extreme Programming)
Enfoque Principal	Gestión visual del flujo de trabajo y mejora continua	Desarrollo rápido y flexible de aplicaciones	Mejora continua y calidad del código, centrada en las prácticas de desarrollo ágil
Tipo de Metodología	Ágil, Lean	Ágil, Iterativa	Ágil, Iterativa
Principales Elementos	Tablero Kanban, tarjetas, límites de trabajo en curso (WIP)	Prototipado, iteraciones cortas, revisiones frecuentes	Programación en parejas, pruebas continuas, revisiones de código
Curva de Aprendizaje	Baja, fácil de implementar y entender	Moderada, requiere habilidad en prototipado y	Alta, requiere disciplina en las prácticas de desarrollo

Característica	KANBAN	RAD (Rapid Application Development)	XP (Extreme Programming)
		adaptabilidad	
Control de Progreso	Visual, mediante el tablero Kanban	Basado en prototipos y entregas rápidas	Mediante iteraciones cortas y retroalimentación continua
Frecuencia de Entregas	Continua, según el flujo del trabajo	Frecuente, con entregas rápidas de prototipos	Muy frecuente, con iteraciones cortas (semanales o bisemanales)
Interacción con el Cliente	Moderada, centrada en la gestión del flujo de trabajo	Alta, con revisiones y retroalimentación frecuente	Muy alta, cliente involucrado en todo el proceso
Documentación	Ligera, centrada en tarjetas y flujo de trabajo	Ligera, centrada en prototipos y entregas	Ligera, centrada en las prácticas y pruebas continuas
Herramientas Comunes	Tableros Kanban físicos o digitales (Trello, Jira)	Herramientas de prototipado rápido, CASE tools	Herramientas de gestión de versiones, entornos de desarrollo integrados (IDEs)
Principal Ventaja	Proporciona una visualización clara del flujo de trabajo y ayuda a identificar cuellos de botella.	Desarrollo rápido y flexible de aplicaciones, adaptándose bien a cambios en los requerimientos.	Enfocado en la calidad del código y la mejora continua, con prácticas sólidas de desarrollo ágil como programación en parejas y pruebas continuas.
Principal Desventaja	No proporcionar suficiente estructura para equipos que necesitan orientación más detallada en el proceso de desarrollo	Requiere habilidad en prototipado y puede ser menos efectivo para proyectos muy grandes y complejos	Requiere un alto nivel de disciplina y compromiso de todo el equipo, lo que puede ser difícil de mantener a largo plazo.

Al investigar las metodologías de la Tabla 17 se optó por la metodología RAD (Desarrollo Rápido de Aplicaciones) debido a su capacidad para manejar requisitos cambiantes de manera iterativa e incremental. RAD permite una planificación organizada y entrega gradual de software, lo que minimiza errores y garantiza la calidad del producto final.

Las fases de la metodología RAD son las siguientes:

- Planificación: en esta fase se identifican los objetivos del sistema, así como los requisitos.
- Diseño: Se crean prototipos rápidos de las funcionalidades que tendrá el sistema, permitiendo refinar el producto.

- Construcción: En esta fase se empieza a desarrollar el software de manera iterativa, entregando el sistema en períodos cortos y frecuentes. Cada iteración incluye la construcción de nuevas funcionalidades y la mejora de las existentes, basándose en la retroalimentación recibida de los prototipos
- Pruebas: Con el usuario se realiza las respectivas pruebas que validen que el sistema ha llegado a su objetivo.
- Implantación: Una vez finalizadas las fases anteriores, se procede a implantar el sistema en el entorno de producción del cliente. Esta fase incluye la configuración final y la transición del sistema al uso diario[42].



Figura 17. Fases RAD

2.1.4 Análisis y selección de herramienta de gestión de proyectos

El uso de herramientas de gestión de proyectos en el desarrollo de software es necesario para garantizar el éxito y la eficiencia en la ejecución de los proyectos. Estas herramientas proporcionan una estructura organizativa que facilita la planificación, coordinación y seguimiento de todas las actividades relacionadas con el desarrollo del software. Permiten a los equipos gestionar recursos, asignar tareas, establecer plazos y realizar un seguimiento del progreso de manera efectiva.

Tabla 18. Herramientas de gestión de proyectos

Características	ASANA	JIRA	TRELLO
Desarrollado por	Asana, Inc.	Atlassian	Atlassian
Tipo de Herramienta	Gestión de proyectos y tareas	Gestión de proyectos y seguimiento de incidencias	Gestión de proyectos visual, estilo Kanban
Interfaz de Usuario	Intuitiva, basada en tareas y proyectos	Completa, con soporte para metodologías ágiles (Scrum, Kanban)	Sencilla, basada en tableros Kanban
Principales	Tareas, subtareas,	Gestión de	Tableros, listas,

Características	ASANA	JIRA	TRELLO
Funcionalidades	dependencias, hitos, calendarios, cronogramas	incidentes, flujos de trabajo personalizables, informes ágiles	tarjetas, etiquetas
Curva de Aprendizaje	Baja, fácil de usar	Moderada a alta, más compleja	Muy baja, muy fácil de usar
Colaboración en Equipo	Alta, con comentarios, adjuntos y seguimiento de tareas	Alta, con herramientas de comunicación y colaboración integradas	Alta, con comentarios y adjuntos
Integraciones	Slack, Google Drive, Microsoft Teams, Zapier	Bitbucket, Confluence, Slack, GitHub	Slack, Google Drive, Microsoft Teams, Zapier
Personalización	Alta, con proyectos, etiquetas y campos personalizados	Muy alta, con flujos de trabajo y campos personalizados	Moderada, con listas y tarjetas personalizables
Soporte para Metodologías Ágiles	Parcial, con soporte para algunos elementos de Scrum	Completo, diseñado para Scrum y Kanban	Parcial, se puede adaptar con listas y etiquetas
Mejor Uso	Gestión de tareas y proyectos simples a medianos	Seguimiento de proyectos complejos y gestión de incidentes	Gestión de proyectos visuales y colaborativos
Principal Ventaja	Interfaz intuitiva y fácil de usar, ideal para la gestión de tareas y proyectos con características avanzadas como dependencias y cronogramas	Gestión de proyectos ágiles y seguimiento de incidentes, con flujos de trabajo personalizables y herramientas de informes avanzadas.	Interfaz muy simple y fácil de usar basada en tableros Kanban, excelente para la visualización y gestión de proyectos colaborativos
Principal Desventaja	informes y analíticas son básicas en comparación con herramientas más avanzadas[43].	Curva de aprendizaje más pronunciada debido a su complejidad y amplitud de funciones[44].	Capacidades de informes y analíticas limitadas y menor personalización comparado con Asana y Jira[45].

Al revisar las características en la Tabla 18, se decidió utilizar Jira como herramienta de gestión de proyectos es ideal para manejar proyectos complejos y necesitan un seguimiento detallado de incidentes y tareas. Jira ofrece una interfaz completa con soporte integrado para metodologías ágiles, incluyendo RAD, la cual fue seleccionada para este proyecto. Esta herramienta se adapta a las necesidades de gestión de proyectos ágiles, facilitando la implementación de flujos de trabajo iterativos y rápidos, característicos de RAD, y permitiendo la personalización para ajustarse a los requisitos específicos del proyecto.

2.1.5 Análisis y selección de herramienta de gestión de versiones

Permite a los equipos de desarrollo mantener un historial detallado de los cambios realizados, lo que facilita la identificación y corrección de errores, así como la comprensión de la evolución del proyecto. La gestión de versiones promueve la colaboración efectiva entre los miembros del equipo al proporcionar un marco estructurado para trabajar en paralelo en diferentes ramas de desarrollo.

Tabla 19. Herramientas de control de versiones

Características	Git	Mercurial	Apache Subversion
Curva de Aprendizaje	Media (Compleja y varios comandos)	Baja (Simple y potente)	Baja (Más fácil para principiantes)
Seguridad	Es adecuada para usuarios experimentados.	Es adecuada para usuarios no experimentados.	Es adecuada para usuarios inexpertos.
Ramificaciones	Eficaz	Puede causar confusión.	Puede ser muy complicado.
Flujo de Trabajo	Bifurcación y Fusión	Ramificación y Fusión	Centralizado
Trabajo Offline	Sí	Sí	Se encuentra limitado
Control de Acceso	Depende de la plataforma donde se aloja.	Depende de la plataforma de hosting.	Control de acceso detallado.
Velocidad	Alta	Alta	Media
Interfaz Gráfica	Sí (varias)	Sí	Sí
Ventajas	Mantiene un historial de cambios muy completo. Su velocidad y rendimiento son altos.	Sencillez en el manejo. Permite buen manejo de archivos binarios grandes.	De fácil administración. Buen soporte con Windows y perfecta integración con herramientas ágiles. Buen manejo de archivos binarios.
Desventajas	Consumo de espacio de almacenamiento. Complejidad al iniciar por su gran cantidad de funciones.	No trabaja bien con extensiones o complementos que no sean programados con Python. Genera repositorios pesados.	Una verificación local requiere mayor espacio en disco.
Costo	Variable dependiendo de la infraestructura[46].	Variable dependiendo de la infraestructura[47].	Costos iniciales más altos debido a la infraestructura centralizada[48].

Después de revisar la Tabla 19, se decidió emplear git como herramienta de gestión de versiones, dada su amplia gama de interfaces gráficas y su eficacia en el manejo de tareas. Al ser una plataforma distribuida, ofrece al desarrollador mayor flexibilidad para efectuar cambios, lo que facilita un control más preciso sobre el proyecto y sus diversas versiones.

2.1.6 Análisis y selección de herramientas de prototipado

Estas herramientas permiten a los equipos de desarrollo visualizar y comprender mejor las funcionalidades y el flujo de la aplicación antes de comenzar la codificación, lo que reduce significativamente el riesgo de malinterpretaciones entre los miembros del equipo y los interesados. Además, los prototipos proporcionan una forma efectiva de recopilar retroalimentación temprana de los usuarios finales, lo que permite realizar ajustes y mejoras antes de invertir tiempo y recursos en el desarrollo completo del software.

Tabla 20. Herramientas de prototipado.

Características	Figma	Adobe XD	Moqups
Colaboración en Tiempo Real	Sí	Sí	Sí
Prototipado	Avanzado, con interacciones complejas	Avanzado, con auto animación	Básico
Integraciones	Jira, Slack, Zeplin, y más	Adobe Creative Cloud, Slack, Zeplin	Jira, Slack, Google Drive, y más
Accesibilidad	Desde cualquier dispositivo con internet	Principalmente en macOS y Windows	Desde cualquier dispositivo con internet
Formatos soportados	Importación de Sketch, exportación a PDF/PNG/SVG	Importación de Sketch, exportación a PDF/PNG/SVG	Exportación a PDF/PNG/SVG
Historial de Versiones	Sí	No, pero permite deshacer múltiples pasos	No
Costo	Plan gratuito, y planes pagos desde \$12/mes	Plan gratuito, y planes pagos desde \$9.99/mes	Plan gratuito, y planes pagos desde \$7/mes
Plataforma	Web, macOS, Windows, Linux	macOS, Windows	Web
Principal ventaja	Permite que múltiples usuarios trabajen simultáneamente en el mismo archivo	Se integra perfectamente con otras herramientas de Adobe	Ofrece una interfaz intuitiva y es una opción económica

Características	Figma	Adobe XD	Moqups
Principal desventaja	Puede experimentar problemas de rendimiento con archivos muy grandes	Algunas funcionalidades avanzadas y colaborativas están limitadas o no disponibles en la versión gratuita	Las capacidades de prototipado son más limitadas en comparación con Figma y Adobe XD
Documentación y Tutoriales	Extensa, con una comunidad activa[49]	Extensa, con integración en Adobe Learn[50].	Adeuada, pero menos extensa[51].

Al analizar la Tabla 20, se optó por Figma como herramienta de prototipado debido a su versatilidad, facilidad de uso y capacidad para generar prototipos interactivos. Figma ofrece herramientas avanzadas de prototipado y bibliotecas de componentes compartidos, lo que asegura la consistencia y eficiencia en los diseños. Con una comunidad activa y una amplia gama de integraciones.

2.1.7 Análisis y selección de base de datos

Las bases de datos permiten almacenar, organizar y gestionar grandes volúmenes de información de manera estructurada y eficiente. Facilitan el acceso rápido a datos específicos, mejorando la operatividad y la toma de decisiones informadas en diversas organizaciones. Además, posibilitan el análisis de tendencias y la generación de informes detallados.

Tabla 21. Bases de datos

Características	MySQL	MariaDB	PostgreSQL
Desarrollador	Oracle Corporation	MariaDB Foundation	PostgreSQL Global Development Group
Tipo de licencia	Open Source/Comercial	Open Source/Comercial	Open Source
Lenguaje soportado	SQL	SQL	SQL, PL/pgSQL
Plataformas soportadas	Windows, Linux, macOS, FreeBSD, Solaris, etc.	Windows, Linux, macOS, etc.	Windows, Linux, macOS
Modelo de datos	Relacional	Relacional	Relacional
Lenguaje de programación	C, C++, Java, PHP, Python, etc.	C, C++, Java, PHP, Python, etc.	C, C++, Java, PL/pgSQL, Python,
Alta disponibilidad	Mediante configuración de réplicas y clustering	Réplicas y clustering	Réplicas, clustering y streaming replication
Seguridad	Usuarios y privilegios,	Usuarios y privilegios,	Usuarios y roles,

Características	MySQL	MariaDB	PostgreSQL
	encriptación de datos	encriptación de datos	encriptación de datos, políticas avanzadas
Herramientas de gestión	MySQL Workbench, phpMyAdmin, HeidiSQL, etc.	MySQL Workbench, phpMyAdmin	pgAdmin, psql, pgAdmin III
Soporte	Comunidad, Oracle Corporation (para versiones comerciales)[52].	Comunidad, MariaDB Foundation[53].	Comunidad activa[54].

Al analizar la Tabla 21 se decidió elegir a MySQL la principal ventaja es su velocidad y rendimiento. MySQL es conocido por ser un sistema de gestión de bases de datos rápido y eficiente, especialmente en operaciones de lectura. Además, tiene una amplia comunidad de usuarios y desarrolladores, lo que facilita encontrar soporte y recursos. También es compatible con muchas plataformas y lenguajes de programación como: C, C++, Java, PHP, Python.

2.1.8 Análisis y selección de framework de desarrollo

Es un conjunto de herramientas y bibliotecas diseñadas para facilitar la creación de aplicaciones móviles para una o múltiples plataformas (como iOS y Android) a partir de una base de código única. Estos frameworks proporcionan componentes de interfaz de usuario, acceso a las funcionalidades del dispositivo y herramientas de desarrollo que permiten a los desarrolladores escribir, probar y desplegar aplicaciones de manera más eficiente.

Tabla 22. Framework de desarrollo

Características	Ionic	Xamarin	Flutter
Tipo de Herramienta	Framework para aplicaciones híbridas	Framework para aplicaciones nativas	SDK para aplicaciones nativas
Desarrollado por	Ionic Framework Team	Microsoft	Google
Lenguaje Base	HTML, CSS, JavaScript/TypeScript	C#	Dart
Compatibilidad Multiplataforma	iOS, Android, Web	iOS, Android, Windows, macOS	iOS, Android, Web, Desktop
Acceso a Funcionalidades Nativas	Capacitor/Cordova	Integrado directamente con .NET	Integrado directamente
Componentes de UI	Amplia biblioteca de componentes	Controles nativos y personalizados	Widgets personalizados
Curva de Aprendizaje	Moderada, especialmente si conoces web	Moderada, especialmente si	Moderada, especialmente por

Características	Ionic	Xamarin	Flutter
		conoces C#	aprender Dart
Desarrollo de UI	Basado en web, con componentes nativos	Basado en XAML y controles nativos	Basado en widgets altamente personalizables
Licencia	Open Source[55]	Open Source[56].	Open Source[57].

Se eligió para este proyecto Ionic ya que permite desarrollar aplicaciones multiplataforma (iOS, Android, Web) usando tecnologías web familiares como HTML, CSS y JavaScript/TypeScript. Con una amplia biblioteca de componentes de UI optimizados para móviles y acceso a funcionalidades nativas a través de Capacitor/Cordova, Ionic facilita el desarrollo rápido y eficiente. Además, su curva de aprendizaje moderada y su naturaleza de código abierto lo convierten en una opción accesible y flexible, aprovechando la experiencia existente en desarrollo web para construir aplicaciones móviles de alta calidad.

2.1.9 Análisis y selección del framework JavaScript

En el desarrollo de aplicaciones hibridas se puede elegir entre varios framework para la lógica y la arquitectura en este caso se describirá los siguientes: React, Angular y Vue.js. Esta tabla comparativa proporciona una visión detallada de las características, ventajas y desventajas de cada una.

Tabla 23. Framework JavaScript

Característica	React	Angular	Vue.js
Desarrollado por	Facebook	Google	Evan You y la comunidad
Lenguaje Base	JavaScript (opcional TypeScript)	TypeScript	JavaScript (opcional TypeScript)
Curva de Aprendizaje	Moderada	Pronunciada	Suave
Arquitectura	Basada en componentes	MVC (Model-View-Controller)	Basada en componentes
Componentes Reutilizables	Sí, con componentes	Sí, con módulos y componentes	Sí, con componentes
Principal ventaja	Es muy flexible y se puede integrar con una amplia variedad de bibliotecas y herramientas	Es un framework completo que incluye todo lo necesario para desarrollar aplicaciones complejas	Es fácil de integrar en proyectos existentes y tiene una curva de aprendizaje suave
Principal desventaja	Por sí solo no es un	Tiene una curva de	Aún tiene una

Característica	React	Angular	Vue.js
	framework completo y requiere configuración adicional para formar una solución completa	aprendizaje más empinada debido a su complejidad	comunidad y un ecosistema más pequeños en comparación con React y Angular
Licencia	Open Source[58].	Open Source[59].	Open Source[60].

Al analizar la Tabla 23 se optó por React debido a su flexibilidad y capacidad para integrarse con una amplia variedad de bibliotecas y herramientas, lo que facilita la creación de interfaces de usuario personalizadas y dinámicas. React tiene una curva de aprendizaje moderada y se basa en una arquitectura de componentes reutilizables, lo que permite a los desarrolladores construir aplicaciones robustas y escalables.

2.1.10 Análisis y selección del lenguaje de programación backend

Son utilizados para crear la lógica del servidor, gestionar bases de datos y proporcionar la funcionalidad central de aplicaciones web y móviles. Estos lenguajes permiten a los desarrolladores escribir el código que ejecuta en el servidor, maneja las solicitudes de los usuarios, procesa datos y comunica con bases de datos para almacenar y recuperar información.

Tabla 24. Lenguajes backend

Característica	Python	JavaScript	PHP
Desarrollado por	Guido van Rossum	Ryan Dahl	Rasmus Lerdorf
Paradigma	Multiparadigma (orientado a objetos, funcional, etc.)	Event-driven, non-blocking I/O	Multiparadigma (procedural, orientado a objetos, etc.)
Curva de Aprendizaje	Suave, fácil de aprender	Moderada, JavaScript tiene una curva más pronunciada en el backend	Suave, ampliamente conocido y utilizado
Uso Principal	Aplicaciones web, ciencia de datos, automatización	Aplicaciones web, servicios en tiempo real	Aplicaciones web, desarrollo de contenido dinámico
Frameworks Populares	Django, Flask	Express, Koa	Laravel, Symfony
Gestión de Paquetes	pip	npm	Composer
Principal Ventaja	Sintaxis clara y legible, lo que lo hace fácil de aprender y usar, especialmente para principiantes	Utiliza un modelo de I/O no bloqueante y orientado a eventos, lo que lo hace eficiente para manejar múltiples conexiones concurrentes	Simplicidad y efectividad en el desarrollo web

Característica	Python	JavaScript	PHP
Principal Desventaja	Generalmente más lento que los lenguajes compilados como Java o C++.	Curva de aprendizaje más pronunciada	Rendimiento y seguridad
Licencia	Open Source[61].	Open Source[62].	Open Source[63].

En la Tabla 24 se analiza las principales características de Python, JavaScript y PHP. Se eligió Python como lenguaje de programación backend ya que es conocido por su sintaxis clara y legible, lo que facilita su aprendizaje y uso, especialmente para principiantes. Además, Python cuenta con una amplia gama de bibliotecas y frameworks robustos como Django y Flask, que aceleran el desarrollo y simplifican la implementación de aplicaciones complejas. Python es ideal para manejar tareas de automatización y ciencia de datos.

2.1.11 Análisis y selección del framework para el backend

Un framework proporciona una estructura sólida y consistente, facilita el desarrollo rápido y eficiente, y ofrece herramientas y bibliotecas integradas que simplifican tareas comunes como la autenticación, el manejo de bases de datos, y la gestión de rutas. Además, un framework mejora la mantenibilidad y escalabilidad del sistema, permitiendo que el proyecto evolucione y se adapte a nuevas necesidades sin problemas.

Tabla 25. Framework backend

Característica	Django	Flask	Pyramid
Descripción	Un framework web de alto nivel que promueve el desarrollo rápido y un diseño limpio y pragmático.	Un micro framework ligero y flexible para aplicaciones web.	Un framework flexible que se adapta a diferentes necesidades de desarrollo web.
Arquitectura	Basado en una arquitectura MTV (Model-Template-View).	Basado en una arquitectura minimalista y extensible.	Basado en una arquitectura MVC (Model-View-Controller).
Facilidad de uso	Altamente estructurado, ideal para proyectos grandes con muchas características integradas.	Simple y fácil de usar, ideal para proyectos pequeños y medianos.	Flexible y escalable, adecuado para proyectos de cualquier tamaño.
Curva de aprendizaje	Moderada, con una curva de aprendizaje más empinada debido	Baja, fácil de aprender para principiantes.	Moderada, con más flexibilidad y opciones de configuración.

Característica	Django	Flask	Pyramid
	a su complejidad y cantidad de funcionalidades.		
Escalabilidad	Adecuado para grandes aplicaciones empresariales.	Puede requerir más configuración manual para proyectos grandes.	Altamente escalable, adaptable a necesidades cambiantes.
Soporte y documentación	Gran comunidad de usuarios.	Buena documentación y una comunidad activa.	Buena documentación y una comunidad en crecimiento.
Manejo de base de datos	Incluye un ORM poderoso, soporta múltiples bases de datos[64].	No incluye un ORM, pero se puede integrar con SQLAlchemy o similares[65].	Compatible con múltiples ORM, incluyendo SQLAlchemy[66].

Al analizar la Tabla 25 Flask fue el framework elegido ya que sirve especialmente para proyectos pequeños, ligero y fácil de aprender, lo que te permitirá desarrollar de manera rápida y eficiente. Su simplicidad permite construir las funcionalidades básicas sin la complejidad adicional de frameworks más grandes.

2.1.12 Análisis y selección de los microprocesadores

Estas plataformas proporcionan un entorno versátil y flexible para la integración de sensores, actuadores y sistemas de comunicación. Gracias a su capacidad de procesamiento, almacenamiento y conectividad, las tarjetas de desarrollo permiten la implementación de algoritmos de control inteligentes y sistemas de automatización. Además, su capacidad de adaptación a diferentes entornos y necesidades específicas del proyecto facilita la personalización y la escalabilidad de las soluciones implementadas.

Tabla 26.Comparación de los microprocesadores

Características	Arduino Uno	915MHz ESP32 LORA V3	Raspberry Pi 3 modelo B
Imagen			
Voltaje	5v	5V	5V (a través del conector micro USB)
Corriente aceptable	400 mA	400 mA	2.5 A (a través del conector micro USB)
Procesador	ATmega328P	Xtensa Dual-Core 32-bit LX6	Broadcom BCM2837B0, Cortex-

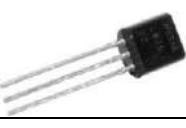
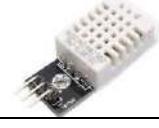
Características	Arduino Uno	915MHz ESP32 LORA V3	Raspberry Pi 3 modelo B
			A53 (ARMv8) 64-bit SoC
Frecuencia del procesador	16 MHz	240 MHz	1.2 GHz
Wifi	No	Si	Si
Interfaces de Comunicación	UART, I2C, SPI	UART, I2C, SPI, Lora	UART, I2C, SPI, HDMI, Ethernet
Digital Pins (I/O Pines	14	26	26
Temperatura de funcionamiento	-40°C a 85°C [67]	-40°C a 85°C [68]	0°C a 50°C[69]
Costo	\$15	\$45	\$145

En la Tabla 26 se realiza una comparación de tres microprocesadores que son posibles candidatos para el uso en el proyecto. Sin embargo, se eligió 915MHz ESP32 LORA V3 por su capacidad para operar en un amplio rango de temperatura de -40°C a 125°C, importante para aplicaciones en entornos extremos como las heladas. Equipada con un procesador dual-core Tensilica LX6 a 240 MHz, 520 KB de SRAM y 4 MB de memoria flash, esta placa ofrece un rendimiento robusto y capacidad de almacenamiento adecuada para tareas complejas. Su tecnología de comunicación Lora permite la transmisión de datos a largas distancias con bajo consumo de energía.

2.1.13 Análisis y selección de los sensores de temperatura

Los sensores de temperatura y humedad desempeñan un papel importante ya que estos sensores proporcionan datos precisos y en tiempo real sobre las condiciones climáticas locales, permitiendo una evaluación detallada de los riesgos de heladas y una respuesta rápida para minimizar los daños en los cultivos.

Tabla 27.Sensores de temperatura

Características	DHT11	LM35	Ky-013
Imagen			
Magnitudes soportadas	Temperatura, Humedad	Temperatura	Temperatura, Humedad

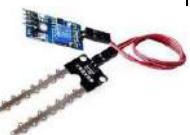
Características	DHT11	LM35	Ky-013
Voltaje de alimentación	3,3V ~ 6V	4V a 30V	5 V
Rango de operación	- 40°C ~ 80°C 0% ~ 100% HR	-10°C a +85°C	-55°C a +125°C
Precisión	±0,5°C ±2% HR	+/- 0.1°C	+/- 0.5°C
Tiempo de respuesta	n/u		
Resolución ADC	16-bit	12 bits	18 bits
Frecuencia de muestreo	0,5 Hz	0,5 Hz	1 Hz
Costo	bajo	medio	alto

Después de analizar las características principales de los sensores en la Tabla 27 se optó por el sensor DHT11 por su bajo costo y facilidad de adquisición. Es capaz de medir tanto la temperatura como la humedad del ambiente, dos variables importantes para monitorear las condiciones climáticas durante las heladas en los cultivos. Su rango de operación va desde -40°C hasta 80°C y de 0% a 100% de humedad relativa, lo hace ideal para adaptarse a diferentes entornos agrícolas.

2.1.14 Análisis y selección de los sensores de humedad del suelo

Estos sensores permiten a los usuarios monitorear el contenido de humedad en el suelo, proporcionando información para la gestión del riego y prevención de enfermedades por exceso de agua.

Tabla 28. Comparación sensores de humedad del suelo

Características	FC-28	YL-69	HD-38
Imagen			
Voltaje de alimentación	3.3V a 5V	3V a 6 V	3.3 a 12V
Rango de operación	20 % a 90 %	0 a 100%	0 a 100%
Precisión	3%	2%	1,50%
Resolución ADC	10 bits	10 bits	10 bits

Características	FC-28	YL-69	HD-38
Temperatura de operación	-40°C a 80°C	-20°C a 70°C	-40°C a 85°C
Aplicaciones comunes	Agricultura, Jardinería	Educativas, Proyectos DIY	Industrial, Investigación
Costo	Bajo	Bajo	Medio

Analizado las características de los sensores de humedad del suelo en la Tabla 28 se optó por el FC-28 debido a su bajo costo, facilidad de uso. Este sensor permite la monitorización en tiempo real de la humedad del suelo. Aunque su rango de operación es de 20% a 90%, es suficiente para aplicaciones agrícolas.

2.1.15 Análisis y selección de los sensores de velocidad del viento

Estos sensores proporcionan datos precisos y oportunos sobre las condiciones atmosféricas, lo que permite a los usuarios anticipar y responder de manera proactiva a la amenaza de las heladas. Estos sensores permiten una monitorización continua de la velocidad del viento en el entorno agrícola.

Tabla 29.Comparación sensores de velocidad del viento.

Características	Renke	JL-FS2	Terrific
Imagen			
Voltaje	10V ~ 30V	9V ~ 24V	No requiere
Rango	0m/s ~ 60m/s	0m/s ~ 30m/s	0m/s ~ 70m/s
Potencia	0,4W	0,3W	n/u
Precisión	± 0,2m/s	± 0,3m/s	± 0,3m/s
Resolución	0,1m/s	0,1m/s	0,1 m/s
Tiempo de respuesta	≥ 0,5s	n/u	n/u
Inicio de detección de viento	≥ 0,1 [48]m/s	≥ 0,4 m/s	≥ 0,4 m/s
Temperatura de trabajo	-20°C ~ 60°C	-40°C ~ 80°C	20°C ~ 60°C
Voltaje de salida	0V ~ 5V	0V ~ 5V	0V ~ 2V
Voltaje de salida	0V ~ 5V	0V ~ 5V	0V ~ 2V
Costo	Alto	Alto	Medio

Al analizar las características de los sensores de velocidad del viento en la Tabla 29, el sensor Terrific fue seleccionado por su amplio rango de medición, que va desde 0m/s hasta 70m/s. Además, su precisión de $\pm 0,3\text{m/s}$ y su resolución de 0,1m/s proporcionan lecturas precisas, lo que permite una respuesta rápida y efectiva ante cambios en las condiciones atmosféricas. La temperatura de trabajo del sensor, que oscila entre -20°C y 60°C, lo hace adecuado para entornos agrícolas variados y cambiantes, asegurando un rendimiento óptimo incluso en condiciones climáticas extremas. También, su bajo consumo de energía y su voltaje de salida de 0V a 2V lo convierten en una buena opción.

2.1.16 Análisis y selección del método de riego

El riego desempeña un papel crucial en la protección de los cultivos de papas durante las heladas al brindar una defensa contra las bajas temperaturas. Durante estos eventos climáticos, el suministro controlado de agua a través del riego puede actuar como un amortiguador térmico, ayudando a mantener una temperatura más estable en el suelo y alrededor de las raíces de las plantas. Este efecto termorregulador del riego puede mitigar los daños causados por las heladas al evitar la congelación del suelo y las raíces, así como al proteger las estructuras celulares de las plantas de los efectos nocivos del frío extremo.

Tabla 30.Comparación del riego

Característica	Aspersión	Goteo
Imagen		
Aplicación	Rociado amplio	Goteo puntual
Eficiencia de agua	Mayor consumo de agua	Menor consumo de agua
Costo de instalación	Bajo	Medio
Costo de mantenimiento	Medio	Medio

Característica	Aspersión	Goteo
Taponamientos	No	Si
Cobertura	Amplio	En cada planta

Tanto el riego por goteo como el riego por aspersión emergen como técnicas para mitigar los efectos de las heladas y preservar la productividad agrícola. El riego por goteo, al aplicar el agua de manera precisa y controlada directamente a las raíces de las plantas, puede ayudar a mantener una temperatura más estable en el suelo y proteger las raíces del frío extremo. Por otro lado, el riego por aspersión, al rociar el agua sobre la superficie del cultivo, puede crear una capa de protección alrededor de las plantas, formando una barrera contra las temperaturas gélidas. Por esta razón se eligió el riego por aspersión para este proyecto.

2.1.17 Análisis y selección de los aspersores

Los aspersores con el riego del agua ayudan a mitigar los daños, pero ahí la forma de aspersión y la cantidad juegan un papel muy importante. Tanto los aspersores agrícolas, con su capacidad para dispersar grandes volúmenes de agua sobre una amplia área, como los micro aspersores, con su capacidad para proporcionar un riego más suave y controlado, son componentes esenciales en la estrategia de protección contra heladas.

Tabla 31. Comparación de tipos de aspersores.

Características	Aspersores agrícolas	Micro aspersores
Imagen		
Tipo de aspersión	En forma de arco o circular.	En forma de nebulización fina.
Uniformidad del riego	Puede variar según la distancia y la presión del agua.	Mayor uniformidad debido a la distribución más fina y controlada del agua.
Caudal	Caudal generalmente más alto.	Caudal más bajo y controlado.
Eficiencia del uso del agua	Menos eficiente debido a posibles pérdidas por evaporación y escurreimiento.	Más eficiente debido a la menor evaporación y mejor penetración en el suelo.
Mantenimiento	Puede requerir un mantenimiento más regular	Requiere menos mantenimiento debido a su

Características	Aspersores agrícolas	Micro aspersores
	debido a la mayor exposición a obstrucciones.	diseño más compacto y menos propenso a obstrucciones.
Durabilidad	Tienden a ser duraderos, pero pueden ser más susceptibles al desgaste debido a su mayor tamaño y peso.	Generalmente duraderos debido a su diseño compacto y menor exposición a daños mecánicos.
Resistencia a obstrucción	Pueden ser más propensos a obstrucciones debido a la mayor área de las boquillas y los filtros.	Menos propensos a obstrucciones debido a las boquillas más pequeñas y los filtros más finos.
Tipo de boquilla	Boquillas grandes que dispersan agua en patrones más amplios.	Boquillas más pequeñas que producen una neblina fina de agua.
Altura de elevación	Altura de elevación variable, generalmente mayor que la de los micro aspersores.	Altura de elevación más baja debido a la nebulización más fina y controlada.
Sistema de drenaje	Puede requerir un sistema de drenaje para evitar acumulación de agua en el terreno.	Menos propenso a requerir sistemas de drenaje debido al menor caudal y mayor eficiencia de penetración en el suelo.

Al analizar la Tabla 31 las principales características de los dos tipos de aspersores se optaron por los micro aspersores los cuales al proporcionar menor caudal de agua evita que la plata se sobre hidrate y provoque enfermedades asociadas, el agua en forma de nebulización fina hace que las hojas de la plata se deslice la escarcha acumulada ayudando a mitigar el daño por el descenso de la temperatura.

2.1.18 Análisis y selección del tipo de bomba de agua

El uso de una bomba de agua en el riego sirve para garantizar una distribución eficiente y uniforme del agua a los cultivos. Una bomba de agua permite que el riego sea más controlado y preciso, asegurando que las plantas reciban la cantidad adecuada de agua según sus necesidades, lo cual es especialmente importante en áreas con suministro de agua irregular o limitado. Además, el uso de bombas facilita la implementación de sistemas de riego modernos, como el riego por aspersión o goteo.

Tabla 32 . Bomba de agua

Características	Bomba Sumergible	Bomba Periférica
Imagen		
Voltaje	110v	110v
Potencia	0.5 HP - 20 HP	0.5 HP - 2 HP
Caudal	Medio	Bajo
Altura máxima	100m	30m
Costo	\$200	\$80-\$100
Mantenimiento	Moderado	Bajo
Aplicaciones	Pozos, desagües, acuicultura	Residencial, sistemas pequeños

A analizar la Tabla 32 la bomba periférica es una opción viable ya que su principal aplicación es en sistemas pequeños debido a su bajo costo y facilidad de instalación. Estas bombas son ideales para tareas que no requieren un caudal alto, como el suministro de agua a una casa o el riego de jardines pequeños. Además, las bombas periféricas son eficientes energéticamente y requieren poco mantenimiento.

2.1.19 Análisis y selección de la comunicación

La protección de los cultivos contra las bajas temperaturas, las soluciones inalámbricas ofrecen una plataforma ágil para monitorear y responder de manera proactiva a las condiciones climáticas cambiantes. Mediante el uso de sensores de temperatura, humedad y velocidad del viento equipados con tecnología inalámbrica, los agricultores pueden obtener datos en tiempo real sobre las condiciones ambientales en sus campos, lo que les permite tomar decisiones informadas y rápidas para proteger sus cultivos contra las heladas. Además, la comunicación inalámbrica facilita la coordinación y la automatización de sistemas de riego y dispositivos de protección, como sistemas de aspersión optimizando así la eficacia de las medidas de mitigación de heladas.

Tabla 33. Tecnologías de comunicación

Características	Wifi	Lora	ZigBee
Frecuencia de trabajo	2.4GHz a 5.4GHz	868MHz a 928MHz	868MHz a 2.4GHz
Velocidad de Tx	150Mbps a 600Mbps	300bps a 5kbps	50kbps
Consumo de energía	Moderado	Bajo	Bajo
Alcance	330m a 820m	2km a 15km	10m a 100m
Protocolo de comunicación	802.11	Lora WAN	Zigbee
Transmisión	Datos, audio, video	Datos	Datos, imágenes
Seguridad	WEP/ WPA2	AES-128	Network keys
Costo	Bajo	Medio	Medio

La elección de la tecnología de Lora, se consideró ya que los terrenos no se encuentran cerca los hogares donde tienen la facilidad del internet por lo cual las distancias son muy amplias. Además, la capacidad de comunicación inalámbrica de largo alcance de estas tecnologías radio frecuencia garantiza una cobertura efectiva incluso en áreas extensas de cultivo.

2.1.20 Análisis y selección del tipo de alimentación

La alimentación energética mediante fuentes renovables es fundamental especialmente en entornos donde el acceso a la energía es limitado o escaso. En un contexto donde los cultivos se encuentran distantes de las zonas habitadas y la infraestructura energética es limitada, las energías renovables, como la solar, la eólica y el biogás, ofrecen una solución sostenible y autónoma para satisfacer las necesidades energéticas requeridas para la protección de los cultivos contra las heladas.

Tabla 34. Energías renovables

Características	Energía solar	Energía eólica	Biogás
Imagen			
Disponibilidad	Alta	Alta	Dependiente de materiales orgánicos

Características	Energía solar	Energía eólica	Biogás
Mantenimiento	Bajo	Moderado	Moderado
Variabilidad	Depende de la ubicación	Depende de la ubicación	Alta
Intermitencia	No	Sí, en función de la disponibilidad de viento	No
Impacto Ambiental	Bajo	Moderado	Dependiente de la fuente de material orgánico
Costos	Inversión inicial baja	Inversión inicial moderada, bajos costos operativos	Inversión inicial alta, bajos costos operativos
Eficiencia	Moderada-alta	Moderada-alta	Variable dependiendo del sistema y la materia prima
Tiempo de Vida Útil	25-30 años	20-25 años	Variable dependiendo del sistema y la materia prima

Al analizar la Tabla 34 se consideró las principales características de tres energías renovables, seleccionó la energía solar como fuente de alimentación ya que el costo y su mantenimiento a lo largo del tiempo es menor. Además, al usar la radiación no es necesario que los días sean soleados.

2.2 Desarrollo de la propuesta

2.2.1 Análisis de las condiciones climáticas

En el cantón Quero, específicamente en la zona alta, donde se cultiva principalmente papa, las heladas representan un desafío significativo para los agricultores. Aunque la altitud promedio en esta área es de aproximadamente 3,000 metros sobre el nivel del mar, lo que brinda condiciones adecuadas para el cultivo de papas, la topografía del terreno influye en la presencia de heladas. Durante los meses más fríos, las temperaturas pueden descender considerablemente, alcanzando valores inferiores a 2°C, lo que puede causar daños graves a los cultivos de papa. La escasez de mecanismos o sistemas de protección contra heladas y un sistema de riego óptimo agravan aún más la situación para los agricultores de la zona alta de Quero.

Para proteger los cultivos de papa contra las heladas, es fundamental establecer umbrales precisos basados en la temperatura, la humedad y la velocidad del viento. Según la investigación y las entrevistas realizadas, se determinó que una temperatura

ambiente inferior a 2 °C es un indicador crítico de posibles heladas. La papa pera, en particular, puede soportar temperaturas hasta 0 °C sin sufrir consecuencias perjudiciales. La humedad relativa también juega un papel crucial; niveles superiores al 70% indican una alta probabilidad de formación de heladas. Por otro lado, la velocidad del viento debe ser cuidadosamente monitoreada. Aunque el viento puede igualar las temperaturas del aire y reducir el riesgo de heladas, ráfagas inferiores a 15 km/h (4.1 m/s), la desactivación del riego no se toma en cuenta ya que puede existir ráfagas inesperadas de viento[70].

Para mitigar estos riesgos, es importante implementar sistemas de monitoreo y alerta temprana que permitan a los agricultores tomar medidas preventivas en caso de condiciones climáticas adversas. La recopilación de datos en tiempo real sobre temperatura, humedad del ambiente y velocidad del viento, junto con la instalación de sistemas de riego adecuados y medidas de protección contra heladas, son aspectos fundamentales para garantizar la sostenibilidad y productividad de los cultivos de papa en la zona alta del cantón Quero.

2.2.2 Fase I: Recopilación de requisitos

En esta fase se reunió con el agricultor donde se va a aplicar el prototipo quien proporciona los requisitos que se espera con la aplicación.

a. Definición de roles

Para el proyecto se contó con 3 roles que se describirán a continuación:

Responsable	Rol	Función
Silvia Villacrés	Desarrollador	Está a cargo de la implementación de la programación en la aplicación móvil, así como en el ensamblaje del prototipo físico de acuerdo con los requerimientos.
Ing. Fernando Ibarra	Verificador	Encargado de dar seguimiento a las actividades del proyecto, así como a la evaluación del progreso de acuerdo con lo planificado.
Leonidas Villacrés	Agricultor/usuario	Solicita los requerimientos para el proyecto además brinda retroalimentación.

b. Requerimientos técnicos

A continuación, se describe las tecnologías que se usan el desarrollo de la aplicación móvil.

Tecnología	Versión	Descripción
React js	18.3.1	Biblioteca de JavaScript para construir interfaces de usuario
Ionic	7.2.0	Framework para el desarrollo de aplicaciones híbridas móviles
MySQL	8.2.0	Sistema de gestión de bases de datos relacional
Python	3.11.9	Lenguaje de programación multiparadigma
Arduino	1.8.18	Entorno de desarrollo integrado para microcontroladores, basado en C/C++

c. Requerimientos funcionales de la aplicación móvil

Para el levantamiento de los requerimientos se utiliza Historias de Usuario las cuales sirven para manejar la información del usuario de manera clara y concisa. A continuación, la plantilla que se aplicará.

Tabla 35. Plantillas de Historias de Usuario[41]

Historia de Usuario	
Número:	Usuario
Nombre:	
Prioridad:	Riesgo:
Puntos estimados:	Responsable:
Descripción:	
Criterios de aceptación:	

Donde:

- **Número:** Un identificador único para la historia de usuario que facilita el seguimiento y la referencia (por ejemplo, US001).
- **Usuario:** El tipo de usuario que se beneficiará de esta funcionalidad (por ejemplo, nuevo usuario, administrador, cliente, etc.).
- **Nombre:** El nombre o título específico de la historia de usuario que describe claramente la funcionalidad (por ejemplo, Registrar Usuario).
- **Prioridad:** La importancia de esta historia de usuario en el contexto del proyecto (por ejemplo, Alta, Media, Baja).

- **Riesgo:** Una evaluación del riesgo asociado con la implementación de esta historia de usuario (por ejemplo, Alto, Medio, Bajo).
- **Puntos estimados:** Una estimación del esfuerzo requerido para completar la historia de usuario, generalmente en puntos de historia.
- **Iteración asignada:** La iteración o sprint en la que se planea trabajar en esta historia de usuario.
- **Responsable:** La persona o equipo responsable de implementar esta historia de usuario.
- **Descripción:** Una narrativa que describe qué necesita el usuario y por qué, estructurada típicamente como: "Como [usuario], quiero [acción], para [beneficio]".
- **Criterios de aceptación:** Una lista de condiciones que deben cumplirse para que la historia de usuario se considere completa. Estos criterios deben ser específicos y verificables.

Para la priorización de las historias de usuario se uso MoSCoW que es una técnica utilizada en la gestión de proyectos y desarrollo de software en entornos ágiles, radica en su capacidad para ayudar a los equipos a centrarse en las características y requisitos más críticos de un proyecto. Se clasifican 4 categorías:

- **Must Have (Debe tener):** Estas son las características y requisitos esenciales que el sistema debe cumplir para ser considerado exitoso. Sin estos elementos, el proyecto no podría funcionar o cumplir con sus objetivos mínimos.
- **Should Have (Debería Tener):** Estos son requisitos importantes que aportan un gran valor, pero no son críticos para la funcionalidad básica del sistema. Aunque son altamente deseables, el proyecto puede seguir siendo viable sin ellos.
- **Could Have (Podría Tener):** Estos son requisitos que son buenos de tener y que pueden proporcionar beneficios adicionales, pero que no son esenciales ni urgentes. Su implementación se puede considerar si hay tiempo y recursos adicionales.
- **Won't Have (No Tendrá):** Estos son requisitos que no se incluirán en la fase actual del proyecto, pero que pueden ser considerados para futuras iteraciones

o versiones. Ayudan a gestionar las expectativas y a concentrar los esfuerzos en lo más importante.

Se decidió asignarle una numeración para asignar la prioridad de la siguiente manera:

- **Must Have:** 4
- **Should Have:** 3
- **Could Have:** 2
- **Won't Have:** 1

Para valoración de los puntos estimados de cada Historia de Usuario se utilizó Three-Point Estimation o Estimación de tres puntos que está basada en PERT (Program Evaluation and Review Technique) proporciona una estimación ponderada teniendo en cuenta la incertidumbre y la variabilidad de la duración de la tarea. Los pasos son los siguientes:

- **Estimar el Tiempo Optimista (O):** El tiempo mínimo en que se puede completar la tarea si todo va perfectamente.
- **Estimar el Tiempo Más Probable (M):** El tiempo en que se espera completar la tarea en circunstancias normales.
- **Estimar el Tiempo Pesimista (P):** El tiempo máximo en que se puede completar la tarea si todo sale mal.
- **Calcular el Tiempo Esperado (E):** Utiliza la fórmula de PERT para calcular el tiempo esperado la cual es la siguiente:

$$E(x) = \frac{O + 4M + P}{6} \quad (4)$$

Donde:

E(x): Tiempo promedio de la tarea.

O: Tiempo optimista.

M: Tiempo más probable.

P: Tiempo pesimista.

La fórmula para calcular la varianza es:

$$Var(x) = \frac{(P - O)^2}{36} \quad (5)$$

Donde:

Var(x): Varianza de la tarea.

P: Tiempo pesimista.

M: Tiempo optimista.

Tabla 36. Historia de usuario- Conexión de Sensores

Historia de Usuario	
Número:001	Usuario: Agricultor
Nombre: Conexión de Sensores	
Prioridad:4	Riesgo: Alto
Puntos estimados:5	Responsable: Desarrollador
Descripción: Como agricultor, quiero que los sensores se conecten en puntos estratégicos del cultivo para medir la temperatura, la humedad del ambiente y la velocidad del viento, para asegurar la recolección de datos precisos.	
Criterios de aceptación: Dado que los sensores deben estar distribuidos en el campo. Cuando se instalan en los puntos estratégicos. Entonces deben medir y enviar datos de temperatura, humedad y velocidad del viento.	

Tabla 37. Historia de usuario- Conexión inalámbrica

Historia de Usuario	
Número:002	Usuario: Agricultor
Nombre: Conexión inalámbrica	
Prioridad:4	Riesgo: Alto
Puntos estimados:5	Responsable: Desarrollador
Descripción: Como agricultor, quiero que el sistema transmita los datos de los sensores de forma inalámbrica.	
Criterios de aceptación: Dado que los sensores están conectados. Cuando los datos se envían a través de Lora. Entonces el sistema debe recibir y mostrar estos datos en la pantalla del módulo Lora.	

Tabla 38. Historia de usuario- Activar el riego automáticamente

Historia de Usuario	
Número:003	Usuario: Agricultor
Nombre: Activar el riego automáticamente	
Prioridad:4	Riesgo: Alta
Puntos estimados:8	Responsable: Desarrollador
Descripción: Como agricultor, quiero que el sistema active el riego automáticamente cuando los sensores detecten que las condiciones lo requieren.	
<p>Criterios de aceptación:</p> <p>Dado que el sistema monitorea las condiciones climáticas.</p> <p>Cuando la temperatura, la humedad o la velocidad del viento alcanzan los umbrales configurados.</p> <p>Entonces el sistema debe activar automáticamente el riego.</p>	
<p>Dado que el sistema ha activado el riego automáticamente.</p> <p>Cuando el riego se activa.</p> <p>Entonces el sistema debe registrar el evento con la fecha y hora de activación, así como las condiciones climáticas que provocaron la activación.</p>	
<p>Dado que las condiciones climáticas han sido monitoreadas.</p> <p>Cuando las condiciones vuelven a niveles normales o se detecta un evento de finalización.</p> <p>Entonces el sistema debe apagar automáticamente el riego y registrar el evento.</p>	

Tabla 39. Historia de usuario- Mostrar datos de los sensores

Historia de Usuario	
Número:004	Usuario: Agricultor
Nombre: Mostrar datos de los sensores	
Prioridad:4	Riesgo: Alto
Puntos estimados:5	Responsable: Desarrollador
Descripción: Como agricultor, quiero ver los datos actualizados de los sensores para monitorear las condiciones de mi cultivo.	
<p>Criterios de aceptación:</p> <p>Dado que un agricultor está en la pantalla principal.</p> <p>Cuando los datos de los sensores se actualizan cada 10 minutos.</p> <p>Entonces el sistema debe mostrar los valores actualizados de humedad, velocidad del viento, temperatura y humedad del suelo.</p>	
<p>Dado el usuario este en el sistema.</p> <p>Cuando el estado de la bomba de riego cambia.</p> <p>Entonces el sistema debe actualizar y mostrar el estado del riego en la pantalla (APAGADO/ENCENDIDO).</p>	
<p>Dado que un agricultor quiere ver la variación de la humedad.</p> <p>Cuando los datos se actualizan.</p> <p>Entonces el sistema debe mostrar un gráfico de la variación de la humedad del suelo.</p>	

Tabla 40. Historia de usuario- Control remoto del riego

Historia de Usuario	
Número:005	Usuario: Agricultor
Nombre: Control remoto del riego	
Prioridad:4	Riesgo: Alto
Puntos estimados:5	Responsable: Desarrollador
Descripción: Como agricultor, quiero poder controlar el sistema de riego de forma remota desde la aplicación móvil, para poder encender y apagar el riego según sea necesario.	
Criterios de aceptación:	
Dado que un agricultor necesita controlar el riego.	
Cuando el agricultor pulsa el botón de encendido en la aplicación.	
Entonces el sistema debe encender el riego y permitir que el agricultor establezca un tiempo de riego.	
Dado que el riego está encendido.	
Cuando el tiempo establecido se cumple.	
Entonces el sistema debe apagar el riego automáticamente.	
Dado que el agricultor desea apagar el riego manualmente.	
Cuando el agricultor pulsa el botón de apagado en la aplicación.	
Entonces el sistema debe mostrar un mensaje de confirmación y apagar el riego si se confirma la acción.	

Tabla 41. Historia de usuario- Listado de actividades del riego

Historia de Usuario	
Número:006	Usuario: Actividades
Nombre: Listado de actividades del riego	
Prioridad:4	Riesgo: Alto
Puntos estimados:4	Responsable: Desarrollador
Descripción: Como agricultor, quiero ver un historial de las actividades de riego para revisar las acciones realizadas.	
Criterios de aceptación:	
Dado que un agricultor está en la pantalla de actividades.	
Cuando se carga la pantalla.	
Entonces el sistema debe mostrar una lista de 10 últimas las actividades del riego (encendido y apagado, manual y automático) con fecha y hora de cada evento.	
Dado que un agricultor está viendo una actividad específica cuando el agricultor selecciona "VER MÁS" en una actividad.	
entonces el sistema debe mostrar detalles adicionales del evento, incluyendo humedad, temperatura y velocidad del viento en el momento de la actividad.	

Tabla 42. Historia de usuario- Notificar al usuario el estado del riego

Historia de Usuario	
Número:007	Usuario: Agricultor
Nombre: Notificar al usuario el estado del riego	
Prioridad:4	Riesgo: Medio
Puntos estimados:4	Responsable: Desarrollador
Descripción: Como agricultor, quiero recibir notificaciones en mi dispositivo móvil sobre el estado del riego.	
Criterios de aceptación: Dado que el sistema de riego cambia de estado. Cuando el riego se enciende o apaga manualmente. Entonces el sistema debe enviar una notificación al dispositivo móvil del usuario indicando el cambio de estado y si fue manual. Dado que el sistema de riego cambia de estado Cuando el riego se enciende o apaga automáticamente Entonces el sistema debe enviar una notificación al dispositivo móvil del usuario indicando el cambio de estado y si fue automático.	

Tabla 43. Historia de usuario- Registrar usuario

Historia de Usuario	
Número:008	Usuario: Agricultor
Nombre: Registrar usuario	
Prioridad:4	Riesgo: Medio
Puntos estimados:5	Responsable: Desarrollador
Descripción: Como nuevo usuario, quiero registrarme en la aplicación móvil para acceder a sus funcionalidades.	
Criterios de aceptación: Dado que un nuevo usuario quiere registrarse. Cuando el usuario ingresa su nombre, apellido, dirección, teléfono, correo electrónico y contraseña. Entonces el sistema debe permitir la introducción de estos datos. Dado que un nuevo usuario se registra. Cuando el usuario ingresa su correo electrónico. Entonces el sistema debe validar que el correo electrónico no esté ya registrado. Dado que un nuevo usuario se registra. Cuando el usuario proporciona sus datos personales. Entonces el sistema debe crear una nueva cuenta y permitir el acceso al usuario.	

Tabla 44. Historia de usuario- Ingreso al Sistema

Historia de Usuario	
Número:009	Usuario: Agricultor
Nombre: Ingreso al Sistema	
Prioridad:3	Riesgo: Medio
Puntos estimados:3	Responsable: Desarrollador
Descripción: Como usuario registrado, quiero poder ingresar al sistema utilizando mis credenciales para acceder a mis datos y controlar el riego.	
<p>Criterios de aceptación:</p> <p>Dado que el usuario está registrado y ha iniciado sesión.</p> <p>Cuando el usuario actualiza su nombre, apellido, dirección, teléfono, correo electrónico o contraseña. Entonces el sistema debe permitir la modificación de estos datos.</p>	
<p>Dado que el usuario ingresa un nuevo correo electrónico.</p> <p>Cuando el sistema verifica el correo.</p> <p>Entonces el sistema debe validar que el correo electrónico actualizado no esté ya registrado.</p>	
<p>Dado que el usuario ha realizado cambios en su información personal.</p> <p>Cuando el usuario guarda los cambios.</p> <p>Entonces el usuario debe recibir una confirmación de actualización exitosa.</p>	
<p>Dado que el usuario desea cambiar su contraseña.</p> <p>Cuando el usuario ingresa su contraseña actual y escribe la nueva contraseña dos veces.</p> <p>Entonces el sistema debe validar la contraseña actual y actualizar la contraseña si las dos nuevas coinciden.</p>	

Tabla 45. Historia de usuario- Actualizar usuario

Historia de Usuario	
Número:010	Usuario: Agricultor
Nombre: Actualizar usuario	
Prioridad:3	Riesgo: Bajo
Puntos estimados:3	Responsable: Desarrollador
Descripción: Como usuario, quiero poder actualizar mis datos personales en la aplicación para mantener mi información actualizada.	
<p>Criterios de aceptación:</p> <p>Dado que el usuario está registrado y ha iniciado sesión.</p> <p>Cuando el usuario actualiza su nombre, apellido, dirección, teléfono, correo electrónico o contraseña. Entonces el sistema debe permitir la modificación de estos datos.</p>	
<p>Dado que el usuario ingresa un nuevo correo electrónico.</p> <p>Cuando el sistema verifica el correo.</p> <p>Entonces el sistema debe validar que el correo electrónico actualizado no esté ya registrado.</p>	
<p>Dado que el usuario ha realizado cambios en su información personal.</p> <p>Cuando el usuario guarda los cambios.</p> <p>Entonces el usuario debe recibir una confirmación de actualización exitosa.</p>	
<p>Dado que el usuario actualiza su información personal,</p> <p>Cuando el sistema guarda los cambios</p> <p>Entonces el sistema debe almacenar los datos actualizados de forma segura.</p>	

Historia de Usuario
Dado que el usuario desea cambiar su contraseña. Cuando el usuario ingresa su contraseña actual y escribe la nueva contraseña dos veces. Entonces el sistema debe validar la contraseña actual y actualizar la contraseña si las dos nuevas coinciden.

Tabla 46. Historia de usuario- Recuperar contraseña

Historia de Usuario	
Número:009	Usuario: Agricultor
Nombre: Recuperar contraseña	
Prioridad:4	Riesgo: Medio
Puntos estimados:3	Responsable: Desarrollador
Descripción: Como usuario, quiero poder recuperar mi contraseña si la olvido para seguir accediendo a la aplicación.	
Criterios de aceptación: Dado que un usuario ha olvidado su contraseña. Cuando el usuario ingresa su correo electrónico en el campo proporcionado y presiona "Enviar". Entonces el sistema debe enviar un correo electrónico con las instrucciones para restablecer la contraseña.	

Tabla 47. Historia de usuario- Implementar Energía Renovable

Historia de Usuario	
Número:010	Usuario: Agricultor
Nombre: Notificar al usuario el estado del riego	
Prioridad:4	Riesgo: Medio
Puntos estimados:5	Responsable: Desarrollador
Descripción: Como agricultor, quiero que el sistema utilice un panel solar para alimentar la placa de emisión de datos, para asegurar que el sistema funcione de manera autónoma y sostenible con energía renovable.	
Criterios de aceptación: Dado que el sistema necesita una fuente de energía. Cuando se instala el panel solar. Entonces el sistema debe ser capaz de funcionar con energía solar, alimentando la placa de emisión de datos. Dado que se usa energía solar. Cuando el panel solar está en funcionamiento. Entonces el sistema debe ser capaz de cargar una batería que permita la operación continua durante la noche o días nublados.	

d. Estimación de historias de usuario usando el método de tres puntos

La tabla a continuación presenta la estimación en horas de las historias de usuario utilizando el método tres puntos. El tiempo promedio se redondeó a un número entero, y la varianza se convirtió a minutos.

Tabla 48. Estimación de historias de usuario método tres puntos

Número	Tiempo Optimista(h)	Tiempo más probable(h)	Tiempo pesimista(h)	Tiempo promedio(h)	Varianza(min)
001	12	15	18	15	60
002	7	9	10	9	20
003	8	10	12	10	30
004	5	6	8	6	20
005	10	12	14	12	30
006	6	8	9	8	20
007	5	7	8	7	20
008	5	7	9	7	30
009	4	6	8	6	30
010	3	5	7	5	30
011	4	6	8	6	30
012	3	5	8	5	50
			Total	96(h)	6(h)

e. Plan de entrega

Finalizado la definición de requisitos, se procede a realizar el plan de entrega tomando en cuenta las prioridades del sistema.

Tabla 49. Plan de entrega

Iteración	Código	Historia de Usuario	Tiempo estimado	
			Horas	Días
Iteración 1	001	Conexión de los sensores	15	2
	002	Comunicación inalámbrica	9	2
	003	Activar riego automáticamente	10	2
Iteración 2	004	Mostrar datos de los sensores	6	1
	005	Control remoto del riego	12	2
	006	Listado de actividades del riego	8	1
Iteración 3	007	Notificar al usuario el estado del riego	7	1
	008	Registrar usuario	7	1
	009	Ingreso al sistema	6	1
	010	Actualizar usuario	5	1
	011	Recuperar contraseña	6	1
	012	Implementar Energía Renovable	5	1

2.2.3 Fase II: Diseño del prototipo

a. Diseño de la base de datos

A continuación, se muestra el diagrama de la base de datos que estará implementada en MYSQL y servirá para gestionar la información del sistema.

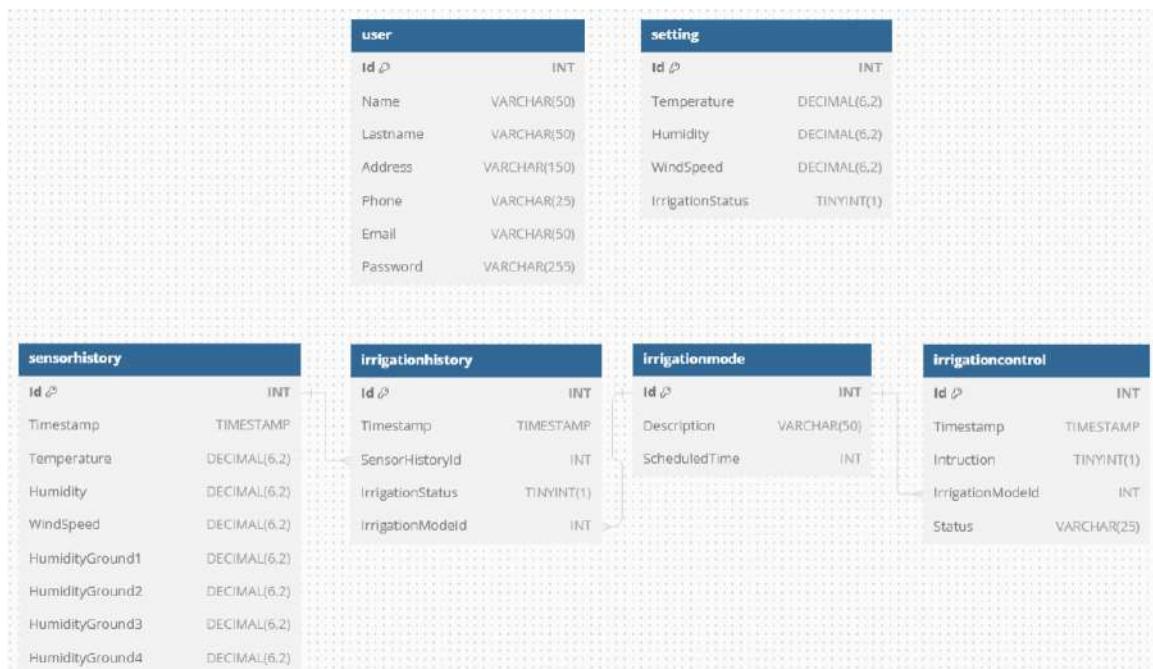


Figura 18. Diagrama de base de datos

En la Figura 18 se muestra el diagrama de la base de datos con las tablas necesarias para el almacenamiento de la información. Además, se tomó en cuenta campos como el inicio de sesión y de los usuarios para el envío de notificaciones.

b. Esquema del sistema

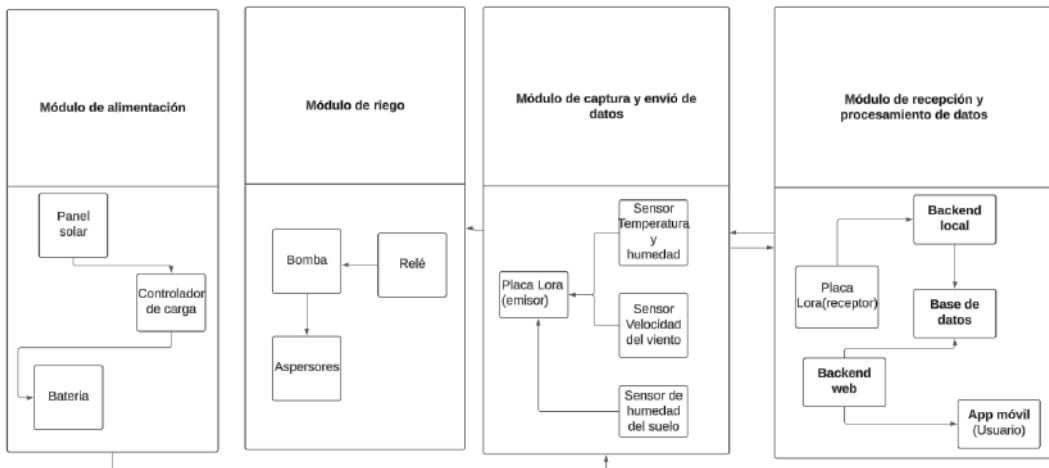


Figura 19. Esquema del proyecto

En la Figura 19 se muestra de forma genérica el esquema del sistema con los diferentes módulos que lo conforman, está dividido en 4: el módulo de alimentación del sistema el cual se encarga de proporcionar la energía a los módulos de riego (bomba), y el módulo de captura y envío de datos que será el encargado de emitir los datos hacia el Gateway para su procesamiento y visualización en la aplicación móvil.

c. Diagrama de la alimentación del sistema

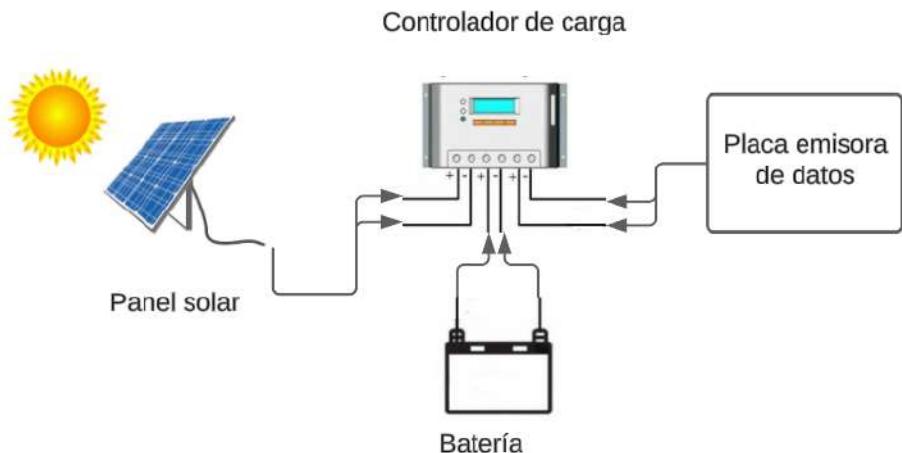


Figura 20. Diagrama de la alimentación del sistema(solar)

El sistema de energía solar diseñado para alimentar una placa emisora de datos. El sistema incluye un panel solar que captura y convierte la energía del sol en electricidad, un controlador de carga que regula la energía del panel solar antes de almacenarla en una batería, y la batería que proporciona una fuente de energía

constante. La energía almacenada en la batería alimenta la placa emisora de datos, permitiendo su funcionamiento continuo incluso en ausencia de luz solar. Este diseño garantiza que la placa emisora funcione de manera autónoma y sostenible, utilizando energía renovable para monitorear condiciones climáticas o de cultivos.

d. Diagrama electrónico de emisor del prototipo

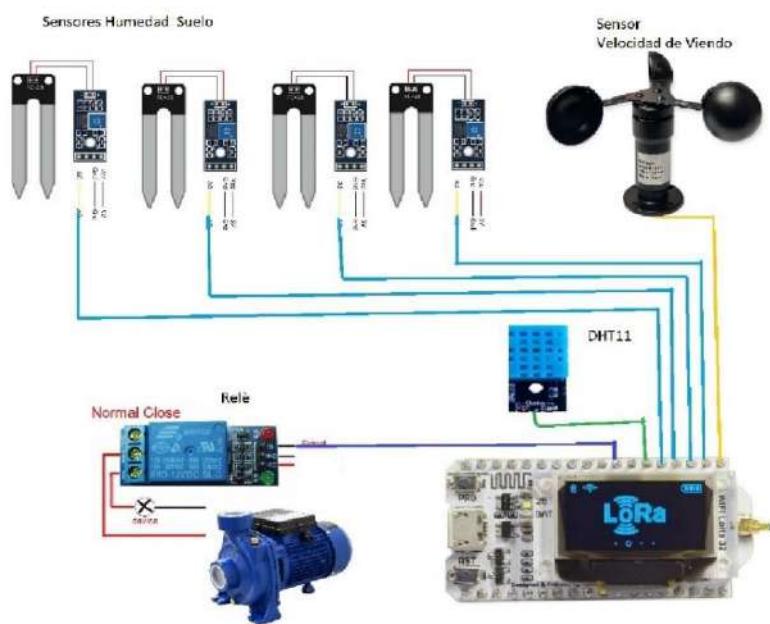


Figura 21. Diagrama de sistema electrónico (Emisor)

El diagrama corresponde al emisor del sistema que es el encargado de la recopilación de datos, utilizando 4 sensores de humedad del suelo lo que sirve para verificar si están funcionando correctamente los aspersores, un módulo de control Lora para el envío de los datos al receptor y un sensor DHT11 que mide temperatura y humedad ambiente. Estos sensores están conectados al módulo Lora, que procesa las señales y controla un módulo de relé. El relé, configurado en modo normalmente cerrado, activa o desactiva una bomba de agua según las necesidades de riego determinadas por los sensores o por el usuario. El sistema permite la gestión remota del riego.

e. Diagrama del receptor del prototipo



Figura 22. Diagrama del receptor

El receptor del sistema está basado en el protocolo MQTT, donde un router proporciona conectividad de red conectando el sistema a internet a través de WAN y a la red local a través de LAN. En el centro, el servidor MQTT actúa como broker, gestionando la transmisión de mensajes entre los clientes MQTT conectados vía LAN. A la derecha, un cliente MQTT, como un microcontrolador, se comunica con el broker para publicar datos de sensores y recibir comandos de control. Las conexiones LAN y WAN indican cómo los dispositivos están interconectados.

f. Arquitectura del sistema

- **Capa de Presentación:** Actúa como la interfaz de usuario (UI), donde los usuarios interactúan con el sistema. Esta capa se encarga de mostrar los datos y recoger las entradas del usuario.
- **Capa de Lógica de Negocio:** Procesan las reglas de negocio y la lógica de la aplicación. Esta capa maneja las solicitudes del usuario, procesa los datos recibidos de la capa de presentación y se comunica con la capa de acceso a datos.
- **Capa de Datos:** Base de Datos MySQL: Gestiona la persistencia de los datos. Esta capa se encarga de las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la base de datos.
- **Capa de Servicios Locales:** Procesan los datos recibidos de los dispositivos IoT. Esta capa se comunica tanto con los dispositivos IoT como con la base de datos.

- **Capa de Dispositivos IoT:** Emisor y Receptor: Capturan datos del entorno (sensores) mediante comunicación Lora y los envían a los servicios locales para su procesamiento. Esta capa actúa como el punto de entrada de los datos del mundo físico al sistema.

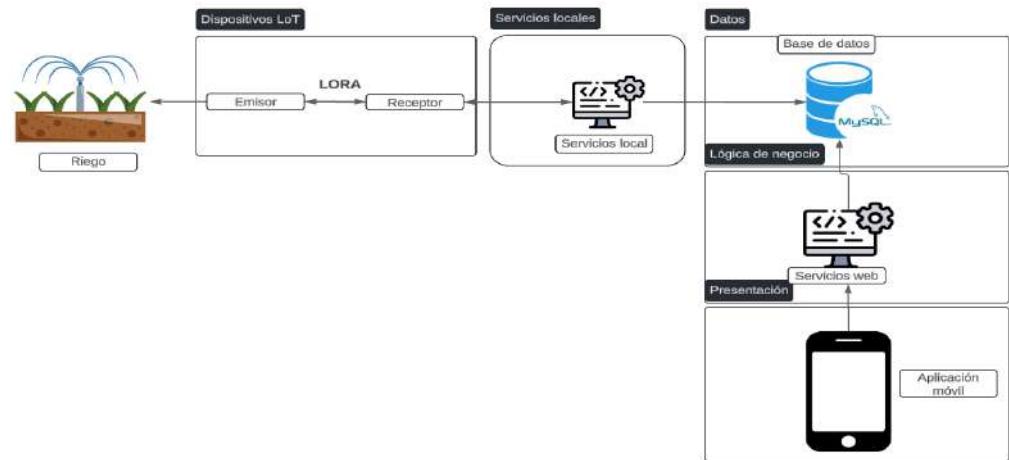


Figura 23. Arquitectura del sistema

g. Diseño de interfaces de usuario

- **Ingreso al sistema**



Figura 24. Prototipo para a inicio de sesión de la aplicación móvil

En la Figura 24 se muestra el diseño del ingreso del sistema en el cual tendrá los campos de correo y contraseña además de la opción recuperar contraseña en caso de

que tenga problemas con recordarla, y en caso de que no tenga cuenta pobra dirigirse a la ventana Registrarse por medio de la opción en pantalla.

- ***Recuperación de la contraseña***



Figura 25. Recuperar Contraseña

En la Figura 25 se muestra el prototipo de la ventana para recuperación de contraseña la cual tendrá el envío de un mensaje con una contraseña temporal para ingresar al sistema la cual podrá cambiarse en el perfil del usuario.

- ***Registro en el sistema***



Figura 26. Prototipo del registro de la aplicación móvil

En la Figura 26 se muestra el formulario para crear una cuenta nueva en la aplicación móvil, en donde están los campos de nombre, apellido, dirección, teléfono, correo y contraseña.

- ***Panel Principal***



Figura 27. Prototipo de la visualización de datos de los sensores

En la Figura 27 se mostrará los datos referentes a los sensores de humedad del ambiente, temperatura, velocidad del viento y la humedad del suelo. Además, se mostrará el estado del riego y usuario pueda manipular de manera remota.

- ***Historial del sistema (Actividades)***

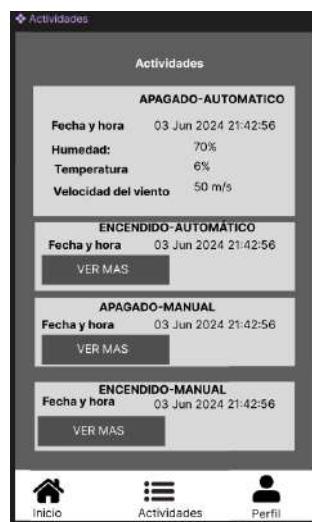


Figura 28. Prototipo del historial de los datos de los sensores

En la Figura 28 se mostrará un historial de cuando se encendió el riego, esto permitirá tener una mejor visualización del consumo de los recursos a lo largo del tiempo.

- *Alertas del sistema*

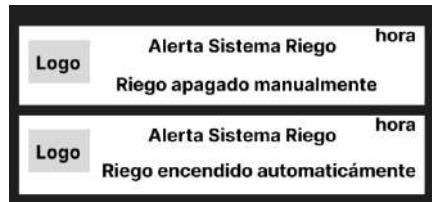


Figura 29. Prototipo de las alertas emitidas por la aplicación móvil.

En la Figura 29 se muestra las alertas del sistema cuando se enciende y se apaga el riego ya sea del modo manual o del automático.

- *Perfil del usuario*



Figura 30. Prototipo del perfil de usuario de la aplicación móvil

En la Figura 30 se muestra el perfil del usuario con los datos que tiene registrado en caso de que requiera modificar alguna información, además cuenta con el proceso de cambio de contraseña en caso de olvidarse y optar por recuperar contraseña. Además de activar el modo oscuro o modo claro.

2.2.4 Fase III: Desarrollo del prototipo

a. Inicialización del proyecto

Se establecieron las tareas en Jira para proporcionar una guía detallada del cronograma, lo cual contribuyó a la organización y gestión del proyecto. Al utilizar Jira, se pudo desglosar el proyecto en tareas específicas y asignarles plazos, facilitando el seguimiento del progreso y los tiempos de entrega.

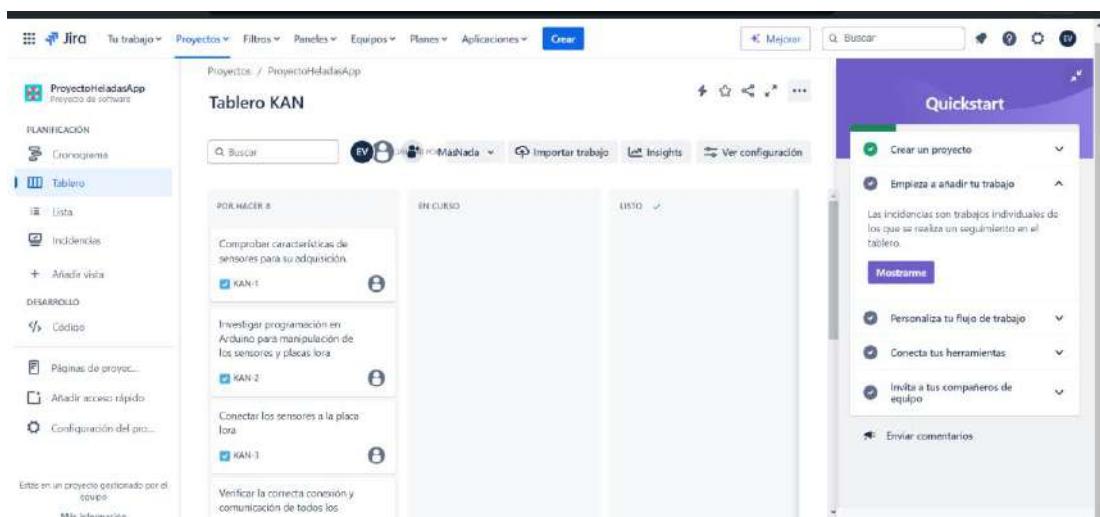


Figura 31.Tareas en Jira

Se creó un repositorio en GitHub para el control de las versiones, el cual estará constituido por 3 ramas: developBackendLocal (servicios para el funcionamiento local), developBackendWeb (servicios para el control remoto), developFront (aplicación móvil).

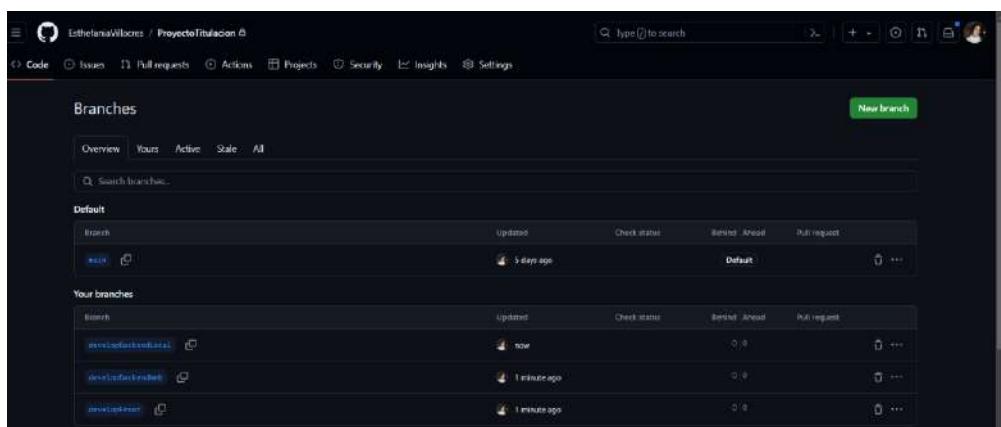
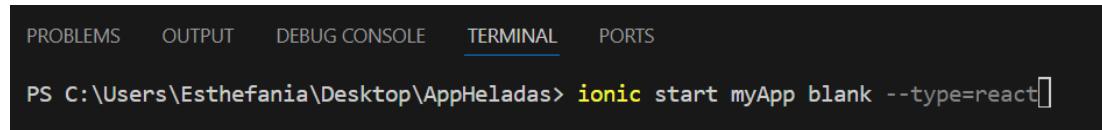


Figura 32. Repositorio GitHub

b. Instalación de herramientas

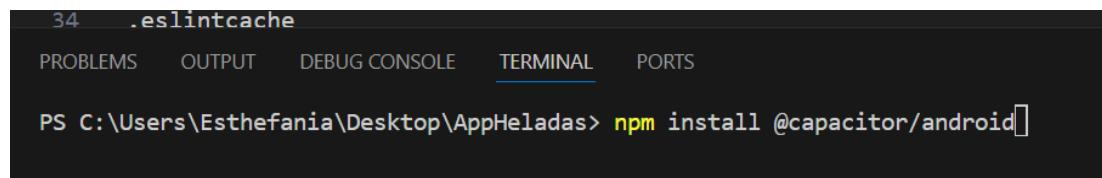
Para el frontend de la aplicación móvil se trabajó con Ionic, además se especifica que se usará React y su instalación se lo realiza a través de siguiente comando:



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
PS C:\Users\Esthefania\Desktop\AppHeladas> ionic start myApp blank --type=react
```

Figura 33. Comando para crear el proyecto Ionic de tipo React

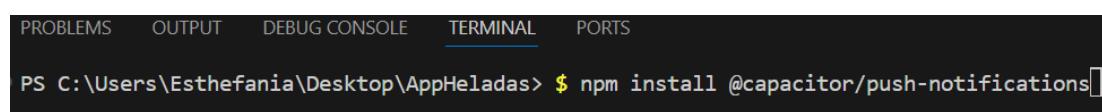
La aplicación al ser hibrida se debe instalar el capacitor de acuerdo la plataforma que se va a usar en este caso se ejecutara en Android por lo cual se añade el siguiente comando al proyecto:



```
34 .eslintcache  
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
PS C:\Users\Esthefania\Desktop\AppHeladas> npm install @capacitor/android
```

Figura 34. Comando para añadir plataforma Android

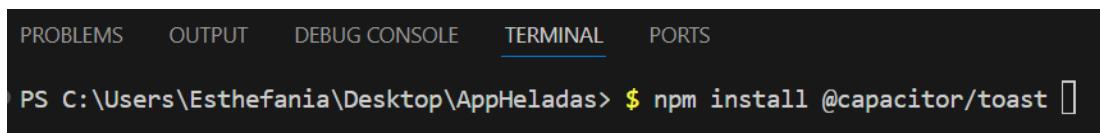
El comando \$ npm install @capacitor/push-notifications se utiliza para instalar el complemento de notificaciones push en un proyecto de Ionic que utiliza Capacitor. Este complemento permite integrar la funcionalidad de notificaciones push, que son mensajes enviados desde un servidor a la aplicación del usuario, incluso cuando la aplicación no está activa.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
PS C:\Users\Esthefania\Desktop\AppHeladas> $ npm install @capacitor/push-notifications
```

Figura 35. Comando para notificaciones push

El comando \$ npm install @capacitor/toast se emplea para instalar el complemento de toast en un proyecto de Ionic que utiliza Capacitor. Este complemento permite mostrar mensajes de toast, que son notificaciones breves y temporales que aparecen en la pantalla del usuario para proporcionar información rápida o feedback sobre acciones realizadas dentro de la aplicación.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\Esthefania\Desktop\AppHeladas> $ npm install @capacitor/toast
```

Figura 36. Comando para toast

c. Codificación del prototipo

• *Emisor*

La importación de librerías es esencial para el funcionamiento del sistema, ya que permite manejar tanto la placa Heltec LoRa ESP32 como los sensores DHT. La librería heltec_unofficial.h proporciona las funciones necesarias para controlar la placa, incluyendo la pantalla OLED, los pines de entrada/salida y la comunicación LoRa. Por otro lado, la librería DHT.h facilita la interacción con los sensores de temperatura y humedad DHT11, proporcionando métodos para inicializar el sensor y leer los valores de temperatura y humedad.

```
#include <heltec_unofficial.h>
#include <DHT.h>
```

Figura 37. Importación de librerías

La definición de constantes y variables configura los diferentes componentes del sistema, como los sensores, los parámetros de comunicación Lora y las variables de control. Las constantes establecen los pines de conexión para los sensores de humedad del suelo, el sensor de velocidad del viento y el control de la bomba de riego. Además, se definen parámetros cruciales para la transmisión Lora, incluyendo la frecuencia, el ancho de banda, el factor de esparcimiento y la potencia de transmisión. Las variables de control se utilizan para manejar la transmisión y recepción de datos, así como para almacenar el tipo de helada detectado.

```

// Configuración de los sensores DHT y otros sensores
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Pines de los sensores de humedad del suelo
const int pinSensorHumedad1 = 6;
const int pinSensorHumedad2 = 5;
const int pinSensorHumedad3 = 4;
const int pinSensorHumedad4 = 3;

// Pin del sensor de velocidad del viento
const int pinSensorVelocidadViento = 7;

// Pin para el control de la bomba
const int pinBomba = 38;

```

Figura 38. Configuración de sensores

```

// Pausa entre paquetes transmitidos en milisegundos
#define PAUSE          2000

// Frecuencia en MHz. Ajusta según tu región.
#define FREQUENCY      915.0

// Ancho de banda de LoRa. Mantén el punto decimal para designar float.
#define BANDWIDTH       125.0

// Factor de esparcimiento de LoRa. Mayor significa más lento pero mayor alcance.
#define SPREADING_FACTOR 7

// Potencia de transmisión en dBm.
#define TRANSMIT_POWER   15

```

Figura 39. Configuración parámetros Lora

```

String rxdata;
volatile bool rxFlag = false;
uint64_t last_tx = 0;
uint64_t tx_time;
uint64_t minimum_pause;

```

Figura 40. Variables de control

La configuración e inicialización del sistema se realizan en la función `setup()`. Aquí, se inicializa la placa Heltec y se configuran los pines de la bomba y los sensores. La comunicación serial se establece a 115200 baudios para depuración. El módulo de radio Lora se configura con los parámetros definidos anteriormente y se inicia la recepción de datos. Finalmente, se inicializa el sensor DHT para comenzar a leer los valores de temperatura y humedad. Esta configuración asegura que todos los componentes estén listos para funcionar correctamente.

```

void setup() {
    heltec_setup();
    pinMode(pinBomba, OUTPUT);
    digitalWrite(pinBomba, LOW);

    Serial.begin(115200);

    Serial.println("Inicializando radio");
    RADIOLIB_OR_HALT(radio.begin());

    // Configurar parámetros del radio
    Serial.printf("Frecuencia: %.2f MHz\n", FREQUENCY);
    RADIOLIB_OR_HALT(radio.setFrequency(FREQUENCY));
    Serial.printf("Ancho de banda: %.1f kHz\n", BANDWIDTH);
    RADIOLIB_OR_HALT(radio.setBandwidth(BANDWIDTH));
    Serial.printf("Factor de espaciamiento: %i\n", SPREADING_FACTOR);
    RADIOLIB_OR_HALT(radio.setSpreadingFactor(SPREADING_FACTOR));
    Serial.printf("Potencia de transmisión: %i dBm\n", TRANSMIT_POWER);
    RADIOLIB_OR_HALT(radio.setOutputPower(TRANSMIT_POWER));

    // Iniciar la recepción
    radio.setDio1Action(rx);
    RADIOLIB_OR_HALT(radio.startReceive(RADIOLIB_SX126X_RX_TIMEOUT_INF));

    // Inicializar el sensor DHT
    dht.begin();

    Serial.println("Emisor LoRa iniciado");
}

```

Figura 41. Configuración e Inicialización en el setup()

El bucle principal (loop()) maneja el envío y la recepción de datos en el sistema. Dentro de este bucle, se verifica si ha pasado suficiente tiempo desde la última transmisión para enviar nuevos datos. Si es así, se llama a la función transmitirDatos() para leer los sensores y enviar la información. Además, el bucle verifica si se han recibido datos mediante el flag rxFlag. Si se ha recibido información, se llama a la función manejarRecepcion() para procesar los datos recibidos. Este bucle continuo asegura que el sistema responda tanto a la recolección de datos como a los comandos recibidos.

La función transmitirDatos() lee los valores de los sensores de temperatura, humedad, velocidad del viento y humedad del suelo, y determina el estado actual de la bomba de riego. También evalúa el tipo de helada basado en las condiciones ambientales. Luego, construye un mensaje en formato JSON con estos datos y lo transmite utilizando Lora. Esta función se encarga de la recopilación y envío de información crítica para el monitoreo y control del riego.

La función manejarRecepcion() procesa los datos recibidos por el sistema. Si se ha recibido un mensaje sin errores, este se analiza y se ejecutan los comandos

correspondientes, como encender o apagar la bomba de riego. Esta función asegura que el sistema responda adecuadamente a las instrucciones enviadas, permitiendo un control remoto efectivo del riego.

La función processCommand() ejecuta comandos específicos basados en los datos recibidos. Actualmente, maneja comandos para encender (BOMBA_ON) o apagar (BOMBA_OFF) la bomba de riego. Al recibir un comando, la función ajusta el estado del pin correspondiente para activar o desactivar la bomba, proporcionando un control directo sobre el sistema de riego.

```
void loop() {
    heltec_loop();

    bool tx_legal = millis() > last_tx + minimum_pause;
    if (tx_legal) {
        transmitirDatos();
    }

    if (rxFlag) {
        manejarRecepcion();
    }
}
```

Figura 42. Bucle Principal (loop())

- ***Receptor***

Este bloque define las constantes y variables utilizadas en el sistema. Configura los parámetros de Lora como frecuencia, ancho de banda, factor de esparcimiento y potencia de transmisión. También define las credenciales de WiFi, la configuración del servidor MQTT y sus tópicos para la publicación y suscripción de datos.

```

// Configuración de LoRa
#define FREQUENCY 915.0
#define BANDWIDTH 125.0
#define SPREADING_FACTOR 7
#define TRANSMIT_POWER 14

String rxdata;
volatile bool rxFlag = false;

// Configuración de WiFi y MQTT
const char* ssid = "ESTHEFANIA";
const char* password = "1804816104@";

const char* mqtt_server = "192.168.0.107"; //ip fija del servidior
const int mqtt_port = 1883; // puerto del mosquitto

// Tópicos MQTT
const char* topicTemp = "sensor/temp";
const char* topicHum = "sensor/hum";
const char* topicViento = "sensor/viento";
const char* topicHum1 = "sensor/hum1";
const char* topicHum2 = "sensor/hum2";
const char* topicHum3 = "sensor/hum3";
const char* topicHum4 = "sensor/hum4";
const char* topicEstadoBomba = "actuator/bomba";
const char* topicCommandBomba = "command/bomba";

WiFiClient espClient;
PubSubClient client(espClient);

```

Figura 43. Definición de Constantes y Variables

La función setup() inicializa la placa Heltec, establece la comunicación serial y llama a las funciones de configuración para el módulo de radio Lora, la conexión Wifi y el cliente MQTT. Asegura que todos los componentes estén listos para operar.

```

void setup() {
    heltec_setup();
    Serial.begin(115200);

    both.println("Inicializando radio");
    setupRadio();

    setupWiFi();
    setupMQTT();
}

```

Figura 44. Configuración e Inicialización

La función loop() es el bucle principal que mantiene la placa Heltec en funcionamiento, verifica y mantiene la conexión MQTT, y procesa los datos recibidos a través de Lora. Si se reciben datos, se llama a la función processReceivedData() para manejarlos

```

void loop() {
    heltec_loop();

    if (!client.connected()) {
        reconnectMQTT();
    }
    client.loop();

    if (rxFlag) {
        rxFlag = false;
        radio.readData(rxdata);
        if (_radiolib_status == RADIOLIB_ERR_NONE) {
            both.printf("Datos recibidos: %s\n", rxdata.c_str());
            processReceivedData(rxdata);
        } else {
            both.printf("Error [%d]\n", _radiolib_status);
        }
        RADIOLIB_OR_HALT(radio.startReceive(RADIOLIB_SX126X_RX_TIMEOUT_INF));
    }
}

```

Figura 45. Bucle Principal

La función setupWiFi() se encarga de conectar el dispositivo a la red WiFi utilizando las credenciales proporcionadas. Intenta conectar cada segundo hasta que se establece la conexión, luego imprime la dirección IP local.

```

void setupWiFi() {
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        both.println("Conectando a WiFi...");
    }
    both.println("Conectado a la red WiFi");
    both.println(WiFi.localIP());
}

```

Figura 46. Función setupWiFi()

La función setupMQTT() configura el cliente MQTT para conectarse al servidor especificado y establece la función de devolución de llamada (mqttCallback) para manejar los mensajes entrantes. Luego intenta conectar al servidor MQTT llamando a reconnectMQTT().

```

void setupMQTT() {
    client.setServer(mqtt_server, mqtt_port);
    client.setCallback(mqttCallback);
    reconnectMQTT();
}

```

Figura 47. Función setupMQTT()

La función reconnectMQTT() maneja la reconexión al servidor MQTT. Intenta conectar repetidamente cada 5 segundos hasta que se establece la conexión, luego se suscribe al tópico topicCommandBomba

```
void reconnectMQTT() {
    while (!client.connected()) {
        both.println("Intentando conexión MQTT...");
        if (client.connect("ESP32Client")) {
            both.println("Conectado a MQTT");
            client.subscribe(topicCommandBomba);
        } else {
            both.println("Falló en la conexión MQTT, reintento en 5 segundos");
            delay(5000);
        }
    }
}
```

Figura 48. Función reconnectMQTT()

La función mqttCallback() se ejecuta cada vez que se recibe un mensaje en un tópico suscrito. Construye el comando recibido en una cadena de caracteres y llama a sendActivationCommand() para procesarlo.

```
void mqttCallback(char* topic, byte* payload, unsigned int length) {
    string command = "";
    for (int i = 0; i < length; i++) {
        command += (char)payload[i];
    }
    sendActivationCommand(command);
}
```

Figura 49.Funció mqttCallback()

La función setupRadio() configura el módulo de radio Lora con los parámetros especificados (frecuencia, ancho de banda, factor de esparcimiento y potencia de transmisión). También establece la acción a realizar al0 recibir datos (rx).

```

void setupRadio() {
    RADIOLIB_OR_HALT(radio.begin());

    both.printf("Frecuencia: %.2f MHz\n", FREQUENCY);
    RADIOLIB_OR_HALT(radio.setFrequency(FREQUENCY));
    both.printf("Ancho de banda: %.1f kHz\n", BANDWIDTH);
    RADIOLIB_OR_HALT(radio.setBandwidth(BANDWIDTH));
    both.printf("Factor de esparcimiento: %i\n", SPREADING_FACTOR);
    RADIOLIB_OR_HALT(radio.setSpreadingFactor(SPREADING_FACTOR));
    both.printf("Potencia de transmisión: %i dBm\n", TRANSMIT_POWER);
    RADIOLIB_OR_HALT(radio.setOutputPower(TRANSMIT_POWER));

    radio.setDio1Action(rx);
    RADIOLIB_OR_HALT(radio.startReceive(RADIOLIB_SX126X_RX_TIMEOUT_INF));
}

```

Figura 50. Función setupRadio()

La función processReceivedData() procesa los datos recibidos en formato JSON, extrayendo información como temperatura, humedad, velocidad del viento, humedad del suelo y estado de la bomba. Imprime estos datos en la consola serial y los publica en los tópicos MQTT correspondientes.

```

void processReceivedData(String data) {
    DynamicJsonDocument doc(1024);
    DeserializationError error = deserializeJson(doc, data);
    if (error) {
        both.printf("Error al parsear JSON: %s\n", error.c_str());
        return;
    }
    float temp = doc["Temp"];
    float hum = doc["Hum"];
    float viento = doc["Viento"];
    float Hum1 = doc["Hum1"];
    float Hum2 = doc["Hum2"];
    float Hum3 = doc["Hum3"];
    float Hum4 = doc["Hum4"];
    String estadoBomba = doc["Bomba"];
    String tipoHelada = doc["TipoHelada"];

    both.println("==== Datos Recibidos ====");
    both.printf("Temperatura: %.2f °C\n", temp);
    both.printf("Humedad: %.2f %%\n", hum);
    both.printf("Velocidad del Viento: %.2f m/s\n", viento);
    both.printf("Hum1: %.2f %%\n", Hum1);
    both.printf("Hum2: %.2f %%\n", Hum2);
    both.printf("Hum3: %.2f %%\n", Hum3);
    both.printf("Hum4: %.2f %%\n", Hum4);
    both.printf("Estado de la Bomba: %s\n", estadoBomba.c_str());
    //both.printf("Tipo de Helada: %s\n", tipoHelada.c_str());
    both.println("====");

    // Enviar datos en formato JSON por MQTT
    client.publish(topicTemp, String(temp).c_str());
    client.publish(topicHum, String(hum).c_str());
    client.publish(topicViento, String(viento).c_str());
    client.publish(topicHum1, String(Hum1).c_str());
    client.publish(topicHum2, String(Hum2).c_str());
    client.publish(topicHum3, String(Hum3).c_str());
    client.publish(topicHum4, String(Hum4).c_str());
    client.publish(topicEstadoBomba, estadoBomba.c_str());
}

```

Figura 51. Función processReceivedData()

La función sendActivationCommand() envía comandos específicos para encender o apagar la bomba de riego. Transmite el comando a través de Lora y maneja el estado del módulo de radio para la recepción continua.

```
void sendActivationCommand(String command) {
    if (command == "BOMBA_ON" || command == "BOMBA_OFF") {
        radio.clearDiolAction();
        RADIOLIB(radio.transmit(command.c_str()));
        if (_radiolib_status == RADIOLIB_ERR_NONE) {
            both.printf("Comando enviado [%s]\n", command.c_str());
        } else {
            both.printf("Error al enviar comando [%d]\n", _radiolib_status);
        }
        radio.setDiolAction(rx);
        RADIOLIB_OR_HALT(radio.startReceive(RADIOLIB_SX126X_RX_TIMEOUT_INF));
    }
}
```

Figura 52. Función sendActivationCommand()

- *Backend local*

Esta función intenta insertar los datos del sensor en la base de datos MySQL. Verifica que todos los valores de los sensores sean válidos antes de realizar la inserción. Utiliza una conexión de pool para gestionar las conexiones a la base de datos de manera eficiente. Si la inserción es exitosa, los valores del sensor se restablecen a None para la próxima inserción.

```
def try_insert_data_into_db(self):
    if all(value is not None for value in self.sensor_data.values()):
        columns = ', '.join(self.sensor_data.keys())
        placeholders = ', '.join(['%s'] * len(self.sensor_data))
        values = tuple(self.sensor_data.values())
        query = f"INSERT INTO sensorhistory ({columns}, Timestamp) VALUES ({placeholders}, NOW())"
        try:
            conn = self.dbpool.get_connection()
            cursor = conn.cursor()
            cursor.execute(query, values)
            conn.commit()
            print("Data inserted successfully")
            # Reset values for next insertion
            for key in self.sensor_data:
                self.sensor_data[key] = None
        except mysql.connector.Error as e:
            print(f"Error inserting data into DB: {e}")
        finally:
            cursor.close()
            conn.close()
```

Figura 53. Inserción de Datos en la Base de Datos

Esta función consulta el estado del riego en la base de datos y envía un comando MQTT para encender o apagar la bomba según el último estado registrado. Se

ejecuta periódicamente gracias a un temporizador, asegurando que el sistema de riego se ajuste automáticamente según los datos más recientes.

```

def check_and_update_irrigation(self):
    query = "SELECT IrrigationStatus FROM irrigationhistory ORDER BY Id DESC LIMIT 1"
    try:
        conn = self.dbpool.get_connection()
        cursor = conn.cursor()
        cursor.execute(query)
        result = cursor.fetchone()
        if result:
            status = result[0]
            command = "BOMBA_ON" if status == 1 else "BOMBA_OFF"
            self.send_command(command)
            print(f"Sending MQTT command based on DB status: {command}")
    except mysql.connector.Error as e:
        print(f"Database error when fetching irrigation status: {e}")
    finally:
        cursor.close()
        conn.close()

```

Figura 54. Verificación y Actualización del Estado del Riego

Esta función controla el sistema de riego automáticamente, tomando decisiones basadas en datos recientes del sensor, umbrales de riego y el historial de riego. Puede encender o apagar el riego automáticamente según las condiciones climáticas y los modos de riego (Automático o Manual). También envía notificaciones cuando se realizan cambios en el estado del riego.

```

def control_irrigation_system(self):
    sensor_data = self.select_last_sensor()
    if not sensor_data:
        print("No se pudo obtener los datos del sensor")
        return
    irrigation_on_data = self.get_irrigation_on_data()
    irrigation_off_data = self.get_irrigation_off_data()
    if not irrigation_on_data or not irrigation_off_data:
        print("No se pudieron obtener los umbrales de encendido o apagado")
        return
    irrigation_history = self.get_irrigation_history_with_modes()
    if not irrigation_history:
        print("No se pudo obtener el historial de riego")
        return
    current_temperature = sensor_data['Temperatura']
    current_humidity = sensor_data['Humedad']
    current_windSpeed = sensor_data['VelocidadViento']
    current_time = sensor_data['FechaHora']
    irrigation_status = irrigation_history['IrrigationStatus']
    irrigation_mode = irrigation_history['Description']
    last_irrigation_time = irrigation_history['Timestamp']

```

```

        scheduled_time = irrigation_history['ScheduledTime']
        if not irrigation_status and current_temperature <=
irrigation_on_data['Temperature'] and
current_humidity>=irrigation_on_data['Humidity'] and
current_windSpeed<=irrigation_on_data['WindSpeed']:
            command = "BOMBA_ON"
            self.send_command(command)
            self.insert_irrigation_history_on(sensor_data['Id'])
            self.send_notify("Riego encendido automáticamente")
            print("El sistema de riego se ha encendido automáticamente")
        elif irrigation_mode == 'Automatic' and irrigation_status and
current_temperature >= irrigation_off_data['Temperature']and
current_humidity<irrigation_off_data['Humidity']:
            command = "BOMBA_OFF"
            self.send_command(command)
            self.insert_irrigation_history_off(sensor_data['Id'])
            self.send_notify("Riego apagado automáticamente")
            print("El sistema de riego se ha apagado automáticamente")
        elif irrigation_mode == 'Manual' and irrigation_status:
            time_diff = (datetime.now() + timedelta(hours=1) -
last_irrigation_time).total_seconds() / 60
            if time_diff >= scheduled_time:
                command = "BOMBA_OFF"
                self.send_command(command)
                self.insert_irrigation_history_off(sensor_data['Id'])
                self.send_notify("Riego apagado automáticamente")
                print("El sistema de riego programado se ha apagado después
del tiempo programado")

```

Figura 55. Control del Sistema de Riego Automático

La función `insert_irrigation_history_on_manual` inserta un registro en la tabla `irrigationhistory` de la base de datos MySQL para documentar la activación manual del riego. Utiliza una consulta SQL con parámetros para incluir el ID del historial del sensor, el estado de riego (encendido), y el modo de riego (manual). La función obtiene una conexión pool de conexiones, ejecuta la consulta y confirma la transacción, asegurando que los cambios se guarden. En caso de error, captura la excepción y muestra un mensaje de error. Finalmente, cierra el cursor y la conexión para liberar recursos.

```

def insert_irrigation_history_on_manual(self, sensor_history_id):
    query = "INSERT INTO irrigationhistory (SensorHistoryID, IrrigationStatus, IrrigationModeID) VALUES (%s, %s, %s)"
    try:
        conn = self.dbpool.get_connection()
        cursor = conn.cursor()
        cursor.execute(query, (sensor_history_id, 1, 2))
        conn.commit()
        print("Registro insertado correctamente en irrigationhistory")
    except mysql.connector.Error as e:
        print(f"Error al insertar datos en MySQL: {e}")
    finally:
        cursor.close()
        conn.close()

```

Figura 56. Función insert_irrigation_history_on_manual

La función control_irrigation_manual gestiona las tareas de riego manual pendientes en el sistema. Primero, obtiene las tareas pendientes de la base de datos y verifica si el estado de la tarea es 'Pending'. Si es así, intenta obtener los datos más recientes del sensor. Si no se pueden obtener los datos del sensor, la función imprime un mensaje y termina. Si los datos del sensor están disponibles, verifica la instrucción de la tarea pendiente. Si la instrucción es activar la bomba, envía un comando para encender la bomba, inserta un registro de activación manual en el historial de riego y envía una notificación. Si la instrucción es desactivar la bomba, envía un comando para apagar la bomba, inserta un registro de desactivación manual en el historial de riego y envía una notificación. Finalmente, la función actualiza el estado de la tarea de riego en la base de datos a 'Completed'. Esta función asegura que las tareas de riego manual se ejecuten y registren correctamente

```

def control_irrigation_manual(self):
    pending_task = self.get_pending_tasks()
    print(pending_task['Status'])
    if pending_task['Status'] == 'Pending':
        sensor_data = self.select_last_sensor()
        if not sensor_data:
            print("No se pudo obtener los datos del sensor")
            return
        if pending_task['Instruction']:
            command = "BOMBA_ON"
            self.send_command(command)
            self.insert_irrigation_history_on_manual(sensor_data['Id'])
            self.send_notify("Riego encendido manualmente")
        else:
            command = "BOMBA_OFF"
            self.send_command(command)
            self.insert_irrigation_history_off_manual(sensor_data['Id'])
            self.send_notify("Riego apagado manualmente")
        self.update_status_irrigation_control()

```

Figura 57. Función control_irrigation_manual

La función initialize_firebase inicializa la conexión con Firebase utilizando un archivo de credenciales. send_notify envía notificaciones push a los dispositivos registrados utilizando Firebase Cloud Messaging (FCM). Esta función obtiene los tokens de los dispositivos desde la base de datos y envía notificaciones con el mensaje especificado, facilitando la comunicación en tiempo real con los usuarios del sistema de riego.

```

def initialize_firebase(self):
    if not firebase_admin._apps:
        cred = credentials.Certificate("Credentials.json")
        firebase_admin.initialize_app(cred)

def send_notify(self, body):
    try:
        conn = self.dbpool.get_connection()
        cursor = conn.cursor(dictionary=True)
        cursor.execute("SELECT DISTINCT Token FROM token")
        rows = cursor.fetchall()
        cursor.close()
        conn.close()
        tokens = [token[ 'Token' ] for token in rows]
        for token in tokens:
            message = messaging.Message(
                notification=messaging.Notification(
                    title='Alerta Sistema Riego',
                    body=body
                ),
                android=messaging.AndroidConfig(
                    notification=messaging.AndroidNotification(
                        sound='default'
                    )
                ),
                token=token,
            )
            messaging.send(message)
        return {"message": "Enviado correctamente el mensaje"}
    except mysql.connector.Error as e:
        print(f"Error al actualizar datos en MySQL: {e}")
    return False

```

Figura 58. Inicialización de Firebase y Envío de Notificaciones

- ***Backend web***

En el proyecto se ha utilizado Flask como el framework de desarrollo web debido a su flexibilidad además se ha implementado la arquitectura MVC (Modelo-Vista-Controlador). Esta estructura se refleja en nuestra disposición de carpetas, donde cada directorio tiene un propósito específico: la carpeta `controller` contiene los controladores que manejan la lógica de negocio, `models` alberga las definiciones de las estructuras de datos y modelos de la base de datos, `routes` define las rutas de la aplicación que conectan las URL con las funciones

apropiadas, `services` incluye los servicios que encapsulan funcionalidades comunes, y `database` gestiona las interacciones con la base de datos.

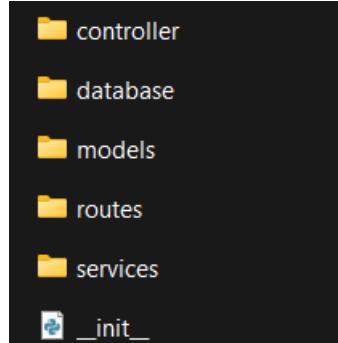


Figura 59. Estructura de carpetas del backend web

A continuación, se describirá las principales funciones que se utilizarán en la aplicación móvil.

El código define un endpoint POST en Flask en la ruta `/user` para agregar un usuario. La función `add_user` recibe datos JSON de la solicitud y verifica si existe caso contrario devuelve un error 400. Si los datos están presentes, se crea un objeto `User` con las propiedades extraídas del JSON, como `Id`, `Name`, `Lastname`, `Address`, `Phone`, `Email` y `Password`. Este objeto se pasa al método `insert_user` del `auth_controller` para su inserción en la base de datos. La respuesta de la operación se maneja devolviendo un código 201 para una inserción exitosa o un 500 si ocurre un error en el servidor.

```
@api_scope.route('/user', methods=['POST'])
def add_user():
    user_data = request.json
    if not user_data:
        return jsonify({"error": "No se proporcionaron datos"}), 400
    user=User(Id=user_data.get("Id"),Name=user_data.get("Name"),
              Lastname=user_data.get("Lastname"),
              Address=user_data.get("Address"),
              Phone=user_data.get("Phone"),
              Email=user_data.get("Email"),
              Password=user_data.get("Password"))
    result = auth_controller.insert_user(user)
    if result['success']:
        return jsonify(result), 201 # Código 201 para indicar creación exitosa
    else:
        return jsonify(result), 500 # Código 500 para indicar error en el servidor
```

Figura 60. Endpoint creación de usuario

Se define una ruta `/forgotPassword` para manejar solicitudes de recuperación de contraseña. La función `forgot_password_route` extrae el campo `email` del JSON recibido en la solicitud. Si el correo electrónico no se proporciona, devuelve una

respuesta JSON con un mensaje de error y un código de estado 401, indicando que es necesario ingresar un correo. Si el correo está presente, se llama al método `handle_forgot_password` del `auth_controller` con el correo proporcionado. La respuesta y el código de estado resultantes de esta operación se devuelven como una respuesta JSON al cliente, proporcionando información sobre el éxito o fracaso del proceso de recuperación de contraseña.

```
@api_scope.route('/forgotPassword', methods=['POST'])
def forgot_password_route():
    email = request.json.get('email')
    if not email:
        return jsonify({"error": "No se ha ingresado un correo"}), 401
    response, status_code = auth_controller.handle_forgot_password(email)
    return jsonify(response), status_code
```

Figura 61. Endpoint recuperar contraseña

La siguiente ruta `/login` para manejar las solicitudes de inicio de sesión de usuarios. La función `handle_login` extrae los datos de usuario (`user_data`) del JSON recibido en la solicitud. Si no se proporcionan datos, se devuelve una respuesta JSON con un mensaje de error y un código de estado 400, indicando que faltan datos requeridos. Si los datos están presentes, se crea una instancia de la clase `User` con los campos `Email` y `Password` obtenidos del JSON. Luego, se pasa este objeto `User` al método `login` del `auth_controller` para manejar el proceso de autenticación. Finalmente, el resultado de esta operación se devuelve como respuesta al cliente, indicando si el inicio de sesión fue exitoso o no.

```
@api_scope.route('/login', methods=['POST'])
def handle_login():
    user_data = request.json
    if not user_data:
        return jsonify({"error": "No se proporcionaron datos"}), 400
    user = User(Email=user_data.get("Email"), Password=user_data.get("Password"))
    result = auth_controller.login(user)
    return result
```

Figura 62. Endpoint ingresar al sistema

Las siguientes rutas se utilizan en el panel principal de la aplicación móvil:

Este endpoint GET en la ruta `/sensor` se obtiene los datos del último sensor registrado. La función `get_last_sensor` crea una instancia vacía de `Sensor` y llama al método `last_sensor` del `sensor_controller`, pasando la instancia del sensor para obtener el último registro. Si se encuentra un registro (`sensor_new` no es `None`), se devuelve una respuesta JSON con los datos del sensor en formato de diccionario

(`asdict()`) y un código de estado 200, indicando éxito. Si no se encuentra un registro, se devuelve una respuesta JSON con un mensaje de error y un código de estado 500, indicando un problema al obtener los datos de la base de datos.

```
@api_scope.route('/sensor', methods=['GET'])
def get_last_sensor():
    sensor = Sensor()
    sensor_new = sensor_controller.last_sensor(sensor)
    if sensor_new is not None:
        return jsonify(sensor_new._asdict()), 200 # Código 200 para indicar éxito
    else:
        return jsonify({"error": "Error al obtener datos de la base de datos"}), 500
```

Figura 63. Endpoint consulta el ultimo registro de los sensores

Este código define un endpoint GET en la ruta `/lastIrrigationStatus` para obtener el estado más reciente del sistema de riego. La función `get_irrigation_status` llama al método `irrigation_status` del `irrigation_controller`, que retorna el estado actual del riego. Si se obtiene un estado (`irrigation_new` no es `None`), se devuelve una respuesta JSON con los datos del estado en formato de diccionario (`asdict()`) y un código de estado 200, indicando éxito. Si no se puede obtener el estado del riego, se devuelve una respuesta JSON con un mensaje de error y un código de estado 500, indicando que hubo un problema al obtener la información.

```
@api_scope.route('/lastIrrigationStatus', methods=['GET'])
def get_irrigation_status():
    irrigation_new = irrigation_controller.irrigation_status()
    if irrigation_new:
        return jsonify(irrigation_new._asdict()), 200
    else:
        return jsonify({"error": "No se pudo obtener el estado del riego"}), 500
```

Figura 64. Endpoint estado del riego

Endpoint GET bajo la ruta `/humidity` para obtener los últimos registros de humedad. La función `get_last_humidity_records` llama al método `select_last_humidity_records` del `sensor_controller` para recuperar los datos de humedad más recientes. Si los registros (`records`) no son `None`, se devuelve una respuesta JSON con los datos y un código de estado 200, indicando que la solicitud fue exitosa. Si no se pueden obtener los registros de humedad, se devuelve una

respuesta JSON con un mensaje de error y un código de estado 500, indicando un problema al acceder a los datos.

```
@api_scope.route('/humidity', methods=['GET'])
def get_last_humidity_records():
    records = sensor_controller.select_last_humidity_records() # Obtener los últimos registros de humedad
    if records is not None:
        return jsonify(records), 200
    else:
        return jsonify({"error": "Error al obtener los datos"}), 500
```

Figura 65. Enpoint Recupera la humedad del suelo

La ruta `/irrigationHistory` recupera el historial reciente de datos de riego. La función `recent_irrigation_data` llama al método `get_recent_irrigation_data` del `irrigation_controller` para obtener los datos de riego más recientes. Si los datos obtenidos (`data`) no son `None`, se devuelve una respuesta JSON con los datos y un código de estado 200, indicando que la operación fue exitosa. Si no se pueden obtener los datos de riego, se devuelve una respuesta JSON con un mensaje de error y un código de estado 500, señalando un problema al acceder a los datos de riego.

```
@api_scope.route('/irrigationHistory', methods=['GET'])
def recent_irrigation_data():
    data = irrigation_controller.get_recent_irrigation_data()
    if data is not None:
        return jsonify(data), 200 # Código 200 para indicar éxito
    else:
        return jsonify({"error": "Error al obtener los datos de riego"}), 500
```

Figura 66. Enpoint Historial del riego

Este endpoint POST bajo la ruta `/irrigationOnManual` sirve para activar manualmente el sistema de riego. La función `irrigation_on_manual` recibe datos en formato JSON de la solicitud mediante `request.get_json()`. Dentro de un bloque `try`, verifica si se han proporcionado datos; si no, retorna un mensaje de error con un código 400. Si los datos están presentes, actualiza la hora programada del riego utilizando `setting_controller.update_scheduled_time` con el valor `2` y la hora proporcionada en `data['scheduled_time']`, y luego activa el riego mediante `setting_controller.on_system_irrigation()`. Si todo se ejecuta correctamente, devuelve un mensaje de éxito con un código 200 indicando que el riego se encendió manualmente. En caso de que ocurra una excepción, se captura y se devuelve un

mensaje de error con el detalle de la excepción y un código 500, señalando un error en el servidor.

```
@api_scope.route('/irrigationOnManual', methods=['POST'])
def irrigation_on_manual():
    data = request.get_json()
    try:
        if not data:
            return jsonify({"error": "No se proporcionaron datos"}), 400
        setting_controller.update_scheduled_time(2,data['scheduled_time'])
        setting_controller.on_system_irrigation()
        return jsonify({"message": "Riego encendido manualmente"}), 200
    except Exception as e:
        return jsonify({"error": str(e)}), 500
```

Figura 67. Enpoint Activación del riego manualmente

Este código define un endpoint PUT en Flask en la ruta `/user/<int:id>` para actualizar la información de un usuario específico identificado por su `id`. La función `update_user` recibe los datos del usuario desde el cuerpo de la solicitud en formato JSON. Primero, verifica si se han proporcionado los datos; de lo contrario, retorna un mensaje de error con un código 400. Si los datos están presentes, genera un hash de la nueva contraseña utilizando `generate_password_hash` y crea una instancia de la clase `User` con los datos proporcionados, incluyendo `Id`, `Name`, `Lastname`, `Address`, `Phone`, `Email` y la contraseña cifrada. Luego, se llama al método `update_user` del `auth_controller` para actualizar la información del usuario en la base de datos. Si la actualización es exitosa (`result['success']`), se devuelve una respuesta JSON con los detalles del resultado y un código 200; de lo contrario, se devuelve un mensaje de error con un código 500, indicando un problema en el servidor.

```
@api_scope.route('/user/<int:id>', methods=['PUT'])
def update_user(id):
    user_data = request.json
    if not user_data:
        return jsonify({"error": "No se proporcionaron datos"}), 400
    passhashed_password = generate_password_hash(user_data.get("Password"))
    user=User(Id=user_data.get("Id"),Name=user_data.get("Name"),
              Lastname=user_data.get("Lastname"),Address=user_data.get("Address"),
              Phone=user_data.get("Phone"),Email=user_data.get("Email"),Password=passhashed_password)
    result = auth_controller.update_user(id, user)
    if result and result['success']:
        return jsonify(result), 200 # Código 200 para indicar éxito
    else:
        return jsonify(result), 500 # Código 500 para indicar error en el servidor
```

Figura 68. Enpoint Actualizar usuario

Este endpoint PUT en la ruta `/updatePassword` se usa para actualizar la contraseña de un usuario. La función `update_password_route` extrae los datos del cuerpo de la solicitud JSON, incluyendo el `user_id` (identificador del usuario), `old_password` (contraseña actual) y `new_password` (nueva contraseña). Luego, llama al método `update_password` del `auth_controller`, pasando estos tres parámetros para actualizar la contraseña del usuario en el sistema. El resultado de esta operación, junto con el código de estado correspondiente, se devuelve como respuesta JSON al cliente, proporcionando información sobre el éxito o fracaso de la actualización de la contraseña.

```
@api_scope.route('/updatePassword', methods=['PUT'])
def update_password_route():
    data = request.json
    user_id = data['user_id']
    old_password = data['old_password']
    new_password = data['new_password']

    response, status_code = auth_controller.update_password(user_id, old_password, new_password)
    return jsonify(response), status_code
```

Figura 69. Endpoint Actualizar contraseña

- *Aplicación móvil*

Dado que se realizó en Ionic y con react la estructura de carpetas es la siguiente en los components estará el menú, y en pages las siguientes:

- Register: Página de registro de nuevos usuarios.
- ForgotPassword: Página para la recuperación de contraseña.
- Login: Página de inicio de sesión.
- Dashboard: Página principal que muestra los datos de los sensores de temperatura, humedad del ambiente, velocidad del viento, así como las gráficas de los sensores de la humedad del suelo.
- Activities: Página que muestra el historial del riego en sus diferentes modos automático y manual.
- Profile: Página del perfil del usuario.

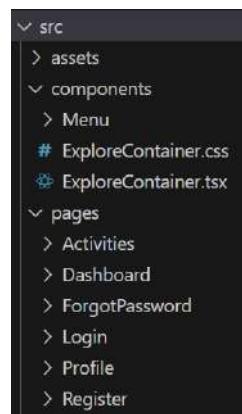


Figura 70. Estructura de las carpetas del sistema.

Register



Figura 71. Formulario para registrar usuario

```

const handleSubmit = async (e: React.FormEvent) => {
  e.preventDefault();

  try {
    const response = await fetch('https://villacrescode.com/pruebappy/user', {
      method: 'POST',
      headers: [
        'Content-Type': 'application/json'
      ],
      body: JSON.stringify(formData)
    });

    const result = await response.json();
    if (response.ok) {
      router.push('/login','root','replace')
    } else {
      setError(`Error al ingresar: ${result.message}`);
    }
  } catch (error) {
    console.error('Error al realizar la solicitud:', error);
  }
};
  
```

Figura 72. Función para registrar usuario- Frontend

En la Figura 71 se muestra el registro que incluye campos para Nombre, Apellido, Dirección, Teléfono, Correo y Contraseña, además de un botón para registrar y un enlace para iniciar sesión si ya se tiene cuenta. El formulario recoge los datos del usuario, que se envían como `formData` en la solicitud POST de la Figura 72. Si el registro es exitoso, el usuario es redirigido; si hay un error, se muestra un mensaje de error.

Forgot password



Figura 73. Recuperar contraseña

```
const handleSend = async () => {
  const email = emailInputRef.current?.value as string;
  if (!email) {
    setAlertMessage('Por favor, ingresa un correo');
    setShowAlert(true);
    return;
  }

  try {
    const response = await fetch('https://villacrescode.com/pruebapay/forgotPassword', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ email })
    });

    const result = await response.json();

    if (result.ok) {
      setAlertMessage('Se envió la información correspondiente a tu correo');
      setShowAlert(true);
      setTimeout(() => {
        history.push('/login');
      }, 3000);
    } else {
      setAlertMessage(result.error || 'Error al enviar el correo');
      setShowAlert(true);
    }
  } catch (error) {
    setAlertMessage('Error al enviar el correo');
    setShowAlert(true);
  }
};
```

Figura 74. Función recuperar contraseña- Frontend

La Figura 73 permite a los usuarios ingresar su correo electrónico para recibir instrucciones de restablecimiento, como se refleja en el campo de texto y el botón "ENVIAR". En la Figura 74 gestiona este proceso al capturar el correo ingresado, verificar su presencia y enviar una solicitud POST a la URL `https://villacrescode.com/pruebapay/forgotPassword`. Si la solicitud es exitosa, se

informa al usuario que la información se ha enviado a su correo y se redirige a la página de inicio de sesión; de lo contrario, se muestra un mensaje de error.

Login

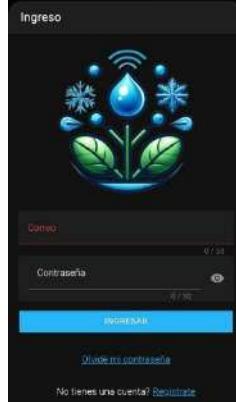


Figura 75. Ingreso al sistema

```
const handleSubmit = async (e: React.FormEvent) => {
  e.preventDefault();
  try {
    const response = await fetch('https://villacrescode.com/pruebapay/login', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(formData)
    });
    const result = await response.json();
    if (response.ok) {
      setError('');
      console.log(result.idUser);
      sessionStorage.setItem('idUser', result.idUser);
      router.push('/dashboard', 'root', 'replace');
    } else {
      throw new Error(result.error || 'Correo o contraseña no son correctos');
    }
  } catch (error) {
    setError(error as Error).message;
  } finally {
    setIsSubmitting(false);
  }
};
```

Figura 76. Función ingresar usuario- Frontend

En el inicio de sesión Figura 75 muestra campos para que los usuarios ingresen su correo electrónico y contraseña, junto con opciones para "INGRESAR" y enlaces para recuperar la contraseña o registrarse si no tienen una cuenta. En la Figura 76 maneja el proceso para enviar una solicitud POST a la URL `https://villacrescode.com/pruebapay/login` con los datos del formulario. Si la respuesta es exitosa, el código almacena el ID del usuario en el almacenamiento de sesión y redirige al usuario al panel de control; si la respuesta es negativa, se muestra un error indicando que el correo o la contraseña son incorrectos.

Dashboard

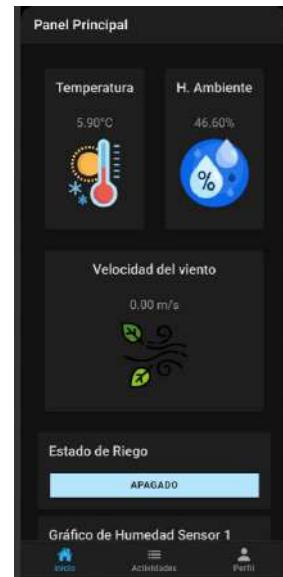


Figura 77. Pantalla principal datos sensores

```
useEffect(() => {
  notifyPush()
  const fetchWeatherData = async () => {
    try {
      const response = await fetch('https://villacrescode.com/pruebapy/sensor');
      const data = await response.json();
      setWeatherData({
        temperatura: `${data.Temperatura}°C`,
        humedadAmbiente: `${data.Humedad}%`,
        velocidadViento: `${data.VelocidadViento} m/s`,
        humedadSuelo: `${data.HumedadSuelo}%`,
        timeStamp: data.FechaHora
      });
    } catch (error) {
      console.error('Error fetching weather data:', error);
    }
  };
};
```

Figura 78. Función obtener los datos de los sensores - Frontend

```
const fetchIrrigationStatus = async () => {
  try {
    const response = await fetch('https://villacrescode.com/pruebapy/lastIrrigationStatus');
    const data = await response.json();
    setIrrigationStatus(data.IrrigationStatus);
  } catch (error) {
    console.error('Error fetching weather data:', error);
  }
};
```

Figura 79. Función obtener el ultimo estado del riego- Frontend

```

const fetchHumidityData = async () => {
  try {
    const response = await fetch('https://villacrescode.com/pruebapay/humidity');
    const data = await response.json();
    const formattedData = data.map((item: any, index: number) => ({
      index: index + 1,
      hum1: item.HumidityGround1,
      hum2: item.HumidityGround2,
      hum3: item.HumidityGround3,
      hum4: item.HumidityGround4,
      formated_date:item.formatted_date
    }));
    setHumidityData(formattedData);
  } catch (error) {
    console.error('Error fetching humidity data:', error);
  }
};

```

Figura 80. Función obtener datos de sensores del suelo- Frontend

En la Figura 77 se muestra los datos sobre el estado del clima y el sistema de riego, como la temperatura, la humedad ambiental, la velocidad del viento y el estado del riego. En la Figura 78 se encarga de obtener datos meteorológicos como la temperatura, humedad ambiental y velocidad del viento desde la URL `https://villacrescode.com/pruebapay/sensor`. En la Figura 79 obtiene el estado del riego desde `https://villacrescode.com/pruebapay/lastIrrigationStatus`. En la Figura 80 recupera los datos de humedad del suelo desde `https://villacrescode.com/pruebapay/humidity` y los formatea adecuadamente. Todos estos datos se utilizan para actualizar la interfaz de usuario en tiempo real, proporcionando a los usuarios información actualizada sobre las condiciones ambientales y el estado del sistema de riego.

Activities



Figura 81. Actividades del riego

```

const Activities: React.FC = () => {
  const [data, setData] = useState<Activity[]>([[]]);
  const [expanded, setExpanded] = useState<number | null>(null);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await fetch('https://villacrescode.com/pruebapay/irrigationHistory');
        const result = await response.json();
        setData(result);
      } catch (error) {
        console.error('Error fetching data:', error);
      }
    };
  });
}

```

Figura 82. Función registrar el historial del riego- Frontend

La pantalla de "Actividades" Figura 81 un historial de eventos relacionados con el estado del sistema de riego, indicando si el sistema fue encendido o apagado de forma manual o automática, junto con detalles como la fecha y hora, humedad, temperatura y velocidad del viento en esos momentos. En la Figura 82 se realiza una solicitud asíncrona a la URL `https://villacrescode.com/pruebapay/irrigationHistory` para obtener el historial de riego, estableciendo estos datos en el estado una vez recibidos.

Profile

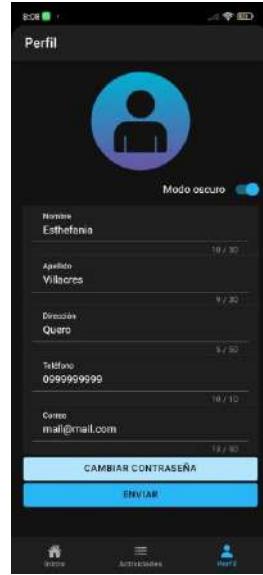


Figura 83. Actualizar usuario

```

useEffect(() => {
  const fetchDataUser = async () => {
    try {
      const idUser = sessionStorage.getItem('idUser');
      const response = await fetch(`https://villacrescode.com/pruebapay/user/${idUser}`, {
        method: 'GET',
        headers: {
          'Content-Type': 'application/json'
        }
      });
      const data = await response.json();
      console.log(data);
      if (response.ok) {
        setUserInfo(data);
      } else {
        console.error('Failed to fetch user data');
      }
    } catch (error) {
      console.error('Error fetching user data:', error);
    }
  };
  fetchDataUser();
}, []);

```

Figura 84. Función editar usuario- Frontend

```

const updatePassword = async () => {
  if (passwordData newPassword !== passwordData confirmPassword) {
    setError('Las contraseñas nuevas no coinciden');
    return;
  }

  try {
    const idUser = sessionStorage.getItem('idUser');
    const response = await fetch(`https://villacrescode.com/pruebapay/updatePassword`, [
      method: 'PUT',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        user_id: idUser,
        old_password: passwordData.oldPassword,
        new_password: passwordData.newPassword
      })
    ]);

    const result = await response.json();
    if (response.ok) {
      setError('');
      setSuccess(true); // Mostramos mensaje de éxito
      setTimeout(() => {
        setSuccess(false); // Ocultar mensaje de éxito después de unos segundos
      }, 1000);
    } else {
      setError(result.error);
    }
  } catch (error) {
    console.error('Error al actualizar la contraseña:', error);
  }
};

```

Figura 85. Función actualizar usuario- Frontend

En el perfil Figura 83 permite a los usuarios ver y editar sus datos personales, incluyendo nombre, apellido, dirección, teléfono y correo electrónico. Además, ofrece la opción de cambiar la contraseña y activar o desactivar el modo oscuro. En la Figura 84 se encarga de obtener la información del usuario al cargar el componente, haciendo una solicitud GET a la URL `https://villacrescode.com/pruebapay/user/` con el ID del usuario almacenado en la sesión. En la Figura 85 permite al usuario actualizar su contraseña, verificando que la nueva contraseña coincide con la confirmación antes de enviar una solicitud PUT a la URL `https://villacrescode.com/pruebapay/updatePassword`, incluyendo en el cuerpo de la solicitud el ID del usuario, la contraseña antigua y la nueva.

2.2.5 Fase IV: Pruebas del prototipo

a. *Pruebas de aceptación*

Las pruebas de aceptación se llevaron a cabo con el propósito de verificar que todas las historias de usuario especificadas al inicio del proyecto (Fase I: Recopilación de requisitos) se cumplieran. Para su ejecución, se organizaron según las iteraciones establecidas.

Tabla 50. Pruebas de aceptación-Iteración 1

Número	Historia de Usuario	Criterio	Estado	Responsable
1	001	Los sensores están correctamente conectados a los pines designados en el microcontrolador.	Aceptado	Desarrollador
2	001	Verificar que cada sensor está enviando datos válidos al sistema.	Aceptado	Desarrollador
3	001	Los datos de los sensores se integran correctamente con la lógica del sistema.	Aceptado	Desarrollador
4	002	Los datos de los sensores se transmiten correctamente a través de la red Lora.	Aceptado	Desarrollador
5	002	Verificar que el receptor recibe y procesa correctamente los datos enviados.	Aceptado	Desarrollador
6	002	Evaluar la distancia máxima a la que los datos se pueden transmitir y recibir sin pérdida de información,	Aceptado	Desarrollador
7	003	Simular condiciones que activen el riego automáticamente según los umbrales establecidos (temperatura, humedad, viento).	Aceptado	Desarrollador
8	003	Comprobar que la bomba de riego se enciende automáticamente cuando se cumplen los umbrales.	Aceptado	Agricultor
9	003	El sistema notifique correctamente el estado de activación del riego	Aceptado	Agricultor

Número	Historia de Usuario	Criterio	Estado	Responsable
10	003	Simular condiciones que desactiven el riego automáticamente y verificar que la bomba se apague.	Aceptado	Desarrollador

Tabla 51. Pruebas de aceptación-Iteración II

Número	Historia de Usuario	Criterio	Estado	Responsable
1	004	Comparar los datos mostrados en la aplicación con los valores reales obtenidos de los sensores.	Aceptado	Desarrollador
2	004	Mostar los datos de los sensores en la interfaz de manera legible y clara.	Aceptado	Desarrollador
3	005	El riego se activa al presionar el botón.	Aceptado	Agricultor
4	005	El riego se desactiva al presionar el botón.	Aceptado	Agricultor
5	005	Muestra mensaje de confirmación para apagar el riego.	Aceptado	Agricultor
6	005	Al activar el riego muestra ventana para establecer el tiempo.	Aceptado	Agricultor
7	005	El tiempo de respuesta del sistema debe ser menor a 1 min.	Aceptado	Agricultor
8	006	Verificar que las actividades de riego (manual y automático) se registran en el sistema.	Aceptado	Agricultor
9	006	Confirmar que el historial de actividades se muestra correctamente en la aplicación.	Aceptado	Agricultor
10	006	Cada entrada en el historial contiene detalles completos (fecha, hora, tipo de riego, valores de los sensores).	Aceptado	Agricultor
11	007	El sistema envía notificaciones al usuario cuando se activa o desactiva el riego.	Aceptado	Agricultor

Número	Historia de Usuario	Criterio	Estado	Responsable
12	007	Las notificaciones contienen información clara y relevante sobre el estado del riego.	Aceptado	Agricultor
13	007	Las notificaciones se reciben correctamente en diferentes dispositivos móviles.	Aceptado	Desarrollador

Tabla 52, Pruebas de aceptación-Iteración III

Número	Historia de Usuario	Criterio	Estado	Responsable
1	008	El formulario de registro acepta y valida todos los campos necesarios (nombre, apellido, dirección, teléfono, correo electrónico, contraseña).	Aceptado	Agricultor
2	008	El sistema valida correctamente los datos ingresados para el cambio de contraseña (contraseña anterior coincide con la de la base y las contraseñas nuevas son iguales).	Aceptado	Agricultor
3	008	El sistema crea una nueva cuenta de usuario y almacena los datos en la base de datos.	Aceptado	Desarrollador
4	008	Probar que el usuario puede iniciar sesión inmediatamente después de registrarse.	Aceptado	Agricultor
5	009	Verificar que el formulario de inicio de sesión acepta y valida el correo electrónico y la contraseña.	Aceptado	Desarrollador
6	009	El sistema autentica correctamente al usuario con las credenciales proporcionadas.	Aceptado	Desarrollador
7	009	El usuario autenticado puede acceder a todas las funcionalidades de la aplicación.	Aceptado	Agricultor
8	010	El formulario de actualización de usuario acepta y valida los cambios en los datos personales.	Aceptado	Desarrollador

Número	Historia de Usuario	Criterio	Estado	Responsable
9	010	El sistema valida correctamente los datos ingresados (formato de correo electrónico, longitud de la contraseña).	Aceptado	Desarrollador
10	010	Los cambios se reflejan correctamente en la base de datos.	Aceptado	Desarrollador
11	011	Verificar que el formulario de recuperación de contraseña acepta el correo electrónico del usuario.	Aceptado	Agricultor
12	011	El sistema verifica si el correo electrónico está registrado en la base de datos.	Aceptado	Agricultor
13	011	Verificar que el sistema envía un correo electrónico con las instrucciones para restablecer la contraseña.	Aceptado	Agricultor
14	011	Probar que el usuario puede restablecer su contraseña y acceder al sistema con la nueva contraseña.	Aceptado	Agricultor
15	012	Verificar que el panel este correctamente instalados y conectados al sistema.	Aceptado	Agricultor
16	012	La placa de emisión de datos esté funcionando con la energía suministrada por el panel solar.	Aceptado	Agricultor

b. Pruebas de usabilidad de la aplicación móvil

A continuación, las pruebas de usabilidad aplicada a la aplicación móvil se van a evaluar 5 aspectos[71].

Tabla 53. Resultado de las pruebas de usabilidad

Id	Aspecto evaluado	Descripción	Observación	Estado
1	Simplicidad y claridad en el diseño	Interfaz de fácil compresión, diseño visual claro y jerarquía en la información.	Las interfaces de la aplicación móvil presentan un diseño limpio con campos claramente etiquetados para ingreso de la información, así como para la visualización.	Aceptado
2	Fácil navegación	Navegación intuitiva, coherente y con botones etiquetados	Los menús y botones están claramente etiquetados, facilitando el acceso a diferentes secciones como inicio, actividades y perfil.	Aceptado
3	Retroalimentación	Mensajes que indiquen que sucede con la aplicación.	Mensajes al cambiar la contraseña o guardar la información del perfil además de mensajes de espera cuando está realizando una acción.	Aceptado
4	Legibilidad y contaste	Fuentes de tamaños uniforme con colores y contrastes visualmente distinguibles.	Las secciones de ingreso, registro y recuperación de contraseña son fáciles de leer, incluso cuando el modo del dispositivo se encuentre en oscuro.	Aceptado
5	Accesibilidad	Se debe ajustar la aplicación para el uso de personas con capacidades especiales.	La aplicación puede ser utilizada por todos, pero no se ha implementado comandos por voz una guía para su manejo.	Por implementar (en un futuro si lo requieren los usuarios)

c. Pruebas de distancia comunicación Lora

Tabla 54.Pruebas del alcance de comunicación Lora

N	Potencia de transmisión(dBm)	Distancia(m)	Condición climática	Tiempo de respuesta
1	15	104	lluvioso	5 seg
2	15	110	lluvioso	5 seg
3	15	150	soleado	5 seg
4	15	265	soleado	5 seg
5	15	300	soleado	6 seg
6	15	425	soleado	7 seg
7	15	450	soleado	7 seg
8	15	510	soleado	8 seg
9	15	610	soleado	10 seg
10	15	1010	soleado	30 seg
11	15	1025	soleado	1 min
12	15	1030	soleado	1 min

Se evaluó el alcance de la transmisión de datos bajo diferentes condiciones climáticas, manteniendo siempre una línea de vista clara. Estas pruebas se realizaron utilizando el receptor que se dejó en la vivienda y el emisor se desplazó a diferentes distancias. En condiciones de lluviosas la distancia máxima de comunicación fue 100 metros y en un día soleado se midió hasta 1030 metros y la comunicación fue satisfactoria con un tiempo de respuesta de 1 min.

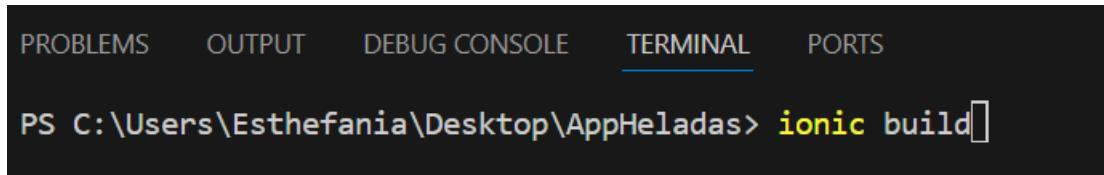
2.2.6 Fase V: Implantación

a. Configuración de la base de datos

La configuración se realizó en un hosting de Banahosting que tiene un almacenamiento de 1GB para la base de datos y para los archivos de 2GB, se procedió a configurar la ip del pc que esta como servidor local para que reciba los datos, donde se ubica la ip de hosting, nombre de usuario y la contraseña por motivos de seguridad no se mostrara en esta sección.

b. Despliegue

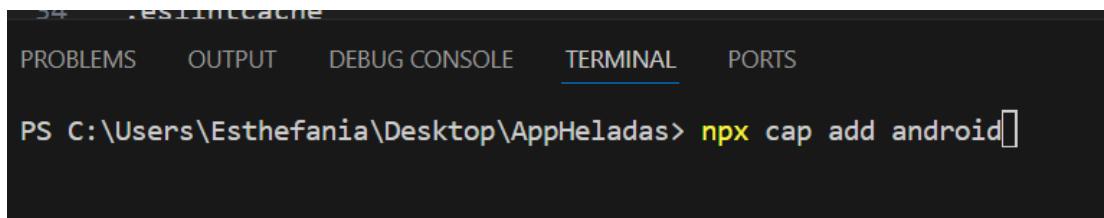
El comando `ionic build` genera una versión optimizada del proyecto para producción. Convierte el código fuente de la aplicación en archivos que se pueden desplegar en un servidor o empaquetar para una tienda de aplicaciones.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
PS C:\Users\Esthefania\Desktop\AppHeladas> ionic build
```

Figura 86. Construcción proyecto Ionic

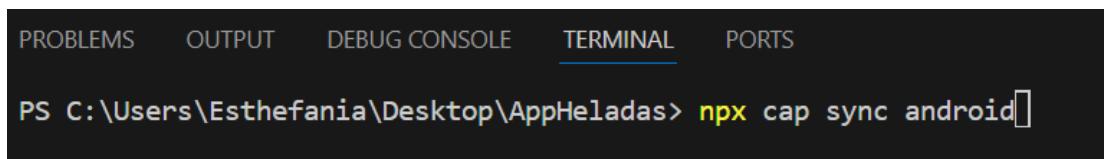
El comando `npx cap add android` se utiliza para añadir la plataforma Android a un proyecto Ionic mediante Capacitor, un runtime que permite ejecutar aplicaciones web como aplicaciones móviles nativas. Este comando configura los archivos y directorios necesarios para compilar y ejecutar la aplicación en dispositivos Android, integrando el proyecto con Android Studio y permitiendo acceder a las funcionalidades nativas de Android desde el código JavaScript/TypeScript.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
PS C:\Users\Esthefania\Desktop\AppHeladas> npx cap add android
```

Figura 87. Capacitor Android

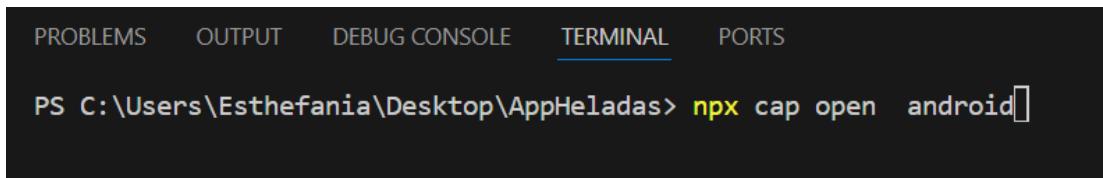
El comando `npx cap sync android` se utiliza para sincronizar los cambios realizados en el proyecto Ionic con la plataforma Android configurada mediante Capacitor. Este comando asegura que los archivos y configuraciones más recientes del proyecto web se copien a la carpeta nativa de Android, manteniendo ambas partes del proyecto en sincronía. Esto incluye actualizaciones en dependencias, archivos estáticos, configuraciones y cualquier cambio relevante que necesite reflejarse en la versión Android de la aplicación.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
PS C:\Users\Esthefania\Desktop\AppHeladas> npx cap sync android
```

Figura 88. Sincronización del proyecto

El comando `npx cap open android` se utiliza para abrir el proyecto Ionic configurado para Android en Android Studio. Este comando lanza automáticamente Android Studio y carga el proyecto en su entorno de desarrollo integrado (IDE), permitiendo a los desarrolladores compilar, depurar y ejecutar la aplicación en un emulador o dispositivo Android. Es una forma conveniente de acceder rápidamente al proyecto Android desde la línea de comandos.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\Esthefania\Desktop\AppHeladas> npx cap open android
```

Figura 89. Abrir Android Studio

Para generar un archivo APK en Android Studio para distribuir e instalar una aplicación en dispositivos Android. En la barra de menú superior selecciona `Build`, elige `Build Bundle(s) / APK(s)` y selecciona `Build APK(s)`. Esto iniciará el proceso de compilación, que puede tardar unos minutos.

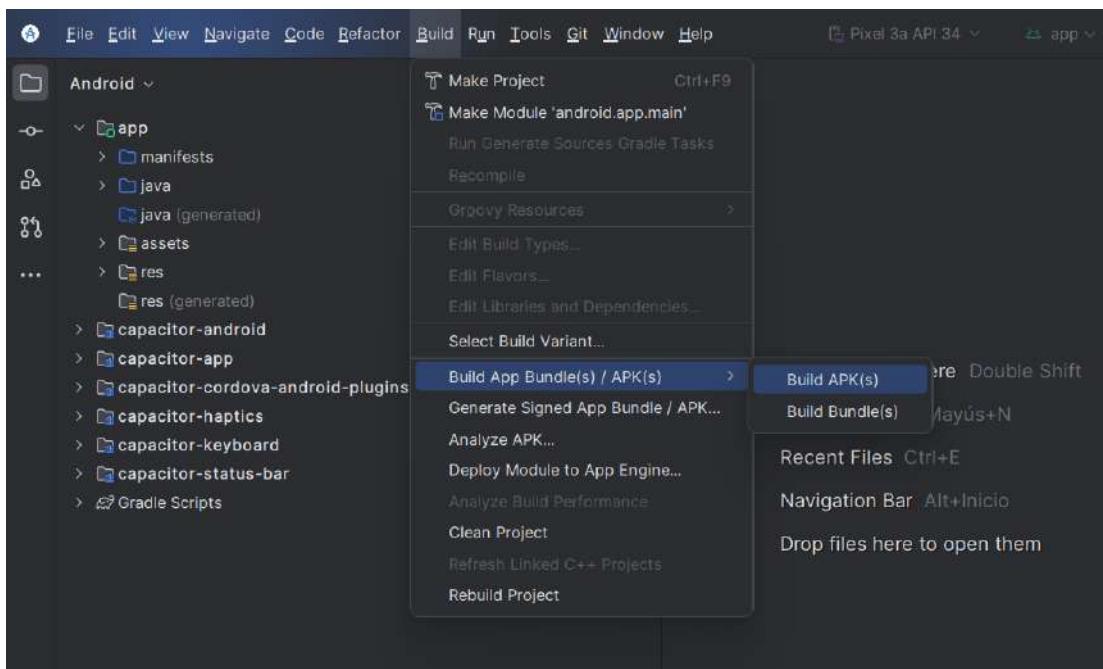


Figura 90. Generación de apk

Una vez completado, Android Studio notificará la ubicación del archivo APK generado, generalmente en el directorio `app/build/outputs/apk/`. Este archivo APK

puede ser utilizado para pruebas en dispositivos reales, distribución a testers o publicación en tiendas de aplicaciones.

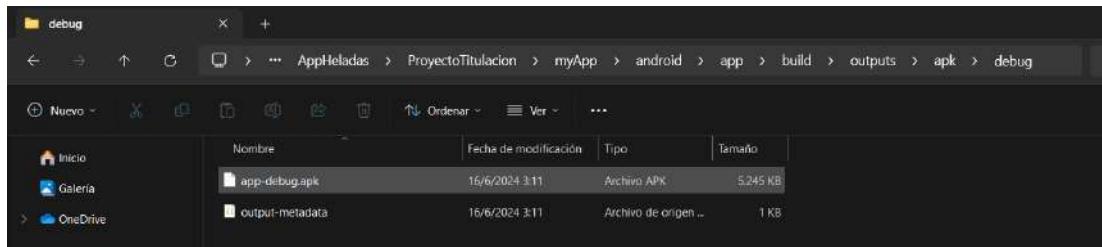


Figura 91. Apk instalable Android

El terreno de pruebas está ubicado en el caserío Jaloa Alto El Guasmo perteneciente al cantón Quero, el cual tiene una superficie aproximada de 200 metros cuadrados.



Figura 92. Terrero de pruebas del prototipo

Para la parte de hardware se procedió a instalar por módulos que se describirán a continuación:

c. Módulo de alimentación

El módulo de alimentación está constituido por un panel solar, un controlador de carga y una batería. El panel está ubicado en la parte superior del techo de la casa, el controlador de carga se lo instaló junto a la batería para que reciba la corriente y la almacene. Y será el responsable de suministrar energía a la placa de envío de datos.



Figura 93. Panel Solar



Figura 94. Controlador de carga



Figura 95. Batería

d. Módulo de captura y envío de datos

Este módulo se los colocó justo al controlador de carga para que le brinde energía necesaria para su funcionamiento, además que los sensores de humedad del suelo están conectados mediante cable al terreno de pruebas, el sensor de humedad y

temperatura relativa se lo ubicó al costado del sembrío y el sensor de velocidad del viento está en la parte superior de la casa.

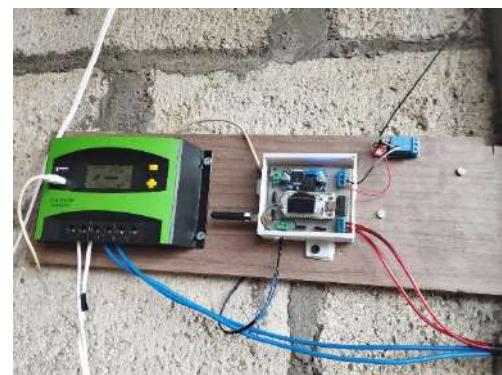


Figura 96. Emisor de datos



Figura 97. Sensor de velocidad del viento



Figura 98. Sensor de humedad del suelo



Figura 99. Sensor de temperatura y humedad del ambiente

e. Módulo de recepción y procesamiento de datos

Está ubicado en la parte interior de la casa ya que la comunicación lora permite el envío de datos incluso a través de obstáculos como en este caso las paredes. Es el encargado de recibir los datos y a través de comunicación MQTT se conecta la backend local que es el responsable del envío de la información a la base de datos del hosting.



Figura 100. Receptor de datos

f. Módulo de riego

Los aspersores se colocaron tomando en cuenta el diámetro del riego de cada uno por lo cual son tres hileras de 3 micro aspersores cada una que cubrió todo el terreno de

pruebas. La bomba esta instala junto al relé que es el encarado del controlar el encendido o pagado de acuerdo con las necesidades climáticas o del usuario.



Figura 101. Bomba de agua



Figura 102. Micro aspersores

2.3 Resultados

Se aplicó el Modelo de Aceptación Tecnológica (TAM) en inglés, para verificar el impacto de la aplicación móvil en los agricultores que en este caso son los únicos usuarios del sistema.

2.3.1 Modelo de aceptación de tecnología TAM

El Modelo de Aceptación Tecnológica (TAM) es un marco teórico utilizado para evaluar la adopción y aceptación de sistemas informáticos o tecnologías por parte de

los usuarios. Este modelo se centra en dos variables principales: la utilidad percibida (Perceived Usefulness, PU) y la facilidad de uso percibida (Perceived Ease of Use, PEU). La utilidad percibida se refiere al grado en que un usuario cree que utilizar una determinada tecnología mejorará su desempeño laboral, mientras que la facilidad de uso percibida se relaciona con la medida en que el usuario considera que la tecnología es fácil de aprender y utilizar[72].

2.3.2 Cuestionario para el modelo TAM

Para las respuestas del cuestionario se empleó una escala de Likert en la que 5 representa el valor más alto y 1 el más bajo. Esta escala permite medir el grado de acuerdo del usuario con las afirmaciones presentadas, distribuyéndose de la siguiente manera:

- 5: Totalmente de acuerdo.
- 4: De acuerdo.
- 3: Neutral.
- 2: En desacuerdo.
- 1: Totalmente en desacuerdo.

Tabla 55. Cuestionario de evolución del modelo TAM

PU	Utilidad percibida
PU1	¿Cree que la aplicación móvil mejora la monitorización y el control de los cultivos de papa frente a las heladas?
PU2	¿En qué medida considera que la aplicación es eficaz para activar automáticamente el riego cuando se detecten condiciones climáticas adversas como una helada?
PU3	¿Qué tan útil considera la funcionalidad de control remoto del sistema de riego a través de la aplicación móvil?
PU4	¿Cree que los datos y reportes generados por la aplicación facilitan la toma de decisiones para la protección de los cultivos de papa?
FUP	Facilidad de uso percibida
PEU1	¿Considera que la interfaz de usuario de la aplicación es intuitiva y fácil de navegar?
PEU2	¿Considera que la curva de aprendizaje para usar la aplicación es baja y no requiere mucho tiempo?
PEU3	¿La aplicación proporciona una retroalimentación clara (mensajes de éxito y error) durante su uso?
PEU4	¿Considera que la aplicación proporciona una experiencia coherente en cuanto a diseño y funcionalidad, sin importar el dispositivo que se utilice (tablet o celular)?

2.3.3 Resultados del modelo TAM

El cuestionario del Modelo TAM de la Tabla 56 se realizó a 10 agricultores de la zona y los resultados obtenidos son los siguientes:

Tabla 56. Resultados modelo TAM-Utilidad percibida

PU	Totalmente de acuerdo	De acuerdo	Neutral	En desacuerdo	Totalmente en desacuerdo
PU1	20%	70%	10%	0%	0%
PU2	40%	40%	20%	0%	0%
PU3	50%	40%	10%	0%	0%
PU4	10%	50%	40%	0%	0%

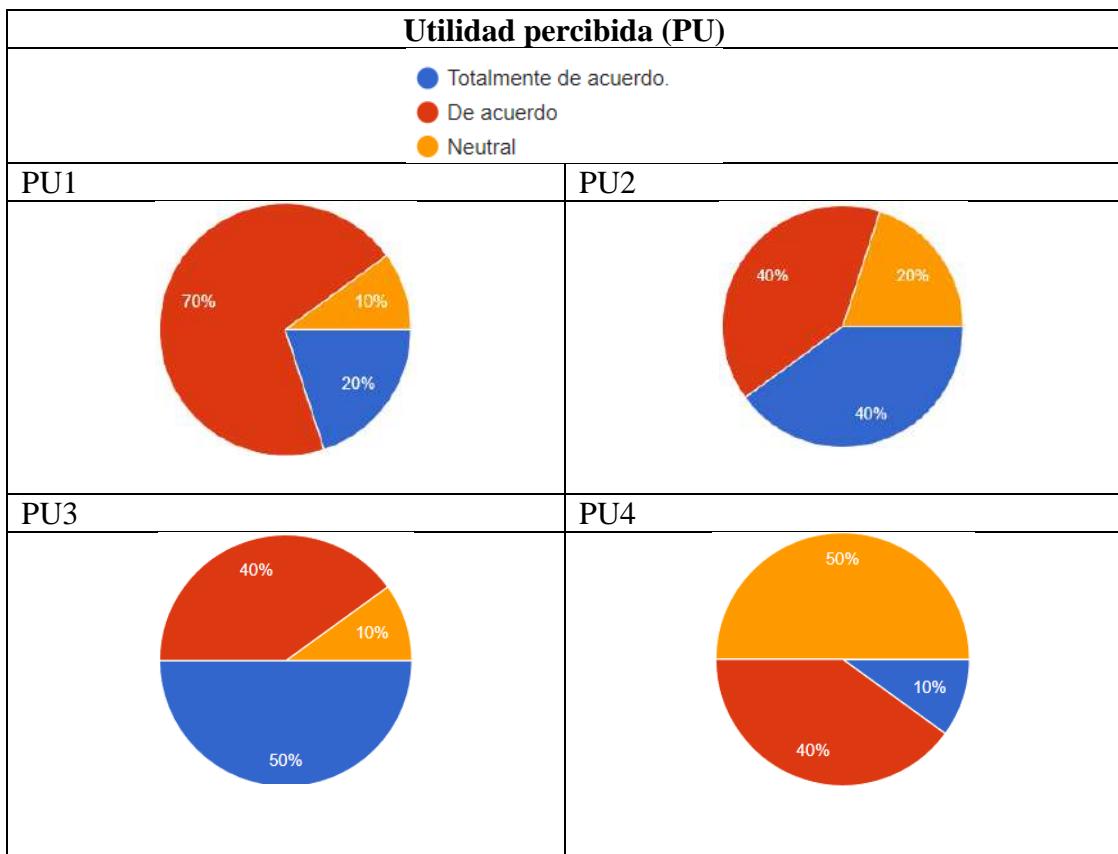


Figura 103. Resultados modelo TAM-Utilidad percibida

Se observa que los agricultores en su mayoría consideran que la aplicación móvil junto con el sistema electrónico es útil para sus cultivos y ayudara a la protección de las heladas.

Tabla 57. Resultados modelo TAM-Utilidad percibida

PEU	Totalmente de acuerdo	De acuerdo	Neutral	En desacuerdo	Totalmente en desacuerdo
PEU1	20%	70%	10%	0%	0%
PEU2	0%	60%	40%	0%	0%
PEU3	10%	70%	20%	0%	0%
PEU4	10%	70%	20%	0%	0%

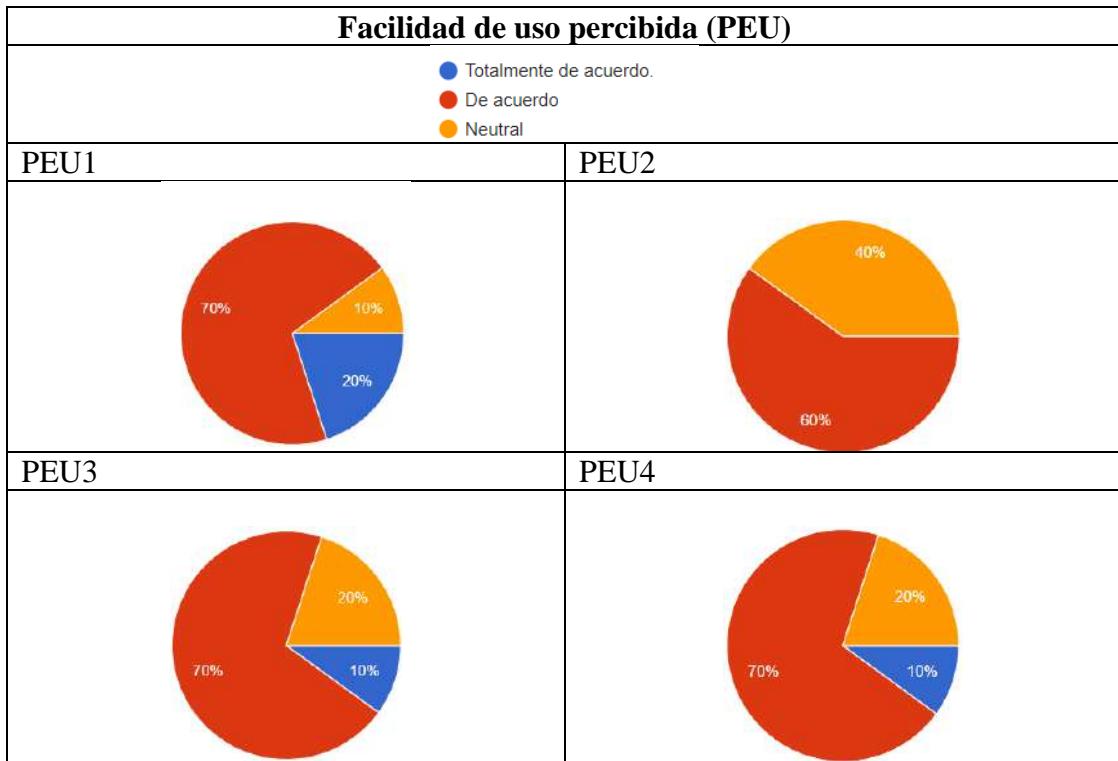


Figura 104. Resultados modelo TAM-factibilidad de uso percibida

Con los resultados de la encuesta sobre la factibilidad de uso percibida se observa que los usuarios están de acuerdo con que la aplicación es intuitiva y proporcionando las retroalimentaciones respectivas para su manejo.

2.4 Presupuesto

A continuación, se lista los materiales que se usaron para la construcción del prototipo y su implantación en un terreno de 200 metros cuadrados aproximadamente.

Tabla 58. Presupuesto del prototipo

Número	Materiales	Medida	Cantidad	Precio Unitario	Total
1	Sensor de Temperatura/humedad DTH11	unidad	1	3,5	3,5
2	Sensor de humedad del suelo FC-28	unidad	4	4,2	16,8
3	Sensor de velocidad del viento TERRIFIC	unidad	1	70	70
4	Placa 915MHz ESP32 LORA V3	unidad	2	45	90
5	Bomba de agua Mariolo 0.5 HP 0.37 KW	unidad	1	80	80
6	Relé 5V	unidad	1	4	4
7	Panel solar 120W 12V	unidad	1	120	120
8	Controlador de carga 100A 12V	unidad	1	50	50
9	Batería 100A 12V	unidad	1	70	70
10	Case placas lora	unidad	2	10	20
11	Baquelita	unidad	1	40	40
12	Cable flexible calibre 12	metros	10	1,8	18
13	Cable flexible calibre 16	metros	100	0,2	20
14	Micro aspersor rondo + estaca	unidad	12	1,5	18
15	Tubería de plástico 3/4	metros	60	0,4	24
16	Tubería de plástico 1/2	metros	80	0,4	32
17	Válvula de retención 1/2	unidad	1	20	20
18	Mano de obra de construcción electrónica	horas	76	1,99	151,24
19	Mano de obra de desarrollador de software	horas	352	1,99	700,48
20	Luz eléctrica	meses	6	20	120
21	Dominio	meses	6	2	12
22	Hosting	meses	6	5	30
Subtotal					1453,517
IVA 15%					256,503
Total					1710,02

Como se muestra en la Tabla 58, el costo total del proyecto es de \$ 1710,03 dólares.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

3.1 Conclusiones

- El sistema ha demostrado ser eficiente para monitorear las condiciones climáticas críticas, como la temperatura, la humedad y la velocidad del viento, que son fundamentales para detectar y mitigar los efectos de las heladas en los cultivos. La implementación de energías renovables, como los paneles solares, ha garantizado una fuente de energía sostenible y autónoma, lo que ha reducido los costos operativos a largo plazo.
- La tecnología Lora ha sido identificada como la opción más adecuada para la transmisión de datos en condiciones de heladas, debido a su capacidad de largo alcance y bajo consumo de energía. Además, su infraestructura de red sencilla y su capacidad de implementación a bajo costo hacen que sea accesible para agricultores.
- Con el desarrollo del prototipo electrónico que mide y registra datos críticos sobre la temperatura, humedad y velocidad del viento. Este sistema permite una monitorización en tiempo real de las condiciones climáticas, proporcionando datos que son esenciales para la gestión del riego en condiciones adversas, como las que se encuentran en la zona alta del cantón Quero.
- La aplicación móvil desarrollada proporciona una plataforma intuitiva y accesible para que los agricultores gestionen la información sobre las condiciones climáticas (heladas) que afectan a sus cultivos. La interfaz de usuario está diseñada para facilitar el acceso y la comprensión de los datos recopilados por el prototipo electrónico, permitiendo a los usuarios recibir alertas y tomar decisiones informadas en tiempo real.

3.2 Recomendaciones

- Se sugiere la expansión del proyecto a múltiples ubicaciones, integrando la funcionalidad de seguimiento geolocalizado para gestionar eficientemente el riego en diferentes parcelas. Esto permitirá a los agricultores monitorear y controlar varios sitios de cultivo desde una sola plataforma, optimizando los recursos y mejorando la gestión agrícola a gran escala.
- Se recomienda la investigación de nuevas tecnologías de transmisión de datos que puedan ofrecer mejoras en términos de eficiencia energética, seguridad de datos y cobertura, especialmente en situaciones donde la transmisión de grandes distancias y la falta de línea de vista sean desafíos significativos. Para estos escenarios, se sugiere la implementación de una red de diferentes endpoints de transmisión, que actúen como repetidores o puntos de recolección intermedios para asegurar la transmisión continua y confiable de los datos.
- Para aumentar la autonomía del sistema, se recomienda proporcionar alimentación a la bomba de riego mediante paneles solares u otras fuentes de energía renovable. Esto garantizará que todo el sistema de riego pueda operar de manera completamente independiente de la red eléctrica, reduciendo aún más los costos operativos y aumentando la sostenibilidad del sistema. También agregar sensores adicionales para medir otras variables relevantes, como la radiación solar.
- Con la acumulación de datos históricos, se pueden desarrollar modelos predictivos que no solo anticipen heladas, sino que también optimicen el riego diario según las condiciones climáticas previstas. Esta funcionalidad mejorará la eficiencia del uso del agua y ayudará a mantener la salud óptima de los cultivos, adaptándose de manera dinámica a las necesidades específicas de cada parcela.

REFERENCIAS BIBLIOGRÁFICAS

- [1] C. A. Estacio Rodriguez y A. M. Ariza Garzón, «Sistema de protección ante las heladas, para los cultivos de papa del territorio cundiboyacense. AMKHA», may 2020, Accedido: 9 de abril de 2024. [En línea]. Disponible en: <http://repository.javeriana.edu.co/handle/10554/49885>
- [2] D. Marmolejo Gutarra y J. E. Ruiz Velásquez, «Tolerancia de papas nativas (*Solanum spp.*) a heladas en el contexto de cambio climático», *Sci. Agropecu.*, vol. 9, n.º 3, pp. 393-400, jul. 2018, doi: 10.17268/sci.agropecu.2018.03.10.
- [3] «Prototipo de alerta temprana para detección de heladas en las zonas agrícolas de las provincias de Cotopaxi y Tungurahua, el cual será desarrollado en conjunto y financiado por el Consorcio de Gobiernos Autónomos Provinciales del Ecuador “CONGOPE”», ELÉCTRICA Y ELECTRÓNICA. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://deee-el.espe.edu.ec/proyectos-de-investigacion-2019/prototipo-de-alerta-temprana-para-deteccion-de-heladas-en-las-zonas-agricolas-de-las-provincias-de-cotopaxi-y-tungurahua-el-cual-sera-desarrollado-en-conjunto-y-financiado-por-el-consorcio-de-gobien/>
- [4] «Ministerio de Agricultura y Ganadería – Ministerio de Agricultura y Ganadería, (MAG) es la institución rectora del desarrollo agrario del país, que promueve acciones para el crecimiento sostenible, impulsando el bienestar de los productores, en particular de la agricultura familiar campesina.» Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.agricultura.gob.ec/>
- [5] «Entregan indemnizaciones en Tungurahua por pérdida de cultivos – Ministerio de Agricultura y Ganadería». Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.agricultura.gob.ec/entregan-indemnizaciones-en-tungurahua-por-perdida-de-cultivos/>
- [6] C. D. Arcos Guamán, «Diseño e implementación de un sistema electrónico para el monitoreo, control de la helada y riego, en huertos de fresas por medio de redes de sensores inalámbricos con hardware libre.», bachelorThesis, Escuela Superior Politécnica de Chimborazo, 2017. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/7592>
- [7] L. A. Rivadeneira Avila, «Desarrollo de un prototipo electrónico para control de heladas en cultivos de arándanos para pequeños productores en la Sierra», bachelorThesis, 2021. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/20712>
- [8] S. P. Salazar Lema, «Sistema electrónico de monitoreo climático y detección de heladas para el GAD del cantón Pujilí», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Telecomunicaciones, 2023. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/38477>
- [9] P. A. Panata Buñay, «Software de control meteorológico para la detección de heladas y emisión de alertas tempranas en el Geoportal del Honorable Gobierno Provincial de Tungurahua», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informáticos, 2020. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/30692>

- [10] J. L. Arias Solis, «Sistema electrónico de monitoreo y control para la prevención de heladas en cultivos de mora, utilizando la Arquitectura IoT.», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Electrónica y Comunicaciones, 2023. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/38342>
- [11] «Construcción de un prototipo electrónico para el monitoreo climatológico en cultivos de aguacate en el municipio de Tona, basado en tecnología IoT.» Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://repository.unad.edu.co/handle/10596/38749>
- [12] «Automated Electronic System to Prevent Freezing Damage to Crops IEEE EDS - SIGHT / Stage 1». Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://repository.usta.edu.co/handle/11634/43010>
- [13] B. E. H. Mauffray, «DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA AUTOMATIZADO DE PROTECCIÓN DE CEREZOS PARA LLUVIAS Y HELADAS».
- [14] F. A. Salamanca Figueroa, C. C. Cárdenas Gamboa, y W. A. Suárez Rodríguez, «Prototipo de monitoreo y alarma para la detección de heladas blancas en sectores rurales de Sogamoso», https://revistas.uptc.edu.co/index.php/ingenieria_sogamoso/article/view/8032/6568, jun. 2018, Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://repositorio.uptc.edu.co//handle/001/2448>
- [15] F. Silva Pérez, «Sistema de control automático contra heladas en huertos de paltos con Raspberry PI 3 y Arduino Uno por medio de Wifi», 2018, Accedido: 9 de abril de 2024. [En línea]. Disponible en: <http://repositorio.unab.cl/xmlui/handle/ria/9431>
- [16] «Processes | Free Full-Text | A Smart Sensors-Based Solar-Powered System to Monitor and Control Tube Well for Agriculture Applications». Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.mdpi.com/2227-9717/10/8/1654>
- [17] G. Chellaian, College, y S. Rosenthaliana, «AGRIOT: Precision Agriculture using IoT and Sensors», ene. 2022.
- [18] U. Iqbal *et al.*, «Application of indigenized soil moisture sensors for precise irrigation of wheat crop under various irrigation methods», *Pak. J. Agric. Res.*, jun. 2022.
- [19] J. Wanyama *et al.*, «Development of a Solar Powered Smart Irrigation Control System Kit», *Smart Agric. Technol.*, vol. 5, p. 100273, jun. 2023, doi: 10.1016/j.atech.2023.100273.
- [20] Y. Lu, Y. Hu, C. Zhao, y R. Snyder, «Modification of Water Application Rates and Intermittent Control for Sprinkler Frost Protection», *Trans. ASABE*, vol. 61, pp. 1277-1285, ene. 2018, doi: 10.13031/trans.12596.
- [21] Y. Lu, Y. Hu, P. Li, K. T. Paw U, y R. L. Snyder, «Prediction of Radiation Frost Using Support Vector Machines Based on Micrometeorological Data», *Appl. Sci.*, vol. 10, n.º 1, Art. n.º 1, ene. 2020, doi: 10.3390/app10010283.
- [22] E. Conlan, T. Borisova, E. Smith, J. Williamson, y M. Olmstead, «The Use of Irrigation for Frost Protection for Blueberry in the Southeastern United States», *HortTechnology*, vol. 28, n.º 5, pp. 660-667, oct. 2018, doi: 10.21273/HORTTECH04008-18.

- [23] S. Cheng, Y. Hu, Y. Lu, P. Qingmin, K. Jin, y J. Zheng, «Investigation on Combination of Airflow Disturbance and Sprinkler Irrigation for Horticultural Crop Frost Protection», *Agric. Sci.*, vol. 5, p. p8, jun. 2023, doi: 10.30560/as.v5n1p8.
- [24] «Emerging Research Advancements to Overcome the Peach Spring Frost». Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.onlinelibrary.wiley.com/journal/view.php?number=1795&viewtype=pubreader>
- [25] R. Yauri, O. Llerena, J. Santiago, y J. Gonzales, «Sprinkler Irrigation Automation System to Reduce the Frost Impact Using Machine Learning», *Int. J. Electr. Comput. Eng. Syst.*, vol. 14, n.º 7, Art. n.º 7, sep. 2023, doi: 10.32985/ijeces.14.7.8.
- [26] C. V. A. Ing, «Desarrollo de aplicación con sensores de temperatura usando una versión del Lenguaje JAVA llamada JAVELIN adecuada para el uso en Microcontroladores que admiten esta tecnología».
- [27] W. P. Jiménez Briones y J. D. Moreira Arreaga, «Desarrollo aplicación móvil orientada a publicación y alerta de datos meteorológicos para consumo urbano, con soporte de prototipo IoT de recolección de almacenamiento de datos.», oct. 2020, Accedido: 14 de diciembre de 2023. [En línea]. Disponible en: <http://repositorio.ug.edu.ec/handle/redug/49598>
- [28] «Firebase | Google's Mobile and Web App Development Platform». Accedido: 15 de diciembre de 2023. [En línea]. Disponible en: <https://firebase.google.com/?hl=es>
- [29] B. Inc, «Back4App - Low-code backend to build modern apps», Back4appTM. Accedido: 15 de diciembre de 2023. [En línea]. Disponible en: <https://www.back4app.com>
- [30] «Welcome to AWS Documentation». Accedido: 14 de diciembre de 2023. [En línea]. Disponible en: <https://docs.aws.amazon.com/>
- [31] «Servicio de aplicaciones web | Microsoft Azure». Accedido: 14 de diciembre de 2023. [En línea]. Disponible en: <https://azure.microsoft.com/es-es/products/app-service/web>
- [32] «¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS», Amazon Web Services, Inc. Accedido: 15 de diciembre de 2023. [En línea]. Disponible en: <https://aws.amazon.com/es/what-is/scrum/>
- [33] Asana, «¿Qué es la programación extrema (XP)? [2022]», Asana. Accedido: 15 de diciembre de 2023. [En línea]. Disponible en: <https://asana.com/es/resources/extreme-programming-xp>
- [34] «Principios agronómicos en el cultivo de la papa | Yara Ecuador». Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.yara.com.ec/nutricion-vegetal/papa/principios-agronomicos-en-el-cultivo-de-la-papa/>
- [35] «Principios agronómicos en el cultivo de la papa | Yara Ecuador», Yara None. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.yara.com.ec/nutricion-vegetal/papa/principios-agronomicos-en-el-cultivo-de-la-papa/>
- [36] Chilebio, «Estudio arroja información clave para desarrollar cultivos resistentes a las heladas», ChileBIO. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.chilebio.cl/2018/10/25/nuevo-estudio-arroja-informacion-clave-para-desarrollar-cultivos-resistentes-a-las-heladas/>
- [37] «Boletín INIA N° 417.pdf». Accedido: 9 de abril de 2024. [En línea]. Disponible en:

- <https://biblioteca.inia.cl/bitstream/handle/20.500.14001/6847/Bolet%C3%ADn%20INIA%20N%C2%B0%20417?sequence=1>
- [38] M. Franquesa, «Tipos de heladas en agricultura: aprende a identificarlas», Agroptima. Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://www.agroptima.com/es/blog/tipos-de-heladas-en-agricultura/>
- [39] «¿Cómo proteger los cultivos de las heladas? - blog.cambiagro.com». Accedido: 9 de abril de 2024. [En línea]. Disponible en: <https://blog.cambiagro.com/2024/01/30/como-proteger-los-cultivos-de-las-heladas/>
- [40] I. N. de E. y Censos, «Estadísticas Agropecuarias», Instituto Nacional de Estadística y Censos. Accedido: 27 de junio de 2024. [En línea]. Disponible en: <https://www.ecuadorencifras.gob.ec/estadisticas-agropecuarias-2/>
- [41] M. Fonseca, «METODOLOGÍAS ÁGILES DEL DESARROLLO DE SOFTWARE».
- [42] «Rapid Application Development (RAD) | Definition, Steps & Full Guide». Accedido: 7 de junio de 2024. [En línea]. Disponible en: <https://kissflow.com/application-development/rad/rapid-application-development/>
- [43] Asana, «Gestiona el trabajo, los proyectos y las tareas de tu equipo en línea • Asana • Asana», Asana. Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://asana.com/es>
- [44] «Jira | Software de seguimiento de proyectos e incidencias | Atlassian». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.atlassian.com/es/software/jira>
- [45] «Gestiona los proyectos de tu equipo desde cualquier lugar | Trello». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://trello.com/es>
- [46] «Git». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://git-scm.com/>
- [47] «Mercurial SCM». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.mercurial-scm.org/>
- [48] «Apache Subversion». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://subversion.apache.org/>
- [49] «Figma Learn - Help Center». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://help.figma.com/hc/en-us>
- [50] «Introducción a Adobe XD». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://helpx.adobe.com/es/xd/get-started.html>
- [51] «Herramienta de Maquetas, Esquemas & Prototipos UI En Línea • Moqups». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://moqups.com>
- [52] «MySQL». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.mysql.com/>
- [53] «MariaDB Server Documentation», MariaDB KnowledgeBase. Accedido: 31 de julio de 2024. [En línea]. Disponible en: <https://mariadb.com/kb/en/documentation/>
- [54] «PostgreSQL: Documentation». Accedido: 31 de julio de 2024. [En línea]. Disponible en: <https://www.postgresql.org/docs/>
- [55] «Ionic Framework - The Cross-Platform App Development Leader», Ionic Framework. Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://ionicframework.com/>

- [56] «Xamarin», Microsoft. Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://dotnet.microsoft.com/es-es/apps/xamarin>
- [57] «Flutter - Build apps for any screen». Accedido: 26 de junio de 2024. [En línea]. Disponible en: [//flutter.dev/](https://flutter.dev/)
- [58] «React – Una biblioteca de JavaScript para construir interfaces de usuario». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://es.legacy.reactjs.org/>
- [59] «Angular». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://docs.angular.lat/>
- [60] «Vue.js - The Progressive JavaScript Framework | Vue.js». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://vuejs.org/>
- [61] «Welcome to Python.org», Python.org. Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.python.org/doc/>
- [62] «JavaScript documentation — DevDocs». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://devdocs.io/javascript/>
- [63] «PHP: ¿Qué es PHP? - Manual». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.php.net/manual/es/intro-whatis.php>
- [64] «Django», Django Project. Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.djangoproject.com/>
- [65] «Welcome to Flask — Flask Documentation (3.0.x)». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://flask.palletsprojects.com/en/3.0.x/>
- [66] «Welcome to Pyramid, a Python Web Framework». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://trypyramid.com/>
- [67]? «Arduino UNO R3 Características, Especificaciones», Proyecto Arduino. Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://proyectoarduino.com/arduino-uno-r3/>
- [68] «Cómo seleccionar y usar el módulo ESP32 con Wi-Fi/Bluetooth adecuado para una aplicación de IoT industrial», DigiKey. Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.digikey.com/es/articles/how-to-select-and-use-the-right-esp32-wi-fi-bluetooth-module>
- [69] «Raspberry Pi 3 Model B+ - Placa base Raspberry - LDLC». Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://www.ldlc.com/es-es/ficha/PB00246555.html>
- [70] «DSpace Angular Universal», Accedido: 27 de junio de 2024. [En línea]. Disponible en: <https://repositorio.inaigem.gob.pe/items/18c6f46c-f37f-4263-a528-bd2b896d002f>
- [71] «Reglas de Usabilidad y Accesibilidad en Aplicaciones Móviles | PDF | Aplicación móvil | Entorno de desarrollo integrado», Scribd. Accedido: 20 de junio de 2024. [En línea]. Disponible en: <https://es.scribd.com/document/715493125/Reglas-de-usabilidad-y-accesibilidad-en-aplicaciones-moviles>
- [72] L. A. Yong Varela, L. A. Rivas Tovar, y J. Chaparro, «Modelo de aceptación tecnológica (TAM): un estudio de la influencia de la cultura nacional y del perfil del usuario en el uso de las TIC», *Innovar*, vol. 20, n.º 36, pp. 187-203, ene. 2010.

ANEXOS

Anexo A. Cuestionario para la encuesta de los agricultores

1. ¿Qué indicadores o señales considera más importantes para determinar cuándo se va a producir una helada?
2. ¿Con qué frecuencia experimenta heladas en su área de cultivo de papas durante la temporada de crecimiento?
3. ¿Qué tan significativas son las pérdidas económicas causadas por las heladas en su cultivo de papas?
4. ¿Qué métodos utilizan actualmente para proteger sus cultivos de papa contra las heladas?
5. ¿Cómo calificaría la efectividad de los métodos actuales para proteger sus cultivos de papas contra las heladas?"
6. ¿Consideraría útil una aplicación que le ayude a monitorear la posibilidad de heladas en sus cultivos de papas?
7. ¿En qué dispositivo le gustaría tener la aplicación?
8. Para proteger sus cultivos de papas contra las heladas, ¿Cuáles de las siguientes opciones considera esenciales o útiles?
9. ¿Qué barreras o limitaciones podrían enfrentar al utilizar esta aplicación móvil? ¿Cuánto estaría dispuesto a invertir en una solución tecnológica que garantice la protección de sus cultivos de papas contra las heladas?

10. ¿Cree usted que el uso de energías renovables es importante en las soluciones tecnológicas para la protección contra heladas en sus cultivos de papas?

11. ¿Qué tipo de tecnologías renovables consideraría más prácticas o accesibles para implementar en sus campos agrícolas?

Anexo B. Solicitudes de validación de la encuesta

VALIDACIÓN DE ENCUESTAS MEDIANTE CONSULTA A EXPERTOS PARA EL DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL EN LA PROTECCIÓN DE CULTIVOS DE PAPA CONTRA HELADAS

Ambato, 20/03/2024

A quien corresponda

Presente. -

De mi consideración:

Yo, Ángel José Beltrán Armendáriz con título de Ingeniero Agrónomo obtenido en la Escuela Superior Politécnica de Chimborazo, he revisado exhaustivamente el contenido de la encuesta presentada. Considero que las preguntas incluidas son claras, pertinentes y adecuadas para el propósito del estudio. Cada ítem está formulado de manera que permite obtener información descriptiva detallada y precisa, necesaria para el análisis propuesto.

Además, las preguntas cubren de manera integral los aspectos críticos del tema de investigación, asegurando que los datos recolectados serán representativos y útiles para alcanzar las conclusiones esperadas. En base a mi experiencia y conocimiento en el área, doy mi aval para el uso de esta encuesta en la recolección de datos del estudio.

Atentamente,



EL SEMBRADOR QUERO
1891803342001

Ing. Ángel José Beltrán Armendáriz
0602710071
0985760522
elsembrioq09@yahoo.com

Figura B 1. Solicitud Ing. Ángel Beltrán

**VALIDACIÓN DE ENCUESTAS MEDIANTE CONSULTA A EXPERTOS PARA
EL DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL EN LA
PROTECCIÓN DE CULTIVOS DE PAPA CONTRA HELADAS**

Ambato, 20/03/2024

A quien corresponda

Presente. -

De mi consideración:

Yo, Claudio Damián Silvia Araujo con título de Ingeniero Agrónomo obtenido en la Universidad Técnica de Ambato, he revisado exhaustivamente el contenido de la encuesta presentada. Considero que las preguntas incluidas son claras, pertinentes y adecuadas para el propósito del estudio. Cada ítem está formulado de manera que permite obtener información descriptiva detallada y precisa, necesaria para el análisis propuesto.

Además, las preguntas cubren de manera integral los aspectos críticos del tema de investigación, asegurando que los datos recolectados serán representativos y útiles para alcanzar las conclusiones esperadas. En base a mi experiencia y conocimiento en el área, doy mi aval para el uso de esta encuesta en la recolección de datos del estudio.

Atentamente,



AGRO-SILVA
Ing. Claudio Damián Silvia Araujo
D.R.C.A. 1804358875
0981309787
claudiodesa1993@yahoo.com

Ing. Claudio Damián Silvia Araujo
D.R.C.A. 1804358875
0981309787
claudiodesa1993@yahoo.com

Figura B 2. Solicitud Ing. Claudio Silva

**VALIDACIÓN DE ENCUESTAS MEDIANTE CONSULTA A EXPERTOS PARA
EL DESARROLLO DE UN PROTOTIPO DE APLICACIÓN MÓVIL EN LA
PROTECCIÓN DE CULTIVOS DE PAPA CONTRA HELADAS**

Ambato, 20/03/2024

A quien corresponda

Presente. -

De mi consideración:

Yo, Robinson Fabricio Moreta Villacres con título en Ingeniero Agrónomo obtenido en la Universidad Técnica de Ambato, he revisado exhaustivamente el contenido de la encuesta presentada. Considero que las preguntas incluidas son claras, pertinentes y adecuadas para el propósito del estudio. Cada ítem está formulado de manera que permite obtener información descriptiva detallada y precisa, necesaria para el análisis propuesto.

Además, las preguntas cubren de manera integral los aspectos críticos del tema de investigación, asegurando que los datos recolectados serán representativos y útiles para alcanzar las conclusiones esperadas. En base a mi experiencia y conocimiento en el área, doy mi aval para el uso de esta encuesta en la recolección de datos del estudio.

Atentamente,



Ing. Robinson Fabricio Moreta Villacres

1804588638

0988258945

robinson.moreta90@gmail.com

Figura B 3. Solicitud Ing. Robinson Moreta

Anexo C. Manual de usuario

Este manual de usuario describe cómo utilizar la aplicación móvil de control y monitoreo para la protección de cultivos de papas contra heladas y sigue la siguiente estructura:

1. Ingreso a la aplicación
2. Registro de usuario
3. Recuperar Contraseña
4. Panel principal datos de los sensores
5. Historial del riego
6. Datos del usuario

1. Ingreso a la aplicación

En esta pantalla, los usuarios pueden acceder a la aplicación proporcionando su correo electrónico y contraseña. Además, hay enlaces para registrar una nueva cuenta o recuperar una contraseña olvidada.

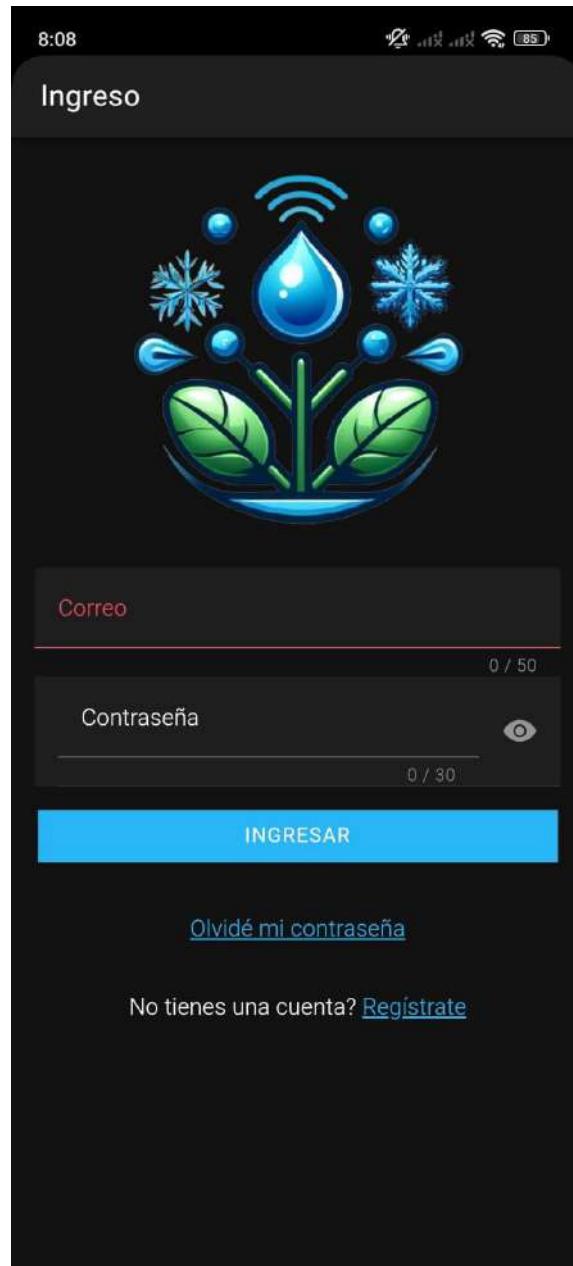


Figura C 1. Ingreso

2. Registro de Usuario

En esta pantalla, los nuevos usuarios pueden registrarse en la aplicación proporcionando la información requerida.

- **Nombre:** Ingrese solo un nombre.

- **Apellido:** Ingrese su apellido.
- **Dirección:** Ingrese la dirección de su cultivo.
- **Teléfono:** Ingrese su número de teléfono.
- **Correo:** Ingrese su dirección de correo electrónico (servirá para el ingreso al sistema).
- **Contraseña:** Cree una contraseña segura de acuerdo con las validaciones del sistema.



Figura C 2. Registro

3. Recuperar Contraseña

Los usuarios pueden recuperar su contraseña ingresando su correo electrónico para recibir una contraseña temporal la cual servirá para el ingreso al sistema se recomienda que una vez ingresado se dirija a la pantalla del perfil de usuario y se procesa con el cambio de contraseña.

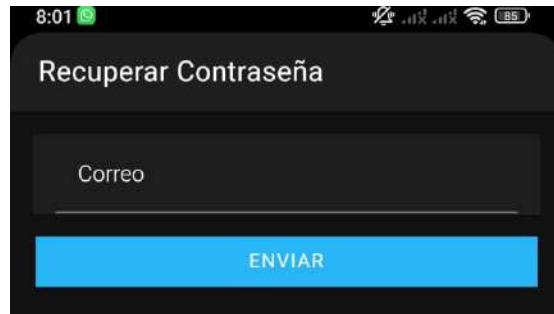


Figura C 3. Recuperar contraseña

4. Panel Principal de Datos de los Sensores

En esta pantalla, se pueden observar los datos de los sensores instalados en el campo, incluyendo temperatura, humedad ambiental, velocidad del viento y el estado del riego. También se muestran gráficos que ilustran la variación de la humedad del suelo la cual se puede comparar para verificar que el riego esté funcionando.

- **Temperatura:** Muestra la temperatura actual del ambiente.
- **Humedad Ambiental:** Muestra el porcentaje de humedad en el aire.
- **Velocidad del Viento:** Muestra la velocidad actual del viento.
- **Estado de Riego:** Indica si el sistema de riego está encendido o apagado.
- **Gráficos de Humedad del Sensor:** Muestra la variación de la humedad del suelo en el tiempo para cada sensor.

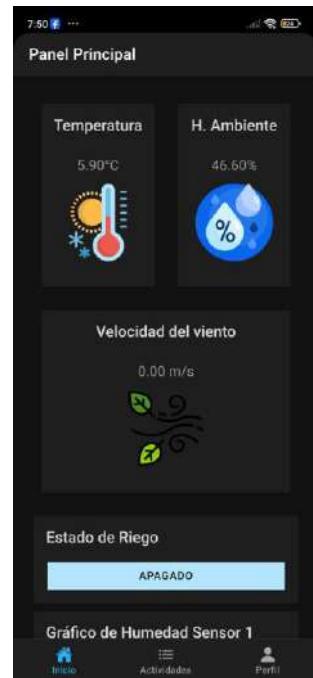


Figura C 4. Panel principal

Al pulsar en el estado del riego aparece una ventana en donde el usuario puede configurar el tiempo que desea que este activo esta medida se puede ingresar en minutos.



Figura C 5. Estado del riego

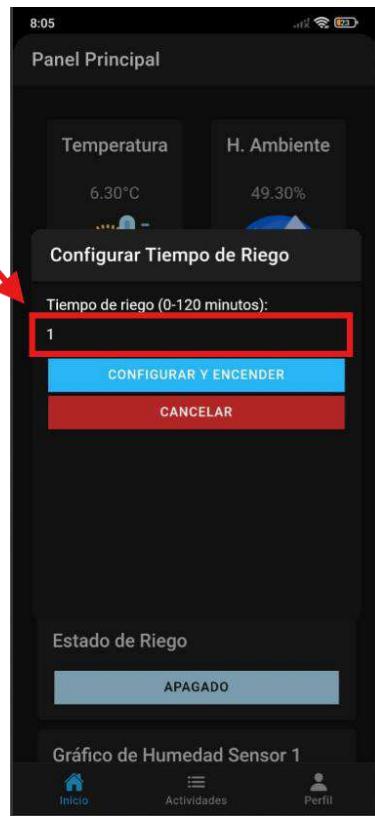


Figura C 6. Delimitación del tiempo de riego

Una vez que el riego este activo y pase el tiempo que el usuario le haya proporcionado se apagara automáticamente, pero si quiere cancelar el riego se le muestra un mensaje de alerta que aún no se ha completado el tiempo o las condiciones no son favorables.



Figura C 7. Actualización del estado del riego

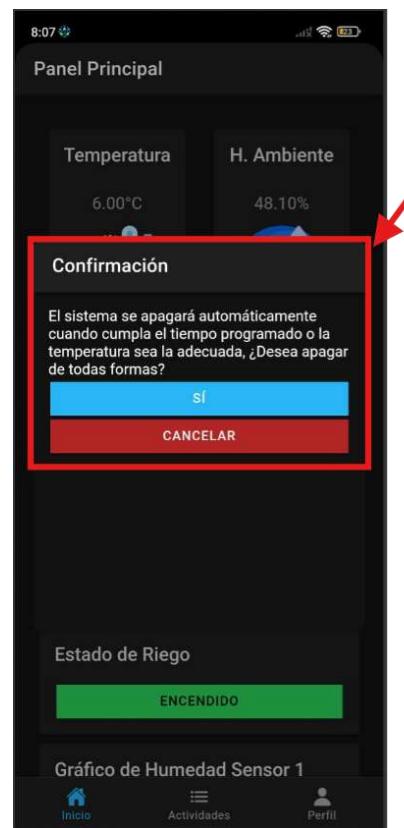


Figura C 8. Mensaje de confirmación de apagado de riego

5. Notificaciones de la aplicación

Por cada acción realizada referente al estado del riego se envía una notificación al usuario, con el estado del riego y el modo.

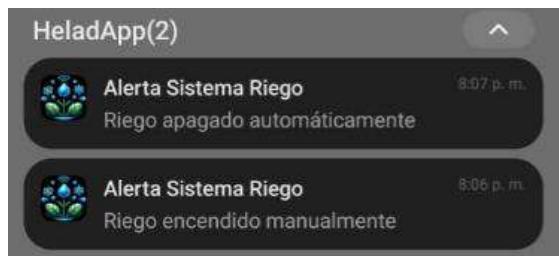


Figura C 9. Notificaciones

6. Historial del Riego

Se puede consultar el historial de actividades del riego de los últimos 10 registros, incluyendo detalles de encendido y apagado del sistema, junto con la fecha, hora, humedad, temperatura y velocidad del viento al momento de cada evento. Presione para ver detalles adicionales de cada actividad de riego.



Figura C 10. Historial de actividades

7. Datos del usuario

En esta pantalla, los usuarios pueden ver y actualizar su información personal registrada en la aplicación. También pueden cambiar su contraseña actual siguiendo los parámetros de validación de contraseñas como son que contenga un carácter especial, numérico, mayúscula, minúscula y un numérico.

Nombre: Muestra el nombre registrado del usuario.

Apellido: Muestra el apellido registrado del usuario.

Dirección: Muestra la dirección registrada del usuario.

Teléfono: Muestra el número de teléfono registrado del usuario.

Correo: Muestra la dirección de correo electrónico registrada del usuario.

Botón Cambiar Contraseña: Presione para cambiar la contraseña actual y tendrá que poner la clave actual y escribir la nueva siendo su validador que coincidan y deban cumplir con la validación.

Modo oscuro: Con esta opción el usuario puede cambiar a una paleta de colores más claros u oscuros que se guardará automáticamente en su perfil. Esto significa que, una vez que seleccione su preferencia de tema, la aplicación recordará esta configuración para futuras sesiones.

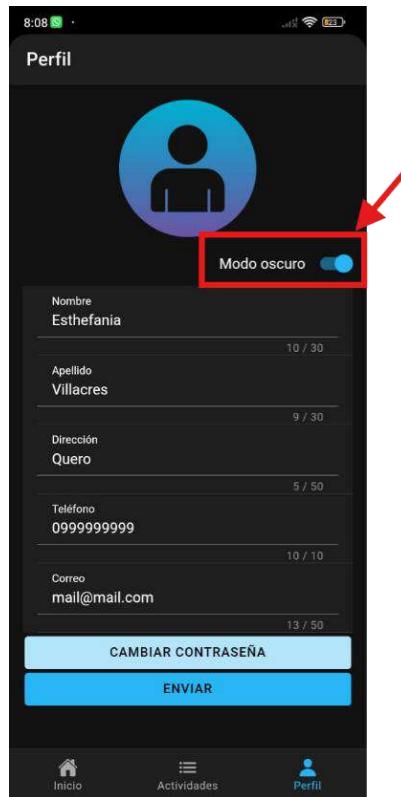


Figura C 11. Perfil Usuario

Anexo D. Pruebas de distancia módulos Lora



Figura D 1.Distancia prueba 1 - Lluvioso



Figura D 2. Distancia prueba 2 - Lluvioso



Figura D 3. Distancia prueba 3 - Soleado



Figura D 4. Distancia prueba 4 - Soleado



Figura D 5. Distancia prueba 5 - Soleado

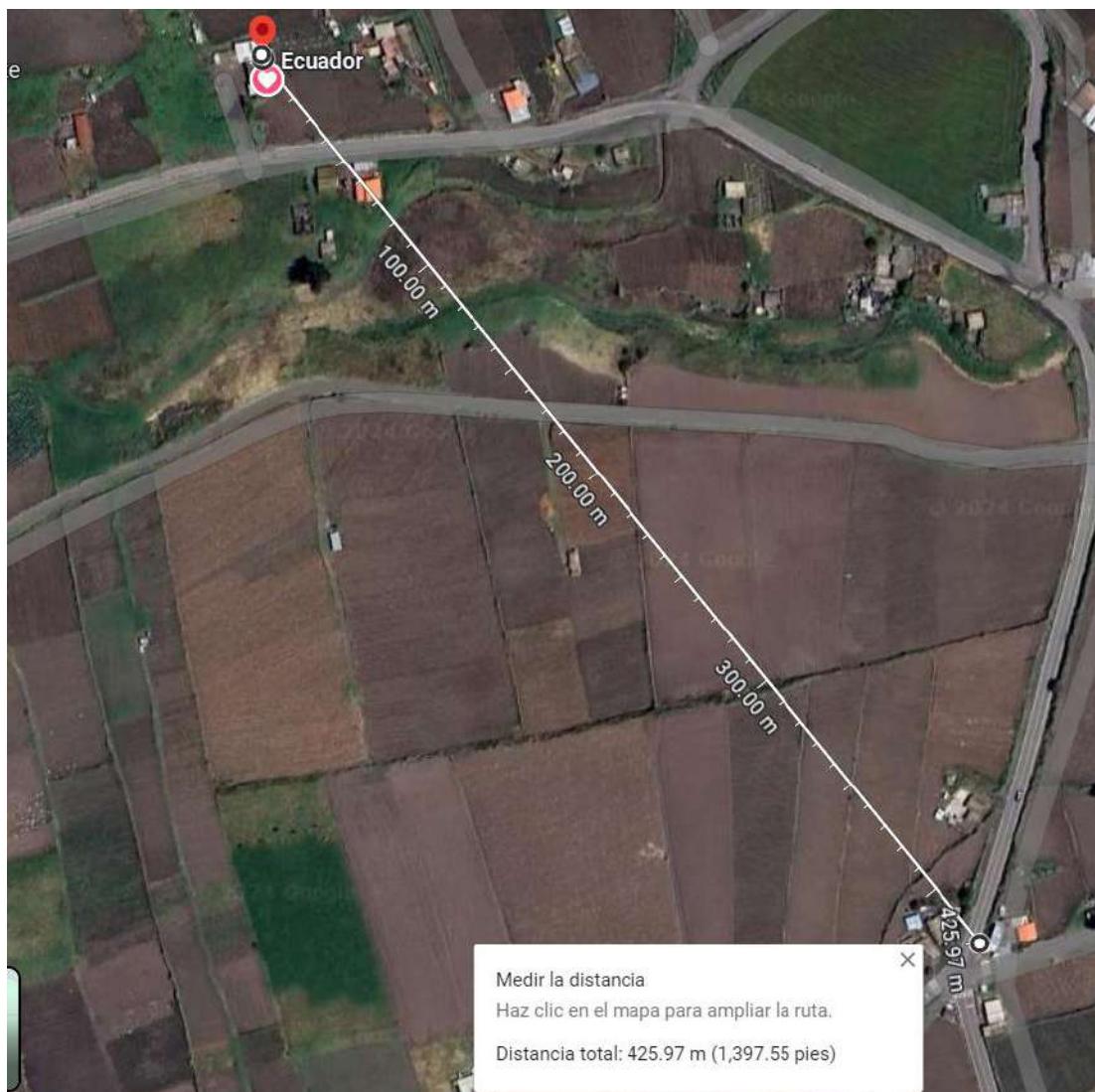


Figura D 6. Distancia prueba 6 - Soleado



Figura D 7. Distancia prueba 7 - Soleado



Figura D 8. Distancia Prueba 8 - Soleado



Figura D 9. Distancia prueba 9 - Soleado

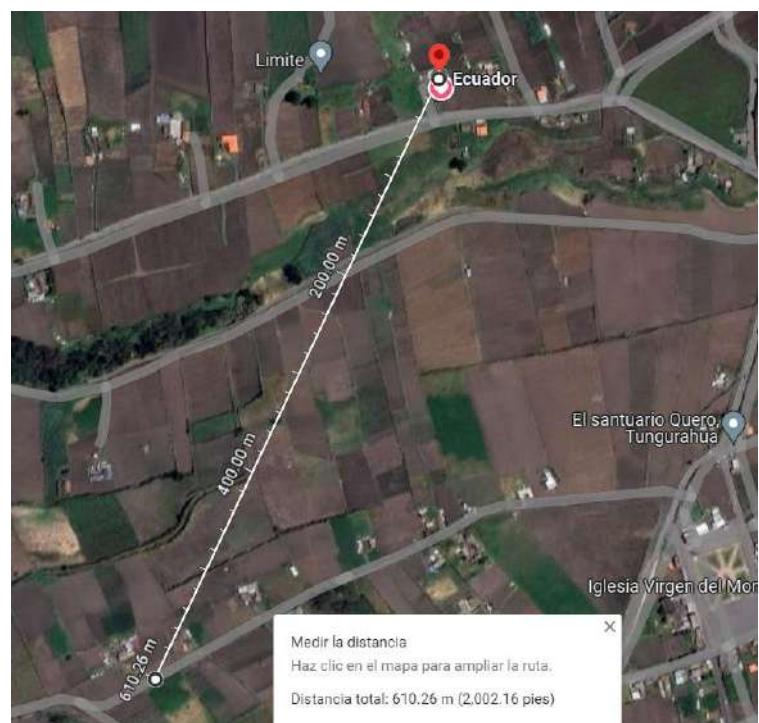


Figura D 10. Distancia prueba 10 - Soleado



Figura D 11. Distancia prueba 11 - Soleado



Figura D 12. Distancia prueba 12 - Soleado



Figura D 13.Distancia prueba - Soleado