



Desarrollo de una aplicación web para la supervisión y control remoto del riego en cultivos de
arroz de la granja UNIMINUTO

Daniel Sebastián Bello Hernández

Jorge Alberto Roncancio Enciso

Samuel David Gómez Piamba

Corporación Universitaria Minuto de Dios

Rectoría Oriente (Orinoquía)

Centro Universitario Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

Agosto de 2025

Desarrollo de una aplicación web para la supervisión y control remoto del riego en cultivos de
arroz de la granja UNIMINUTO

Daniel Sebastián Bello Hernández

Jorge Alberto Roncancio Enciso

Samuel David Gómez Piamba

Trabajo de investigación e innovación presentado como requisito para optar al título de
Tecnólogo en Desarrollo de Software

Asesor(a)

Luis Carlos Vásquez Chala

Ingeniero de Sistemas – Ingeniero Industrial

Corporación Universitaria Minuto de Dios

Rectoría Oriente (Orinoquía)

Centro Universitario Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

Agosto de 2025

Dedicatoria

- Para mi familia, fuente inagotable de fuerza e inspiración.

Este trabajo es el reflejo de su apoyo constante.

Agradecimientos

Los autores de este proyecto de grado desean expresar su más sincero agradecimiento a todas las personas e instituciones que contribuyeron a su realización.

En primer lugar, a la Corporación Universitaria Minuto de Dios (UNIMINUTO), por brindarnos el espacio y los recursos para nuestra formación profesional. A nuestro asesor, el Ingeniero Luis Carlos Vásquez Chala, por su invaluable orientación, paciencia y compromiso con nuestro aprendizaje; su guía fue esencial para dar dirección y calidad a este trabajo. Extendemos este agradecimiento al cuerpo docente del programa de Tecnología en Desarrollo de Software, cuyas enseñanzas forjaron nuestras competencias.

Un agradecimiento especial y profundo a nuestras familias. Su apoyo incondicional, comprensión durante las largas jornadas de estudio y sacrificio constante fueron el motor que nos impulsó a alcanzar esta meta. Este logro es también un reconocimiento a su esfuerzo y confianza en nosotros.

Finalmente, agradecemos a los amigos y compañeros que nos brindaron su apoyo y ánimo a lo largo de este camino.

A todos quienes, de una u otra forma, hicieron parte de este proceso, Gracias.

Contenido

Lista de tablas	7
Lista de figuras	8
Lista de anexos.....	9
Resumen	10
Abstract.....	11
1 Introducción.....	12
2 CAPÍTULO I	13
2.1 Objetivo General.....	13
2.2 Objetivos Específicos	13
2.3 Planteamiento Del Problema.....	14
2.3.1 Formulación Del Problema	16
2.4 Justificación.....	16
3 CAPITULO II	18
3.1 Marco Teórico	18
3.2 Marco Conceptual.....	20
4 CAPITULO III	24
4.1 Metodología de Desarrollo de Software.....	24
4.2 Metodología de Investigación	24
4.3 Muestra.....	25
Número de Unidades	25
Selección de Unidades:	25
Criterios de Inclusión:	26
Criterios de Exclusión:.....	27
Tamaño de la Muestra:	27
Tipo de Muestreo:.....	28
4.4 Arquitectura de la Solución: Enfoque Offline-First.....	28
4.5 Flujo de Datos del Sistema.....	29
4.6 Tecnologías Descartadas.....	30
5 CAPITULO IV	32

5.1	Sobre el Desarrollo.....	32
5.1.1	La programación e instalación Arduino	32
6	CAPÍTULO V	44
6.1	Conclusiones	44
6.2	Trabajo Futuro y Recomendaciones	45
	CAPÍTULO I	¡Error! Marcador no definido.
7	Título nivel 1.....	¡Error! Marcador no definido.
7.1	Subtítulo nivel 2	¡Error! Marcador no definido.
7.1.1	Subtítulo nivel 3	¡Error! Marcador no definido.
	CAPÍTULO II	¡Error! Marcador no definido.
8	Título nivel 1.....	¡Error! Marcador no definido.
8.1	Subtítulo nivel 2	¡Error! Marcador no definido.
8.1.1	Subtítulo nivel 3	¡Error! Marcador no definido.
	Referencias.....	¡Error! Marcador no definido.
	Anexos.....	¡Error! Marcador no definido.

Lista de tablas

Lista de figuras

Lista de anexos

Resumen

El presente trabajo de investigación e innovación tiene como objetivo diseñar e implementar una **aplicación web progresiva (PWA)** que gestione el sistema automatizado de riego para cultivos de arroz de la granja UNIMINUTO, con el propósito de optimizar el uso del recurso hídrico, reducir la intervención manual y mejorar la eficiencia del proceso agrícola. La propuesta busca responder a una problemática concreta identificada en la práctica: la conectividad intermitente en el campo, que limita el uso de aplicaciones tradicionales.

Para ello, se plantea una solución tecnológica con una arquitectura *Offline-First*, basada en sensores de humedad, controladores Arduino y una aplicación desarrollada con **Next.js (React)**. La aplicación utiliza **Service Workers** para funcionar sin conexión y almacena los datos localmente en **IndexedDB**, una base de datos en el navegador. El despliegue se realiza en la nube mediante **Vercel**. Este sistema permite realizar mediciones en tiempo real y activar el riego de manera automatizada, sincronizando los datos con un servidor central cuando se recupera la conexión, contribuyendo así a la sostenibilidad y tecnificación del campo.

El enfoque metodológico es cuantitativo, con un método experimental que permitirá comparar los resultados entre parcelas controladas y aquellas intervenidas con la tecnología, midiendo variables como el ahorro de agua, la frecuencia de riego y la mejora en la productividad del cultivo.

Palabras clave: riego automatizado, agricultura de precisión, Progressive Web App (PWA), Offline-First, Next.js, Arduino.

Abstract

This research project aims to design and implement a **Progressive Web Application (PWA)** to manage the automated irrigation system for rice crops at the UNIMINUTO farm. Its purpose is to optimize water use, reduce manual intervention, and improve the efficiency of the agricultural process. The proposal responds to a specific problem identified in practice: intermittent connectivity in the field, which limits the use of traditional applications.

To address this, a technological solution with an *Offline-First* architecture is proposed, based on humidity sensors, Arduino controllers, and an application developed with **Next.js (React)**. The application uses **Service Workers** to operate offline and stores data locally in **IndexedDB**, a browser-based database. The system is deployed in the cloud through **Vercel**. This setup allows for real-time measurements and automated irrigation activation, synchronizing data with a central server once connectivity is restored, thus contributing to the sustainability and modernization of the agricultural field.

The methodological approach is quantitative, with an experimental method that will compare the results between control plots and those using the technological solution, measuring variables such as water savings, irrigation frequency, and improvements in crop productivity.

Keywords: automated irrigation, precision agriculture, Progressive Web App (PWA), Offline-First, Next.js, Arduino.

1 Introducción

Desde los inicios de la civilización sedentaria, la agricultura ha sido un pilar fundamental en el desarrollo de las sociedades humanas. Débase a que es el sustento de energía y nutrientes para los seres humanos, es de vital importancia el desarrollo de buenos sistemas agrícolas tanto ganaderos como *en base a las plantas*. Esta temática hoy día toma creciente interés debido a los problemas que enfrenta la sociedad actual: sobrepoblación, mayor brecha entre clases socioeconómicas, carencia de recursos ecológicos y la concentración de poder y oportunidades alrededor del mundo.

Desde las primeras civilizaciones y concentraciones sociales se han buscado métodos y tecnologías para mejorar la eficiencia y calidad de los sistemas agrícolas respecto al riego. Estos han ido evolucionando conforme las herramientas y conocimientos disponibles para cada una de las épocas. Para civilizaciones indígenas como la Maya, los sistemas de acueducto permitían un sistema de riego y optimización hídrica basados en canales de agua, en los que basaban los diseños de sus ciudades. Nótese la predilección de aquella civilización de priorizar los sistemas hídricos sobre los diseños estéticos y funcionales de sus estructuras.

Dejando en claro la importancia de los sistemas de riego a lo largo de la historia, hoy se presenta una propuesta de implementar tecnologías avanzadas en un sistema de riego para los cultivos de la granja UNIMINUTO. Para este caso puede hacerse uso del término “micro-irrigación” que, según (Agbetuyi, 2016), define como el suplemento artificial de agua a las raíces de una planta. Históricamente se ha apoyado en la irrigación para asistir en el crecimiento de cultivos agro-culturales, mantenimiento de paisajes y la revegetación de tierras secas. (Agbetuyi, 2016)

2 CAPÍTULO I

2.1 Objetivo General.

Diseñar y desarrollar una **aplicación web progresiva (PWA)** para la supervisión y control remoto del sistema de riego en los cultivos de arroz en UNIMINUTO, empleando tecnologías como **Next.js (React)** y una arquitectura *Offline-First* para garantizar su funcionamiento en condiciones de conectividad intermitente.

2.2 Objetivos Específicos

Verificar los sensores y demás dispositivos electrónicos del sistema de riego, comprobando su correcto funcionamiento en condiciones reales de campo, con el fin de garantizar que los datos recolectados reflejen de manera precisa el estado del terreno y permitan una respuesta automatizada coherente con las necesidades del cultivo.

Analizar los requerimientos funcionales y no funcionales del sistema, partiendo del análisis del contexto agrícola y de las necesidades identificadas por la investigación en curso, con el propósito de establecer una base clara que oriente la fase de diseño y asegure la coherencia del desarrollo con el entorno de implementación.

Modelar la arquitectura general del sistema, integrando tanto la lógica del riego automatizado como la estructura de funcionamiento de la aplicación web, procurando una interacción eficiente entre hardware y software, y respondiendo a los objetivos técnicos establecidos por la metodología seleccionada.

Desplegar **la aplicación web** desarrollada, verificando su integración con el sistema físico de riego, el acceso remoto mediante conexión a internet y su validación preliminar en campo, de

forma que pueda ser utilizada como una herramienta efectiva de monitoreo y control por los usuarios finales.

2.3 Planteamiento Del Problema

La granja UNIMINUTO es un espacio agroecológico en el que estudiantes universitarios desarrollan clases teórico-prácticas, experimentos y actividades formativas relacionadas con la agricultura, la biología y otras ramas de las ciencias naturales. Además, en este entorno se desarrollan investigaciones científicas tanto por estudiantes —en el marco de sus trabajos de grado— como por docentes y personal académico, con fines de publicación y avance del conocimiento científico.

Actualmente, en la granja se desarrolla un proyecto de investigación enmarcado en una convocatoria académica que exige la implementación de sistemas de riego avanzados mediante el aprovechamiento de aguas pluviales. El objetivo de esta implementación es resolver la ausencia de un sistema automatizado de riego en la granja y, al mismo tiempo, analizar las relaciones entre el riego, la calidad del agua y la productividad de los cultivos. Esta convocatoria contempla evaluar dos variantes del sistema de riego: uno conectado a la red eléctrica y otro con bombeo independiente. (Gómez, 2024)

La participación en dicha convocatoria permite sentar las bases para un proyecto de investigación independiente, el cual busca proponer soluciones aplicables a escenarios de irrigación básicos que requieran automatización inteligente. La oportunidad radica en diseñar un sistema de riego automatizado que registre de forma continua variables críticas como la humedad del suelo, la temperatura ambiente y el consumo de agua.

La ausencia de estos datos automatizados genera diversos problemas en entornos agrícolas experimentales:

- Variabilidad en la frecuencia y cantidad de riego, con riesgos de estrés hídrico o encharcamiento de las plantas.
- Desperdicio de recurso hídrico y aumento de la carga de trabajo manual para técnicos y estudiantes.
- Inconsistencias en los resultados de investigación y dificultades para replicar ensayos.

Como lo plantea Agbetuyi(2016),

Existe una necesidad imperante en mejorar los métodos antiguos y actuales de irrigación. Un sistema de irrigación automatizado requiere ser desarrollado para optimizar el uso hídrico en cultivos agrícolas. Un sistema automático de irrigación debe tener todos los componentes que monitoreen y controlen autónomamente el nivel de agua disponible para las plantas sin fallo alguno o intervención humana. (...) Las ventajas de la irrigación automática para las plantas incluye un ahorro de dinero, agua, una disminución de carga laboral y una conveniencia general.

Actualmente, el cultivo de arroz se encuentra en un sistema de riego manual con 2 líneas principales de riego. Cada una de estas es administrada y supervisada personalmente por una persona encargada de realizar el riego según los horarios preestablecidos y observaciones del comportamiento de los cultivos. Si bien, es un sistema de riego existente para los cultivos este continúa siendo manual, permitiendo que exista un amplio margen de optimización. Elementos como recursos hídricos, económicos y de tiempo podrían verse mejorados gracias a lo que permite un control de manera remota.

En consecuencia, la implementación de un sistema de riego que permita la medición y control remoto y automático del recurso hídrico se vuelve esencial para garantizar eficiencia operativa, precisión investigativa y sostenibilidad en los procesos de formación y producción agrícola de la granja UNIMINUTO.

2.3.1 *Formulación Del Problema*

¿Cómo desarrollar una **aplicación web progresiva (PWA)** con una arquitectura *Offline-First* basada en **Next.js (React)** y Arduino para la supervisión y control remoto del riego en los cultivos de la Granja UNIMINUTO, que sea resiliente a fallos de conectividad?

2.4 Justificación

Se desarrollará una **aplicación web progresiva (PWA)** para la supervisión y control remoto del riego en cultivos de arroz de la Granja UNIMINUTO. Este sistema contará con un despliegue en la nube que permitirá el acceso desde cualquier dispositivo con un navegador web, de modo que pueda ser administrado según las necesidades del supervisor de los cultivos. Una ventaja clave de la arquitectura PWA es su capacidad para **funcionar sin conexión a internet**, permitiendo la captura de datos en campo y su posterior sincronización automática.

Además, la aplicación ofrecerá estadísticas relevantes sobre el estado del cultivo y el funcionamiento del sistema de riego, como la humedad individual de cada sensor, la humedad promedio por la línea de riego, la humedad total del sistema, la temperatura ambiente, entre otros datos clave.

Según (Gutierrez, Villa-Medina, Nieto, & Porta, 2014), los sistemas de riego automatizados demuestran que el uso optimizado del agua puede reducirse significativamente para una determinada cantidad de producción de biomasa fresca. Además del ahorro monetario, la importancia de preservar este recurso natural justifica ampliamente la implementación de este tipo de tecnologías. Esta afirmación respalda la necesidad urgente de establecer sistemas de riego automatizados orientados a la optimización de recursos en escenarios agrícolas.

Este proyecto surge ante la oportunidad de facilitar los procesos de irrigación en los cultivos de arroz de la Granja UNIMINUTO, permitiendo sentar un precedente que acerque a los demás cultivos institucionales a sistemas controlados de manera remota por sus supervisores. Se busca que la aplicación no solo permita al responsable del cultivo automatizar el riego, sino también conocer con mayor profundidad las estadísticas y valores de las magnitudes ambientales que influyen en su producción, superando las limitaciones de los sistemas manuales.

De esta manera, la propuesta no solo resuelve un problema operativo específico, sino que también aporta al cuerpo de conocimiento de la universidad y establece una base escalable para futuras implementaciones en otros cultivos y contextos agrícolas.

3 CAPITULO II

3.1 Marco Teórico

Hoy en día, muchas actividades que antes se hacían completamente de manera manual, han empezado a modernizarse gracias al uso de tecnologías automatizadas. En el caso de la agricultura, esto no ha sido la excepción. En cultivos como el arroz, por ejemplo, uno de los procesos más importantes es el riego, y cuando no se maneja bien puede causar tanto desperdicio de agua como pérdida de producción. Por eso mismo, en los últimos años se ha buscado integrar soluciones tecnológicas que ayuden a hacer más eficiente ese proceso sin quitarle el control al agricultor.

Los sistemas automatizados de riego han sido diseñados para controlar cuándo y cuánto regar, basándose en datos reales como la humedad del suelo o la temperatura. Esto se logra conectando sensores de humedad a un microcontrolador como Arduino, que a su vez se comunica con una aplicación o plataforma que pueda gestionar toda la información. Como señalan [OBJ-08], usar este tipo de sistemas no solo ayuda a ahorrar agua, sino que también permite conservarla mejor, lo cual es clave en este momento que aquel recurso se está volviendo cada vez más limitado.

En diferentes investigaciones se han probado modelos parecidos que ayudan a confirmar la utilidad de este tipo de tecnología. (Hernández, 2021) por ejemplo, diseñó un sistema que funciona con energía solar y que además de regar según la humedad detectada, también notifica al usuario por celular cuando hay cambios importantes. Algo similar hizo (Domínguez, 2022), quienes usaron sensores y Arduino Uno para automatizar el riego de distintas parcelas, mostrando que estos dispositivos se pueden adaptar a muchos tipos de suelo y ambiente sin que

eso signifique complicaciones técnicas. A su vez, (Pastrana, 2025) propuso un sistema más completo que no solo activa o desactiva el riego, sino que también muestra datos en tiempo real a través de una interfaz. Este tipo de soluciones facilita mucho el trabajo de técnicos o supervisores que quieren monitorear el cultivo sin estar pegados todo el tiempo al terreno.

Con base en eso, se plantea el desarrollo de una **aplicación web progresiva (PWA)** que controle y supervise el riego automatizado de los cultivos de arroz en la granja UNIMINUTO. La idea surge porque el sistema de riego actual no cuenta con ningún tipo de automatización ni monitoreo remoto, y además, la conectividad en la zona puede ser inestable.

Lo que se propone es una herramienta desarrollada con **Next.js (React)** que, gracias a su arquitectura *Offline-First*, puede funcionar incluso sin conexión a internet. La aplicación utiliza **Service Workers** para cachear la interfaz y **IndexedDB** para almacenar localmente los datos de los sensores. Cuando el dispositivo recupera la conexión, los datos se sincronizan automáticamente con la base de datos remota. Este sistema no solo permitirá regar según las condiciones reales del suelo, sino también visualizar estadísticas importantes como la humedad por sensor, promedios por la línea de riego y temperatura ambiente, ayudando a tomar decisiones más acertadas desde cualquier lugar.

Aunque al comienzo se aplique solo en un cultivo puntual, se espera que esta propuesta pueda adaptarse más adelante a otros escenarios dentro o fuera de la granja UNIMINUTO.

3.2 Marco Conceptual

Agronomía: Se denomina agronomía a los saberes y las técnicas que permiten el desarrollo de cultivos. Se trata de una disciplina que, basándose en datos de diferentes ciencias, contribuye a la explotación de la ganadería y la agricultura.

La finalidad de la agronomía es optimizar los procesos que posibilitan obtener y transformar productos agropecuarios. Para esto se dedica a analizar diferentes cuestiones vinculadas a la biología, la química, la física, la sociología y la economía que, de una forma u otra, inciden sobre los diferentes procesos de producción. (Julián Pérez Porto, 2022)

Sistema de riego e irrigación: Se define como el sistema a partir del cual se logra un aporte de agua en los cultivos, capaz de suplir las necesidades hídricas de las plantas, que no se ven cubiertas únicamente por las precipitaciones. Este aporte puede realizarse con el fin de incrementar la producción de una parcela, transformando zonas agrícolas de secano en zonas de regadío. (Innovatione, 2019) Por su parte, la irrigación en el campo agroecológico hace referencia al aporte de agua que se realiza a un cultivo para cubrir sus necesidades hídricas. Debido a que las precipitaciones pueden resultar inefficientes, la irrigación permite suplir esa carencia y posibilita que las plantas crezcan de manera saludable. (Julián Perez Porto, 2022)

Micro-irrigación: es un tipo de riego de baja presión y bajo caudal que puede reducir la probabilidad de riego excesivo en un jardín. Este tipo de riego suministra el agua directamente donde más se necesita: la zona radicular de las plantas. Además, distribuye el agua lentamente y durante un periodo más prolongado, evitando la escorrentía y reduciendo la evaporación. (Agency, 2024)

Automatización: La automatización es la aplicación de tecnología, programas, robótica o procesos para lograr resultados con una intervención humana mínima. Estas se usan para aumentar la productividad y la rentabilidad, mejorar el servicio de atención, reducción de costes y los errores operativos. (IBM, s.f.)

Internet de las cosas (IoT): El concepto describe objetos físicos (o grupos de estos) con sensores, capacidad de procesamiento, software y otras que se conectan e intercambian datos con otros dispositivos y sistemas a través de internet u otras redes de comunicación. (Yasar, 2024)

Software: Es el conjunto de programas, datos, instrucciones u procedimientos que permiten realizar tareas específicas en un sistema informático. A diferencia del hardware, el software es intangible y esencial para el funcionamiento de cualquier computadora o dispositivo digital. (Significados, 2024)

Hardware: Es la parte física de un ordenador o sistema informático. Está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos tales como circuitos de cables y luz, memorias, discos duros, dispositivos periféricos y cualquier otro material en estado físico que sea necesario para que el equipo funcione. (Significados, Hardware, 2023)

Arduino: Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla. (Arduino, s.f.)

Sensores: Es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos aparatos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas. (Ana Gardey, 2021)

UI: Es el punto de interacción y comunicación humano-computadora en un dispositivo. Esto puede incluir pantallas de visualización, teclados, un mouse y la apariencia de un escritorio. (Churchville, 2020) La interfaz de usuario o UI es el conjunto de elementos de la pantalla que permiten al usuario interactuar con una página web. (More, 2023)

Control remoto: Se trata de un dispositivo que se utiliza para administrar el funcionamiento de una máquina o de un sistema a distancia. Es un elemento que facilita el poder de control y evita acciones manuales directas en el sistema que controla. (Merino, 2018)

Aplicación web: Una aplicación web se refiere a un software de aplicación que opera en servidores web y se accede a través de navegadores web utilizando tecnologías web como HTML, CSS y JavaScript. (Sarvaiya, 2023)

Progressive Web App (PWA): Es un tipo de aplicación de software que se entrega a través de la web, creada utilizando tecnologías web comunes como HTML, CSS y JavaScript. Está destinada a funcionar en cualquier plataforma que utilice un navegador compatible con los estándares. Las PWA ofrecen funcionalidades similares a las de una aplicación nativa, como la capacidad de trabajar sin conexión, recibir notificaciones push y ser instaladas en la pantalla de inicio del dispositivo, mejorando la experiencia del usuario y la resiliencia de la aplicación. (MDN, 2025)

Next.js: Es un framework de desarrollo de código abierto basado en React que permite crear aplicaciones web rápidas y optimizadas. Proporciona funcionalidades listas para usar, como el renderizado del lado del servidor (SSR) y la generación de sitios estáticos (SSG), lo que mejora el rendimiento y el SEO. Además, incluye un sistema de enrutamiento basado en el sistema de archivos y rutas de API integradas, simplificando la creación de aplicaciones web completas y escalables. (Barragán, 2025)

IndexedDB: Es una API de bajo nivel del lado del cliente para almacenar cantidades significativas de datos estructurados en el navegador. Funciona como una base de datos NoSQL basada en objetos JavaScript y es la tecnología estándar para implementar el almacenamiento de datos en aplicaciones web progresivas (PWA) con capacidad offline. (IndexedDB, 2025) A diferencia de LocalStorage, IndexedDB permite realizar consultas complejas y manejar grandes volúmenes de datos de manera asíncrona, sin bloquear la interfaz de usuario.

Service Worker: Es un script que el navegador ejecuta en segundo plano, separado de la página web, y que actúa como un proxy de red. Es la tecnología central que habilita las funcionalidades de una PWA, como el funcionamiento sin conexión, al interceptar y gestionar las solicitudes de red, permitiendo cachear recursos y datos. También gestiona las notificaciones push y las actualizaciones en segundo plano. (Harb, 2018)

4 CAPITULO III

4.1 Metodología de Desarrollo de Software

Una metodología para el desarrollo de software es un modo sistemático de realizar, gestionar, y administrar un proyecto para llevarlo a cabo con altas probabilidades de éxito. Esta comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado.

Para este proyecto de desarrollo se ha optado por utilizar la flexible, pero muy fiable metodología **Kanban**. Esta metodología la podemos definir como un marco de trabajo enfocado en las tareas y los estados de realización de estas. Entre estos estados dinámicos propios de cada tarea (o funcionalidad, dependiendo del tipo de tablero) encontramos las pendientes, en ejecución o proceso, finalizadas y en revisión. Cada estado es atribuible a un elemento de un tablero, el cual se divide en columnas que representan el conjunto de tareas agrupadas según sus etapas. Una de sus restricciones recae en la columna del WIP "Work In Progress" o las tareas en ejecución, la cual indica que no pueden haber más de 3 tareas en etapa de realización. El propósito de esta restricción es mejorar el enfoque, evitar cuellos de botella y acelerar la finalización de cada tarea, garantizando un flujo de trabajo más eficiente y predecible. (Maida & Pacienza, 2015).

4.2 Metodología de Investigación

Este proyecto se desarrollará bajo un enfoque de investigación cuantitativo, ya que se trabajará principalmente con datos numéricos recolectados a través de sensores de humedad del suelo, temperatura y consumo de agua. El análisis de estos datos permitirá observar de manera

objetiva el comportamiento del sistema de riego, identificar mejoras, y tomar decisiones basadas en resultados concretos.

A su vez, el tipo de investigación será experimental, porque se implementará un sistema automatizado real dentro del entorno agrícola de la granja UNIMINUTO, con el fin de evaluar su efectividad en condiciones de uso reales. Este tipo de enfoque nos permite comprobar cómo la automatización puede impactar directamente en variables como el uso del agua, la frecuencia de riego y la carga de trabajo manual.

4.3 Muestra

La muestra de este estudio se conformará a partir de las unidades de observación disponibles dentro del contexto de la Granja UNIMINUTO, que es el lugar donde se implementará el sistema de riego automatizado. Dado que el proyecto está enfocado en un sistema tecnológico de riego, la muestra estará compuesta por:

4.3.1 Número de Unidades

La muestra estará formada por un total de 18 materas de cultivo de arroz en la Granja UNIMINUTO. Las materas estarán interconectadas mediante 3 líneas independientes de riego, las cuales suplirán de agua a 6 materas cada una. Se entiende según la documentación del proyecto multidisciplinar que cada línea contará con propiedades de humedad y suelo diferentes con propósito de estudio de comportamiento y propiedades.

4.3.2 Selección de Unidades

La selección de las materas se realizará de manera intencional (o por juicio), aprovechando la estructura existente de las tres líneas de riego para garantizar que la muestra sea la más

adecuada para la validación experimental del sistema. Esta selección se basará en la disponibilidad de las parcelas para la instalación de los sensores y la adecuación a las condiciones de control requeridas por un estudio de enfoque cuantitativo. Se elegirá el conjunto de 18 macetas que mejor permita la comparación de propiedades (humedad y suelo) y la eficiencia operativa del software de automatización.

4.3.3 Criterios de Inclusión

Las parcelas seleccionadas deben cumplir rigurosamente con los siguientes criterios, esenciales para el desarrollo de un experimento controlado:

- Tener cultivos de arroz en una fase de crecimiento activa, evitando interferencia con las etapas iniciales (siembra) o finales (recolección).
- Estar equipadas con un sistema de riego convencional (manual o automático básico) para establecer un grupo de control o referencia, permitiendo la comparación de datos cuantitativos.
- Estar ubicadas dentro del área de la granja que garantice el acceso y cobertura de la red inalámbrica, indispensable para el monitoreo remoto y el control del sistema por la aplicación web.
- Formar parte de las 3 líneas de riego independientes necesarias para el estudio de las propiedades diferenciadas de humedad y suelo.

4.3.4 Criterios de Exclusión

Se excluirán las parcelas que presenten condiciones que comprometan la validez o la viabilidad de la implementación:

- Parcelas que no cuenten con un sistema de riego funcional previo.
- Parcelas que se encuentren en fase de siembra o de recolección, ya que estas etapas introducen variables que podrían sesgar los resultados de eficiencia hídrica.
- Parcelas que no tengan una conexión de red inalámbrica estable para el control y la transmisión de datos a la base de datos remota.
- Cualquier macetera que, por su ubicación o estado de cultivo, no pueda ser asignada claramente a una de las 3 líneas de riego requeridas para la estructura de la muestra.

4.3.5 Tamaño de la Muestra

El tamaño de la muestra es de 18 materas (distribuidas en 3 líneas de 6 materas cada una). Este tamaño se considera adecuado para un análisis preliminar de la efectividad del sistema de riego automatizado en un entorno de prueba controlado, a pesar de las limitaciones de tiempo y recursos del proyecto. La interconexión en 3 líneas proporciona una base suficiente para obtener datos representativos y diferenciados del comportamiento del sistema frente a variables como humedad y consumo, sin comprometer la viabilidad del proyecto a gran escala.

4.3.6 *Tipo de Muestreo*

Se utilizará un muestreo no probabilístico por conveniencia. La elección se fundamenta en que la selección de las 18 materas no se realiza al azar, sino que se basa en:

- **Conveniencia:** Su disponibilidad y accesibilidad dentro de la Granja UNIMINUTO para la instalación de hardware y software.
- **Juicio/Intención:** El investigador selecciona intencionalmente el conjunto de 18 macetas que cumplen con los criterios técnicos y de infraestructura (las 3 líneas de riego, conexión de red) que son críticos para la validación experimental del sistema de control remoto.

Este método es coherente con el enfoque experimental y cuantitativo, ya que prioriza la viabilidad técnica y la obtención de datos medibles en un entorno controlado.

4.4 **Arquitectura de la Solución: Enfoque Offline-First**

La principal restricción del proyecto es la potencial conectividad intermitente en zonas rurales. Para abordar este desafío, se diseñó una arquitectura

Offline-First basada en una Progressive Web App (PWA), que combina la accesibilidad de un sitio web con las funcionalidades de una aplicación nativa, especialmente la capacidad de operar sin conexión a internet.

La selección del stack tecnológico se basó en requisitos clave como la operación en conectividad intermitente, desarrollo rápido, despliegue sencillo en Vercel y la generación de reportes del lado del cliente.

El stack tecnológico seleccionado es el siguiente:

Framework Frontend: Se utilizó Next.js (React) por su capacidad para el desarrollo web rápido, su optimización para despliegues en Vercel y sus ventajas en SEO.

Capacidades Offline: Se implementaron Service Workers, el núcleo de una PWA, para permitir que la aplicación funcione sin conexión a internet, cacheando la interfaz y los recursos necesarios.

Base de Datos Local: Se eligió IndexedDB para el almacenamiento robusto de datos en el navegador, permitiendo guardar registros de riego, mediciones de sensores y otra información de forma segura mientras no hay conexión.

Backend: Se aprovecharon las API Routes de Next.js para crear un backend integrado en el mismo proyecto, simplificando la arquitectura y la comunicación con la base de datos remota.

Generación de Reportes: Para la creación de informes en formato PDF directamente en el navegador, se seleccionó la librería jsPDF/React-PDF, garantizando que esta funcionalidad también opere sin conexión.

4.5 Flujo de Datos del Sistema

El sistema opera siguiendo un ciclo de vida diseñado para la resiliencia:

Carga Instantánea: El Service Worker guarda una copia local de la aplicación. De esta forma, la aplicación siempre carga y es funcional, incluso sin acceso a internet.

Captura de Datos Offline: El usuario puede registrar datos de riego, consultar lecturas de sensores y realizar otras operaciones sin conexión. Toda esta información se guarda de forma segura en la base de datos local (IndexedDB) del navegador.

Sincronización Automática: Una vez que el dispositivo recupera la conexión a internet, una lógica de sincronización se activa automáticamente. Esta lógica se encarga de enviar todos los datos pendientes desde IndexedDB hacia la base de datos remota a través de las API Routes de Next.js.

4.6 Tecnologías Descartadas

Para asegurar una arquitectura ágil y eficiente para el MVP, se evaluaron y descartaron otras tecnologías populares:

Flutter Web: Aunque es una tecnología potente, su versión web tiende a generar un bundle (paquete de archivos) más pesado en comparación con una aplicación web nativa en React. Next.js ofrece una experiencia más ligera y un ecosistema de librerías web más maduro y optimizado.

Backend Separado (Python/Node.js): Mantener un backend independiente con frameworks como Django o Express introduce una mayor complejidad arquitectónica, requiriendo la gestión de un segundo proyecto, un despliegue adicional y configuraciones de CORS. Las API Routes de Next.js integran esta funcionalidad en un solo proyecto, lo que es ideal para un MVP.

LocalStorage: A diferencia de IndexedDB, LocalStorage es muy limitado. Solo permite guardar pares de clave-valor como texto y tiene un tope de almacenamiento de aproximadamente

5MB. IndexedDB es una base de datos NoSQL completa, diseñada para almacenar grandes volúmenes de datos estructurados de manera eficiente en el navegador.

5 CAPITULO IV

5.1 Sobre el Desarrollo

El proyecto ha sido propuesto y abarcado mediante un concepto que los integrantes del grupo concibieron como “los tres pilares”. Este concepto significa que el desarrollo de esta aplicación se verá realizado gracias a tres actores abarcando los tres escenarios de trabajo que se podrán observar en el proyecto: la programación e instalación del Arduino, la programación del back-end y la configuración de una base de datos en la nube, y el desarrollo del front-end junto a su debido despliegue mediante el PWA. Es de esta manera que se permitirá abarcar la descripción del desarrollo.

Haciendo uso de la metodología Kanban, se ha documentado la realización de tareas e implementación funciones a la aplicación. Esto permite realizar un monitoreo de actividades realizadas y una auditoria de todos los elementos realizados y pendientes por finalizar. Esta metodología permite que el grupo de desarrolladores encuentre un flujo de trabajo constante y continuo para suplir las necesidades instantáneas de cada etapa del desarrollo.

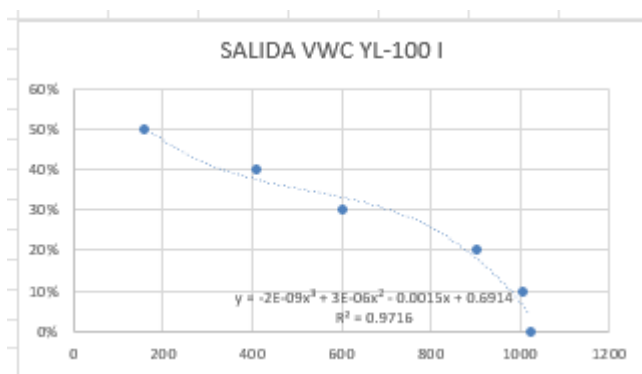
5.1.1 La programación e instalación Arduino

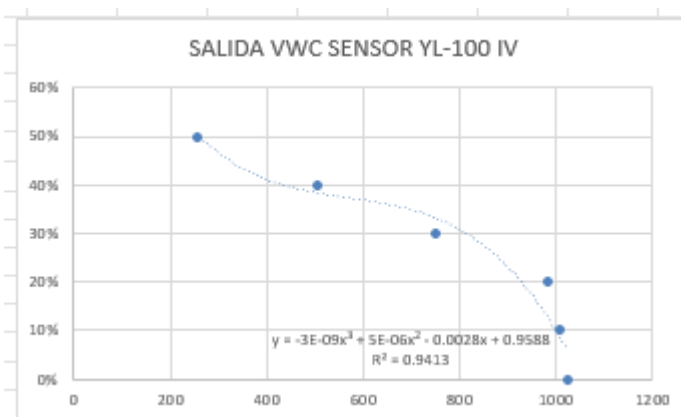
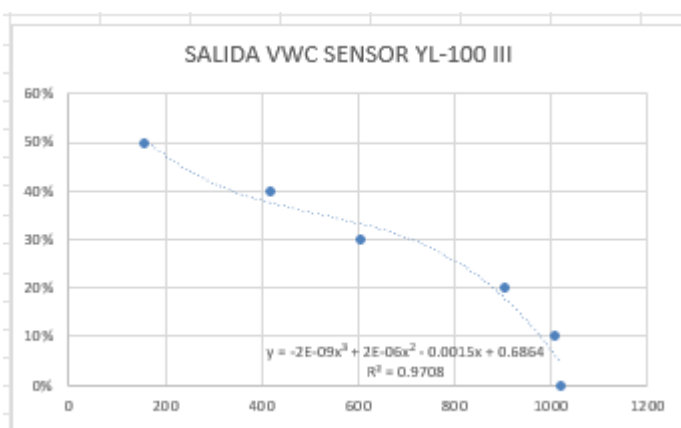
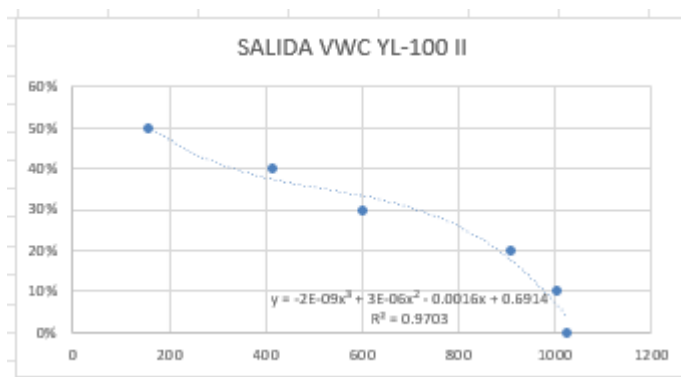
Para la instalación Arduino se hizo uso de los siguientes componentes y tecnologías:

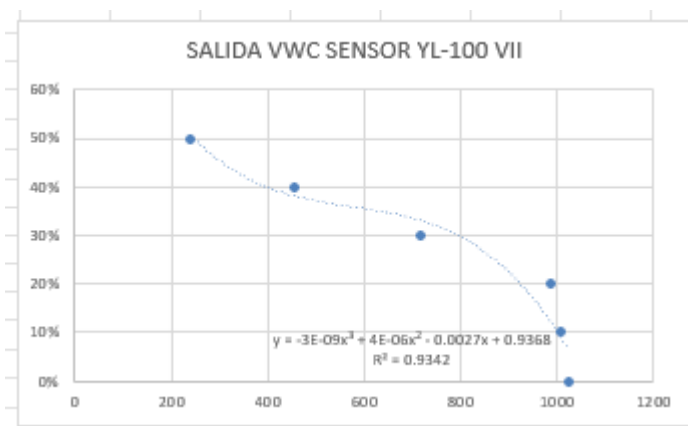
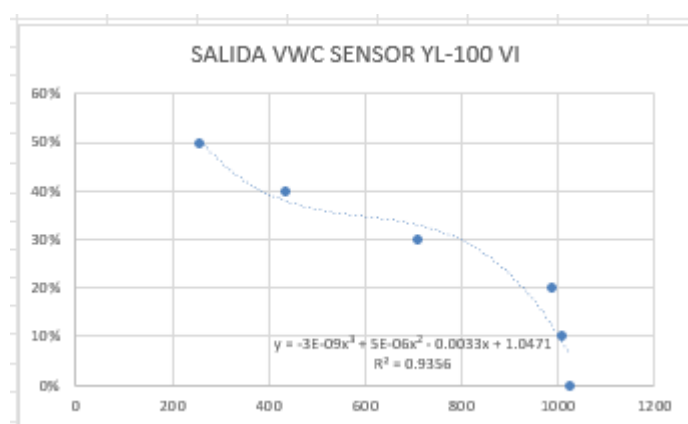
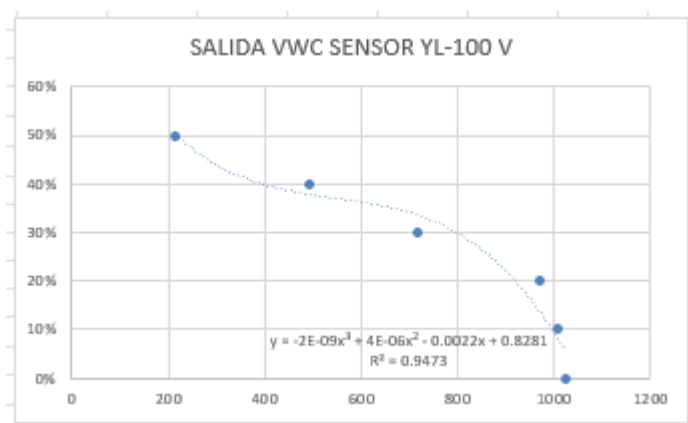
- Placa Arduino UNO
- Sensores de humedad YL-100
- Placa de WiFi []
- 2 Multiplexores CD74HC4067

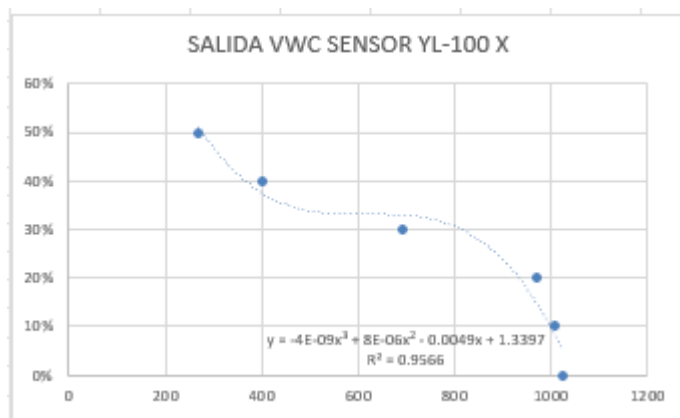
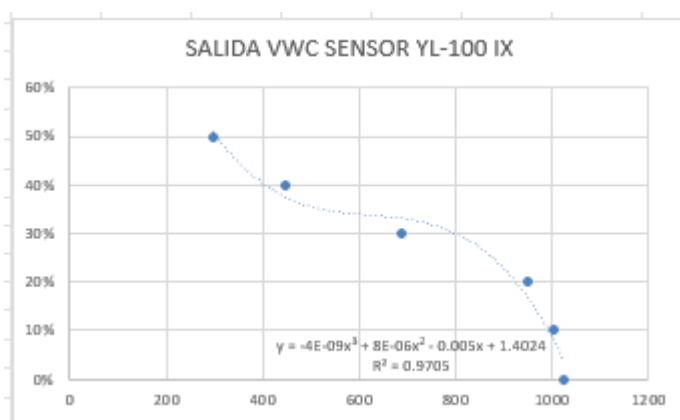
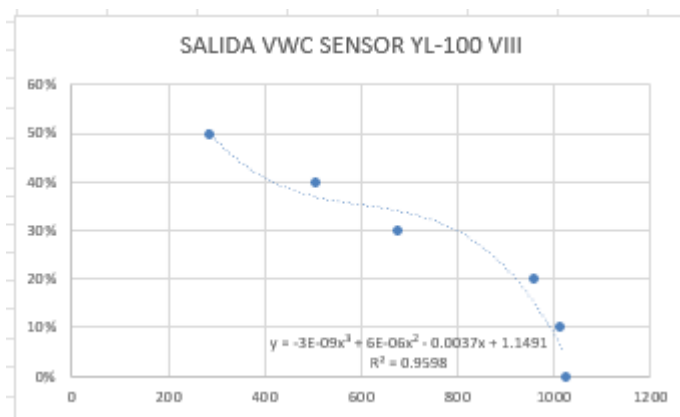
- Arduino IDE
- Lenguaje de programación C++

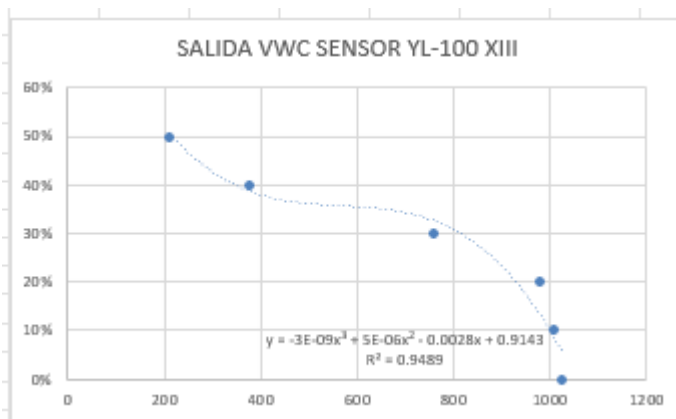
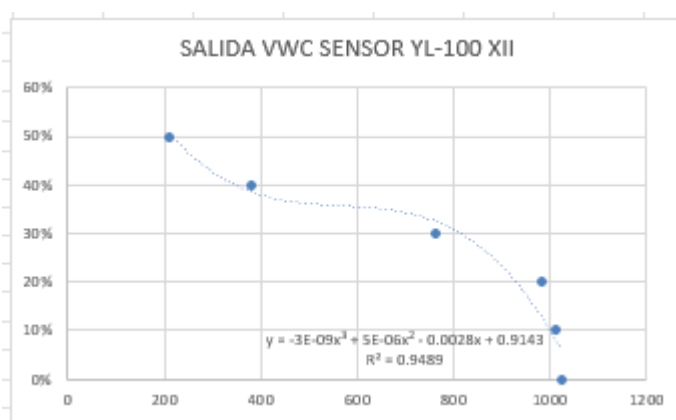
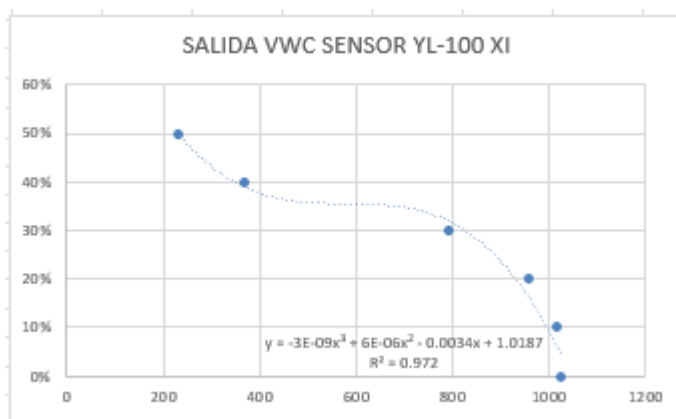
Antes de iniciar el desarrollo dentro de Arduino IDE y cargar los sketches con los que va a trabajar el programa, se realizó una curva de calibración comparativa entre los valores de los sensores de humedad y sus lecturas analógicas con los valores arrojados por mediciones mediante lectores de humedad digitales y experimentos controlados. Se realizaron varias lecturas con cada sensor, para realizar una estimación promedio de los valores obtenidos según los escenarios de humedad en los suelos del experimento permitiendo así establecer equivalencias aproximadas de los valores de lectura analógica con los esperados en Humedad Volumétrica. Estas lecturas se pueden encontrar en las siguientes tablas, cada una correspondiente a un sensor. Se calibraron 13 de 18 sensores según indicaciones de la Doctora que dirige el proyecto multidisciplinar.











Realizando las lecturas de estas curvas, podemos encontrar un patrón de calibración que representa las equivalencias matemáticas de las lecturas análogas y los valores esperados de humedad volumétrica en porcentaje (VWC). Estas se realizaron con el propósito de añadir al

programa lecturas no solo en un formato entendible para técnicos relacionados con el hardware de Arduino sino para que individuos relacionados al ámbito académico del agro pudieran tener lecturas comprensibles según sus campos de estudio.

Tras la fase de calibración de los sensores, se implementó la lógica de control dentro del programa desarrollado en Arduino IDE, utilizando lenguaje C++. Este programa se pensó para realizar la lectura continua de las señales analógicas provenientes de los 18 sensores YL-100, conectados al microcontrolador mediante dos multiplexores CD74HC4067. Gracias a estos multiplexores, fue posible ampliar el número de entradas analógicas disponibles en la placa Arduino UNO, optimizando el uso de pines y permitiendo el monitoreo de tres líneas de riego independientes, cada una conformada por seis de sensores.

El código se diseñó de manera modular, garantizando estabilidad en las lecturas y reduciendo interferencias eléctricas entre canales. Para cada sensor, el programa convierte la señal analógica en un valor de humedad volumétrica (%VWC) aplicando la ecuación de calibración obtenida experimentalmente. Estos valores son agrupados en tres arreglos, correspondientes a las líneas de riego 1, 2 y 3, con el fin de calcular el promedio de humedad de cada una. Este promedio permite determinar, en tiempo real, si la línea requiere activación de riego.

El sistema de control de riego se basa en la lógica de comparación por umbrales que se establecen premeditadamente. Si el valor promedio de humedad de una línea se encuentra por debajo del umbral preestablecido (definido en el código para cada línea, o manipulado manualmente según el usuario), la electroválvula correspondiente se activa automáticamente, permitiendo el paso de agua a los tubos de irrigación de esa zona. Cuando el valor promedio

supera el umbral, la válvula se desactiva, deteniendo el flujo de agua. Esta lógica garantiza que cada línea reciba el riego necesario en función de las condiciones reales del suelo, evitando el exceso o déficit hídrico.

Las lecturas de humedad, los promedios por línea y los estados de las electroválvulas se imprimen constantemente en el Monitor Serial del Arduino y, paralelamente, son almacenadas en formato JSON para ser enviadas a la base de datos en la nube mediante la capa de comunicación establecida en el segundo pilar del proyecto (back-end en Python). Esta sincronización permite que la aplicación web muestre en tiempo real los datos de humedad y el estado operativo del sistema de riego, facilitando el monitoreo remoto desde cualquier dispositivo.

En las pruebas de campo realizadas en la Granja UNIMINUTO y en simulaciones de laboratorio controladas, se verificó el funcionamiento correcto de los sensores, multiplexores y electroválvulas. El sistema fue capaz de detectar las variaciones de humedad del suelo, activar las válvulas según las condiciones de cada línea y mantener un ciclo de riego automatizado estable durante las sesiones de prueba. Estos resultados validaron la efectividad de la lógica implementada y sentaron la base para la integración del módulo Arduino con la infraestructura de software desarrollada en las siguientes fases del proyecto.

5.1.2 La programación del back-end y la configuración de la base de datos en la nube

El segundo pilar del proyecto se centró en la creación del back-end y la configuración de la base de datos en la nube, con el propósito de establecer una comunicación estable y segura entre el sistema físico de riego (Arduino UNO) y la aplicación web de supervisión y control remoto. Esta capa intermedia cumple un rol fundamental dentro del sistema, ya que actúa como el núcleo lógico encargado de

almacenar, procesar y distribuir la información proveniente de los sensores y de los módulos de control de riego.

Para el desarrollo de esta capa se empleó el lenguaje de programación **Python**, debido a su robustez, legibilidad y extensa compatibilidad con librerías orientadas al manejo de datos y servicios web. En conjunto con Python, se implementó la base de datos **CockroachDB**, una solución distribuida en la nube que ofrece alta disponibilidad, escalabilidad automática y tolerancia a fallos, características esenciales para garantizar la confiabilidad del sistema.

El back-end fue diseñado para recibir los paquetes de datos en formato **JSON** enviados desde el Arduino, los cuales contienen las lecturas individuales de cada sensor, los promedios de humedad volumétrica por línea, los estados de las electroválvulas y marcas temporales correspondientes a cada evento. Una vez recibidos los datos, el servidor ejecuta procesos de validación y almacenamiento en la base de datos, asegurando la integridad de la información y manteniendo un registro histórico de las mediciones obtenidas en el campo de cultivo.

Además, el sistema cuenta con una **API RESTful**, la cual permite el acceso controlado y estructurado a los datos almacenados. Dicha API constituye el punto de comunicación entre el back-end y la aplicación web, proporcionando las funcionalidades necesarias para consultar los valores de humedad en tiempo real, visualizar las tendencias históricas, y ejecutar comandos remotos de activación o desactivación de las válvulas de riego.

Durante la fase de pruebas, se implementaron endpoints específicos para la lectura y escritura de datos, así como mecanismos de autenticación que limitan el acceso a los usuarios autorizados. La estructura del modelo de datos fue definida con base en un esquema relacional, en el cual cada registro conserva información clave como el identificador del sensor, la línea de riego a la que pertenece, el valor de humedad medido, el estado de la válvula asociada y la fecha y hora de la lectura.

Esta arquitectura del back-end, junto con la base de datos en la nube, garantiza la persistencia y disponibilidad continua de los datos del sistema de riego, incluso en escenarios de desconexión temporal del hardware o la aplicación web. Gracias a esta configuración, el sistema no solo logra automatizar el proceso de irrigación, sino que además permite un monitoreo confiable, histórico y centralizado de las condiciones hídricas del cultivo.

5.1.3 Desarrollo del front-end y despliegue del PWA

El tercer pilar del proyecto se orientó hacia el desarrollo del front-end, entendido como la capa de interacción directa entre el usuario y el sistema de riego automatizado. En esta etapa se buscó diseñar y construir una interfaz intuitiva, funcional y accesible desde cualquier dispositivo conectado a internet, que permitiera tanto la supervisión en tiempo real de las lecturas de humedad del suelo como el control manual o automatizado del riego de cada línea del cultivo.

Para cumplir con estos objetivos, se optó por desarrollar una Aplicación Web Progresiva (PWA), una tecnología moderna que combina las ventajas de las aplicaciones móviles y las páginas web tradicionales. Este tipo de aplicación permite ser instalada directamente desde el navegador, operar en modo offline gracias al uso de Service Workers, y ofrecer una experiencia fluida y responsiva tanto en computadoras como en teléfonos móviles y tabletas.

La aplicación se diseñó siguiendo principios de usabilidad y diseño centrado en el usuario, priorizando la claridad de la información y la facilidad de acceso a las funciones principales. La interfaz principal presenta de forma visual y organizada las tres líneas de riego, mostrando para cada una el promedio de humedad volumétrica (VWC), el estado de su respectiva electroválvula, y la posibilidad de activar o desactivar manualmente el riego en caso de requerirse una intervención directa por parte del operador.

Además, el sistema incluye un apartado de historial y monitoreo, en el cual el usuario puede consultar las variaciones de humedad registradas a lo largo del tiempo. Estos datos, que provienen directamente de la base de datos en la nube gestionada por el back-end, son mostrados mediante gráficos y tablas dinámicas que facilitan la interpretación de las condiciones del suelo en diferentes periodos de observación. Esta característica es especialmente útil para los técnicos y académicos que realizan estudios comparativos sobre la humedad del suelo y su relación con las prácticas de riego.

Desde el punto de vista técnico, la aplicación fue desarrollada empleando tecnologías web modernas y desplegada en la plataforma Vercel, la cual ofrece un entorno estable y escalable para la ejecución de aplicaciones web progresivas. La comunicación con la API del back-end se realiza a través de peticiones HTTP, lo que permite actualizar la información de manera constante sin requerir recargas completas de la página. Esta arquitectura garantiza una experiencia fluida, similar a la de una aplicación nativa, y mantiene sincronizados los datos entre la interfaz del usuario y la base de datos alojada en la nube.

En términos de usabilidad práctica, la implementación del PWA representa un avance significativo en la forma en que los operarios, investigadores y estudiantes de la Granja UNIMINUTO pueden interactuar con el sistema de riego. Gracias a esta herramienta, el monitoreo y control del riego deja de depender exclusivamente del acceso físico a las instalaciones, posibilitando la gestión remota, en tiempo real y desde cualquier lugar, siempre que se disponga de conexión a internet. De igual manera, la capacidad de la aplicación para funcionar en modo offline asegura la continuidad operativa incluso ante fallos temporales de conectividad.

En conclusión, el desarrollo del front-end consolidó la integración del sistema completo, brindando una solución práctica, accesible y sostenible para la supervisión y control de riego en cultivos de arroz. Su diseño centrado en la experiencia del usuario refleja la intención del proyecto de ofrecer

una herramienta útil tanto para el ámbito académico como para la aplicación agrícola real, cumpliendo con el propósito de innovación tecnológica que motivó la creación del sistema automatizado de riego.

6 CAPÍTULO V

6.1 Conclusiones

El desarrollo de este proyecto permite concluir que la implementación de una arquitectura **Progressive Web App (PWA)** con un enfoque **Offline-First** fue una estrategia acertada para cumplir con los objetivos planteados. Se logró construir un sistema de gestión de riego robusto, resiliente a fallos de conectividad y de desarrollo ágil, validando la idoneidad del stack tecnológico seleccionado para entornos agrícolas con acceso intermitente a internet.

La elección de:

Next.js (React) junto con **Service Workers e IndexedDB** ofreció un balance ideal entre desarrollo rápido y una experiencia de usuario funcional en todo momento. El sistema demostró las siguientes ventajas clave:

Arquitectura Simplificada: La integración del front-end y el backend en un solo proyecto mediante las API Routes de Next.js redujo significativamente la complejidad del despliegue y mantenimiento, lo cual es ideal para un Producto Mínimo Viable (MVP).

Tolerancia a Fallos: El diseño centrado en la funcionalidad offline garantizó que la captura de datos en campo nunca se viera interrumpida por la falta de conexión, almacenando la información de manera segura en el dispositivo del usuario hasta su posterior sincronización automática.

Curva de Aprendizaje Rápida: Al basar el desarrollo en un ecosistema conocido como React, se pudo acelerar la fase de construcción y enfocar los esfuerzos en la lógica de negocio específica del sistema de riego.

Se cumplió con el objetivo general de desarrollar una aplicación web para la supervisión y control del riego, sentando una base tecnológica sólida que no solo resuelve la problemática actual, sino que también es fácilmente escalable.

6.2 Trabajo Futuro y Recomendaciones

Si bien el MVP actual es completamente funcional, la arquitectura implementada abre la puerta a futuras mejoras que podrían potenciar aún más el impacto del proyecto. Se recomienda explorar las siguientes líneas de trabajo:

Análítica Avanzada de Datos: Incorporar un Dashboard más detallado que permita visualizar históricos de humedad, consumo de agua por parcela y correlacionar estos datos con variables climáticas externas.

Modelos Predictivos: Utilizar los datos recolectados para entrenar un modelo de Machine Learning sencillo que pueda predecir las necesidades de riego futuras, optimizando aún más el uso del recurso hídrico.

Notificaciones en Tiempo Real: Implementar un sistema de alertas (vía push o email) que notifique a los supervisores sobre eventos críticos, como niveles de humedad demasiado bajos o altos, o fallos en algún sensor.

Integración de Más Sensores: Ampliar el sistema para incluir no solo sensores de humedad, sino también de pH del suelo, niveles de nutrientes (NPK) y medidores de caudal, ofreciendo una visión agronómica mucho más completa.

Gamificación y Usabilidad: Mejorar la interfaz de usuario con elementos de gamificación para incentivar las buenas prácticas agrícolas y facilitar la adopción de la herramienta por parte de personal con distintos niveles de habilidad tecnológica.

7 Referencias

- Agbetuyi, A. F. (2016). *Design and implementation of an automatic irrigation system based on monitoring soil moisture*. ResearchGate.
- Agency, U. S. (4 de Junio de 2024). *Microirrigación*. Obtenido de EPA: https://www-epa.gov.translate.google.com/translate?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sge#:~:text=El%20microrriego%20es%20un%20tipo,zona%20radicular%20de%20las%20plantas.
- Ana Gardey, J. P. (21 de Diciembre de 2021). *Sensor*. Obtenido de Definición.de: <https://definicion.de/sensor/>
- Arduino. (s.f.). *Qué es arduino*. Obtenido de Arduino: <https://arduino.cl/que-es-arduino/>
- Barragán, A. (10 de Febrero de 2025). *Por qué Next.js es el framework perfecto para aplicaciones React*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/por-que-nextjs-es-el-framework-perfecto-para-aplicaciones-react/>
- Churchville, F. (2020). *Interfaz de usuario (UI)*. Obtenido de TechTarget: <https://www.computerweekly.com/es/definicion/Interfaz-de-usuario-UI>
- Domínguez, R. (2022). *Sistema de riego inteligente con sensores y monitoreo automatizado de cultivo: Sedes regionales San Miguel y San Salvador*. Universidad Albert Einstein.
- Gómez, A. d. (Marzo de 2024). Presentación de proyecto de investigación a profesores asesores de TCDS.
- Gutierrez, J., Villa-Medina, J., Nieto, A., & Porta, M. (2014). *Automated irrigation system using a wireless sensor network and GPRS module*. IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Harb, A. (2018). *Service Workers: ¿Qué son y cómo se comen?* Obtenido de Medium:

https://medium.com/@anuar_harb/service-workers-qu%C3%A9-son-y-c%C3%B3mo-se-comen-a78166b0d88e

Hernández. (2021). *Sistema de riego automático para el agro basado en Arduino e IoT alimentado por un sistema fotovoltaico*. Revista Universitaria UNIVO. Obtenido de

<https://www.camjol.info/index.php/revunivo/article/view/11377>

IBM. (s.f.). *Automatización*. Obtenido de IBM: <https://www.ibm.com/es-es/topics/automation>

IndexedDB. (2025). Obtenido de MDB Web Docs:

https://developer.mozilla.org/es/docs/Web/API/IndexedDB_API

Innovatione. (8 de Julio de 2019). *Sistemas de riego I*. Obtenido de Innovatione:

<https://innovatione.eu/2019/07/08/riego-agricultura/>

Julián Pérez Porto, A. G. (22 de Julio de 2022). *Agronomía - Qué es, utilidad, definición y concepto*.

Obtenido de Definición.de: <https://definicion.de/agronomia/>

Julián Perez Porto, M. M. (7 de Julio de 2022). *Irrigación - Qué es, en la biología, definición y concepto*.

Obtenido de Definicion.de: <https://definicion.de/irrigacion/>

Maida, E. G., & Pacienza, J. (2015). *Metodologías de Desarrollo de Software*. Universidad Católica de Argentina.

MDN. (2025). *Aplicaciones Web Progresivas*. Obtenido de MDN Web Docs:

https://developer.mozilla.org/es/docs/Web/Progressive_web_apps

Merino, M. (20 de Diciembre de 2018). *Control Remoto*. Obtenido de Definicion.de:

<https://definicion.de/control-remoto/>

More, M. (16 de Noviembre de 2023). *¿Qué es la UX y la UI?* Obtenido de IEBS:

<https://www.iebschool.com/hub/que-es-la-ux-y-la-ui-analitica-usabilidad/>

Paessler. (2025). *Bases de datos*. Obtenido de Paessler: <https://www.paessler.com/es/it-explained/database>

Pastrana, M. (2025). *Diseño de sistema de riego automático con sensores de humedad de suelo, temperatura y calidad del agua*. Obtenido de <https://repository.unad.edu.co/handle/10596/66860>

Python.org. (s.f.). *FAQ General*. Obtenido de Python: <https://docs.python.org/es/3.10/faq/general.html>

Sarvaiya, D. (2023). *Web App*. Obtenido de Intelivita: <https://www.intelivita.co.uk/blog/web-app>

Significados, E. (11 de Noviembre de 2023). *Hardware*. Obtenido de Significados: <https://www.significados.com/hardware/>

Significados, E. (9 de Diciembre de 2024). *Software*. Obtenido de Significados: <https://www.significados.com/software/>

Yasar, K. (Junio de 2024). *Internet Of Things (IoT)*. Obtenido de TechTarget: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>