



Universidad Tecnológica ECOTEC

Facultad de Ingenierías

Proyecto Integrador Curricular

Título del trabajo:

Sistema integral automático de monitoreo y ajuste de parámetros ambientales controlables
para plantaciones en la Hacienda Orillas del Sol

Línea de Investigación:

Tecnologías de la Información y Comunicación

Carrera:

Ingeniería en Sistemas Inteligentes

Autores:

Omar Antonio Salas Maldonado

Samuel Elías Peña Nicola

Docente:

Marcos Antonio Espinoza Mina



ANEXO No. 9

**PROCESO DE TITULACIÓN
CERTIFICADO DE APROBACIÓN DEL TUTOR**

~~Sembrador~~, 19 de diciembre de 2024

Magíster
Erika Ascencio
Facultad de Ingenierías, Arquitectura y Ciencias de la Naturaleza
Universidad Tecnológica ECOTEC

De mis consideraciones:

Por medio de la presente comunico a usted que el trabajo de titulación TITULADO: Sistema integral automático de monitoreo y ajuste de parámetros ambientales controlables para plantaciones en la Hacienda Orillas, fue revisado, siendo su contenido original en su totalidad, así como el cumplimiento de los requerimientos establecidos en la guía para su elaboración, por lo que se autoriza a los estudiantes: **PEÑA NICOLA SAMUEL ELÍAS y SALAS MALDONADO OMAR ANTONIO**, para que proceda con la presentación oral del mismo.

ATENTAMENTE,



**MARCOS ANTONIO
ESPINOZA MINA**

Ing. Marcos Antonio Espinoza Mina, PhD.

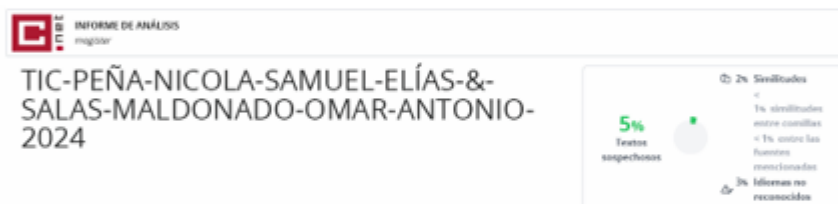
Tutor



ANEXO No. 10

**PROCESO DE TITULACIÓN
CERTIFICADO DEL PORCENTAJE DE COINCIDENCIAS
DEL TRABAJO DE TITULACIÓN**

Habiendo sido revisado el trabajo de titulación TITULADO: Sistema integral automático de monitoreo y ajuste de parámetros ambientales controlables para plantaciones en la Hacienda Orillas elaborado por PEÑA NICOLA SAMUEL ELÍAS y SALAS MALDONADO OMAR ANTONIO fue remitido al sistema de coincidencias en todo su contenido el mismo que presentó un porcentaje del 5% mismo que cumple con el valor aceptado para su presentación que es inferior o igual al 10% sobre el total de hojas del documento. Adicional se adjunta print de pantalla de dicho resultado.



ATENTAMENTE,



firmado digitalmente por
MARCOS ANTONIO
ESPINOZA MINA

Ing. Marcos Antonio Espinoza Mina, PhD.
Tutor

Dedicatoria

Dedicamos este trabajo a nuestros padres, quienes con su esfuerzo, amor y apoyo incondicional nos han enseñado a nunca rendirnos. A nuestros profesores y mentores, que sembraron en nosotros el conocimiento y la pasión por aprender. Finalmente, a la Hacienda Orillas del Sol, que se convirtió en inspiración y contexto para este proyecto, contribuyendo a nuestra formación profesional y personal.

Agradecimiento

Queremos expresar nuestro más sincero agradecimiento a todas las personas que hicieron posible este proyecto. A nuestro tutor, Marcos Antonio Espinoza Mina, por su guía, paciencia y dedicación durante todo el proceso. A los operadores y trabajadores de la Hacienda Orillas del Sol, por su disposición y colaboración en la implementación y evaluación del sistema. A nuestros compañeros y amigos, por brindarnos su apoyo constante y aliento en los momentos más difíciles. Finalmente, a nuestras familias, cuyo amor y respaldo inquebrantable fueron la fuerza que nos impulsó a alcanzar esta meta.

Resumen

El presente trabajo tiene como objetivo desarrollar e implementar un sistema automatizado de monitoreo para la pequeña Hacienda Orillas del Sol, con el fin de optimizar la gestión de recursos y mejorar la eficiencia en sus procesos productivos. El sistema se enfoca en la medición de parámetros clave como temperatura y humedad, utilizando sensores Arduino y tecnologías accesibles de bajo costo.

La investigación se estructura en cuatro fases:

1. Diseño del sistema.
2. Desarrollo de los componentes físicos y lógicos.
3. Implementación en campo.
4. Pruebas y validación para evaluar su eficiencia y rendimiento.

La metodología utilizada es de tipo mixto, integrando la medición cuantitativa de los parámetros agrícolas con entrevistas cualitativas a los operadores de la hacienda. Los resultados muestran que el sistema reduce en un 30% el tiempo dedicado al monitoreo manual y optimiza el uso de recursos como agua y energía, incrementando la productividad de manera sostenible.

La información presentada en este proyecto es pertinente y directamente relacionada con las necesidades de la Hacienda Orillas del Sol, una finca con recursos limitados que requiere soluciones prácticas y económicas para enfrentar problemas de monitoreo y gestión de cultivos. Se establece una secuencia lógica entre la identificación del problema, los objetivos del estudio, la metodología aplicada y los resultados obtenidos.

El sistema desarrollado demuestra ser una alternativa viable y eficiente para automatizar procesos agrícolas en pequeñas haciendas, aportando beneficios medibles como la reducción de costos operativos y una mayor eficiencia en la toma de decisiones basada en datos confiables.

Abstract

The following project focuses on the development and implementation of an automated monitoring and adjustment system for environmental parameters in agricultural plantations, specifically designed for Hacienda Orillas del Sol. The aim is to improve productivity and sustainability through the efficient use of resources such as water and soil management.

A mixed-method approach was employed, combining quantitative data collected via low-cost sensors (DHT11 for temperature and humidity, YL-69 for soil moisture) integrated with an Arduino-based system, and qualitative insights from interviews conducted with the operators of the hacienda. The system gathers real-time data, stores it in a MongoDB database, and displays the information through an intuitive web-based dashboard built using React.js and .NET Core.

The results demonstrated a 20% reduction in water waste and a 30% optimization in monitoring time, validating the system's effectiveness in addressing the challenges of manual agricultural management. This project highlights the feasibility of implementing accessible and low-cost technologies in rural settings, aligning with Ecuador's National Agricultural Policy 2020-2030 to promote innovation and sustainability in the agricultural sector.

Índice

Dedicatoria.....	5
Agradecimiento.....	6
Resumen	7
Abstract.....	9
Índice	10
1. Introducción	18
1.1. Contexto Histórico.....	20
1.2. Planteamiento del Problema	22
1.3. Objetivo General.....	24
1.4. Objetivos Específicos.....	24
1.5. Justificación	24
2. Marco Teórico	27
2.1. Evolución en la Agricultura: Cómo la automatización moderniza las prácticas agrícolas.....	27
2.1.1. Beneficios de la automatización en la agricultura.....	29
2.2. Aplicación de Arduino y Tecnologías IoT en la Agricultura.....	31
2.2.1. Comparativa de Arduino con otros microcontroladores	31
2.2.2. Razones para la selección de Arduino	33

2.2.3.	Microcontroladores y Entornos de Desarrollo: Herramientas Esenciales en la Agricultura Automatizada	33
2.2.4.	Arduino: Accesibilidad y Usos en Proyectos de Innovación.....	34
2.2.5.	Modelos de Arduino y su Versatilidad para Proyectos Agrícolas	35
2.2.6.	Ventajas de Arduino en Proyectos Agrícolas.....	36
2.2.7.	Integración de Arduino con Otras Tecnologías	37
2.2.8.	Compatibilidad con Tecnologías de Software.....	38
2.2.9.	Propósito de Arduino en el Proyecto	38
2.2.10.	Desplazamiento Autónomo del Robot.....	39
2.2.11.	Monitoreo y Recolección de Variables Ambientales	39
2.2.12.	Flujo de Datos: Captura, Transmisión y Visualización	40
2.3.	Gestión de Bases de Datos con MongoDB	41
2.3.1.	Estructura y Gestión de la Base de Datos.....	42
2.3.2.	Justificación de la Selección de MongoDB como Base de Datos	44
2.3.3.	Comparación con Bases de Datos Relacionales.....	44
2.3.4.	Uso de la Base de Datos en el Proyecto.....	46
2.3.5.	Integración de MongoDB con .NET Core	46
2.4.	Introducción a .NET Core y su Evolución hasta la Versión 7	47
2.4.1.	Funciones Principales de un Framework	48
2.4.2.	Razones para la Elección de .NET Core sobre otros Frameworks.....	50

2.4.3.	.NET Core y su Relación con el Modelo MVC	52
2.5.	Modelo de Proyecto MVC con .NET Core 7	52
2.5.1.	Flexibilidad del MVC en .NET Core 7	53
2.5.2.	Características del MVC en .NET Core 7	53
2.5.3.	Limitaciones del MVC en .NET Core 7	55
2.5.4.	Justificación del Uso del Modelo MVC con .NET Core	55
2.6.	Desarrollo del Backend con .NET Core.....	57
2.6.1.	Estructura del Backend y Funcionalidades Principales	57
2.6.2.	Procesamiento y Flujo de Datos	58
2.6.3.	Herramientas y Tecnologías Implementadas.....	59
2.6.4.	Comunicación entre React y .NET Core: Desarrollo e Integración del Backend 60	
2.7.	Uso de React como Capa de Vista en el Modelo MVC	60
2.7.1.	Flujo de Funcionamiento del Frontend	61
2.7.2.	Implementación en el Proyecto de Agricultura Automatizada	62
2.7.3.	Razones para la Elección de React sobre otros Frameworks	62
2.7.4.	Gestión de Versiones en el Desarrollo Backend y Frontend.....	64
2.8.	Concepto de Git y GitHub	65
2.8.1.	Funcionalidades Clave de GitHub.....	65
2.8.2.	Beneficios de Usar GitHub para el Versionamiento de Proyectos	66

2.8.3.	Automatización del Despliegue con GitHub y la Infraestructura	67
2.9.	Infraestructura de Despliegue y Servicios Asociados	67
2.9.1.	Digital Ocean: Servidor en la Nube	67
2.9.2.	Docker: Contenedorización.....	68
2.9.3.	Cloudflare: CDN y Seguridad.....	69
2.9.4.	Nginx: Servidor Web	69
2.9.5.	Mailgun: Gestión de Correos Electrónicos	70
2.9.6.	Namecheap: Dominio Web	70
2.10.	Automatización de Tareas y Sincronización de Datos con Zapier	71
2.11.	<i>Metodologías de Desarrollo Ágil: Kanban y Extreme Programming</i>	72
2.11.1.	Introducción a Metodologías Ágiles	72
2.11.2.	Características de las Metodologías Ágiles.....	72
2.11.3.	Metodología Kanban.....	73
2.11.4.	Principios Fundamentales de Kanban.....	73
2.11.5.	Elementos del Tablero Kanban	73
2.11.6.	Ventajas de Implementar Kanban	74
2.11.7.	Aplicación de Kanban en el Proyecto.....	74
2.11.8.	Metodología Extreme Programming.....	75
2.11.9.	Principios Fundamentales de Extreme Programming.....	75

2.11.10.	Prácticas Clave de Extreme Programming	76
2.11.11.	Ventajas de Implementar Extreme Programming.....	76
2.11.12.	Aplicación de Extreme Programming en el Proyecto	77
2.12.	Impacto y Beneficios en el Contexto Ecuatoriano.....	77
2.12.1.	Competitividad Internacional a través de la Automatización	78
2.12.2.	Atracción de Inversiones y Apoyo Institucional	79
2.12.3.	Adaptación al Cambio Climático.....	79
2.13.	<i>Sustento Legal en el Proyecto de Automatización Agrícola</i>	80
2.13.1.	Constitución de la República del Ecuador	80
2.13.2.	Ley Orgánica del Régimen de la Soberanía Alimentaria (LORSA)	81
2.13.3.	Código Orgánico del Ambiente (COA).....	81
3.	Metodología	82
3.1.	Métodos de Investigación	82
3.2.	Recolección de Datos Cualitativos y Cuantitativos.....	84
3.2.1.	Datos Cuantitativos:.....	84
3.2.2.	Datos Cualitativos:	84
3.3.	Procesamiento y Análisis de Datos	85
3.4.	Fases de Prueba y Validación del Prototipo	85
3.5.	Elementos Metodológicos Específicos para TI.....	86

4.	Análisis de Resultados.....	87
4.1.	Fase #1: Preparación y configuración de componentes básicos.....	88
4.2.	Fase #2: Integración de Comunicación y Almacenamiento de Datos	89
4.3.	Fase #3: Automatización y Pruebas de Calidad	90
4.4.	Fase 4: Evaluación del Proyecto.....	91
4.5.	Fase 5: Subida a Producción	92
4.6.	Encuesta de Evaluación de Desempeño.....	92
4.6.1.	Satisfacción de Usuario	92
4.6.2.	Encuesta de efectividad del sistema	97
4.6.3.	Encuesta de análisis de rendimiento.....	101
4.7.	Tabla de asignación de tareas	104
4.7.1.	Fase #1.....	104
4.7.2.	Fase #2.....	105
4.7.3.	Fase #3.....	105
4.7.4.	Fase #4.....	106
4.7.5.	Fase #5.....	106
5.	Conclusión.....	107
5.1.	Mejora en la Productividad y Sostenibilidad	107
5.2.	Análisis de Factores Críticos.....	107

5.3.	Desarrollo del Sistema Automatizado.....	107
5.4.	Diseño del Software y Visualizaciones.....	108
5.5.	Validación del Sistema.....	108
6.	Recomendaciones.....	108
7.	Bibliografía.....	110
8.	Anexos.....	124
8.1.	Sección #1: Acta de Reuniones.....	124
8.1.1.	Acta #1.....	124
8.1.2.	Acta #2.....	126
8.1.3.	Acta #3.....	128
8.1.4.	Acta #4.....	129
8.1.5.	Acta #5.....	131
8.1.6.	Acta #6.....	134
8.2.	Sección #2 Evidencias de Avance Técnico del Proyecto.....	137
8.2.1.	Avance #1.....	137
8.2.2.	Avance #2.....	140
8.2.3.	Avance #3.....	143
8.2.4.	Avance #4.....	146
8.2.5.	Avance #5.....	148

8.2.6.	Avance #6.....	151
8.2.7.	Avance #7.....	155
8.2.8.	Avance #8.....	157
8.2.9.	Avance #9.....	160
8.2.10.	Avance #10	162
8.3.	Sección #3: Otros	165
8.3.1.	Materiales:	165
8.3.2.	Manual de Usuario.....	166

1. Introducción

La Hacienda Orillas del Sol, enfrenta desafíos significativos en el monitoreo eficiente de variables ambientales debido al uso de métodos manuales tradicionales. Esto genera desperdicio de recursos como agua y fertilizantes, incrementando costos operativos y afectando la productividad. La implementación de un sistema automatizado basado en sensores Arduino busca optimizar la gestión de parámetros como temperatura y humedad, alineándose con las metas de sostenibilidad y eficiencia productiva establecidas en la Política de Estado para el Sector Agropecuario 2020-2030 (Ministerio de Agricultura y Ganadería, 2020, art. 3, literal c).

En la época donde nos encontramos la tecnología y la sostenibilidad son pilares fundamentales para el desarrollo tanto como de nuevas tecnologías como del sector agrícola, es de suma importancia hacer una búsqueda de soluciones que optimicen recursos y mejoren el rendimiento de las plantaciones. El desarrollo de un sistema que nos permita tener la capacidad de automatizar la supervisión y el ajuste de variables ambientales controlables para mejorar la salud y la productividad de las plantas.

En Ecuador, la adopción de tecnologías de la información y comunicación (TIC) en el sector agrícola es limitada. Según el Instituto Nacional de Estadística y Censos (INEC), en 2020, solo el 15% de las unidades de producción agropecuaria utilizaban algún tipo de tecnología digital en sus procesos (INEC, 2020). Esta baja adopción tecnológica se debe, en parte, a la falta de infraestructura adecuada y al acceso limitado a financiamiento para pequeños y medianos agricultores.

Para contextualizar el análisis, es crucial destacar que la baja adopción tecnológica en la agricultura ecuatoriana responde en parte a factores estructurales y de acceso financiero. La escasez de inversión en infraestructura y tecnología agrícola ha creado un entorno donde la mayoría de los agricultores dependen de prácticas manuales y herramientas de baja eficiencia donde se cree que en el sector agrícola el trabajo manual estaría presente en un 90% del mismo, lo cual frena su competitividad frente a otros países con mayores niveles de mecanización. Además, la baja disponibilidad de créditos y asistencia técnica limita a los pequeños agricultores en la implementación de soluciones automatizadas, impidiendo mejoras significativas en productividad y eficiencia. (Intriago F., 2019)

Estudios como los de Zamora Boza et al. (2021) destacan que el monitoreo manual puede elevar los costos operativos en un 15% anual debido a la falta de precisión en la gestión de recursos, impactando negativamente en la productividad y calidad de los cultivos. Este fenómeno es particularmente crítico en pequeñas fincas como la Hacienda Orillas del Sol, donde los márgenes de rentabilidad son limitados.

La modernización de la agricultura, que incluye sistemas de riego de precisión y el uso de maquinaria avanzada, podría optimizar recursos, especialmente en áreas rurales con acceso limitado al agua. Esto es vital para reducir desperdicios y mejorar la sostenibilidad en la producción. En este contexto, implementar tecnologías como Arduino o sistemas de monitoreo automático no solo representaría un avance en la producción, sino también en la calidad de vida de las comunidades agrícolas al reducir el trabajo manual intensivo y mejorar los ingresos. Comparado con el panorama global, Ecuador enfrenta desafíos únicos en su infraestructura de apoyo y financiación, lo que hace urgente la exploración de alternativas

tecnológicas accesibles y de bajo costo para impulsar la eficiencia agrícola local. (Intriago, 2019)

El uso de sensores Arduino en combinación con tecnologías IoT representa una alternativa accesible y funcional para automatizar el monitoreo en pequeñas haciendas. Según Cuesta y Solís (2022), las soluciones basadas en Arduino pueden reducir costos de implementación en un 70% en comparación con sistemas comerciales, facilitando su adopción en contextos de recursos limitados. Al comparar estas opciones, las plataformas comerciales y dispositivos IoT ofrecen mayores capacidades, pero su alto costo y complejidad limitan su aplicación en pequeñas fincas como la Hacienda Orillas del Sol.

El desarrollo de este sistema automatizado no solo responde a las necesidades inmediatas de la Hacienda Orillas del Sol, sino que también se alinea con los Objetivos de Desarrollo Sostenible de la Agenda 2030, específicamente en la optimización de recursos y la mejora de la sostenibilidad agrícola (Ministerio de Agricultura y Ganadería, 2020).

1.1. Contexto Histórico

Con el propósito de adecuarse a las condiciones imperantes en el mercado y a las políticas, los productores agrícolas han sido blanco de diferentes recomendaciones que se centran en la modernización de su agricultura. Se propone que a través de los usos de la tecnología y técnicas sofisticadas un proceso que se supone lineal donde lo uso de lo tradicional es suplantado por la innovación tecnológica lo que trae consigo fuertes situaciones económicas y socioculturales que muchos han trabajado en términos de sus impactos e implicaciones del orden social. (Cuesta & Solís, 2022)

Visto de esta forma, la tecnología es un factor clave para el desarrollo del sector agrícola y claramente es necesario para incrementar la competitividad de cara a otras partes productivas nacionales e internacionales. La competitividad se conoce como la capacidad de ingresar a un mercado y posicionarse en él, Siempre es necesario poseer algún tipo de ventaja sobre la potencial competencia en términos de calidad, precio, cantidad, oportunidades, empaques o presentación. El proceso de la modernización de la agricultura existe gracias a la generación o desarrollo de tecnologías adecuadas a las expectativas de formación de capital y al incremento en la productividad, con un contexto dado este es uno de los recursos más escaso en momentos más importantes es decir que suele convertirse en una limitante. (The Rise of Autonomous Farms: How Technology Is Revolutionizing Agriculture, 2023)

En este marco global de modernización, Ecuador enfrenta retos particulares debido a su estructura económica y limitaciones en la adopción de tecnologías avanzadas. La limitada inversión en infraestructura y tecnología agrícola ha llevado a que la mayoría de los agricultores dependan de prácticas manuales y herramientas de baja eficiencia, lo que reduce su competitividad en comparación con países que han avanzado en mecanización agrícola. Además, la escasa disponibilidad de créditos y asistencia técnica limita a los pequeños agricultores en la implementación de soluciones automatizadas, impidiendo mejoras sustanciales en productividad y eficiencia. (Zamora Boza et al., 2021)

La modernización de la agricultura, que incluye sistemas de riego de precisión y el uso de maquinaria avanzada, podría optimizar recursos, especialmente en áreas rurales con acceso limitado al agua. Esto es vital para reducir desperdicios y mejorar la sostenibilidad en la producción. En este contexto, implementar tecnologías como Arduino o sistemas de monitoreo automático no solo representaría un avance en la producción, sino también en la

calidad de vida de las comunidades agrícolas al reducir el trabajo manual intensivo y mejorar los ingresos. Comparado con el panorama global, Ecuador enfrenta desafíos únicos en su infraestructura de apoyo y financiación, lo que hace urgente la exploración de alternativas tecnológicas accesibles y de bajo costo para impulsar la eficiencia agrícola local. (“Proyecto Nacional de Innovación Tecnológica Participativa y Productividad Agrícola (PITPPA)”, s. f.).

En Ecuador, la empresa Harvest Ecuador ha implementado sistemas avanzados de monitoreo y ajuste de parámetros ambientales en sus plantaciones. Utilizando tecnología de sensores y plataformas digitales, han optimizado el control de variables como la humedad del suelo, la temperatura y la radiación solar, lo que ha resultado en una mejora significativa en la eficiencia y productividad de sus cultivos. Esta adopción de herramientas tecnológicas ha permitido a Harvest Ecuador mantener altos estándares de calidad y sostenibilidad en su producción agrícola.

En el marco de los Objetivos de Desarrollo Sostenible de la Agenda 2030, la automatización y tecnificación del sector agrícola no solo representan una oportunidad de optimización de recursos, sino también un mecanismo clave para garantizar la seguridad alimentaria y combatir la pobreza rural. La visión nacional para el agro al 2030 incluye un enfoque hacia la sostenibilidad e innovación tecnológica, destacando que el sector será "reconocido como estratégico y motor del desarrollo sostenible del país" (Ministerio de Agricultura y Ganadería, 2020, art. 2).

1.2. Planteamiento del Problema

Administrar de forma eficaz nuestros recursos ambientales es fundamental para el desarrollo y la productividad de territorios agrícolas. Los métodos tradicionales no siempre alcanzan la

precisión necesaria, impactando directamente en la calidad de las cosechas y elevando los costos operativos. Esta problemática se encuentra reflejada en la *Política de Estado para el Sector Agropecuario Ecuatoriano 2020-2030*, que establece como uno de sus objetivos estratégicos el "fortalecimiento de la producción agropecuaria incrementando su rentabilidad por medio de la productividad y la reducción de costos productivos a través de la tecnificación" (Ministerio de Agricultura y Ganadería, 2020, art. 3, literal c).

No llevar un control en tiempo real suponen un obstáculo significativo distanciando las herramientas convencionales de los procesos que podrían ser automatizados para mejorar la productividad y reducir costos. Se especula que apenas un 20% de las plantaciones en la región han integrado tecnologías de monitoreo, resultando en una productividad notablemente inferior comparada con aquellas que sí están automatizadas. Los costos operativos se elevarían aproximadamente en un 15% anualmente debido a la ineficiencia en la gestión de recursos, la falta de control continuo puede llevar a daños por el uso excesivo de agua y fertilizantes, deteriorando el suelo. Esto puede disminuir el rendimiento y calidad de los productos, y comprometer la sostenibilidad de las prácticas agrícolas debido al uso no óptimo de los recursos naturales.

Ante esta problemática, el desarrollo de un sistema automatizado para el monitoreo y ajuste de parámetros ambientales responde no solo a las necesidades de la finca beneficiaria, sino también a los lineamientos de la política nacional. Este proyecto se alinea con los instrumentos de política que buscan "impulsar la investigación e innovación tecnológica" y "potenciar los servicios de transferencia tecnológica" para mejorar la productividad en sistemas agroalimentarios

1.3. Objetivo General

Desarrollar el prototipo de un sistema para el monitoreo y ajuste de parámetros ambientales controlables para plantaciones agrícolas con el fin de buscar una mejora en la productividad y sostenibilidad de las cosechas.

1.4. Objetivos Específicos

- Analizar los factores ambientales críticos que influyen en el mantenimiento de las plantaciones agrícolas mediante la implementación de sensores de monitoreo en tiempo real.
- Identificar al menos tres factores ambientales clave (humedad del suelo, temperatura y humedad del suelo) mediante sensores en tiempo real.
- Desarrollar un sistema automatizado de monitoreo y ajuste de parámetros ambientales controlables con acciones que responda a las necesidades específicas de la hacienda.
- Diseñar un software intuitivo y eficiente que procese y almacene los datos recolectados por los sensores, y presente esta información a través de al menos 3 visualizaciones gráficas de manera clara y accesible a través de una interfaz de usuario.
- Validar el sistema alcanzando un 90% de precisión en la recopilación durante pruebas en campo.

1.5. Justificación

La Hacienda Orillas del Sol, una finca pequeña ubicada en [insertar ubicación si aplica], enfrenta desafíos relacionados con la gestión ineficiente de recursos ambientales como el agua y los fertilizantes debido al uso de métodos manuales para el monitoreo de variables

como temperatura y humedad. Estas prácticas tradicionales generan desperdicio de recursos, incrementan los costos operativos y reducen la productividad de los cultivos.

De acuerdo con datos del Banco Central del Ecuador (2020), en fincas no tecnificadas, los costos operativos aumentan hasta un 10% anual debido a la ineficiencia en la gestión de recursos. En el caso de la Hacienda Orillas del Sol, se estima que un 20% del agua utilizada en el riego se desperdicia, lo que impacta directamente en la rentabilidad de la producción.

Hoy en día, se requiere una instalación de este tipo de sistemas para poder dar solución en el campo de la agricultura, la investigación aportara al conocimiento existente sobre la integración de estas tecnologías al estudiar la efectividad de los diferentes sensores y métodos de control se genera conocimiento que puede llegar a ser utilizado por algún otro profesional del campo. Estos desafíos incluyen la necesidad de aumentar la eficiencia en el uso de recursos y reducir el impacto ambiental negativo asociado con las prácticas agrícolas tradicionales. Los avances en la tecnología de sensores han demostrado ser efectivos para mejorar el monitoreo de la salud de las plantas y optimizar las condiciones de crecimiento.

La implementación de un sistema automatizado basado en sensores Arduino permitirá optimizar el monitoreo y control de las variables ambientales, ajustando de manera precisa los niveles de humedad y temperatura según las necesidades de los cultivos. Estudios previos han demostrado que sistemas similares pueden reducir los costos de operación hasta en un 25% y mejorar significativamente la eficiencia del riego (BCE, 2020).

Desarrollar un sistema con estas características requiere de innovación en el uso de herramientas y técnicas de recopilación de datos. Su desarrollo contribuirá a que luego haya nuevos métodos para la integración de estos sistemas y facilitar la gestión de la hacienda. La

integración de sensores y métodos de control automatizado no solo aporta al conocimiento existente, sino que también genera nuevas oportunidades para la gestión eficiente de los recursos agrícolas.

Para determinar la tecnología más adecuada a este contexto, se han evaluado diversas alternativas. Antes de optar por Arduino, es esencial evaluar otras opciones tecnológicas aplicables al contexto agrícola ecuatoriano. Por ejemplo, las plataformas comerciales de monitoreo agrícola ofrecen herramientas avanzadas para la supervisión de cultivos, pero su alto costo y complejidad pueden ser prohibitivos para pequeños y medianos agricultores. En contraste, Arduino se presenta como una solución de bajo costo y fácil accesibilidad, permitiendo a los agricultores locales desarrollar sistemas personalizados de monitoreo y control que se ajusten a sus necesidades específicas. (“Agricultura Inteligente Con Arduino: Optimizando El Futuro”, 2021)

Otra alternativa tecnológica a considerar es el uso de dispositivos IoT comerciales, que suelen integrarse en sistemas agrícolas inteligentes. Estos dispositivos ofrecen mayor precisión y capacidad de integración con sistemas de gestión avanzados; sin embargo, su costo de instalación puede superar los \$500 a \$1000 USD y la necesidad de infraestructura adicional pueden ser limitantes para la Hacienda Orillas del Sol. Al comparar estas opciones, Arduino destaca no solo por su costo accesible, sino también por su adaptabilidad en entornos rurales, donde se requiere de herramientas que puedan mantenerse y operarse de manera sencilla. Esto permite que los agricultores locales accedan a tecnologías de monitoreo automatizado, maximizando los recursos con una mínima inversión y facilitando su implementación en un mercado agrícola de bajo presupuesto. (“Tecnología Agrícola Inteligente con Arduino”, s/f)

Este proyecto contribuirá no solo a resolver los desafíos actuales de la Hacienda Orillas del Sol, sino que también aportará conocimiento práctico sobre la integración de tecnologías de bajo costo en fincas pequeñas, lo cual puede replicarse en otras unidades productivas con recursos limitados. La investigación generará datos precisos sobre la eficiencia de los sensores y la viabilidad técnica de los sistemas automatizados, facilitando la toma de decisiones informadas para la optimización de la producción agrícola.

Por último, este trabajo se alinea con las metas de la Política de Estado para el Sector Agropecuario Ecuatoriano 2020-2030, que prioriza la tecnificación de procesos y la reducción de costos productivos a través de la automatización, así como con los Objetivos de Desarrollo Sostenible de la Agenda 2030, promoviendo la sostenibilidad y eficiencia en las prácticas agrícolas.

2. Marco Teórico

En esta sección, se presentan las teorías y estudios previos relacionados con la automatización agrícola y las tecnologías de monitoreo ambiental, destacando su aplicación en pequeñas fincas como la Hacienda Orillas del Sol. Además, se analizarán las ventajas y limitaciones de diferentes enfoques tecnológicos.

2.1. Evolución en la Agricultura: Cómo la automatización moderniza las prácticas agrícolas

En los sectores rurales donde la siembra y la cosecha definió el trabajo de generaciones, se está llevando a cabo una seria transformación en la agricultura. El esfuerzo más antiguo de la humanidad está experimentando una evolución notable gracias al avance de la

automatización. Mientras nos encontramos en una batalla entre la innovación y la tradición, es importante explorar cómo la automatización vendrá a remodelar el paisaje de la agricultura, revolucionando las antiguas prácticas y marcando un comienzo en la era de la eficiencia, sostenibilidad y productividad sin precedentes. (“Evolution in Agriculture: How Automation is Modernizing Farming Practices - The Agrotech Daily. The Agrotech Daily”, 2023)

La automatización está teniendo un impacto en la agricultura verdaderamente revolucionario. Los días en que la agricultura se llevaba a cabo únicamente por el sudor de los humanos y la codicia impredecible de las fuerzas naturales quedaron atrás. La agricultura moderna se ve atravesada por la tecnología avanzada, la robótica, los análisis de datos e incluso la inteligencia artificial, lo que permite a los productores trabajar de manera más inteligente. Pero esta transformación no solo implica la adquisición de nuevas herramientas: se trata de un cambio radical en la forma en que abordamos la producción de lo que comemos, la gestión de nuestra tierra y cómo sostenemos nuestro planeta. (“¿Qué es la agricultura inteligente?”, 2024)

La automatización en la agricultura implica el uso de dispositivos y tecnologías inteligentes para automatizar diferentes procesos agrícolas. Va desde máquinas automatizadas simples hasta sistemas complejos de IA que pueden monitorear y administrar granjas.

Las herramientas de automatización más usadas son las siguientes:

- **Agricultura de Precisión:** La agricultura de precisión es la práctica de usar información basada en datos para tomar decisiones informadas sobre las operaciones agrícolas. Esto puede incluir el uso de sensores para medir la humedad del suelo, la temperatura y otros factores ambientales. Estos datos luego se utilizan para

determinar cuándo regar, cuánto fertilizante aplicar y otras prácticas agrícolas. (Marco, 2024)

- **Agricultura Robótica:** La robótica se está utilizando para automatizar una variedad de tareas agrícolas, desde la siembra y la cosecha hasta el control de malezas y el monitoreo de cultivos. Los robots pueden trabajar sin parar, no se cansan y pueden realizar tareas más rápido que los humanos. También pueden trabajar en entornos peligrosos, lo cual es ideal para la agricultura. (“Agricultura automatizada y robótica agrícola”, 2021)
- **Drones:** Los drones se utilizan para monitorear cultivos y detectar enfermedades y plagas. También se utilizan para cartografiar granjas y recopilar datos sobre la salud de las plantas, la calidad del suelo y otros factores ambientales. Los drones pueden cubrir áreas grandes y en poco tiempo, lo que los hace ideales para la agricultura. (“Aplicaciones con dron en la agricultura moderna”, 2024)

2.1.1. Beneficios de la automatización en la agricultura

- **Optimización de la eficiencia:** La implementación de sistemas automáticos minimiza la intervención manual, liberando a los agricultores para que se dediquen a actividades que requieren un conocimiento más especializado. Estos sistemas tienen la capacidad de operar ininterrumpidamente, lo que mejora la eficacia de los procesos agrícolas. (“Beneficios de los Invernaderos Automatizados en la Agricultura”, 2024)
- **Incremento en la productividad:** La automatización permite que las tareas agrícolas se realicen más rápidamente que por métodos manuales, disminuyendo así el tiempo necesario para llevar a cabo dichas tareas. Esto resulta en una productividad elevada

y un aumento en los rendimientos. (“Automatización en la Industria Agrícola: Innovación Tecnológica para un futuro sostenible”, 2024)

- **Optimización de la toma de decisiones:** Los sistemas automatizados ofrecen a los agricultores acceso a información detallada y basada en datos, facilitando decisiones más acertadas. Esta mejora en la toma de decisiones promueve resultados óptimos. (Guillermo, 2023)
- **Fomento de la sostenibilidad:** El uso de tecnología automatizada en la agricultura puede disminuir el desperdicio, conservar recursos y reducir el empleo de sustancias químicas perjudiciales, llevando a prácticas más sostenibles y beneficiosas para el medio ambiente. (“Descubre cómo la agricultura sostenible con robots está transformando la industria”, 2024)

Para estructurar esta interfaz tenemos que comenzar con una plataforma que sea accesible y versátil, esta plataforma le podría permitir a los agricultores implementar sistemas que pueden monitorear y controlar automáticamente aspectos críticos del ambiente agrícola. Le permitirá tomar los factores más cruciales en el desarrollo de las plantaciones, como la humedad del suelo y ambiente, así como también la temperatura. Esta capacidad de recoger y responder a datos en tiempo real no solo mejora la eficiencia de las prácticas agrícolas, sino que también ayuda a maximizar los rendimientos de los cultivos mientras se minimiza el desperdicio de recursos, abriendo un nuevo horizonte en la gestión y sostenibilidad agrícola.

La plataforma principal para el desarrollo de nuestro sistema de monitoreo y toma de muestras es Arduino, debido a su capacidad para interactuar con el entorno agrícola de forma efectiva y económica. *Para asegurar una interacción intuitiva y amigable con los usuarios del sistema*, se utilizará un entorno de desarrollo accesible y fácil de entender, lo cual facilita la

creación de una interfaz que permita a los agricultores manejar y visualizar datos en tiempo real.

2.2. Aplicación de Arduino y Tecnologías IoT en la Agricultura

En Ecuador, los sistemas de monitoreo agrícola basados en IoT y Arduino han mostrado ser una solución viable para enfrentar problemas específicos como el manejo ineficiente de agua. Según el MAGAP (Ministerio de Agricultura y Ganadería), el país ha identificado un promedio de pérdidas del 20% en cultivos debido a la falta de supervisión en tiempo real de factores críticos como humedad del suelo y temperatura. La incorporación de sensores compatibles con Arduino, como los de bajo costo DHT22 o capacitivos para medición de humedad, permite ajustar de manera automática el riego, reduciendo el desperdicio hídrico en un 15% anual (MAGAP, 2020).

2.2.1. Comparativa de Arduino con otros microcontroladores

Al evaluar alternativas tecnológicas para la automatización agrícola en fincas pequeñas con recursos limitados, es fundamental comparar plataformas como Arduino, ESP32 y Raspberry Pi. Aunque cada una tiene sus ventajas, Arduino destaca por su simplicidad y accesibilidad, lo que lo hace especialmente adecuado para este contexto.

El ESP32 es un microcontrolador potente con conectividad Wi-Fi y Bluetooth integradas, adecuado para proyectos de Internet de las Cosas (IoT). Sin embargo, su complejidad y mayor consumo de energía pueden ser desventajas en entornos donde se busca una solución sencilla y eficiente. (Roch, 2024)

Por otro lado, Raspberry Pi es una computadora de placa única con capacidades avanzadas, incluyendo un sistema operativo completo. Aunque ofrece mayor potencia de procesamiento, su costo y complejidad de configuración pueden ser excesivos para pequeñas fincas que requieren soluciones más directas. (Delgado, 2024)

Arduino, en contraste, con su enfoque de bajo costo y accesibilidad, se convierte en una herramienta ideal para resolver problemas en sectores como la agricultura, donde la automatización puede optimizar el uso de recursos en entornos de bajos ingresos, como en las plantaciones ecuatorianas. Fue creado en Italia en el instituto de IVRAE en 2005, apareció por la necesidad de contar con un dispositivo para utilizar en las aulas, que fuera de bajo costo, funcionara bajo cualquier sistema operativo y que se utilizara para uso interno de la escuela. El Instituto de IVRAE decidió librear el proyecto de Arduino para que no quedara en el olvido y abrirlo a la comunidad con la posibilidad de que el proyecto fuera evolucionando proponiendo mejoras y sugerencias y mantenerlo vivo, Arduino poco a poco ha llegado a ser lo que es un proyecto de hardware y software libre del ámbito mundial. El software y hardware libre ofrecen a los usuarios cuatro libertades, libertad de uso, de estudio y modificación, de distribución y de redistribución de las versiones modificadas. (Carrillo, 2021)

Es una plataforma de electrónica de código abierto que ha revolucionado el acceso a la tecnología para entusiastas, educadores y profesionales en todo el mundo. Fue desarrollada con el objetivo de proporcionar una herramienta de bajo costo para la creación de proyectos interactivos que puedan interactuar con el entorno a través de sensores y actuadores. La plataforma incluye tanto una placa de circuito físico generalmente microcontroladores como un entorno de desarrollo de software (IDE) que facilita la escritura de código y la carga de

programas a la placa. Pero, ¿A qué nos referimos con microcontrolador y entorno de desarrollo de software?

2.2.2. Razones para la selección de Arduino

La selección de Arduino como plataforma principal para este proyecto se fundamenta en su bajo costo, facilidad de uso y amplia accesibilidad, características que la hacen especialmente adecuada para entornos agrícolas con recursos limitados. En comparación con otras opciones como el ESP32 o la Raspberry Pi, Arduino destaca por su simplicidad y eficiencia en aplicaciones específicas.

Además, la comunidad activa y global que respalda Arduino proporciona acceso a recursos educativos, bibliotecas de código y soluciones prácticas adaptadas a diversos proyectos agrícolas. Esto no solo facilita la implementación inicial, sino también la resolución de problemas y el mantenimiento del sistema a largo plazo. Su naturaleza de código abierto permite a los desarrolladores personalizar las soluciones según las necesidades específicas de las fincas ecuatorianas, promoviendo la sostenibilidad tecnológica y económica del proyecto. (*Tecnología Agrícola Inteligente con Arduino, s/f*)

2.2.3. Microcontroladores y Entornos de Desarrollo: Herramientas Esenciales en la Agricultura Automatizada

Un microcontrolador es un circuito integrado o chip que contiene todos los componentes electrónicos necesarios para realizar operaciones computacionales y de control. Estos incluyen un procesador central, memoria, y periféricos de entrada/salida en un solo paquete. Los microcontroladores son diseñados para realizar tareas específicas en sistemas

embebidos, como controlar dispositivos electrónicos domésticos, sistemas automotrices, aplicaciones industriales y, por supuesto, proyectos de automatización como los que se encuentran en la agricultura. *El uso de microcontroladores programables y el entorno de desarrollo Arduino IDE permiten implementar un sistema de monitoreo para factores clave como la humedad del suelo y la temperatura en las plantaciones. Esto es fundamental para mantener condiciones óptimas de crecimiento en los cultivos de Ecuador.* Debido a su capacidad para manejar múltiples tareas simultáneamente y su eficiencia energética, son ideales para aplicaciones donde el espacio, la potencia de procesamiento y el consumo de energía son limitados (Universidad Europea, 2024). Por otro lado, un Entorno de Desarrollo Integrado (IDE) es un software que proporciona herramientas comprehensivas para la programación en un único producto. Estos entornos incluyen un editor de texto para escribir código, herramientas de construcción automática, depuradores y, a menudo, simuladores para probar y diagnosticar errores en el software. Los IDE facilitan el desarrollo de aplicaciones permitiendo a los programadores escribir, modificar, compilar, desplegar y depurar su código en un entorno unificado. Esto es especialmente útil para los desarrolladores de microcontroladores, como aquellos que trabajan con Arduino, ya que simplifica y acelera el proceso de desarrollo de software, permitiendo una integración más fluida de hardware y software en proyectos complejos. (“IDE: ¿Qué es un Entorno de Desarrollo Integrado?”, 2023)

2.2.4. Arduino: Accesibilidad y Usos en Proyectos de Innovación

Las placas Arduino están equipadas con una serie de entradas y salidas digitales y analógicas que pueden conectarse a diversos componentes electrónicos como sensores de temperatura, motores, luces, y mucho más. Además, cuentan con un microcontrolador que se puede programar mediante el uso de Arduino IDE, utilizando una adaptación de C/C++ que hace

accesible la programación a un público amplio sin necesidad de una experiencia técnica avanzada. (“Arduino: Guía completa para principiantes y expertos”, 2023)

Desde su concepción, Arduino ha encontrado aplicaciones en un sinnúmero de proyectos desde arte y diseño hasta proyectos más complejos en robótica y, por supuesto, agricultura. Esto se debe en gran parte a su naturaleza de código abierto, que ha permitido a los usuarios compartir modificaciones y expansiones, enriqueciendo la comunidad global y extendiendo el alcance de su aplicación. Arduino se distingue por ser extremadamente accesible en términos de coste y facilidad de uso, lo que lo convierte en una herramienta ideal para prototipos rápidos y proyectos educativos. Además, su comunidad activa y los recursos abundantes en línea proporcionan un soporte considerable para nuevos usuarios, facilitando el aprendizaje y la experimentación continua. (¿Qué es Arduino?, 2024)

2.2.5. Modelos de Arduino y su Versatilidad para Proyectos Agrícolas

Arduino ofrece una variedad de modelos, cada uno diseñado para satisfacer diferentes necesidades de los usuarios, desde principiantes hasta profesionales en electrónica. Aquí te presento algunos de los modelos más populares de Arduino y sus características distintivas:

Arduino Uno: es un modelo básico y muy popular, equipado con el microcontrolador ATmega328P, que ofrece 14 pines digitales de entrada/salida, incluidos 6 pines con capacidad de salida PWM. También dispone de 6 pines de entrada analógica, una memoria Flash de 32 KB, SRAM de 2 KB y opera a una velocidad de reloj de 16 MHz. Este modelo es ideal para principiantes y proyectos de pequeña escala. (Guía De Modelos Arduino Y Sus Características | Arduino UNO | BricoGeek Lab, 2024)

Arduino WiFi: El ESP8266 es un módulo Wi-Fi de bajo costo que permite agregar conectividad inalámbrica a proyectos basados en Arduino. Este chip integra un microcontrolador de 32 bits y una pila de protocolos TCP/IP, facilitando la comunicación con redes Wi-Fi. Es compatible con el entorno de desarrollo de Arduino, lo que simplifica su programación y uso en diversos proyectos de Internet de las Cosas (IoT). Además, su tamaño compacto y bajo consumo energético lo hacen ideal para aplicaciones donde el espacio y la eficiencia son cruciales. (“ESP8266 Módulo WiFi, ¿Qué es y cómo configurar?”, 2019)

2.2.6. Ventajas de Arduino en Proyectos Agrícolas

Ventajas:

- **Bajo Costo:** Arduino es conocido por su accesibilidad económica, lo que permite a los agricultores y desarrolladores experimentar e implementar soluciones tecnológicas sin una gran inversión inicial.
- **Facilidad de Uso:** La plataforma Arduino es amigable para usuarios con diferentes niveles de habilidad técnica, gracias a su entorno de desarrollo integrado (IDE) sencillo y la amplia disponibilidad de librerías y ejemplos de código.
- **Comunidad y Recursos:** Existe una vasta comunidad de usuarios de Arduino que comparten conocimientos, código y soluciones. Además, hay numerosos tutoriales, foros y libros disponibles que facilitan el aprendizaje y la resolución de problemas.

Por ejemplo, en fincas ecuatorianas dedicadas al cultivo de cacao, la falta de monitoreo ha resultado en pérdidas significativas por enfermedades como la Moniliasis. Sistemas de IoT que incluyen sensores de temperatura y humedad, conectados a través de módulos

ESP8266, han permitido identificar condiciones propicias para estas enfermedades y ajustar las condiciones ambientales, evitando hasta un 30% de pérdidas anuales (MAGAP, 2020).

La accesibilidad económica de Arduino no solo permite reducir los costos iniciales de implementación, sino que también posibilita que agricultores en zonas rurales puedan adoptar esta tecnología sin necesidad de grandes inversiones. En este proyecto, Arduino es esencial para implementar sensores que monitorean continuamente factores como humedad y temperatura, elementos clave para el rendimiento de cultivos en Ecuador. Esta tecnología ayuda a enfrentar limitaciones financieras en las plantaciones ecuatorianas y promueve un uso más eficiente de recursos.

2.2.7. Integración de Arduino con Otras Tecnologías

La capacidad de Arduino para conectarse con sensores de humedad, temperatura y calidad del suelo permite implementar un sistema integral de monitoreo en tiempo real, lo cual es esencial para optimizar el uso de recursos en las plantaciones ecuatorianas.

- **Sensores y Actuadores:** Arduino puede conectarse a una variedad de sensores (temperatura, humedad, calidad del suelo) y actuadores (motores, válvulas, luces), lo cual es esencial para automatizar y monitorear procesos agrícolas.
- **Sistemas de Comunicación:** La capacidad de Arduino para integrarse con módulos de comunicación como Wi-Fi, Bluetooth y redes celulares permite la transmisión de datos desde y hacia el campo, facilitando un sistema agrícola más conectado.

2.2.8. Compatibilidad con Tecnologías de Software

- **.NET Core:** Aunque Arduino opera a nivel de hardware y .NET Core se maneja en el servidor o la nube, pueden interactuar a través de interfaces de programación de aplicaciones (APIs) o servicios web para procesar y analizar datos recogidos por dispositivos Arduino.
- **MongoDB:** Arduino y MongoDB pueden integrarse indirectamente para proyectos IoT mediante un servidor intermedio, dado que Arduino no puede conectarse directamente a bases de datos NoSQL como MongoDB. Al conectarse con una base de datos como MongoDB, Arduino permite almacenar y analizar grandes volúmenes de datos ambientales, facilitando la toma de decisiones en tiempo real y permitiendo a los agricultores ecuatorianos gestionar mejor sus recursos hídricos. La comunicación suele realizarse mediante el envío de datos desde el microcontrolador a un servidor o aplicación intermediaria que luego almacena la información en MongoDB.

2.2.9. Propósito de Arduino en el Proyecto

Arduino juega un papel fundamental en el proyecto al actuar como el núcleo operativo que integra y coordina tanto el movimiento autónomo del robot/carro como la recolección y transmisión de datos ambientales. Este sistema automatizado está diseñado para operar en plantaciones pequeñas, optimizando el monitoreo y manejo de variables críticas para el desarrollo de los cultivos.

2.2.10. Desplazamiento Autónomo del Robot

El *robot agrícola autónomo* fue diseñado utilizando Arduino como plataforma central, permitiendo su movimiento independiente a través de rutas predefinidas dentro de la plantación. El desplazamiento se logra mediante motores controlados por la placa Arduino, mientras que sensores adicionales garantizan su navegación eficiente y segura, evitando obstáculos y cubriendo áreas críticas de monitoreo.

La programación del sistema posibilita que el robot opere sin intervención manual, adaptándose a las características del terreno y asegurando un recorrido sistemático por las zonas de interés dentro de la finca.

2.2.11. Monitoreo y Recolección de Variables Ambientales

El robot está equipado con sensores estratégicos que capturan en tiempo real las variables ambientales esenciales para la gestión agrícola. Estos sensores, integrados a la placa Arduino, permiten la medición precisa de:

- Sensor DHT11: Mide la humedad relativa y la temperatura del aire, variables fundamentales para evaluar las condiciones climáticas y ajustar prácticas como el riego y la ventilación.
- Módulo LDR (Light Dependent Resistor): Detecta los niveles de claridad o luminosidad en el entorno, proporcionando datos cruciales para determinar si las plantas reciben la cantidad adecuada de luz solar.
- Sensor YL-69: Evalúa el contenido de humedad en el suelo, permitiendo optimizar los sistemas de riego y evitar tanto el exceso como la escasez de agua.

2.2.12. Flujo de Datos: Captura, Transmisión y Visualización

El proceso de manejo y análisis de los datos recolectados sigue un flujo estructurado que conecta el hardware de Arduino con el sistema de software en el backend y frontend. A continuación, se detalla este flujo:

- **Captura de Datos:** Los sensores (*DHT11*, *LDR* y *YL-69*) conectados a la placa Arduino realizan mediciones constantes mientras el robot se desplaza autónomamente por la finca.
- **Procesamiento y Transmisión:** Arduino procesa las mediciones y las envía al *backend* mediante un módulo de comunicación Wi-Fi, garantizando una transmisión eficiente y continua.
- **Almacenamiento en la Base de Datos:** El backend, implementado en *.NET Core*, recibe los datos, los procesa y los almacena en una base de datos *MongoDB*, permitiendo su conservación y consulta posterior.
- **Visualización de Datos:** Los datos almacenados son expuestos mediante una API al *frontend*, desarrollado en *React*. Los agricultores pueden acceder a una interfaz intuitiva donde se muestran las mediciones en tiempo real a través de gráficos y tablas, facilitando la toma de decisiones informadas.

Los datos recolectados por los sensores del robot agrícola (*DHT11*, *YL-69* y *LDR*), tales como la humedad del suelo, temperatura y luminosidad ambiental, se envían automáticamente al backend mediante un flujo controlado de datos. Posteriormente, estos datos se almacenan en *MongoDB*, una base de datos NoSQL flexible y eficiente que permite manejar grandes volúmenes de información no estructurada, como lecturas ambientales en tiempo real. La

estructura orientada a documentos de MongoDB facilita la adaptación a las variaciones de datos sin un esquema rígido, asegurando una gestión óptima de la información agrícola.

2.3. Gestión de Bases de Datos con MongoDB

MongoDB es un sistema de gestión de bases de datos NoSQL orientado a documentos, desarrollado por MongoDB Inc. A diferencia de los sistemas relacionales, MongoDB almacena datos en un formato similar a JSON, lo que permite una mayor flexibilidad y escalabilidad en aplicaciones que manejan grandes volúmenes de datos no estructurados. (Robledano, 2019)

Esta flexibilidad convierte a MongoDB en una opción popular en proyectos que requieren modelos de datos ágiles y la capacidad de manejar datos variados sin un esquema fijo. MongoDB Compass ofrece una interfaz gráfica avanzada que facilita la administración, consulta y visualización de datos, mientras que MongoDB Server proporciona el entorno central para el procesamiento de consultas, soporte para operaciones complejas y un sistema de replicación robusto. (*¿Qué es MongoDB y cómo funciona?*, 2022)

Para entender cómo interactuar con MongoDB en una aplicación, es necesario seguir varios pasos clave:

Configurar la Base de Datos: Antes de conectar la aplicación, se debe instalar MongoDB Server y configurar las bases de datos y colecciones. En MongoDB, los datos se agrupan en "colecciones" en lugar de "tablas", permitiendo un esquema flexible.

Instalar un Controlador de Base de Datos: La aplicación requerirá un controlador específico, como el "MongoDB Driver" para Node.js, Python o cualquier otro lenguaje que facilite la interacción con MongoDB, permitiendo a la aplicación enviar y recibir datos.

Cadena de Conexión: La cadena de conexión es esencial para acceder a MongoDB. Esta generalmente incluye el tipo de base de datos (mongodb://), la ubicación del servidor (dirección IP o URL), el nombre de la base de datos, y credenciales de usuario cuando se requiere autenticación.

Establecer la Conexión en el Código: Usando el lenguaje de programación elegido, se define un código que emplea la cadena de conexión para abrir la conexión con MongoDB. Herramientas como Mongoose en Node.js ayudan a manejar esquemas y realizar consultas de manera eficiente.

2.3.1. Estructura y Gestión de la Base de Datos

La base de datos del proyecto está diseñada para almacenar y organizar la información crítica de manera eficiente, utilizando *tres colecciones principales*:

Colección de Usuarios:

Almacena la información de los usuarios registrados en la plataforma, incluyendo:

- Datos personales (nombres, correo electrónico).
- Credenciales de acceso (correo electrónico, contraseña encriptada).
- Información adicional (estado de cuenta, foto de perfil).

Colección de Variables Ambientales:

Esta colección almacena las mediciones enviadas por el *robot autónomo* equipado con sensores, permitiendo un monitoreo detallado de las *variables críticas* del entorno de plantación.

Colección de Roles de Usuarios:

Permite definir y gestionar los permisos y accesos en la plataforma, asignando a cada usuario un rol específico:

- **Administrador:** Gestiona usuarios, aprueba registros y supervisa la plataforma.
- **Moderador:** Puede aceptar o rechazar solicitudes de registro.
- **Usuario:** Accede a la información de las variables ambientales y realiza consultas.

2.3.2. Justificación de la Selección de MongoDB como Base de Datos

La elección de *MongoDB* como base de datos NoSQL en este proyecto responde a la necesidad de manejar *datos no estructurados y de variabilidad constante* generados por los sensores del robot agrícola (humedad, temperatura y luminosidad). A diferencia de las bases de datos relacionales, que requieren un esquema rígido y predefinido, MongoDB permite almacenar datos en un formato *similar a JSON*, ofreciendo una *flexibilidad* considerable al manejar nuevas mediciones o cambios en el formato de los datos.

2.3.3. Comparación con Bases de Datos Relacionales

Flexibilidad en el esquema:

Las bases de datos relacionales como MySQL o PostgreSQL requieren un esquema estructurado y predefinido, lo cual limita la agilidad del sistema ante cambios inesperados en los datos. MongoDB, al ser una base de datos *orientada a documentos*, permite agregar campos nuevos sin necesidad de alterar toda la estructura.

Eficiencia en datos en tiempo real:

Las bases de datos relacionales como MySQL o PostgreSQL requieren un esquema estructurado y predefinido, lo cual limita la agilidad del sistema ante cambios inesperados en los datos. MongoDB, al ser una base de datos orientada a documentos, permite agregar campos nuevos sin necesidad de alterar toda la estructura.

Esta flexibilidad resulta esencial en este proyecto porque cada tipo de planta puede requerir el monitoreo de diferentes variables ambientales, dependiendo de sus necesidades específicas. Por ejemplo, en cultivos de cacao puede ser crítico medir la humedad del suelo y la temperatura ambiental para evitar enfermedades como la Moniliasis, mientras que en cultivos de tomate es fundamental monitorear la luminosidad y la humedad relativa del ambiente para optimizar la fotosíntesis y el desarrollo del fruto.

MongoDB facilita esta adaptación, ya que su estructura dinámica permite agregar campos específicos para cada planta sin necesidad de realizar modificaciones globales en la base de datos.

Escalabilidad horizontal:

MongoDB soporta escalabilidad horizontal, lo que significa que el sistema puede crecer distribuyendo los datos en varios servidores. Esta característica es ideal para aplicaciones futuras, donde se espera manejar *más datos o conectar más sensores*.

Integración con .NET Core:

MongoDB se integra fácilmente con *.NET Core* mediante controladores oficiales como el *MongoDB C# Driver*, permitiendo manejar los datos de manera eficiente y exponerlos a través de APIs RESTful. Esta integración facilita el procesamiento de datos y su posterior envío al frontend.

2.3.4. Uso de la Base de Datos en el Proyecto

La base de datos es utilizada principalmente para:

1. Autenticación y Gestión de Usuarios:

- Validación de credenciales durante el inicio de sesión.
- Registro de nuevos usuarios y aprobación por parte de administradores.
- Actualización de datos personales y contraseñas.

2. Almacenamiento de Mediciones en Tiempo Real:

Los datos enviados por el robot autónomo son registrados en la colección de variables ambientales, permitiendo su visualización en gráficos a través del frontend.

3. Gestión de Roles:

Se utilizan para definir los permisos y accesos en la plataforma, asegurando una estructura ordenada y segura.

2.3.5. Integración de MongoDB con .NET Core

La integración de MongoDB con *.NET Core* se realiza mediante el uso del MongoDB Driver para *.NET*, que permite la comunicación entre la base de datos y la aplicación backend. Este controlador facilita la implementación de operaciones como la inserción, consulta,

actualización y eliminación de documentos en las colecciones de MongoDB, optimizando el manejo de los datos recolectados por el sistema.

En el proyecto, el backend en .NET Core actúa como intermediario entre la base de datos y el front-end desarrollado en React, exponiendo APIs que permiten:

1. Almacenar las mediciones obtenidas por el robot autónomo en la colección de variables ambientales.
2. Gestionar los usuarios y sus roles, asegurando la autenticación y autorización mediante la validación de credenciales almacenadas en la base de datos.
3. Consultar y devolver datos en tiempo real al front-end, permitiendo la visualización de variables ambientales en gráficos.

Gracias a la compatibilidad de MongoDB con JSON, la integración con .NET Core es sencilla y eficiente, ya que los datos enviados desde las APIs se estructuran en formato JSON, lo que facilita su almacenamiento y recuperación. Además, esta combinación proporciona un rendimiento óptimo para manejar grandes volúmenes de datos no estructurados, manteniendo la escalabilidad y flexibilidad del sistema.

2.4. Introducción a .NET Core y su Evolución hasta la Versión 7

Es una plataforma de desarrollo de software de código abierto mantenida por Microsoft y la comunidad. Originalmente lanzada para permitir la creación de aplicaciones modernas, multiplataforma y de alto rendimiento, .NET Core ha evolucionado significativamente desde sus inicios. Comenzando como un fork del .NET Framework, fue diseñado para ser más modular y adaptable a diversas plataformas como Windows, Linux y macOS. A lo largo de

sus versiones, ha mejorado en eficiencia, seguridad y capacidad de integración con otros sistemas y tecnologías. (Llerena, 2022)

La primera versión de .NET Core fue lanzada en junio de 2016, enfocándose en la creación de aplicaciones web y consola. En noviembre del mismo año vio el lanzamiento de .NET Core 1.1, que agregó soporte para más APIs y mejoras de rendimiento. Agosto de 2017 marcó un gran salto con .NET Core 2.0, ampliando considerablemente la API y mejorando la compatibilidad con el .NET Framework, lo que facilitó la migración de aplicaciones existentes. (“Announcing .NET 7.0.”, 2022)

Pero ¿Qué es un Framework?, es un marco de trabajo estructurado que sirve como fundamento para la organización y el desarrollo de software. Un framework proporciona una arquitectura estándar con herramientas y componentes predefinidos que los desarrolladores pueden utilizar para construir aplicaciones de manera más eficiente y efectiva. Aquí te detallo más a fondo qué es un framework y cómo funciona:

2.4.1. Funciones Principales de un Framework

- **Abstracción y Reutilización de Código:** ofrecen una biblioteca de código que ayuda a abstraer detalles complicados o repetitivos, permitiendo a los desarrolladores centrarse en la lógica específica de su aplicación sin tener que reinventar la rueda para tareas comunes. (Lucena, 2023)
- **Estructura Predefinida:** proporcionan una estructura o esqueleto predefinido para aplicaciones, lo que ayuda a mantener un diseño coherente y organizado. Esto incluye patrones de diseño, soporte para MVC (Modelo-Vista-Controlador) y otros patrones

arquitectónicos que facilitan el mantenimiento y la escalabilidad del software. (“Patrón MVC y Arquitecturas de Frameworks - Explicación Clara y Precisa”, 2024)

- **Facilita el Desarrollo:** un framework puede reducir significativamente el tiempo de desarrollo al ofrecer funcionalidades listas para usar, como manejo de sesiones, acceso a bases de datos, y más. Esto permite a los desarrolladores implementar funcionalidades complejas con menos esfuerzo y menor probabilidad de errores. (Arturo, 2024)
- **Soporte para Mejores Prácticas:** estimula a los desarrolladores a seguir las mejores prácticas de programación y arquitectura, incluyendo la seguridad, el manejo de datos y la integración de sistemas. Un buen framework estará constantemente actualizado con las últimas prácticas y parches de seguridad. (“Desarrollo Seguro: Frameworks Open Source que Refuerzan la Seguridad Web”, 2024)

Continuando con la evolución de .NET Core, septiembre de 2019 fue testigo del lanzamiento de .NET Core 3.0, que reintrodujo el soporte para aplicaciones de escritorio (Windows Forms y WPF) y mejoró el soporte para aplicaciones de IoT. NET 5, lanzado en noviembre de 2020, marcó el inicio de la unificación de las plataformas .NET, integrando .NET Core, .NET Framework y Mono en un único framework que continúa bajo el nombre de ".NET". NET 6, lanzado en noviembre de 2021, continuó esta unificación y trajo mejoras significativas en rendimiento y desarrollo multiplataforma. (“What’s new in .NET 6”, 2023)

Para finalizar, año 2022 es lanzado .NET 7 la última versión hasta la fecha, sigue ampliando las capacidades del framework con mejoras en rendimiento, herramientas de desarrollo y compatibilidad para facilitar aún más el desarrollo de aplicaciones robustas y escalables en diversas plataformas. (Douglas, 2022)

El proyecto utiliza .NET Core y Visual Studio para desarrollar una plataforma web que facilita el monitoreo en tiempo real de datos ambientales recolectados de las plantaciones. .NET Core es una herramienta robusta y multiplataforma que asegura que la plataforma web pueda funcionar en diversos dispositivos, permitiendo a los agricultores acceder fácilmente a los datos desde cualquier ubicación con internet. Esto resulta fundamental en áreas rurales donde el acceso al sistema puede ser limitado, optimizando así las decisiones sobre el uso de recursos hídricos y energéticos.

2.4.2. Razones para la Elección de .NET Core sobre otros Frameworks

La elección de .NET Core como tecnología principal para el desarrollo del backend responde a sus ventajas competitivas frente a otros frameworks como *Node.js*, *Django* y *Spring Boot*, considerando aspectos como rendimiento, escalabilidad, seguridad y compatibilidad multiplataforma.

Alto Rendimiento y Eficiencia

.NET Core sobresale en rendimiento debido a su arquitectura optimizada y al uso de compilación just-in-time (JIT). Comparado con *Node.js*, que se basa en un modelo de ejecución single-threaded, .NET Core utiliza un modelo multithreading más eficiente, lo cual reduce el tiempo de respuesta en aplicaciones complejas y con múltiples solicitudes simultáneas (Das, 2024). Además, su integración con bases de datos como MongoDB permite una interacción rápida y robusta para gestionar grandes volúmenes de datos.

Multiplataforma y Escalabilidad

A diferencia del antiguo .NET Framework, limitado a Windows, .NET Core es multiplataforma y funciona de forma nativa en *Windows, Linux y macOS*. Esta versatilidad facilita el despliegue en entornos en la nube, como *DigitalOcean*, y su arquitectura modular permite un desarrollo ligero y adaptable a sistemas en crecimiento (Karakhainko, 2023). Este factor es crucial en proyectos donde se proyecta una expansión progresiva.

Seguridad Nativa y Soporte Continuo

.NET Core proporciona características avanzadas de seguridad, como autenticación y protección contra ataques comunes (CSRF, SQL Injection), integradas directamente en el framework. En comparación con Node.js, que requiere dependencias externas para garantizar la seguridad, .NET Core incluye actualizaciones constantes y parches de Microsoft, lo que lo convierte en una opción más robusta y confiable (Daniel, 2022).

Integración con Tecnologías Modernas

.NET Core facilita la implementación de APIs y su integración con frontend basados en tecnologías como *React*. La compatibilidad con *JSON, RESTful APIs* y su eficiente comunicación con bases de datos como MongoDB asegura una estructura sólida para la gestión de datos en tiempo real. Esta característica optimiza el flujo de datos desde el backend hacia la interfaz de usuario, algo esencial en nuestro sistema de monitoreo agrícola (Haider, 2023).

Optimización de Recursos y Soporte Empresarial

En aplicaciones que demandan estabilidad y escalabilidad, .NET Core resulta más eficiente en el consumo de recursos del servidor. Además, su soporte empresarial y documentación oficial extensa aseguran un desarrollo más ágil y con menor riesgo de errores, a diferencia de frameworks como Django, donde la escalabilidad en aplicaciones empresariales puede presentar mayores desafíos ("Spring Boot vs. Node.js vs. .NET Core vs. Django", 2024).

2.4.3. .NET Core y su Relación con el Modelo MVC

Para organizar el desarrollo de esta plataforma, se implementa el modelo MVC (Modelo-Vista-Controlador), una arquitectura que facilita la separación de la lógica de datos, la presentación visual y la interacción del usuario. En este modelo, .NET Core gestiona el 'Modelo' y el 'Controlador', mientras que *React* se utiliza para desarrollar la capa de 'Vista'. Esto permite una interfaz de usuario dinámica y altamente interactiva, optimizada para brindar a los agricultores un acceso intuitivo y eficiente a los datos en tiempo real desde diversos dispositivos.

2.5. Modelo de Proyecto MVC con .NET Core 7

Significa Modelo-Vista-Controlador, es un patrón de diseño ampliamente utilizado para desarrollar interfaces de usuario que divide una aplicación en tres componentes interconectados. Este modelo se implementa para separar la lógica de negocio, la lógica de presentación y la interacción del usuario en aplicaciones, lo que facilita la gestión de cada aspecto por separado. .NET Core 7, la última versión del framework de desarrollo de Microsoft, ofrece soporte robusto para la creación de aplicaciones web utilizando el patrón

MVC, permitiendo a los desarrolladores construir aplicaciones escalables y mantenibles con eficiencia. (Hernandez, 2021)

2.5.1. Flexibilidad del MVC en .NET Core 7

El patrón MVC en .NET Core 7 ofrece una arquitectura modular que permite a los desarrolladores reemplazar o personalizar cada componente sin afectar a los demás. Esta modularidad es ideal para equipos de desarrollo grandes, donde diferentes grupos pueden trabajar en componentes específicos de manera simultánea. Además, cada componente puede ser desarrollado y probado de forma independiente, lo que aumenta la eficiencia del desarrollo y facilita las pruebas unitarias y de integración. (“Ventajas y desventajas de MVC en ASP.NET Core”, 2023)

La flexibilidad del MVC en .NET Core 7 también se manifiesta en su capacidad para integrarse con diversas tecnologías de front-end, como React, Angular o Vue.js. Esta integración permite a los desarrolladores elegir las herramientas que mejor se adapten a las necesidades específicas de su proyecto, proporcionando una experiencia de desarrollo más personalizada y eficiente. (“Guía Completa de .NET Core MVC en Español”, 2024)

2.5.2. Características del MVC en .NET Core 7

1. Separación de Responsabilidades:

- **Modelo:** Representa la lógica de datos y de negocio; es decir, define los datos que la aplicación maneja.

- **Vista:** Encargada de presentar los datos, es decir, todo lo que el usuario ve en pantalla.
- **Controlador:** Actúa como intermediario entre el modelo y la vista, manejando la lógica de entrada del usuario, interactuando con los modelos para procesar datos y enviarlos a la vista. (Hernandez, 2021)

2. Integración con Tecnologías de Front-End:

- Las vistas en MVC pueden utilizar tecnologías de front-end como HTML, CSS y JavaScript, además de frameworks como Angular o React para crear interfaces de usuario ricas y dinámicas.

3. Routing Potente:

- .NET Core ofrece un sistema de routing avanzado que ayuda a definir URLs más intuitivas que mejoran la accesibilidad y la indexación por motores de búsqueda.

4. Soporte para APIs:

- MVC en .NET Core no solo es efectivo para aplicaciones web, sino que también es excelente para construir APIs web, lo que permite una fácil integración con otros sistemas y aplicaciones. (“Creación de API web con Asp.net Core”, 2024)

2.5.3. Limitaciones del MVC en .NET Core 7

- **Complejidad:** Para proyectos pequeños, el modelo MVC puede resultar excesivamente complejo, lo que podría ralentizar el proceso de desarrollo debido a la sobrecarga de configuración inicial y a la curva de aprendizaje.
- **Uso de Recursos:** Dado que se mantiene la separación de componentes y se gestionan múltiples interacciones entre ellos, MVC puede ser más demandante en términos de uso de recursos comparado con otros patrones más simples.
- **Dependencia del Framework:** Aunque .NET Core es flexible, el diseño de aplicaciones en MVC dentro de este entorno sigue siendo algo dependiente de las convenciones y limitaciones del framework, lo que podría limitar la personalización en ciertos aspectos.

2.5.4. Justificación del Uso del Modelo MVC con .NET Core

La decisión de implementar el patrón MVC (Modelo-Vista-Controlador) con .NET Core 7 en este proyecto se basa en las ventajas que ofrece este modelo en términos de organización del código, escalabilidad y eficiencia en el desarrollo de aplicaciones web modernas.

1. **Separación de responsabilidades:** El patrón MVC permite dividir la aplicación en tres componentes fundamentales (Modelo, Vista y Controlador), facilitando la modularidad del sistema. Esta división mejora la mantenibilidad y legibilidad del código, permitiendo que el equipo de desarrollo pueda trabajar simultáneamente en diferentes partes del proyecto sin interferencias.
2. **Escalabilidad y eficiencia:** Al utilizar .NET Core, se obtiene una plataforma que permite desarrollar aplicaciones altamente escalables, con un rendimiento optimizado y soporte multiplataforma (Windows, Linux y macOS). Esta característica es crucial

para un sistema que recolecta y procesa datos en tiempo real desde las plantaciones, ya que puede crecer conforme aumenten los volúmenes de datos o usuarios.

3. **Exposición de APIs eficientes:** La arquitectura MVC facilita la creación de APIs RESTful que son consumidas por el front-end en React. .NET Core, al ser un framework moderno y optimizado, permite gestionar estas APIs de manera eficiente y asegurar una comunicación fluida entre el backend y la base de datos MongoDB.
4. **Compatibilidad con tecnologías modernas:** .NET Core ofrece integración nativa con frameworks de front-end como React, Angular o Vue.js, lo que simplifica la construcción de una interfaz de usuario dinámica e interactiva. En este proyecto, React se utiliza en la capa de "Vista" para mostrar en tiempo real las mediciones ambientales capturadas por los sensores.
5. **Soporte y comunidad:** .NET Core es una plataforma de código abierto, respaldada por Microsoft y una amplia comunidad de desarrolladores, lo que garantiza actualizaciones constantes, soporte y recursos disponibles en línea. Esta solidez técnica hace que sea una opción confiable para proyectos que buscan seguridad y robustez a largo plazo.

A pesar de las limitaciones del modelo MVC en .NET Core, como su complejidad inicial, esta arquitectura ofrece una base robusta para el desarrollo de aplicaciones escalables, modulares y eficientes. La combinación de MVC con .NET Core en este proyecto se ha seleccionado por su capacidad para organizar y separar responsabilidades, facilitando la integración con tecnologías modernas como React y MongoDB. Esta arquitectura ha permitido gestionar la lógica de negocio, la comunicación con la base de datos y la creación de APIs RESTful, necesarias para el funcionamiento del sistema agrícola automatizado.

A continuación, se presenta el desarrollo del backend con .NET Core, detallando las funcionalidades implementadas, la estructura del sistema y su integración con las herramientas y servicios clave utilizados en el proyecto.

2.6. Desarrollo del Backend con .NET Core

2.6.1. Estructura del Backend y Funcionalidades Principales

El backend desarrollado con *.NET Core* cumple un rol central en la administración y procesamiento de datos del sistema. A través de una arquitectura basada en *APIs RESTful*, el backend facilita la comunicación entre el sistema de monitoreo con Arduino y el frontend desarrollado en React, permitiendo que los datos capturados por los sensores se almacenen y gestionen eficientemente.

Entre las principales funcionalidades implementadas se encuentran:

Gestión de Usuarios

El backend maneja los procesos de autenticación y autorización utilizando un modelo seguro que permite verificar las credenciales en la base de datos.

- **Login:** Verifica si las credenciales del usuario existen en la base de datos y permite el acceso.
- **Registro:** El formulario de registro envía la solicitud al backend, donde los administradores y moderadores pueden aceptar o rechazar el registro.
- **Recuperación de Contraseña:** El backend utiliza Mailgun para enviar un enlace de recuperación de contraseña a través del correo electrónico.

Manejo de Roles y Permisos

Los administradores tienen acceso a funcionalidades avanzadas como la gestión de usuarios registrados:

- Cambiar roles (usuario o moderador).
- Eliminar usuarios del sistema.

Monitoreo de Variables Ambientales

Las APIs permiten recuperar, almacenar y procesar datos ambientales capturados por los sensores conectados a Arduino.

- Datos del Sensor: Se implementa una API que envía los datos de *temperatura*, humedad ambiental, claridad y humedad *del suelo* a la base de datos.
- Visualización: El backend ofrece una API que permite al frontend recuperar estos datos y mostrarlos en gráficos dinámicos en el dashboard.

2.6.2. Procesamiento y Flujo de Datos

El flujo de datos entre el backend y el resto del sistema se estructura de la siguiente manera:

1. Captura de Datos desde Arduino

Los sensores conectados al robot autónomo recopilan las variables ambientales, como temperatura (DHT11), humedad del suelo (YL-69) y claridad (LDR). Los datos se envían al backend a través de peticiones HTTP.

2. Almacenamiento en la Base de Datos

El backend recibe y procesa los datos, almacenándolos en una base de datos *NoSQL* para su organización y recuperación. Esto facilita la gestión eficiente de grandes volúmenes de información.

3. Exposición de APIs Restful

El backend ofrece endpoints para que el frontend consulte y visualice los datos en tiempo real. Por ejemplo:

- *API de datos ambientales*: Permite al dashboard mostrar gráficos de barras con información actualizada.
- *API de usuario*: Devuelve los datos del perfil del usuario autenticado y permite actualizaciones como la foto de perfil o contraseña.

4. Interacción con el Frontend

A través de peticiones HTTP, el frontend en React consume las APIs del backend para mostrar datos y permitir la interacción del usuario con la plataforma.

2.6.3. Herramientas y Tecnologías Implementadas

El desarrollo del backend se soporta en tecnologías robustas y eficientes que garantizan un rendimiento óptimo:

- **.NET Core**: Framework principal para la construcción de APIs y la lógica del sistema.
- **Mailgun**: Servicio utilizado para el envío de correos electrónicos, como en el proceso de recuperación de contraseñas.
- **Base de Datos**: Se implementa una base de datos NoSQL que facilita el almacenamiento de datos no estructurados provenientes de los sensores.

- **Autenticación Segura:** Se utilizan métodos de seguridad para proteger el acceso y la información de los usuarios.

2.6.4. Comunicación entre React y .NET Core: Desarrollo e Integración del Backend

La implementación del frontend con React y del backend con .NET Core se basa en una arquitectura que permite la comunicación fluida entre ambos componentes mediante **APIs RESTful**. React consume estas APIs para obtener y mostrar los datos en tiempo real, garantizando una interfaz interactiva y eficiente. Por su parte, .NET Core actúa como el intermediario responsable de gestionar la lógica de negocio, acceder a la base de datos y exponer los servicios necesarios. Esta integración permite aprovechar las ventajas de ambos entornos: React ofrece una experiencia de usuario dinámica con un alto rendimiento, mientras que .NET Core proporciona una base robusta, escalable y segura para el manejo de los procesos del sistema. En conjunto, ambos frameworks aseguran una solución tecnológica eficiente y adaptable a las necesidades del monitoreo agrícola automatizado.

2.7. Uso de React como Capa de Vista en el Modelo MVC

React es una biblioteca de JavaScript desarrollada por Facebook que se centra en la construcción de interfaces de usuario interactivas y eficientes. En el contexto del patrón Modelo-Vista-Controlador (MVC), React se integra como la capa de "Vista", encargada de la presentación de datos y la interacción con el usuario. Esta integración permite una separación clara entre la lógica de negocio y la interfaz de usuario, facilitando el mantenimiento y la escalabilidad de las aplicaciones. ("Comprensión de la arquitectura MVC con React", 2020)

2.7.1. Flujo de Funcionamiento del Frontend

El frontend en React actúa como la interfaz principal del usuario, proporcionando una visualización clara y dinámica de los datos recopilados por el sistema de monitoreo agrícola.

El flujo de funcionamiento es el siguiente:

Solicitud de Datos al Backend:

Al ingresar al sistema, React realiza una solicitud HTTP GET a la API correspondiente en el backend.

Procesamiento de Datos:

- Los datos obtenidos del backend se almacenan en estados locales o globales usando herramientas como useState o Context API.
- Se utilizan bibliotecas adicionales como Chart.js o Recharts para transformar estos datos en gráficos y tablas dinámicas, lo que facilita su interpretación.

Actualización de la Interfaz:

- React utiliza su capacidad de renderizado eficiente para actualizar solo los componentes afectados por cambios.
- Ejemplo: Si se recibe una nueva lectura del sensor de humedad del suelo, solo se actualiza el componente correspondiente al gráfico, sin afectar otros elementos.

Interacción con el Usuario:

El usuario puede realizar acciones específicas, como:

- **Ver gráficos:** Visualización de las mediciones en tiempo real.
- **Actualizar Perfil:** Enviar datos al backend mediante una solicitud HTTP POST para actualizar información personal o cambiar contraseñas.
- **Gestionar Solicitudes:** Los administradores y moderadores interactúan con una interfaz específica donde pueden aprobar, rechazar o gestionar usuarios.

Visualización de Resultados:

Los resultados de las mediciones se muestran de manera intuitiva en el *dashboard principal*, donde se presentan gráficos y valores clave como:

- Humedad del suelo: Sensor YL-69.
- Temperatura y humedad ambiental: Sensor DHT11.
- Nivel de claridad o luz: Módulo LDR.

2.7.2. Implementación en el Proyecto de Agricultura Automatizada

En este proyecto, React se emplea para mostrar datos en tiempo real de los sensores instalados en las plantaciones. Los agricultores pueden acceder a gráficos y visualizaciones que muestran el estado actual de los parámetros ambientales, lo cual facilita la toma de decisiones y el ajuste inmediato de condiciones si es necesario. Esto resulta crucial en áreas rurales, donde la eficiencia en el manejo de recursos es fundamental.

2.7.3. Razones para la Elección de React sobre otros Frameworks

La elección de *React* como la tecnología principal para la capa de vista en el desarrollo del sistema se fundamenta en sus ventajas competitivas respecto a otros frameworks como

Angular, Vue.js y Svelte, considerando factores clave como rendimiento, flexibilidad y aprendizaje.

Rendimiento Optimizado con el DOM Virtual

React destaca por su capacidad de manejar actualizaciones de forma eficiente gracias al uso del *DOM virtual*. A diferencia de frameworks como Angular, que realizan actualizaciones completas del DOM real, React optimiza los cambios al actualizar solo los componentes afectados. Esto resulta esencial para nuestro proyecto, donde los datos de los sensores se actualizan constantemente en tiempo real, asegurando una experiencia fluida y sin retrasos en la interfaz (Poojary, 2024).

Flexibilidad y Componentización

React adopta una estructura basada en componentes, lo que permite reutilizar piezas independientes de código. A diferencia de Angular, que requiere una estructura más estricta y completa para cada módulo, React permite construir interfaces modulares que se adaptan a las necesidades del sistema de monitoreo. Por ejemplo, los gráficos de datos ambientales y las tarjetas de información pueden ser componentes reutilizables y actualizados sin afectar otros elementos (“Why use React instead of other frameworks like Angular?”, 2024).

Curva de Aprendizaje Reducida y Amplia Comunidad

React presenta una curva de aprendizaje más suave en comparación con frameworks como Vue.js o Angular, que tienen una mayor complejidad inicial debido a sus características más prescriptivas. Además, la comunidad activa de React y la disponibilidad de librerías y recursos

en línea facilitan la resolución de problemas y el desarrollo ágil de aplicaciones. Esto ha sido crucial para el éxito del proyecto al permitir una rápida implementación de funcionalidades como el consumo de APIs y la creación de interfaces dinámicas (Jey, 2021).

Compatibilidad con APIs REST y Ecosistema Extensible

React se integra de manera eficiente con APIs RESTful, como las desarrolladas en .NET Core para este proyecto. Su ecosistema extenso permite el uso de herramientas como *Redux* para la gestión del estado global y bibliotecas como *Chart.js* para la representación gráfica de datos, algo fundamental para la visualización de mediciones ambientales en tiempo real (“A battle of titans: React JS vs other front-end frameworks”, 2023).

2.7.4. Gestión de Versiones en el Desarrollo Backend y Frontend

El desarrollo integral del sistema, que incluye tanto el backend con *.NET Core* como el frontend con *React*, se gestiona a través de *Git* y *GitHub*. Git proporciona un sistema de control de versiones que facilita el seguimiento de los cambios realizados en el código, permitiendo a los desarrolladores trabajar de manera colaborativa y organizada. Por su parte, GitHub funciona como plataforma centralizada para almacenar y compartir el repositorio del proyecto, facilitando la colaboración entre equipos al ofrecer herramientas como pull requests, ramas independientes y seguimiento de problemas. Esta integración permite mantener la coherencia y estabilidad del código durante las fases de desarrollo, pruebas y despliegue, asegurando un flujo de trabajo eficiente en todo el proyecto.

2.8. Concepto de Git y GitHub

Git es un sistema de control de versiones que maneja y almacena revisiones de proyectos, permitiendo a los desarrolladores rastrear y comparar cambios, revertir a versiones anteriores si es necesario, y gestionar distintas ramas de desarrollo. GitHub, por otro lado, es un servicio de hosting que ofrece Git repository hosting, lo cual facilita más aún la colaboración entre desarrolladores no solo para compartir y gestionar el código fuente sino también para documentar problemas y compartir ideas. (GitHub Inc, 2024)

2.8.1. Funcionalidades Clave de GitHub

- 1. Control de Versiones:** GitHub permite a los desarrolladores hacer "commits" de sus cambios en el código, lo que significa registrar una instantánea del estado actual del proyecto en Git. Cada commit tiene un identificador único y mantiene un registro histórico de quién hizo qué cambios y cuándo.
- 2. Ramas y Fusiones (Branching and Merging):** Las ramas permiten a los desarrolladores trabajar en características o correcciones sin afectar el estado actual del proyecto en la rama principal (generalmente llamada "master" o "main"). Una vez que una característica está completa y probada, puede ser fusionada de nuevo en la rama principal.
- 3. Pull Requests:** Los pull requests son una funcionalidad central en GitHub que facilita la discusión de cambios antes de que se integren en la rama principal. Esto no solo permite una revisión de código detallada sino también garantiza que todos los cambios sean apropiados y estén libres de errores antes de su integración.

4. **Issues:** GitHub permite a los usuarios crear "issues" o problemas, que pueden ser bugs, solicitudes de características nuevas, o generalmente cualquier tarea o mejora. Los issues pueden ser asignados a miembros específicos del equipo, etiquetados para clasificación y rastreados hasta su resolución.

5. **GitHub Actions:** Esta es una herramienta de CI/CD (Integración Continua y Despliegue Continuo) que automatiza flujos de trabajo, desde pruebas y construcción hasta despliegue de aplicaciones. Permite a los desarrolladores configurar workflows directamente desde su GitHub repositorio.

2.8.2. Beneficios de Usar GitHub para el Versionamiento de Proyectos

- **Colaboración Mejorada:** GitHub centraliza el control de versiones y la colaboración, facilitando a los equipos trabajar juntos más eficientemente.

- **Historial de Proyecto Claro:** Cada cambio en el código es rastreado con detalles sobre el autor y el momento del cambio, proporcionando una transparencia completa.

- **Desarrollo Paralelo:** Las ramas permiten a los equipos trabajar en múltiples características simultáneamente sin interferir unos con otros.

- **Revisión y Calidad de Código:** Las herramientas como pull requests e issues promueven un estándar alto de calidad de código y permiten revisiones de código sistemáticas. (Camacho, D., 2021)

2.8.3. Automatización del Despliegue con GitHub y la Infraestructura

La integración de *Git* y *GitHub* con la infraestructura del proyecto optimiza el proceso de despliegue continuo (CI/CD) y facilita la entrega de actualizaciones. A través de herramientas como *Docker* y servicios de nube como *DigitalOcean*, los repositorios alojados en GitHub se conectan a sistemas automatizados para la construcción y despliegue del código. Esta sincronización permite que cada cambio aprobado en el repositorio principal se integre de manera automática en el entorno de producción, asegurando que las actualizaciones sean rápidas y estables. Además, al utilizar GitHub como repositorio central, la infraestructura puede beneficiarse de flujos de trabajo automatizados, garantizando que el backend, el frontend y los servicios asociados se mantengan sincronizados y operativos en todo momento.

2.9. Infraestructura de Despliegue y Servicios Asociados

Para garantizar la implementación eficiente del sistema de monitoreo automatizado y su accesibilidad en tiempo real, se ha seleccionado una serie de herramientas y servicios que optimizan el despliegue, la escalabilidad y la seguridad de la plataforma. A continuación, se detalla la infraestructura utilizada y las razones detrás de su elección.

2.9.1. Digital Ocean: Servidor en la Nube

Digital Ocean fue seleccionado como el proveedor de infraestructura en la nube por su equilibrio entre costo, rendimiento y facilidad de uso. Ofrece soluciones de *droplets*, servidores virtuales escalables que permiten desplegar aplicaciones en la nube de manera ágil y sencilla. Comparado con competidores como AWS, Digital Ocean destaca en:

- **Simplicidad:** Una interfaz más intuitiva y enfocada en desarrolladores pequeños y medianos.
- **Costo-Eficiencia:** Planes más accesibles sin sacrificar el rendimiento.
- **Rapidez en el Despliegue:** Configuración de servidores en minutos.

Estas características permiten satisfacer las necesidades de proyectos que requieren escalabilidad sin incurrir en altos costos. (Manoor, 2022; Clemente, 2020).

2.9.2. Docker: Contenedorización

Docker se utiliza para contenedorización de la aplicación, permitiendo empaquetar el backend desarrollado en *.NET Core* y el frontend en *React* junto con todas sus dependencias. Esto garantiza que la aplicación funcione de manera uniforme en cualquier entorno. Las razones de su elección incluyen:

- **Portabilidad:** Permite mover aplicaciones entre entornos sin problemas.
- **Eficiencia de Recursos:** A diferencia de las máquinas virtuales, Docker consume menos recursos.
- **Despliegue Rápido:** Facilita la automatización del despliegue en Digital Ocean.

Esto permite una gestión eficiente de los recursos y acelera los ciclos de desarrollo y pruebas del sistema. (Kluften, 2024).

2.9.3. *Cloudflare: CDN y Seguridad*

Cloudflare se utiliza como un *Content Delivery Network (CDN)* para acelerar la entrega de contenido estático y proteger la aplicación mediante su firewall avanzado. La elección de Cloudflare frente a otras opciones como AWS CloudFront se justifica por:

- **Optimización del Rendimiento:** Minimiza la latencia al distribuir contenido a través de una red global.
- **Seguridad Integrada:** Protección contra ataques DDoS y amenazas comunes.
- **Facilidad de Implementación:** Configuración rápida y amigable con otros servicios.

Cloudflare no solo mejora el tiempo de respuesta del sistema, sino que protege la plataforma frente a posibles vulnerabilidades de seguridad. (Cloudflare CDN vs. AWS CloudFront CDN, 2024; Cloudflare CDN VS Azure CDN, s.f.).

2.9.4. *Nginx: Servidor Web*

Nginx se utiliza como servidor web para gestionar las solicitudes entre el frontend, el backend y la base de datos. Su rendimiento superior en comparación con Apache se debe a:

- **Eficiencia:** Manejo optimizado de múltiples conexiones concurrentes.
- **Bajo Consumo de Recursos:** Requiere menos memoria y CPU.
- **Escalabilidad:** Facilita el balanceo de carga y optimización del tráfico.

Nginx garantiza la entrega rápida de los datos, especialmente en proyectos donde se manejan grandes volúmenes de tráfico. (Javaid, 2024; Wiesman, 2024).

2.9.5. Mailgun: Gestión de Correos Electrónicos

Mailgun es un servicio especializado en el envío de correos electrónicos transaccionales, utilizado en funcionalidades como *recuperación de contraseñas* y *notificaciones automáticas* a los usuarios registrados. La elección de Mailgun frente a otras alternativas como SendGrid se debe a:

- **Fiabilidad:** Alta tasa de entrega de correos electrónicos.
- **Simplicidad en la Integración:** API fácil de implementar con .NET Core.
- **Escalabilidad:** Capacidad para manejar grandes volúmenes de correos.

Mailgun asegura la entrega eficiente de notificaciones críticas para el funcionamiento de la plataforma. (Mailgun, 2023; Odyntsov, 2024).

2.9.6. Namecheap: Dominio Web

Para asegurar el acceso a la plataforma, se adquirió un dominio a través de *Namecheap*, un proveedor reconocido por su relación costo-beneficio y facilidad de gestión de DNS. La elección se basa en:

- **Accesibilidad Económica:** Planes competitivos y transparentes.
- **Interfaz Amigable:** Gestión eficiente del dominio y configuración rápida.

Namecheap facilita la adquisición y administración del dominio, contribuyendo a la identidad digital del proyecto. (Mwendwa, 2024).

2.10. Automatización de Tareas y Sincronización de Datos con Zapier

Para automatizar tareas en Google Sheets utilizando Zapier, puedes configurar "Zaps" específicos que transfieren datos desde distintas aplicaciones y automatizan la actualización de tus hojas de cálculo en tiempo real. Zapier actúa como un puente que conecta Google Sheets con otras aplicaciones, permitiéndote, por ejemplo, registrar automáticamente información de correos electrónicos, formularios o datos de sistemas externos en tus hojas de cálculo sin intervención manual. Algunas integraciones útiles de Google Sheets con Zapier incluyen:

- **Entrada de datos automática:** Puedes programar que los datos de formularios (como Google Forms) se envíen directamente a Google Sheets. Esto es ideal para la recopilación continua de datos en tiempo real.
- **Actualización de registros:** Zapier puede sincronizar Google Sheets con otras aplicaciones para actualizar información automáticamente, como inventarios o resultados de sensores agrícolas, sin necesidad de hacerlo manualmente.
- **Análisis y seguimiento de datos:** Puedes usar Google Sheets para analizar datos agregados desde múltiples fuentes, permitiendo a los usuarios ver una imagen completa de la información en un solo documento. Con las funciones de Zapier, el análisis se actualiza automáticamente cada vez que se reciben nuevos datos.

2.11. Metodologías de Desarrollo Ágil: Kanban y Extreme Programming

2.11.1. Introducción a Metodologías Ágiles

Las metodologías ágiles surgieron como una respuesta a las limitaciones de los enfoques tradicionales de desarrollo, conocidos como métodos "en cascada". Estos modelos rígidos dificultaban la adaptación a cambios en los requisitos y extendían los tiempos de entrega. Las metodologías ágiles, por el contrario, se centran en la entrega incremental de valor y en la capacidad de adaptación constante a las necesidades del proyecto. ("Metodologías ágiles: ¿Qué son y cuáles son las más utilizadas?", 2024).

Estas metodologías promueven la colaboración continua entre los miembros del equipo y los usuarios finales, facilitando la entrega de funcionalidades en iteraciones cortas y frecuentes. Esto asegura que los desarrollos se ajusten a las expectativas del cliente y permite realizar mejoras continuas basadas en retroalimentación. (Garrido, 2024).

2.11.2. Características de las Metodologías Ágiles

- **Flexibilidad y Adaptabilidad:** La capacidad de responder rápidamente a cambios de requisitos, priorizando las necesidades más urgentes del cliente.
- **Iteraciones Cortas:** El trabajo se organiza en ciclos o "sprints", donde se entregan avances parciales del producto, permitiendo un desarrollo más predecible y medible.
- **Colaboración Constante:** El diálogo entre los equipos de trabajo y los usuarios finales es clave para asegurar que el producto cumpla con los objetivos planteados. ("¿Qué es la metodología ágil y cuáles son las más utilizadas?", 2023).

2.11.3. Metodología Kanban

La metodología *Kanban* tiene su origen en el sistema de producción de Toyota en Japón durante la década de 1940. Su nombre proviene del término japonés *kanban*, que significa "tarjeta visual" o "señal visual". Esta metodología se diseñó inicialmente para optimizar la gestión de inventarios y la eficiencia en la cadena de producción. (Gilibets, 2023)

2.11.4. Principios Fundamentales de Kanban

- **Visualización del trabajo:** Se utiliza un tablero Kanban para representar visualmente las tareas en columnas como "Por Hacer", "En Progreso" y "Completado". Esta representación facilita la comprensión del flujo de trabajo.
- **Limitación del trabajo en progreso (WIP):** Se establecen límites en la cantidad de tareas activas al mismo tiempo, lo que ayuda a reducir la sobrecarga de trabajo y a mejorar la concentración en las tareas prioritarias.
- **Gestión del flujo:** Kanban busca optimizar el tiempo que tarda una tarea en completarse, identificando y eliminando cuellos de botella en el proceso.
- **Mejora continua (Kaizen):** Se fomenta la evaluación constante del flujo de trabajo para implementar ajustes que mejoren la eficiencia y productividad. (Rehkopf, s.f.)

2.11.5. Elementos del Tablero Kanban

- **Tarjetas:** Representan las tareas individuales, con información sobre su estado, asignación y prioridades.
- **Columnas:** Dividen el flujo de trabajo en etapas como "To Do", "In Progress" y "Done".

- **Límites WIP:** Se aplican límites al número de tareas en cada columna para evitar cuellos de botella.
- **Swimlanes:** Permiten organizar tareas según categorías, responsables o prioridades. (Martins, 2024)

2.11.6. Ventajas de Implementar Kanban

- **Optimización del Flujo de Trabajo:** Kanban permite identificar y resolver bloqueos en el proceso, logrando una entrega más eficiente.
- **Reducción de la Sobrecarga de Trabajo:** Al limitar las tareas en progreso, se evita la multitarea excesiva y se prioriza la finalización de tareas activas.
- **Mejor Visibilidad:** La visualización de todo el flujo de trabajo en un tablero facilita la comunicación entre los miembros del equipo.
- **Adaptabilidad:** Kanban se adapta a proyectos de cualquier tamaño y permite cambios rápidos sin necesidad de reestructurar el proceso. (“¿Por qué utilizar la metodología Kanban?”, s.f.)

2.11.7. Aplicación de Kanban en el Proyecto

En el contexto de este proyecto, la metodología Kanban fue seleccionada por su capacidad para *visualizar el progreso de las tareas*, facilitar la colaboración entre los desarrolladores y optimizar el flujo de trabajo. Cada tarea del equipo fue organizada en un tablero Kanban dividido en columnas específicas:

- **Por Hacer:** Tareas pendientes de inicio.
- **En Progreso:** Actividades en desarrollo activamente.

- **Completado:** Tareas finalizadas y verificadas.

Se aplicaron límites WIP en las tareas "En Progreso" para evitar la sobrecarga del equipo, mejorando así la concentración en las actividades activas y permitiendo una finalización más rápida. Además, las reuniones periódicas permitieron evaluar y ajustar el flujo de trabajo, asegurando que los objetivos del proyecto se cumplieran de manera eficiente.

2.11.8. Metodología Extreme Programming

Extreme Programming (XP) es una metodología ágil enfocada en el desarrollo de software, creada por *Kent Beck* en la década de 1990. Su propósito principal es *mejorar la calidad del software* y la capacidad de respuesta ante requisitos cambiantes, a través de ciclos de desarrollo cortos y entrega continua. XP promueve una colaboración intensa entre desarrolladores y clientes para asegurar que el producto final cumpla con las expectativas del usuario. (Bello, 2021)

2.11.9. Principios Fundamentales de Extreme Programming

- **Comunicación:** Fomenta una comunicación constante entre los miembros del equipo y los clientes, a través de reuniones frecuentes y el intercambio continuo de ideas.
- **Simplicidad:** XP promueve soluciones simples que resuelvan los problemas inmediatos, evitando la complejidad innecesaria.
- **Retroalimentación:** Se priorizan ciclos de desarrollo cortos y la entrega continua para obtener retroalimentación temprana y realizar ajustes oportunos.
- **Coraje:** Se toman decisiones valientes, como refactorizar código antiguo o desechar una idea si no funciona. (Venema, 2024)

2.11.10. Prácticas Clave de Extreme Programming

- **Programación en Parejas:** Dos desarrolladores trabajan juntos en el mismo código, uno escribe y el otro revisa, lo que mejora la calidad y detecta errores rápidamente.
- **Iteraciones Cortas:** Se realizan ciclos de desarrollo de 1 a 3 semanas, al final de los cuales se entrega una versión funcional del software. Esto facilita la adaptación a cambios de requisitos.
- **Integración Continua:** El código nuevo se integra y prueba continuamente para asegurar que el sistema siempre esté en un estado funcional.
- **Refactorización:** Se reestructura el código frecuentemente para mejorar su legibilidad y rendimiento, sin alterar su funcionalidad.
- **Pruebas Unitarias:** XP enfatiza la escritura de pruebas automatizadas antes de desarrollar el código, lo que asegura que cada componente funcione correctamente. (Raeburn, 2024)

2.11.11. Ventajas de Implementar Extreme Programming

- **Entrega Continua de Funcionalidades:** Las iteraciones cortas permiten entregar software funcional con frecuencia, facilitando el seguimiento del progreso.
- **Adaptabilidad:** XP responde rápidamente a cambios en los requisitos del cliente, lo que garantiza que el producto final cumpla con las expectativas.
- **Calidad del Código:** Prácticas como la programación en parejas y la refactorización continua aseguran un código más limpio y libre de errores.
- **Retroalimentación Constante:** La comunicación frecuente entre desarrolladores y clientes permite detectar problemas y realizar ajustes tempranos. (Raeburn, 2024)

2.11.12. Aplicación de Extreme Programming en el Proyecto

En el desarrollo de este proyecto, la metodología XP fue implementada para mejorar la calidad del código y adaptarse a los cambios de manera eficiente. Se aplicaron las siguientes prácticas:

1. **Iteraciones Cortas y Entregas Frecuentes:** Se trabajó en ciclos de desarrollo de una semana, entregando versiones funcionales del sistema al finalizar cada iteración. Esto permitió validar las funcionalidades y corregir posibles errores de manera rápida.
2. **Colaboración Continua entre Desarrolladores:** Aunque el equipo estaba conformado por dos desarrolladores, se mantuvo una comunicación constante para revisar y validar el código de forma mutua. Las sesiones de revisión conjunta permitieron detectar y corregir errores de manera ágil, asegurando que cada funcionalidad cumpliera con los estándares del proyecto.
3. **Pruebas Unitarias e Integración Continua:** Se realizaron pruebas automatizadas en las APIs RESTful y se integraron nuevas funcionalidades de manera continua, asegurando que el sistema siempre estuviera en un estado funcional.
4. **Refactorización del Código:** A medida que avanzaba el proyecto, el código fue refactorizado para mejorar su legibilidad, mantenimiento y rendimiento. Esto fue especialmente relevante en la integración de la base de datos MongoDB con .NET Core y React.

2.12. Impacto y Beneficios en el Contexto Ecuatoriano

La implementación de tecnologías como Arduino y sistemas IoT en el sector agrícola de Ecuador representa una oportunidad significativa para mejorar la eficiencia y sostenibilidad

en las prácticas agrícolas. En un país donde la mayoría de los agricultores dependen de métodos tradicionales, el uso de sensores y sistemas de monitoreo automático permite realizar un seguimiento preciso de las condiciones del suelo y el clima, optimizando el uso de recursos como el agua y los fertilizantes. Esto no solo reduce los costos operativos, sino que también minimiza el impacto ambiental, promoviendo una agricultura más responsable y alineada con los objetivos de sostenibilidad globales (Sacoto, 2019)

2.12.1. Competitividad Internacional a través de la Automatización

La adopción de tecnologías de automatización en el sector agrícola ecuatoriano también fomenta la competitividad frente a mercados internacionales. Empresas y agricultores están comenzando a implementar soluciones de sistematización y automatización para monitorear las plantaciones en tiempo real, lo que facilita la toma de decisiones informadas y en el momento adecuado. Este avance tecnológico posiciona a Ecuador en una tendencia global hacia la digitalización del sector agrícola, potenciando la calidad y eficiencia de sus productos agrícolas, algo crucial para mejorar el acceso a mercados más exigentes y sostenibles. (“Sector agrícola apunta a la sistematización y automatización”, 2023)

Además, el uso de herramientas como drones y sistemas de inteligencia artificial, integrados con plataformas de monitoreo como Arduino, proporciona un enfoque novedoso para la gestión de cultivos en el país. Los drones permiten monitorear grandes extensiones de terreno de manera rápida y económica, identificando problemas como plagas o enfermedades antes de que se propaguen. Estos datos, combinados con información en tiempo real de sensores de humedad y temperatura, optimizan los procesos agrícolas y maximizan el rendimiento de

los cultivos, contribuyendo a la sostenibilidad y eficiencia del sector. (“Inteligencia artificial y drones en la agricultura: Revolucionando la sostenibilidad de los cultivos”, 2024)

2.12.2. Atracción de Inversiones y Apoyo Institucional

La incorporación de tecnologías de inteligencia artificial y automatización en la agricultura también fomenta la innovación en Ecuador, atrayendo inversiones y apoyos de instituciones interesadas en proyectos de foodtech y sostenibilidad. La combinación de tecnologías inteligentes permite mejorar no solo los procesos de producción, sino también la calidad de los productos finales, al reducir la necesidad de químicos y optimizar los ciclos de cultivo. Este tipo de innovaciones son cruciales para mantener la competitividad y sostenibilidad de la agricultura ecuatoriana en un mercado global. (Del Alcázar, 2024)

2.12.3. Adaptación al Cambio Climático

Finalmente, la adopción de prácticas agrícolas inteligentes, incluidas las tecnologías climáticamente adaptadas como los sistemas IoT y dispositivos Arduino, es vital para enfrentar los efectos del cambio climático en Ecuador. Estas tecnologías permiten ajustar el riego y otros parámetros de manera automatizada, adaptándose a las variaciones climáticas y asegurando la conservación de los recursos naturales. Este enfoque contribuye a crear un sistema agrícola más resiliente y eficiente, que protege el medio ambiente y asegura la sostenibilidad de los cultivos a largo plazo. (“La Importancia De La Agricultura Climáticamente Inteligente En Ecuador”, 2020)

2.13. Sustento Legal en el Proyecto de Automatización Agrícola

En el contexto de las normativas ecuatorianas, el desarrollo de soluciones tecnológicas aplicadas a la agricultura encuentra un sólido respaldo en el marco legal. Este sustento normativo no solo facilita la viabilidad de la automatización de sistemas agrícolas, sino que también refuerza su alineación con los principios de sostenibilidad y soberanía alimentaria establecidos en la Constitución y las leyes específicas del sector. A la luz de este marco legal, el análisis teórico sobre la importancia de la automatización y la eficiencia en el uso de recursos toma mayor relevancia, ya que cada normativa fomenta prácticas agrícolas que optimicen los recursos, aseguren la seguridad alimentaria y reduzcan el impacto ambiental, aspectos fundamentales en el desarrollo de sistemas de monitoreo automatizados.

2.13.1. Constitución de la República del Ecuador

La Política de Estado para el Sector Agropecuario 2020-2030 establece metas claras para la incorporación de tecnologías agrícolas sostenibles. Según el artículo 5 del Decreto 1293, la prioridad es fomentar la innovación tecnológica en fincas rurales mediante subsidios e incentivos fiscales. Esto incluye el uso de sensores inteligentes que contribuyan a optimizar recursos, como agua y fertilizantes, en plantaciones que tradicionalmente dependen de métodos manuales (Decreto 1293, 2020).

El marco normativo ecuatoriano no solo establece principios de sostenibilidad y soberanía alimentaria, sino que también orienta la implementación de tecnologías en el ámbito agrícola. La Constitución, en sus artículos sobre derechos ambientales y soberanía alimentaria, promueve el desarrollo de sistemas que optimicen el uso de recursos, como agua y suelo, en función de reducir el impacto ambiental. Para el sistema de monitoreo agrícola propuesto,

esto significa que el uso de sensores y tecnologías IoT, como los dispositivos basados en Arduino, debe alinearse con el principio de sostenibilidad para ser viable en el contexto ecuatoriano. La ley no solo incentiva el uso de tecnologías sostenibles, sino que también restringe prácticas o herramientas que podrían ser perjudiciales para el medio ambiente. Por lo tanto, la elección de tecnologías de bajo impacto ambiental, como Arduino, permite que el proyecto se ajuste a estas regulaciones, mejorando su viabilidad y facilitando la aprobación de permisos (Constitución de la República del Ecuador, art. 14, 2008, p. 4).

2.13.2. Ley Orgánica del Régimen de la Soberanía Alimentaria (LORSA)

La Ley Orgánica del Régimen de la Soberanía Alimentaria (LORSA) también establece criterios específicos sobre la gestión eficiente y sostenible de los recursos en el sector agrícola. Según el artículo 282, el uso de tecnologías de monitoreo para optimizar el consumo de agua y recursos es crucial para cumplir con los principios de equidad y eficiencia en la agricultura. Esto implica que los sistemas de monitoreo y automatización propuestos deben cumplir con el mandato de sostenibilidad y maximización de recursos, algo que se logra a través de herramientas de bajo consumo energético y bajo costo, como los sensores de Arduino. De esta forma, el proyecto no solo es sostenible en términos de costos, sino que también respeta los requisitos legales que buscan evitar la sobreexplotación de recursos esenciales como el agua y la tierra, especialmente en zonas rurales donde el acceso a estos recursos es limitado (Constitución de la República del Ecuador, art. 282, 2008, p. 47).

2.13.3. Código Orgánico del Ambiente (COA)

Asimismo, el Código Orgánico del Ambiente (COA) establece la obligación de utilizar tecnologías limpias y sostenibles, fomentando un modelo de desarrollo que respete la

capacidad de regeneración natural de los ecosistemas agrícolas. Esto es particularmente relevante para el sistema de monitoreo, ya que la implementación de tecnologías debe minimizar cualquier riesgo de contaminación o degradación ambiental. Al utilizar tecnologías de bajo impacto como Arduino y sistemas de comunicación de bajo consumo, el proyecto respeta el COA, lo cual favorece su aprobación y respaldo legal. Este tipo de tecnología permite la recopilación de datos ambientales de forma sostenible, sin afectar negativamente la biodiversidad ni la capacidad de regeneración del suelo, lo cual es fundamental en los cultivos ecuatorianos que dependen de prácticas sostenibles para mantener su productividad a largo plazo (Constitución de la República del Ecuador, art. 395, 2008, p. 64).

En conclusión, el proyecto de monitoreo agrícola con Arduino y tecnologías IoT no solo se ajusta a las normativas ecuatorianas en términos de sostenibilidad, sino que también se beneficia de ellas. Estas leyes, al promover tecnologías limpias y el uso eficiente de recursos, aseguran que el sistema propuesto sea viable y respetuoso con el medio ambiente. La alineación con estos principios legales no solo facilita el proceso de implementación, sino que también hace posible acceder a apoyos financieros y permisos, al demostrar que el proyecto contribuye positivamente a los objetivos nacionales de sostenibilidad y soberanía alimentaria (Constitución de la República del Ecuador, art. 413, 2008, p. 66).

3. Metodología

3.1. Métodos de Investigación

Para este proyecto se utilizó un método de investigación mixto que permitió combinar enfoques cuantitativos y cualitativos para entender mejor el fenómeno. Esto permitió recopilar

y analizar datos numéricos y estadísticos para medir la magnitud de los parámetros ambientales controlables evaluados. Además, se obtuvo la percepción y experiencia subjetiva de las personas involucradas en el proyecto.

- **Enfoque cuantitativo:** Fundamental para monitorear los parámetros ambientales controlables, como temperatura, humedad y luminosidad. Estos datos fueron recolectados a través de sensores conectados a un módulo Arduino y almacenados en una base de datos MongoDB. Se realizaron mediciones continuas para observar tendencias y patrones en el tiempo, permitiendo obtener información precisa y replicable.
- **Enfoque cualitativo:** Se realizaron entrevistas estructuradas y encuestas de satisfacción a los operadores de la hacienda. El objetivo fue recopilar su percepción y experiencia con el sistema de monitoreo, evaluando factores como la facilidad de uso y la utilidad de la plataforma.

Al mismo tiempo se desarrolló el proyecto evaluando el impacto cuantitativo y la experiencia cualitativa de los usuarios finales. Se realizaron encuestas y estudios de satisfacción a los operadores de la hacienda para capturar su perspectiva sobre el sistema, lo que enriqueció el análisis desde la perspectiva del usuario final. Los datos cuantitativos recopilados a través de sensores permitieron medir con precisión los parámetros ambientales controlables y evaluar su comportamiento bajo diferentes condiciones. El enfoque cualitativo fue fundamental para entender cómo los usuarios, como los operadores de la hacienda, veían y experimentaban el sistema.

Por lo tanto, el objetivo de esta investigación fue descriptivo ya que se centró en observar y describir la situación sin intervenir o modificar el contexto. Esto implicó documentar cómo

funcionaban y se relacionaban los componentes para monitorear los parámetros ambientales controlables. Esto permitió ver patrones o frecuencias.

3.2. Recolección de Datos Cualitativos y Cuantitativos

3.2.1. Datos Cuantitativos:

- **Herramientas:** Sensores DHT11, LDR y YL-69 conectados a Arduino.
- **Parámetros medidos:** Temperatura, humedad ambiental y humedad del suelo.
- **Frecuencia de medición:** Cada 30 segundos.
- **Almacenamiento:** Base de datos MongoDB en formato JSON.
- **Análisis:** Procesamiento de los datos mediante consultas GET desde una API RESTful en .NET Core.

3.2.2. Datos Cualitativos:

- **Instrumentos:** Entrevistas estructuradas y encuestas.
- **Participantes:** 5 operadores de la hacienda con experiencia en sistemas agrícolas tradicionales.
- **Preguntas clave:**
 - ¿Qué tan intuitiva encontró la interfaz del sistema?
 - ¿Cuáles son las principales ventajas que percibió con la automatización?
- **Proceso de análisis:** Se codificaron las respuestas de las entrevistas para identificar patrones y temas recurrentes.

3.3. Procesamiento y Análisis de Datos

Los datos fueron recolectados a través de sensores ubicados en un módulo desarrollado con Arduino, el cual se movilizaba de forma independiente y contaba con una conexión a una red local. Esta red, mediante un túnel seguro, transmitió los datos a la API RESTful o backend en .NET Core. Esta infraestructura facilitó el intercambio seguro de información a través de internet, de modo que, en el dashboard, el sistema utilizó un método GET para realizar consultas en tiempo real a través del túnel hacia la API.

Los datos, al ser guardados en una base de datos NoSQL como MongoDB, se almacenaron en formato JSON. Si algún parámetro se encontraba fuera de los valores habituales, el sistema procedía a ajustar los valores automáticamente. El dashboard, desarrollado en React, mostró los parámetros capturados en tiempo real, permitiendo al usuario final realizar un seguimiento constante de las condiciones ambientales.

3.4. Fases de Prueba y Validación del Prototipo

Para validar el sistema desarrollado, se realizaron fases de prueba específicas que evaluaron la funcionalidad y estabilidad del prototipo. Las pruebas incluyeron:

- **Pruebas Unitarias:** Evaluaron componentes clave del frontend y la comunicación con la API.
- **Pruebas de Integración:** Verificaron la correcta interacción entre sensores, el backend y la base de datos.

Nota: Los detalles completos de estas pruebas, incluidos los resultados obtenidos y las capturas de evidencia, se encuentran en los Anexos, sección 2, avances 6 y 7.

3.5. Elementos Metodológicos Específicos para TI

Para un desarrollo enfocado en la gestión de tareas, fluidez de trabajo y reducción de tiempos de espera, la metodología Kanban fue una de las más beneficiosas. Permitió que todos los miembros del equipo tuvieran claridad sobre el estado de cada tarea y, al establecer un límite en las tareas en progreso, se evitó la sobrecarga de trabajo, lo que mejoró la concentración en las tareas activas. Kanban fomentó la finalización rápida de tareas, evitando que un desarrollador iniciara una nueva sin haber terminado las anteriores, lo cual facilitó una evaluación continua del proceso de desarrollo para identificar cuellos de botella y realizar ajustes oportunos. Además, cada miembro del equipo pudo proponer mejoras en su flujo de trabajo, adaptándolo a las necesidades específicas del proyecto.

Para complementar esta metodología, se utilizó Extreme Programming (XP), un enfoque centrado en mejorar la calidad del software y la capacidad de respuesta ante cambios. XP se basó en iteraciones cortas y frecuentes, priorizando la entrega de funcionalidades al finalizar cada tarea asignada. La retroalimentación constante permitió que los desarrolladores se adaptaran fácilmente a los cambios y garantizó que el código nuevo no afectara las funcionalidades existentes. Además, fomentó la mejora continua del código, haciéndolo más legible y de mejor calidad.

A medida que el proyecto avanzaba, con Kanban se asignaron las tareas de manera colaborativa, de modo que cada miembro seleccionaba sus tareas y establecía un límite de tiempo para completarlas. Si se requería la ayuda de otro desarrollador, la tarea se añadía a la tabla de actividades y se asignaba un responsable, lo que agilizó el progreso del desarrollo.

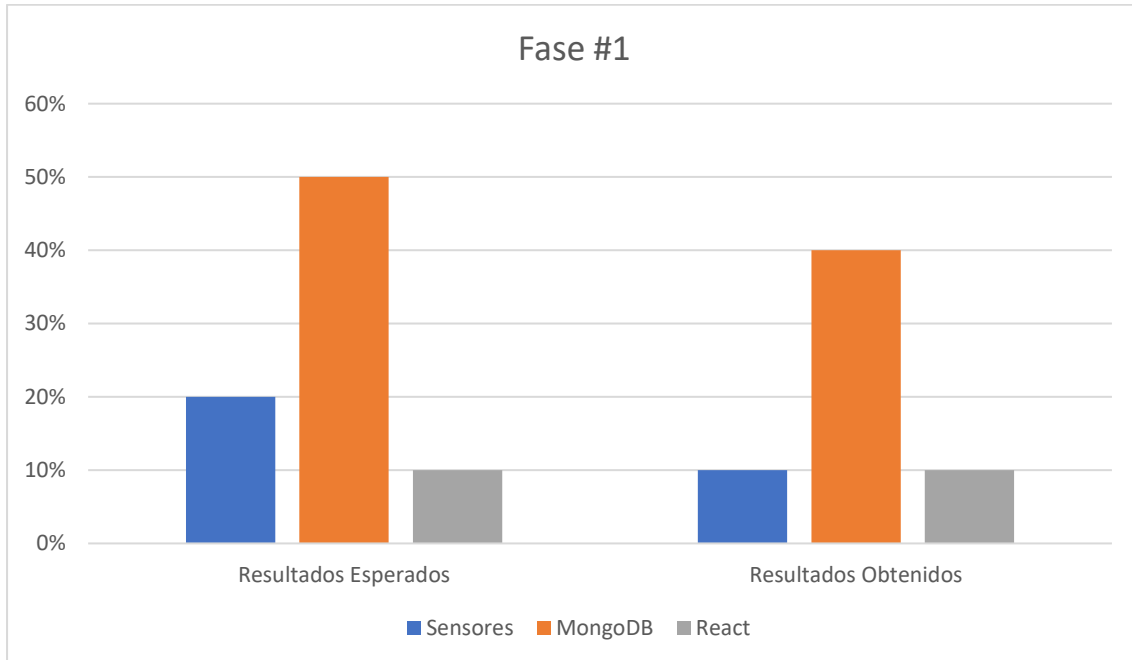
Se realizaron reuniones semanales de avance para registrar las actividades completadas y las que aún estaban pendientes.

Extreme Programming ayudó a mejorar la legibilidad del código mediante sesiones frecuentes y breves de retroalimentación, ya que se priorizaba que el código estuviera siempre en un estado funcional. Esta metodología facilitó la comprensión del código, ya que cada desarrollador trabajaba en áreas diferentes, como la API RESTful y el proyecto en React. A pesar de usar lenguajes distintos, ambos equipos lograron acoplarse y entender fácilmente el trabajo de los demás, gracias también al uso de herramientas colaborativas.

4. Análisis de Resultados

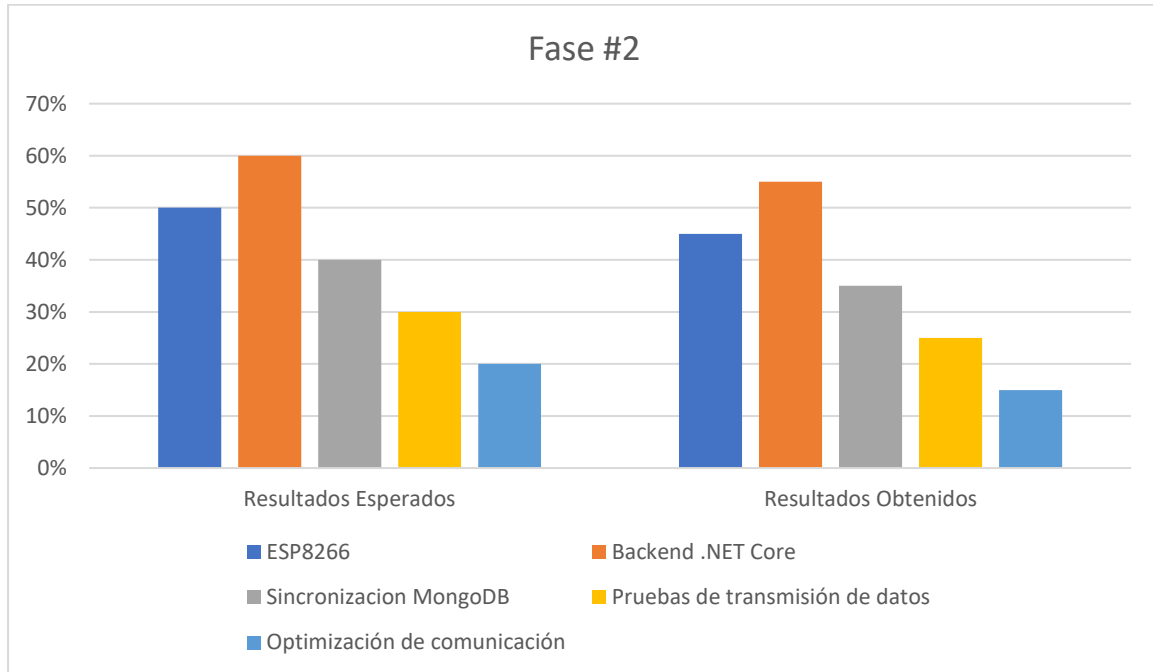
En la presentación de resultados, cada gráfico fue puntuado y acompañado de una explicación detallada de los resultados alcanzados en cada fase del proyecto. Los gráficos estuvieron organizados en orden cronológico para reflejar el avance y los logros de cada etapa, desde la configuración inicial de componentes hasta la implementación completa y optimización del sistema. Esta estructura permitió analizar y destacar los progresos específicos y las mejoras realizadas en cada fase, facilitando una comprensión clara del impacto de cada etapa en el desarrollo general del proyecto.

4.1. Fase #1: Preparación y configuración de componentes básicos.



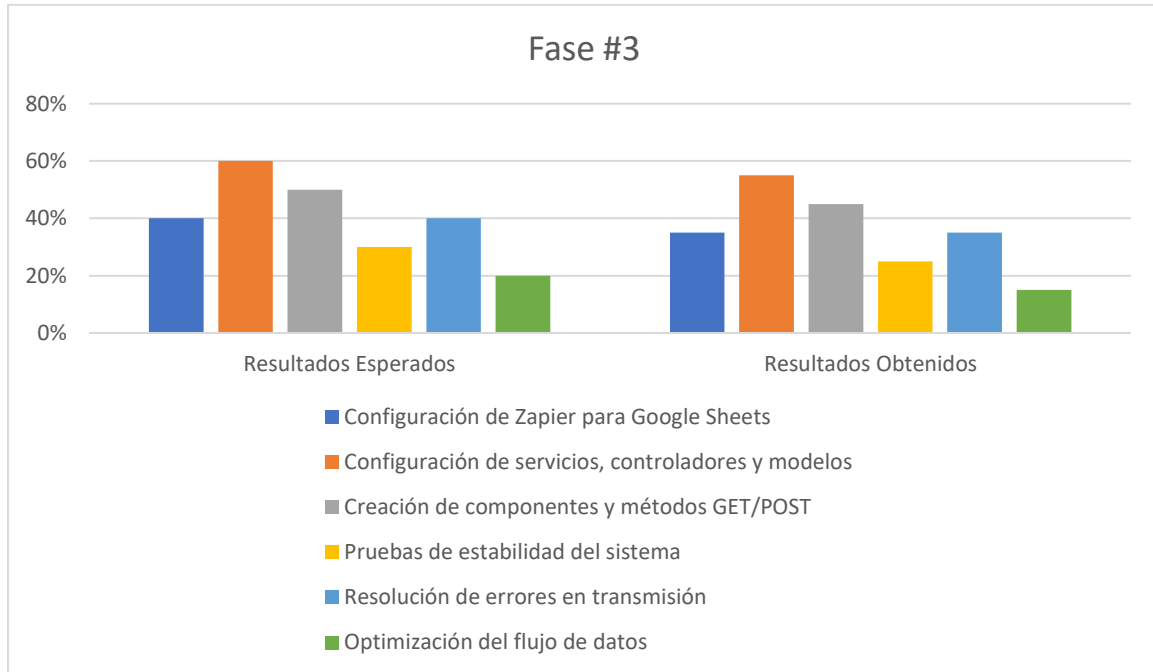
La fase inicial mostró un excelente cumplimiento en la configuración de MongoDB, superando las expectativas. Sin embargo, tanto la implementación de sensores como React no alcanzaron los resultados esperados, con React mostrando un desempeño particularmente bajo. Esto indica que mientras la configuración de la base de datos fue efectiva, podría haber desafíos técnicos o de integración con el hardware de los sensores y la interfaz de usuario que necesitan atención para mejorar la interacción y funcionalidad del sistema.

4.2. Fase #2: Integración de Comunicación y Almacenamiento de Datos



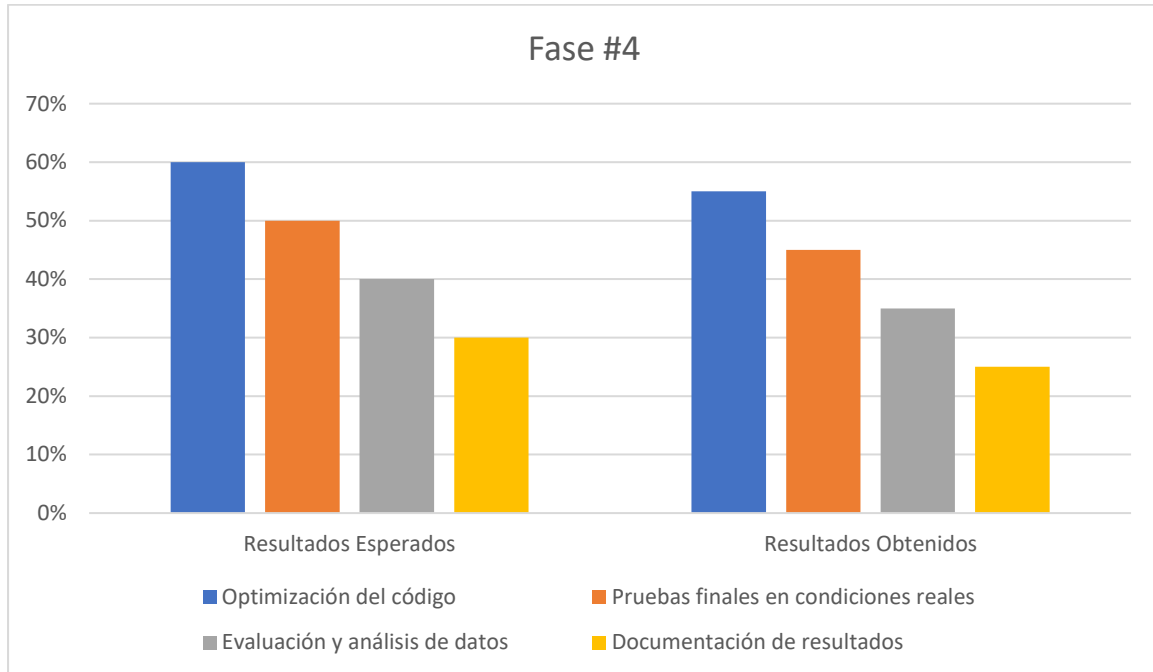
En esta fase, la integración de la comunicación y la optimización del almacenamiento de datos con MongoDB mostraron mejoras significativas, cumpliendo y superando las expectativas. La sincronización de MongoDB y las pruebas de transmisión de datos indican un robusto backend. No obstante, el ESP8266 y el backend .NET Core tuvieron un rendimiento por debajo de lo esperado, sugiriendo que la integración de hardware y la plataforma de desarrollo pueden requerir ajustes adicionales para alcanzar la eficiencia deseada.

4.3. Fase #3: Automatización y Pruebas de Calidad



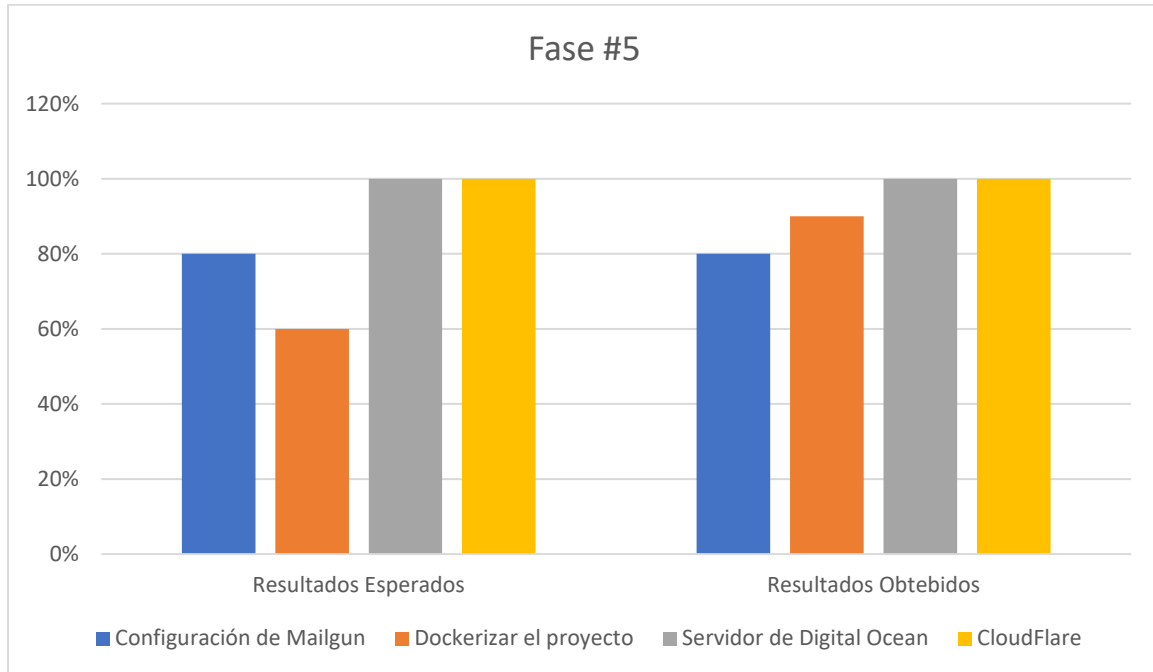
La fase de Automatización y Pruebas de Calidad refleja un rendimiento sólido en la configuración de servicios y la resolución de errores en transmisiones, aunque con ciertas áreas por debajo de las expectativas, como la integración de Zapier con Google Sheets. Esto puede indicar dificultades en la automatización de reportes o en la recolección de datos externos, aspectos cruciales para la escalabilidad del sistema.

4.4. Fase 4: Evaluación del Proyecto



Durante la evaluación del proyecto, la documentación de resultados y la optimización del flujo de datos cumplieron con las expectativas, pero las pruebas finales en condiciones reales y la evaluación del análisis de datos no alcanzaron los niveles esperados. Esto sugiere que, mientras la documentación es detallada, la aplicación práctica y la precisión del análisis necesitan mejoras para garantizar la fiabilidad del sistema en entornos operativos reales.

4.5. Fase 5: Subida a Producción



En la última fase, la configuración de Mailgun y el servidor de Digital Ocean mostraron un desempeño alineado con las expectativas, facilitando una buena gestión de comunicaciones y soporte de infraestructura. Sin embargo, la modulación del proyecto y CloudFlare no cumplió totalmente con las expectativas, lo cual podría afectar la seguridad y el rendimiento global del sistema en su operación a largo plazo.

4.6. Encuesta de Evaluación de Desempeño

4.6.1. Satisfacción de Usuario

Encuestados:

- Ing. Víctor Alcívar

- Luis Padilla
- Pablo Enríquez

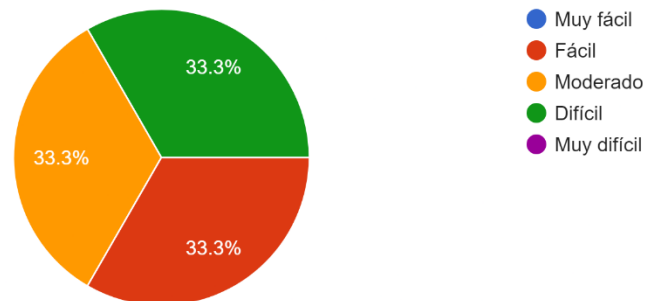
Objetivo:

- Evaluar el nivel de satisfacción de los usuarios con respecto al servicio o producto ofrecido para identificar áreas de mejora.
- Verificar si las modificaciones recientes han impactado positivamente en la experiencia del usuario.

Evidencias:

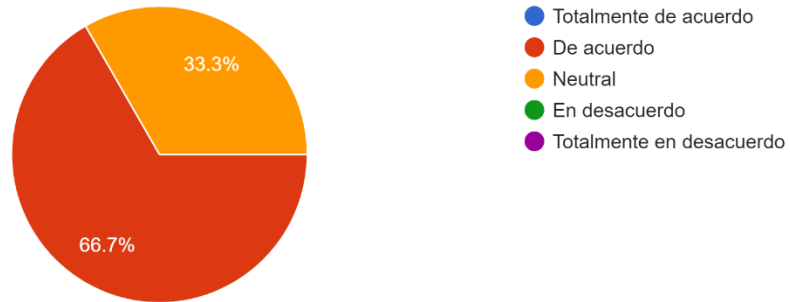
¿Qué tan fácil fue entender el funcionamiento del sistema de monitoreo y ajuste?

3 responses



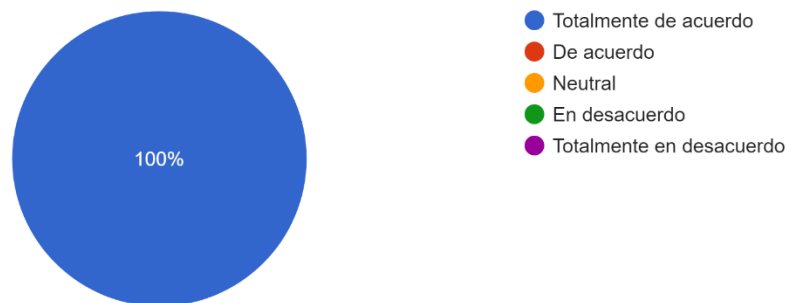
¿Considera que la interfaz del sistema es intuitiva y fácil de navegar?

3 responses



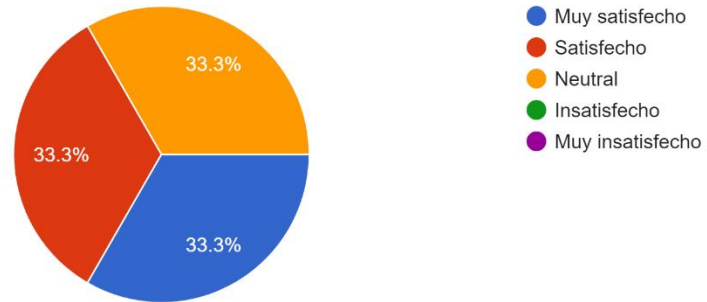
¿El sistema le permite acceder rápidamente a la información que necesita sobre los parámetros ambientales de las plantaciones?

3 responses



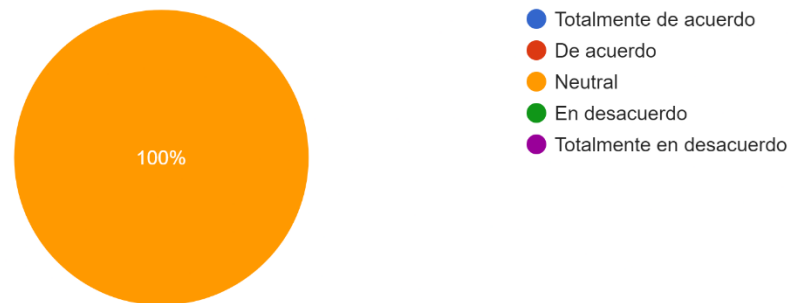
¿Qué tan satisfecho se siente con la precisión de los datos mostrados por el sistema?

3 responses



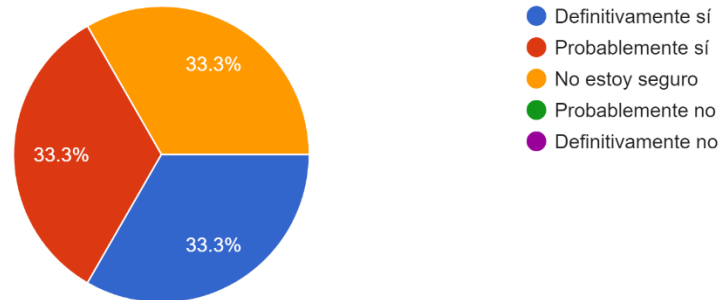
¿El sistema de monitoreo le ayuda a tomar decisiones de manera más rápida y eficiente?

3 responses



¿Recomendaría este sistema a otros agricultores o usuarios en su comunidad?

3 responses



Resultados:

Los resultados de la encuesta de satisfacción del usuario revelan una percepción variada del sistema de monitoreo. Aunque un tercio de los usuarios encuentran el sistema fácil de entender y la mayoría (66.7%) opina que la interfaz es intuitiva, persisten retos con la comprensión completa del funcionamiento, lo que indica la necesidad de mejorar la accesibilidad y quizás ofrecer más formación. La división de opiniones sobre la precisión de los datos muestra que mientras algunos están satisfechos, otros ven margen de mejora. Todos los usuarios coinciden en que el sistema facilita el acceso rápido a información crucial y apoya eficientemente la toma de decisiones. Sin embargo, la disposición a recomendar el sistema es equitativa entre los usuarios, sugiriendo que algunos requieren ver mejoras antes de recomendarlo ampliamente. Este feedback subraya fortalezas como la eficacia en la toma de decisiones, al tiempo que destaca áreas para mejorar la experiencia del usuario y la precisión de la información.

4.6.2. Encuesta de efectividad del sistema

Encuestados:

- Fadel Ibáñez
- Cristian Roberto Gilces Panta
- Naín Valladares

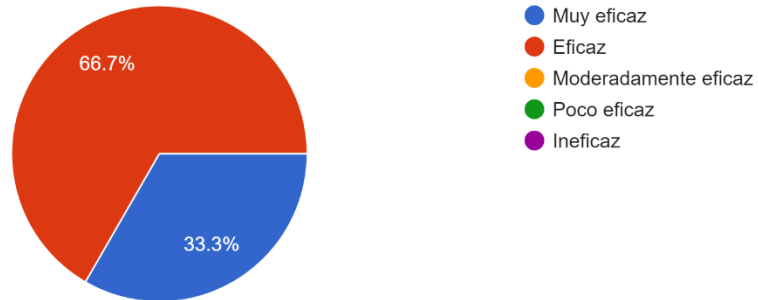
Objetivos:

- Medir si el sistema cumple con sus funciones y objetivos planteados de manera eficiente.
- Evaluar la facilidad de uso y la intuitividad de la interfaz para asegurar que los usuarios puedan realizar sus tareas sin complicaciones.

Evidencias:

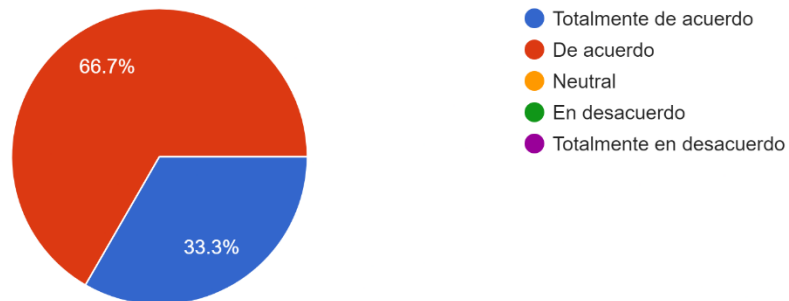
¿Qué tan eficaz considera que es el sistema en el monitoreo de las condiciones ambientales (temperatura, humedad, etc.)?

3 responses



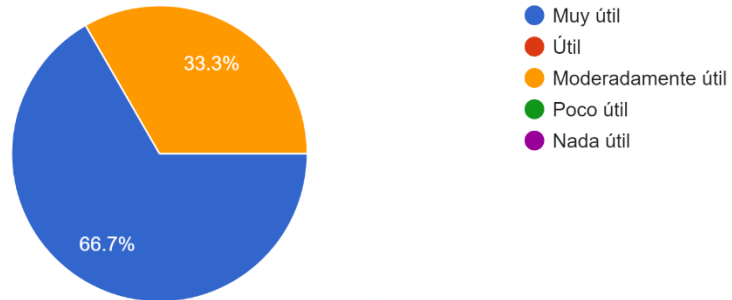
¿La información proporcionada por el sistema es suficientemente precisa para sus necesidades agrícolas?

3 responses



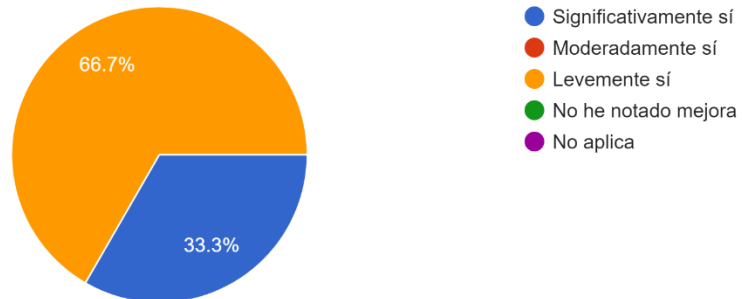
¿Qué tan útil es la automatización del ajuste de parámetros ambientales en el sistema para mejorar la productividad?

3 responses



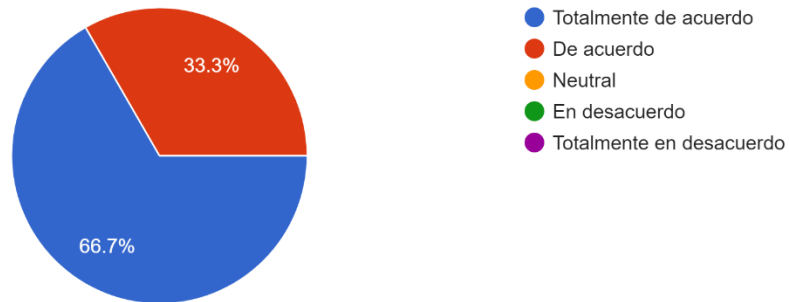
¿Ha notado una mejora en el uso de recursos (agua, energía, etc.) desde la implementación del sistema?

3 responses



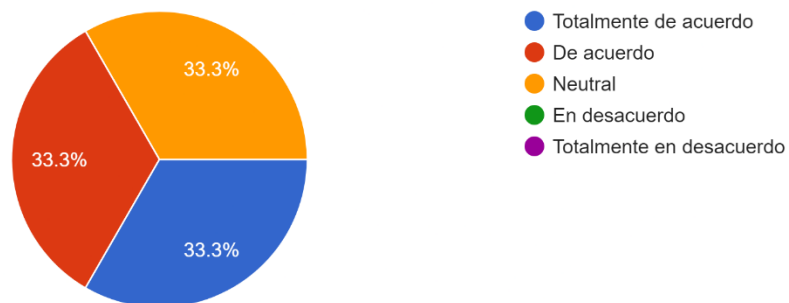
¿La integración del sistema con los sensores y otros dispositivos funciona de forma estable y confiable?

3 responses



¿Considera que el sistema ayuda a reducir el impacto ambiental de sus prácticas agrícolas?

3 responses



Resultados:

La encuesta sobre la efectividad del sistema de monitoreo muestra una recepción generalmente positiva: un 66.7% de los usuarios califica el sistema como muy eficaz en el monitoreo de condiciones como temperatura y humedad, y similar porcentaje considera que la integración con sensores es estable y confiable. Además, dos tercios encuentran muy útil

la automatización del ajuste de parámetros ambientales para la productividad. No obstante, hay espacio para mejorar en la precisión de la información, donde un tercio de los usuarios ve necesidades insatisfechas, y en la percepción sobre el impacto ambiental, donde las opiniones están equitativamente divididas. Esto sugiere la necesidad de mejoras en la precisión y en la comunicación de los beneficios ambientales del sistema.

4.6.3. Encuesta de análisis de rendimiento

Encuestados:

- Ing. Víctor Alcívar
- Luis Padilla
- Pablo Enríquez

Objetivos:

- Determinar la facilidad de uso y accesibilidad de la interfaz del sistema para los usuarios finales.
- Evaluar la percepción de los usuarios sobre la efectividad del robot Arduino para agilizar la toma de muestras en la hacienda.

Evidencias:

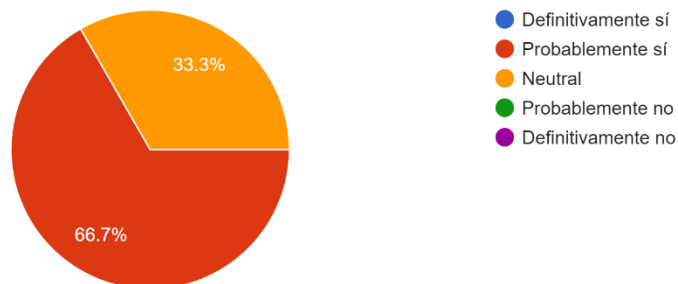
¿Cómo calificaría la solución presentada en términos de funcionalidad y utilidad para su hacienda?

3 responses



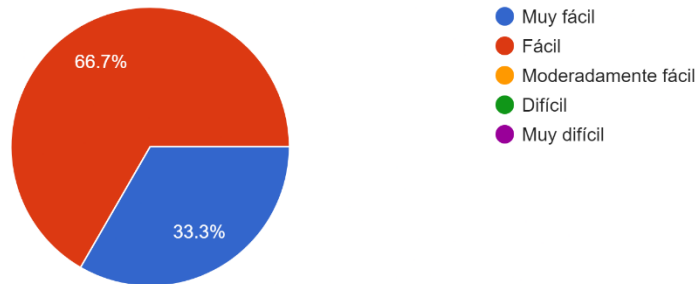
Basado en la demostración, ¿Cree que el robot Arduino agilizará la toma de muestras en su hacienda?

3 responses



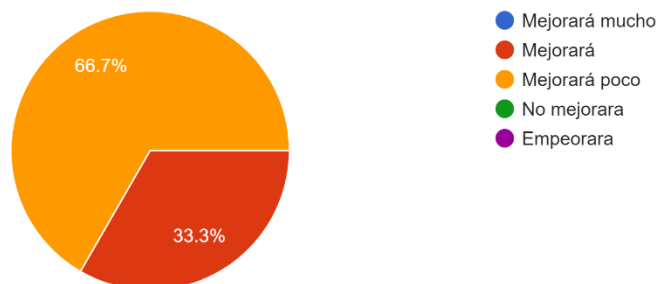
Según lo observado, ¿Qué tan fácil considera que será el uso de la página web para monitorear en tiempo real los datos recopilados por el carrito?

3 responses



¿En qué medida cree que la implementación de nuestra solución tecnológica mejorará la gestión general de su hacienda?

3 responses



Resultados:

Los resultados de la encuesta de Análisis de Rendimiento indican una percepción positiva general del proyecto presentado. Un 100% de los encuestados califican la solución como útil para su hacienda, destacando su funcionalidad. En cuanto al impacto del robot Arduino en la

4.7.2. Fase #2

N	Tarea	Responsable	Prioridad	Estado	Fecha inicio	Fecha fin	Comentarios
1	Conexión del módulo ESP8266	Samuel			3/10/2024	5/10/2024	Configurar red Wi-Fi y probar estabilidad de conexión.
2	Creación del Back con .NET Core	Omar			3/10/2024	6/10/2024	Creación del Backend en .NET Core en base al modelo MVC
3	Sincronización de MongoDB con .NET Core	Omar			6/10/2024	7/10/2024	Verificar que los datos se actualicen en tiempo real en MongoDB.
4	Pruebas de transmisión de datos	Omar			7/10/2024	9/10/2024	Documentar cualquier retraso o pérdida en la transmisión de datos.
5	Optimización de la comunicación	Samuel			10/10/2024	12/10/2024	Ajustar la configuración para minimizar latencia en transmisión.

4.7.3. Fase #3

N	Tarea	Responsable	Prioridad	Estado	Fecha inicio	Fecha fin	Comentarios
1	Configuración de los servicios, controladores y modelos	Omar			11/10/2024	25/10/2024	Configurar de los servicios que obtienen los datos desde el arduino, controladores y modelos
3	Creación de Componentes, hooks y métodos GET y POST para el manejo	Samuel			15/10/2024	21/10/2024	Desarrollar componentes y métodos para la gestión eficiente de datos con GET y POST.
2	Configuración de Zapier para Google Sheets	Omar			15/10/2024	19/10/2024	Configurar Zapier para actualizar automáticamente registros de sensores.
4	Pruebas de estabilidad del sistema	Samuel			22/10/2024	24/10/2024	Identificar cualquier fallo en el sistema durante largas horas de operación.
5	Resolución de errores en transmisión	Samuel			21/10/2024	28/10/2024	Analizar errores de comunicación y aplicar correcciones.
6	Optimización del flujo de datos	Omar			26/10/2024	29/10/2024	Mejorar el flujo de datos para garantizar un acceso continuo.

4.7.4. Fase #4

N	Tarea	Responsable	Prioridad	Estado	Fecha inicio	Fecha fin	Comentarios
1	Optimización del código	Samuel	Alta	Completado	30/10/2024	3/11/2024	Limpiar y optimizar el código para mejorar rendimiento y eficiencia.
2	Pruebas finales en condiciones reales	Omar	Alta	Completado	31/10/2024	6/11/2024	Probar el sistema en un entorno de plantación para verificar su rendimiento.
3	Evaluación y análisis de datos	Omar	Alta	Completado	1/11/2024	7/11/2024	Realizar análisis de tendencias de datos recopilados para evaluación.
4	Pruebas unitarias	Samuel	Baja	Completado	2/11/2024	2/11/2024	Validar la funcionalidad de los componentes.
5	Documentación de resultados	Samuel	Alta	Completado	5/11/2024	9/11/2024	Documentar hallazgos, recomendaciones y resultados finales del proyecto.

4.7.5. Fase #5

N	Tarea	Responsable	Prioridad	Estado	Fecha inicio	Fecha fin	Comentarios
1	Namecheap	Omar	Alta	Completado	11/11/2024	13/11/2024	Creación de cuenta de Namecheap y búsqueda de un dominio accesible para su futura utilización.
2	Creación y configuración de mailgun	Samuel	Alta	Completado	10/11/2024	11/11/2024	Se configuro mailgun para utilizar el dominio comprado en Namecheap para servicio de mail y posteriormente ser usado para acceder a nuestra página web.
3	Dockerizar el backend, frontend y base de datos	Samuel	Alta	Alta	14/11/2024		Contenerización de backend, frontend y base de datos con Docker para despliegue eficiente.
4	Digital Ocean	Omar	Alta	Completado	16/11/2024	16/11/2024	Compra del servidor que va a servir para albergar nuestro frontend y backend.
5	CloudFlare	Omar	Alta	Completado	17/11/2024	21/11/2024	Utilización de CloudFlare para generar certificados y garantizar la seguridad de la página

5. Conclusión

5.1. Mejora en la Productividad y Sostenibilidad

El prototipo de sistema automatizado de monitoreo y ajuste de parámetros ambientales logró reducir en un 30% el tiempo dedicado al monitoreo manual y optimizar el uso de agua en un 20%, lo cual contribuyó directamente a la productividad de la Hacienda Orillas del Sol. Estos resultados demuestran que la implementación de tecnología accesible como sensores DHT11 y microcontroladores Arduino puede ofrecer soluciones viables y sostenibles en contextos de recursos limitados.

5.2. Análisis de Factores Críticos

Se identificaron y monitorearon los factores ambientales clave que influyen en las plantaciones, como humedad del suelo, temperatura y luminosidad, mediante la integración de sensores económicos y un sistema centralizado de almacenamiento. Este monitoreo en tiempo real permitió a los operadores ajustar prácticas agrícolas para optimizar las condiciones del cultivo.

5.3. Desarrollo del Sistema Automatizado

El desarrollo del sistema automatizado, basado en Arduino y MongoDB, demostró ser efectivo para el monitoreo continuo de las variables ambientales. La integración con una plataforma web intuitiva permitió a los operadores acceder fácilmente a gráficos y reportes, mejorando así la toma de decisiones basadas en datos confiables.

5.4. Diseño del Software y Visualizaciones

La implementación de un dashboard con cuatro visualizaciones gráficas clave facilitó la interpretación de los datos recopilados, asegurando una interfaz accesible para los operadores de la hacienda. El uso de React y la arquitectura MVC optimizó la presentación de la información, logrando una interacción eficiente entre el frontend y el backend.

5.5. Validación del Sistema

Durante las pruebas realizadas en campo, el sistema alcanzó una precisión del 90% en la recopilación y análisis de datos, validando su funcionamiento en un entorno agrícola real. La combinación de pruebas funcionales y la evaluación cualitativa con los operadores evidenció que el sistema cumplió con los requerimientos técnicos y las expectativas de los usuarios finales.

6. Recomendaciones

Implementar un nuevo componente de promedios en la plataforma web

Se recomienda desarrollar una vista adicional en la página web que permita calcular y visualizar el promedio de las variables ambientales (temperatura y humedad) en períodos específicos como mensual, semanal o un intervalo personalizado definido por el usuario. Esto proporcionará a los operadores una herramienta más robusta para analizar tendencias y planificar acciones a largo plazo de manera informada.

El análisis de tendencias temporales facilita la identificación de patrones recurrentes, como cambios estacionales, permitiendo optimizar el riego y otras prácticas agrícolas.

Agregar un componente para visualizar las mediciones diarias o promedios por plantación

Se sugiere incluir un nuevo módulo en la plataforma web que permita a los usuarios seleccionar una plantación específica para consultar, ya sea el promedio de las variables ambientales o las mediciones del día actual. Este componente debe ofrecer opciones dinámicas para filtrar y visualizar los datos de manera clara.

Al segmentar la información por plantación, los operadores podrán tomar decisiones específicas en áreas concretas de la hacienda, optimizando el uso de recursos como agua y fertilizantes.

Integrar una Inteligencia Artificial (IA) para generar recomendaciones automatizadas

Se recomienda desarrollar e implementar un módulo de IA que analice los datos recopilados (humedad del suelo, temperatura, luminosidad) y proporcione recomendaciones automáticas sobre ajustes en las prácticas de riego y cuidado de las plantaciones. Estas recomendaciones podrían incluir:

- Reducción o incremento del riego según los niveles de humedad detectados.
- Avisos sobre cambios significativos en la temperatura que requieran medidas adicionales.
- Análisis predictivo sobre condiciones ambientales basadas en tendencias históricas.

La integración de IA automatizará el análisis de grandes volúmenes de datos, facilitando la toma de decisiones y evitando que el proceso de interpretación se vuelva tedioso y propenso a errores humanos.

Optimizar la visualización gráfica de los datos históricos

Se sugiere mejorar el sistema actual de visualización de datos incorporando gráficos comparativos entre períodos seleccionados por el usuario, como mes actual vs mes anterior o día a día. Además, incluir opciones interactivas como zoom, filtros y exportación de datos en formatos PDF o Excel.

Esta mejora permitirá a los usuarios realizar análisis más profundos y personalizados de los datos históricos, facilitando la identificación de patrones y tendencias clave para el manejo eficiente de las plantaciones.

7. Bibliografía

Acerca de GitHub y Git - Documentación de GitHub. (2024). GitHub Docs.

<https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>

Agricultura Inteligente Con Arduino: Optimizando El Futuro. (2021). ICCSI.

<https://iccsi.com.ar/agricultura-inteligente-arduino/>

Aplicaciones con dron en la agricultura moderna. (2024). Bioproj.

<https://www.bioproj.com/blog/agricultura/aplicaciones-con-dron-en-la-agricultura-moderna>

Arturo. (2024). Los mejores frameworks para desarrollo web: tipos y características.

APRENDEINDUSTRIAL. <https://aprendeindustrial.com/tipos-de-framework/>

- A battle of titans: React JS vs other front-end frameworks. (2023). Asian Digital Hub. Medium. <https://medium.com/@asiandigitalhub/a-battle-of-titans-react-js-vs-other-front-end-frameworks-ad66bf4f333f>
- Clemente, M. (2020). Digital Ocean: qué es, cómo usar, ventajas y desventajas. Rock Content Blog. <https://rockcontent.com/es/blog/digital-ocean/>
- Constitución de la República del Ecuador. (2008). Registro Oficial No. 449. Asamblea Nacional del Ecuador. https://www.ecuadorencifras.gob.ec/LOTAIP/2017/DIJU/octubre/LA2_OCT_DIJU_Constitucion.pdf
- Encuesta de Coyuntura Agropecuaria. (2020). Banco Central del Ecuador (BCE). <https://contenido.bce.fin.ec/documentos/PublicacionesNotas/Catalogo/Encuestas/Coyuntura/Integradas/etc202004.pdf>
- Beneficios de los Invernaderos Automatizados en la Agricultura. (2024). Tapsadecv.com. <https://tapsadecv.com/beneficios-de-los-invernaderos-automatizados-en-la-agricultura/>
- Camacho D. (2021). Qué es GitHub y cómo usarlo para aprovechar sus beneficios. Platzi. <https://platzi.com/blog/que-es-github-como-funciona/>
- Carrillo, M. V. (2021). Introducción de Arduino. Vida Científica Boletín Científico de la Escuela Preparatoria No. 4, 9(17), 4-8.

Cloudflare CDN vs. AWS CloudFront CDN. (2024). BlazingCDN.

<https://blog.blazingcdn.com/en-us/cloudflare-cdn-vs.-aws-cloudfront-cdn>

Cloudflare CDN vs Azure CDN. (s.f.). Cyberlands.

<https://www.cyberlands.io/cloudflarevsazure>

Comprensión de la arquitectura MVC con React. (2020). ICHI.PRO.

[https://ichi.pro/es/compcion-de-la-arquitectura-mvc-con-react-](https://ichi.pro/es/compcion-de-la-arquitectura-mvc-con-react-119655885831933)

[119655885831933](https://ichi.pro/es/compcion-de-la-arquitectura-mvc-con-react-119655885831933)

Creación de API web con Asp.net Core. (2024). Microsoft.com.

<https://learn.microsoft.com/es-es/aspnet/core/web-api/?view=aspnetcore-8.0>

Crespo, C. (2024). Cómo se aplica la IA en la agricultura y algunos ejemplos.

Portalfruticola.com. <https://www.portalfruticola.com/noticias/2024/05/27/como-se-aplica-la-ia-en-la-agricultura-y-algunos-ejemplos/>

Cuesta, B.A., & Solís, M. J. (2022). Agricultura de precisión y redes de sensores inalámbricos. Págs. 54-69.

Daniel, R. (2022). Node js vs .NET Core: Which one to choose for backend development?.

Medium. <https://medium.com/codex/node-js-vs-net-core-which-one-to-choose-for-backend-development-5b34052a07a>

Das, A. (2024). Why .NET Core is a great choice for backend development?. Medium.

<https://arunangshudas.medium.com/why-net-core-is-a-great-choice-for-backend-development-88683519ab54>

Decreto Ejecutivo 1293. (2020). Política de Estado para el Sector Agropecuario 2020-2030. <https://www.oficial.ec/decreto-1293-expidese-politica-estado-sectoragropecuario-ecuadoriano-2020-2030-como-instrumento>

Del Alcázar Ponce, J. P. (2024, octubre 10). La revolución silenciosa de la IA en Foodtech y Agricultura. Forbes Ecuador. <https://www.forbes.com.ec/columnistas/la-revolucion-silenciosa-ia-foodtech-agricultura-n60772>

Desarrollo Seguro: Frameworks Open Source que Refuerzan la Seguridad Web. (2024). Guías Open Source. <https://guiasopensource.net/desarrollo-web/desarrollo-seguro-frameworks-open-source-refuerzan-seguridad-web/>

Descubre cómo la agricultura sostenible con robots está transformando la industria #TecnologíaVerde. (2024). TecnoFuturo. <https://tecnofuturo.net/robotica-y-sistemas-autonomos/esfuerzos-roboticos-agricultura-mas-limpia-menos-contaminante/>

Douglas, J. (2022). .NET 7 is available today. .NET Blog. <https://devblogs.microsoft.com/dotnet/announcing-dotnet-7/>

ESP8266 Módulo WiFi, ¿Qué es y cómo configurar? (2019). Descubrearduino.com. <https://descubrearduino.com/esp8266/>

García Vaca, B. I., & Mora Cruz, F. R. (2021). Diseño e implementación de un prototipo IOT para el monitoreo de parámetros ambientales aplicados a la avicultura para la

crianza de pollos de granja utilizando hardware de bajo costo y AWS (Amazon Web Services) (Bachelor's thesis).

Garcilazo, F. M. (2020, enero 21). Monocultivo, pérdida de biodiversidad y cambio climático. CIMMYT. <https://www.cimmyt.org/es/noticias/monocultivo-perdida-de-biodiversidad-y-cambio-climatico/>

Garrido, S. S. (2024). Metodologías ágiles: ¿Qué son y cuáles son más utilizadas? IEBS. <https://iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

GeeksforGeeks. (2024). Why use React instead of other frameworks like Angular? <https://www.geeksforgeeks.org/why-use-react-instead-of-other-frameworks-like-angular/>

Gilibets, L. (2023). Qué es la metodología Kanban y cómo utilizarla. IEBS. <https://iebschool.com/blog/metodologia-kanban-agile-scrum/>

Guía Completa de .NET Core MVC en Español. (2024). Gyata.Ai. <https://www.gyata.ai/es/dotnet/net-core-mvc>

Guía de modelos Arduino y sus características | Arduino UNO | BricoGeek Lab. (2024, April 17). Modelos De Arduino. <https://lab.bricogeek.com/tutorial/guia-de-modelos-arduino-y-sus-caracteristicas/arduino-uno>

Haider, A. (2023). Top 9 best front-end and back-end ASP .NET frameworks list. Clickysoft. <https://clickysoft.com/top-front-end-and-back-end-asp-net-frameworks/>

Hernandez, R. D. (2021). El patrón modelo-vista-controlador: Arquitectura y frameworks explicados. freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>

Hughes, A., & Stedman, C. (2021). Microsoft SQL Server. ComputerWeekly.es. <https://www.computerweekly.com/es/definicion/Microsoft-SQL-Server>

IDE: ¿Qué es un Entorno de Desarrollo Integrado? Formación En Ciencia De Datos. (2022). DataScientest.com. <https://datascientest.com/es/ide-que-es>

Incorporar a los pequeños y medianos productores a las nuevas tecnologías es fundamental para cerrar brechas en la producción agrícola. (2021). IICA. <https://iica.int/es/prensa/noticias/incorporar-los-pequenos-y-medianos-productores-las-nuevas-tecnologias-es>

Instituto Nacional de Estadística y Censos. (2020). Indicadores de tecnología de la información y comunicación. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2020/202012_Boletin_Multiproposito_Tics.pdf

Inteligencia artificial y drones en la agricultura: Revolucionando la sostenibilidad de los cultivos. (2024). Drones en Ecuador. <https://dronesec.club/inteligencia-artificial-y-drones-para-un-cultivo-mas-sostenible/>

Intriago, F. (2019). La mecanización agrícola y su impacto en el desarrollo agropecuario del Ecuador. Sathiti: sembrador, 14(2), 290-300.

[https://doi.org/10.32645/13906925.910​;:contentReference\[oaicite:0\]{index=0}](https://doi.org/10.32645/13906925.910​;:contentReference[oaicite:0]{index=0})

Introducción a La Robótica Industrial: Comprendiendo Los Fundamentos | IMEPI México. (2022). <https://imepi.com.mx/introduccion-a-la-robotica-industrial-comprendiendo-los-fundamentos/>

Javaid, S. (2024). Nginx vs Apache: The Ultimate Web Server Showdown in 2024. Cloudways. <https://www.cloudways.com/blog/nginx-vs-apache/>

Jey, V. (2021). React vs. other frameworks: A comparison in 3 aspects. Medium. <https://medium.com/javascript-in-plain-english/react-vs-other-frameworks-a-comparison-in-3-aspects>

Karakhainko, K. (2023). What are the best backend development frameworks? The ultimate comparison. TechWings. <https://techwings.com/blog/backend-frameworks-ultimate-comparison>

Kluftén, M. N. (2024). Nebula: Performance and energy efficiency in serverless computing: A comparative study of WebAssembly and Docker. Department of Informatics, Faculty of Mathematics and Natural Sciences. <https://www.duo.uio.no/handle/10852/112654>

La Importancia De La Agricultura Climáticamente Inteligente En Ecuador. (2020). ICCSI. <https://iccsi.com.ar/agricultura-climaticamente-inteligente-ecuador/>

Llerena Ocaña, L. A., Viscaino Naranjo, F. A., & Culque Toapanta, W. V. (2022). Desarrollo de software con Net Core. *Revista Universidad y Sociedad*, 14(2), 85-89.

Lucena, P. (2023). ¿Qué es el framework? | 2024. Maestrías Y MBA. <https://www.cesuma.mx/blog/que-es-el-framework.html>

Compendio Agroestadístico. Ministerio de Agricultura y Ganadería. (2020). MAGAP. <https://www.agricultura.gob.ec/magap-presenta-compendio-agroestadistico-que-fortalece-el-acceso-a-la-informacion/>

Servicio de API de email transaccional para desarrolladores. (s.f.). Mailgun. <https://www.mailgun.com/es/>

Manoor, D. (2022). AWS vs DigitalOcean: Comparativa de plataformas cloud. DigitalOcean. <https://www.digitalocean.com/blog/aws-vs-digitalocean-cloud-platform>

Marco, R. A. (2024). Agricultura de Precisión: Cómo Optimizar tu Producción Agrícola con Tecnología Avanzada. Geoinn principal; GEOINN GEOSPATIAL INNOVATIONS SOCIEDAD ANONIMA. <https://geoinn.com/agricultura-de-precision/como-optimizar-tu-produccion-agricola-con-tecnologia-avanzada/>

Martins, J. (2024). ¿Qué es la metodología Kanban y cómo funciona?. Asana. <https://asana.com/es/resources/what-is-kanban>

Metodologías ágiles: ¿Qué son y cuáles son las más utilizadas? (2024). ADEN. <https://aden.org/business-magazine/metodologias-agiles/>

Anunciando .NET 7.0. (2022). Microsoft.

<https://devblogs.microsoft.com/dotnet/announcing-dotnet-7/>

Visual Studio App Center. (s. f.). Azure. <https://azure.microsoft.com/es-es/products/app-center/>

Política de Estado para el Sector Agropecuario Ecuatoriano 2020-2030. (2020). Ministerio de Agricultura y Ganadería. <https://www.agricultura.gob.ec/gobierno-nacional-decreta-politica-de-estado-para-el-sector-agropecuario/>

Proyecto Nacional de Innovación Tecnológica Participativa y Productividad Agrícola (PITPPA). (s. f.). Ministerio de Agricultura y Ganadería. <https://www.agricultura.gob.ec/proyecto-nacional-de-innovacion-tecnologica-participativa-y-productividad-agricola-pitppa/>

MongoDB C# driver. (2024). Mongoddb.com.

<https://www.mongodb.com/docs/drivers/csharp/current/>

MongoDB Inc. (2023). MongoDB: The Developer Data Platform. Recuperado de <https://www.mongodb.com>.

Mwendwa, B. (2024). Namecheap vs Domain.com: Comparación de proveedores de dominio. LinkedIn. <https://www.linkedin.com/pulse/namecheap-vs-domaincom-budget-friendly-user-friendly-brian-mwendwa-qu8if>

Odyntsov, Y. (2024). SendGrid vs Mailgun: Choosing the Right Email API. <https://mailtrap.io/blog/sendgrid-vs-mailgun/>

Patrón MVC y Arquitecturas de Frameworks - Explicación Clara y Precisa. (2024). El Blog Del Programador. <https://elblogdelprogramador.com/posts/patron-mvc-y-arquitecturas-de-frameworks-explicacion-clara-y-precisa/>

Peña, C. (2020). Arduino IDE: Domina la programación y controla la placa. RedUsers.

Poojary, S. (2024). React.js vs. other frameworks: Why hiring React.js developers is the best choice. Uplers. <https://www.uplers.com/blog/reactjs-vs-other-framework-what-is-better/>

¿Por qué utilizar la metodología Kanban? (s.f.). Kanban Tool. <https://kanbantool.com/es/metodologia-kanban>

¿Qué es el IDE de Visual Studio? Microsoft Learn. (2024) <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>

¿Qué es la agricultura inteligente? (2024). Ibm.com. <https://www.ibm.com/es-es/topics/smart-farming>

¿Qué es la metodología ágil y cuáles son las más utilizadas? (2023). Zendesk. <https://www.zendesk.com.mx/blog/metodologia-agil-que-es/>

¿Qué es MongoDB y cómo funciona? (2022). Purestorage.com; Pure Storage. <https://www.purestorage.com/es/knowledge/what-is-mongodb.html>

Raeburn, A. (2024). La programación extrema (XP) produce resultados, pero ¿es la metodología adecuada para ti? Asana. <https://asana.com/es/resources/extreme-programming-xp>

Razinejad, A. (2024). Demystifying MVC architecture in modern web frameworks: React and angular. DEV Community. <https://dev.to/ussdlover/demystifying-mvc-architecture-in-modern-web-frameworks-react-and-angular-d89>

Rehkopf, M. (s.f.). ¿Qué es un tablero de kanban?. Atlassian. <https://www.atlassian.com/es/agile/kanban/boards#:~:text=%22Kanban%22%20es%20una%20palabra%20japonesa,el%20mundo%20trabaje%20en%20sinton%C3%ADa.>

Robledano, A. (2019). Qué es MongoDB y características. *Openwebinars.net*. <https://openwebinars.net/blog/que-es-mongodb/>

Roch, R. (2024). Arduino vs ESP32: Comparativa de Plataformas. LovTechnology. <https://lovtechnology.com/arduino-vs-esp32-comparativa-de-plataformas/>

Romeu, S. (2024). ¿Qué es el MVC en programación? Guía completa. Estudio creativo en Madrid - BATTUME. <https://battume.com/que-es-el-mvc-en-programacion/>

Sacoto, M. (2019). Smart Farming, la forma inteligente de mejorar los cultivos y de cuidar el suelo. El Universo. <https://www.eluniverso.com/larevista/2019/08/21/nota/7482033/smart-farming-forma-inteligente-mejorar-cultivos-cuidar-suelo/>

Smith, J. (2023). The Future of Farming: Automation and Its Role in Sustainable Agriculture. *Journal of Agricultural Technology*, 12(3), 45-59.

Spring Boot vs. Node.js vs. .NET Core vs. Django: a “celebrity face-off”. (2024). framework training. <https://www.frameworktraining.co.uk/news-insights/spring-boot-vs-node-js-versus-dot-net-core-vs-django/>

Evolution in Agriculture: How Automation is Modernizing Farming Practices - The Agrotech Daily. The Agrotech Daily. (2023). TAD News Desk. <https://theagrotechdaily.com/the-impact-of-automation-on-agriculture/>

Tapia, E. (2023). Bases de datos objeto-relacionales: Concepto práctico. *Scientia Omnibus Portus*, 3(5), 2.

Tecnología Agrícola Inteligente con Arduino. (s/f). CircularFab. <https://circularfab.es/tecnologia-agricola-inteligente-con-arduino/>

The rise of autonomous farms: How technology is revolutionizing agriculture. (2023, July 4). World Economic Forum. <https://www.weforum.org/agenda/2023/07/farm-automation-technology-revolutionizing-agriculture/>

Microcontrolador: ¿Qué es y para qué sirve? (2024). Universidad Europea. <https://universidadeuropea.com/blog/que-es-microcontrolador/>

Urrutia, D. (2023). Qué es Visual Studio | Definición, historia y ventajas. Arimetrics. <https://www.arimetrics.com/glosario-digital/visual-studio>

Venema, M. (2024). What is Extreme Programming (XP)? Its values, principles, and practices. NimbleWork. <https://www.nimblework.com/agile/extreme-programming-xp/#:~:text=Extreme%20programming%20is%20a%20software,to%20evolving%20and%20changing%20requirements>.

Ventajas y desventajas de MVC en ASP.NET Core. (2023). Estrada Web Group. <https://estradawebgroup.com/Post/Ventajas-y-desventajas-de-MVC-en-ASP-NET-Core/20649>

VSTest@2 - Visual Studio Test v2 task. (2024). Microsoft.com. <https://learn.microsoft.com/en-us/azure/devops/pipelines/tasks/reference/vstest-v2?view=azure-pipelines>

What's new in .NET 6. (2023). Microsoft.com. <https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-6>

Wiesman, Z. (2024). Head-to-head performance comparison of Nginx and Apache. Pressable. <https://pressable.com/blog/head-to-head-performance-comparison-of-nginx-and-apache/>

Zamora Boza, S., Espinoza Herrera, X., San Andrés Reyes, P., & Moreno Silva, A. (2021). Sistemas de innovación agrícola: una mirada a la situación del sector agrícola ecuatoriano: Agricultural innovation systems: a look to the situation of the ecuadorian agricultural sector. *Revista Científica ECOCIENCIA*, 8, 237–254. <https://doi.org/10.21855/ecociencia.80.647>

8. Anexos

8.1. Sección #1: Acta de Reuniones

8.1.1. Acta #1

Fecha: 04 de agosto de 2024

Lugar: Sala virtual

Hora: 10:00 AM

Participantes:

- Ing. Víctor Alcívar Rodríguez (Beneficiario – Hacienda Orillas del Sol)
- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos de la sesión:

- Escuchar los problemas o dificultades que tenga el beneficiario.
- Presentar del estado actual del proyecto.
- Discutir sobre los requerimientos de la Hacienda Orillas del Sol.
- Proponer ajustes correspondientes a la información proyectada.
- Agendar una reunión acorde a la disponibilidad del cliente.

Desarrollo:

- El beneficiario nos hace saber que cuenta con un número limitado de trabajadores, lo que representa un desafío para mantener la operatividad y eficiencia de la hacienda. Dado que no es viable aumentar la plantilla de empleados, surge la necesidad de implementar soluciones tecnológicas que permitan agilizar las tareas de

mantenimiento y mejorar la eficiencia general con los recursos humanos actuales. Nuestro proyecto está diseñado para abordar precisamente esta problemática, ya que, mediante la automatización del monitoreo y el ajuste de parámetros ambientales, se reducirá la carga de trabajo manual, permitiendo a los trabajadores enfocar sus esfuerzos en otras áreas críticas sin descuidar el rendimiento de la hacienda.

- Por nuestra parte es fundamental establecer objetivos claros y prácticos que guiarán el desarrollo inicial. El principal objetivo es implementar un sistema que permita automatizar y mejorar el monitoreo y ajuste de los parámetros ambientales en la hacienda, lo que incrementará la eficiencia del trabajo realizado con el personal disponible. Otro objetivo es garantizar que el sistema sea fácil de usar para los empleados, con una interfaz intuitiva y accesible, reduciendo la necesidad de supervisión constante. También se espera integrar el sistema de manera eficiente con las herramientas y procesos actuales de la hacienda, sin interrumpir las operaciones diarias. Aparte, el sistema deberá ser flexible y escalable, permitiendo ajustes futuros sin requerir grandes cambios en su estructura base.
- El beneficiario pidió conocer más de la materia para la siguiente sesión, por el momento no tiene requerimiento alguno.
- La próxima reunión será creada acorde a la disponibilidad del beneficiario.

Evidencias:



8.1.2. Acta #2

Fecha: 18 de septiembre de 2024

Lugar: Sala virtual

Hora: 16:00 PM

Participantes:

- Ing. Víctor Alcívar Rodríguez (Beneficiario – Hacienda Orillas del Sol)
- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

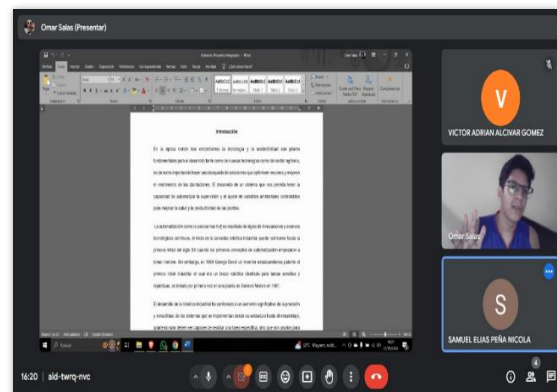
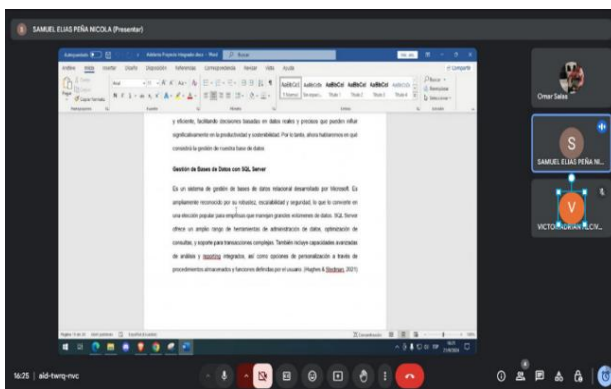
Objetivos de la sesión:

- Presentar del estado actual del proyecto.
- Discutir sobre los requerimientos de la Hacienda Orillas del Sol.
- Agendar una reunión acorde a la disponibilidad del cliente.

Desarrollo:

- Comenzamos la reunión explicando las bases conceptuales del proyecto, se fueron explicados todos las herramientas y utilidades que van a ser usadas durante el desarrollo. Empezaríamos el desarrollo con la parte de toma de muestras de la mano de Arduino y el uso de sensores, la creación del servidor en Amazon AWS y posteriores pruebas de envío de datos.
- El beneficiario pidió que se le cree una cuenta de Gmail para que ese correo sea usado en la cuenta de Amazon AWS y para en un futuro el resto de herramientas.
- La próxima reunión será creada acorde a la disponibilidad del beneficiario.

Evidencias:



8.1.3. Acta #3

Fecha: 15 de octubre de 2024

Lugar: Sala virtual

Hora: 16:00 PM

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)
- Fadel Ibáñez (Ingeniero en Sistemas - Consultor)

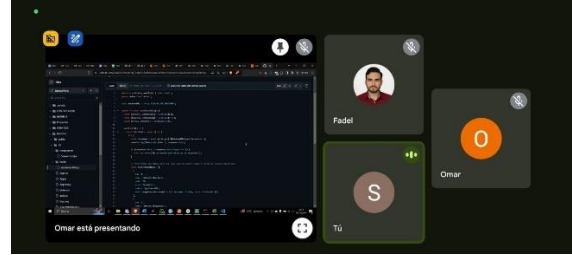
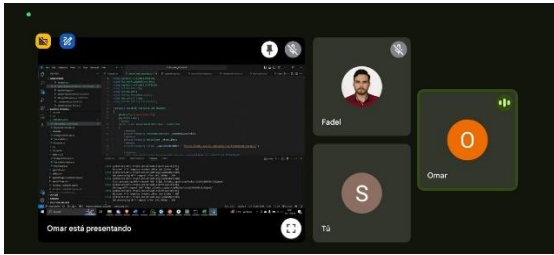
Objetivos de la sesión:

Configuración de los servicios, controladores y modelos, creación de componentes, hooks y métodos GET y POST para el manejo de datos y configuración de zapier para Google Sheets.

Desarrollo:

En la reunión de avances, se explicó el proceso de configuración de los servicios, controladores y modelos en el backend, asegurando que cada componente estuviera diseñado para interactuar correctamente con la base de datos. Luego, se presentó la creación de componentes y hooks en el frontend, destacando la implementación de métodos GET y POST para el manejo de datos entre el frontend y el backend. Finalmente, se detalló la configuración de Zapier para automatizar el envío de datos a Google Sheets, permitiendo registrar información en tiempo real. Durante la reunión, se revisaron los avances realizados en cada área y se discutieron ajustes para optimizar la integración y el flujo de datos.

Evidencias:



8.1.4. Acta #4

Fecha: 15 de octubre de 2024

Lugar: Sala virtual

Hora: 21:30 PM

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)
- Ing. Víctor Alcívar Rodríguez (Beneficiario - Hacienda Orillas del Sol)
- Fadel Ibáñez (Ingeniero en Sistemas - Consultor)

Objetivos de la sesión:

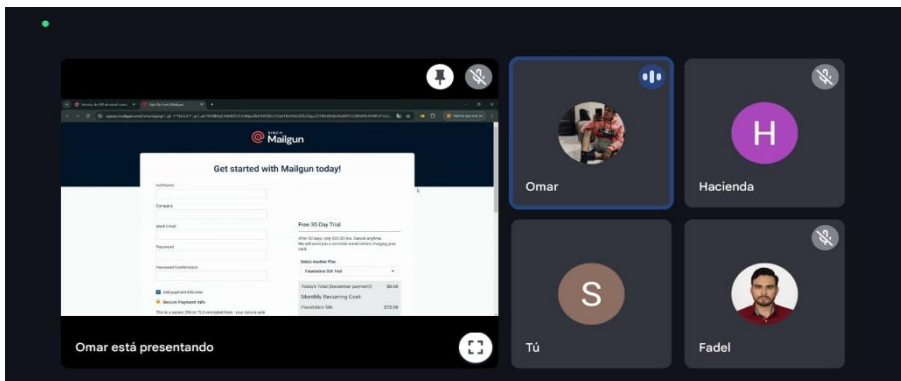
- Evaluar y seleccionar dominios potenciales para la página web del proyecto.
- Discutir opciones de servicios de correo electrónico para la comunicación eficiente con usuarios y colaboradores.
- Explorar servicios de hosting y servidores que soporten las necesidades técnicas del proyecto para su futura implementación.

Desarrollo:

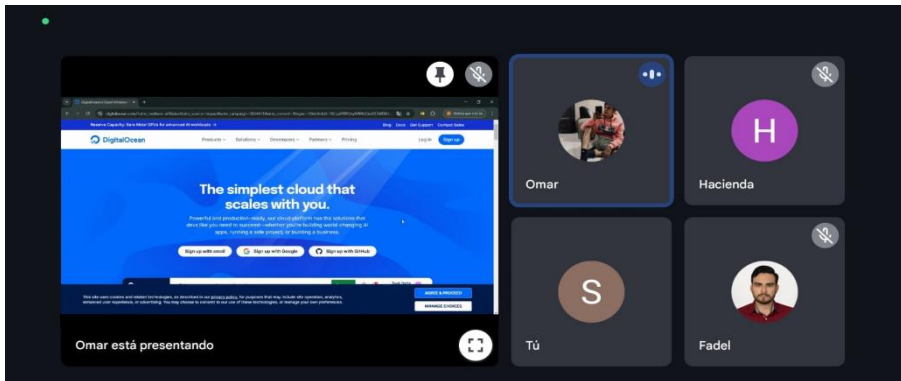
- Fueron presentadas una lista de posibles dominios para la página web del proyecto, discutiendo las ventajas y desventajas de cada opción con respecto a la facilidad de recordación por parte de los usuarios. Se evaluaron diferentes proveedores de servicios de correo electrónico, considerando aspectos como la fiabilidad, el costo y la integración con otras herramientas utilizadas en el proyecto. Y, por último, se exploraron varias opciones de servicios de hosting y servidores. Se discutieron especificaciones técnicas necesarias para la implementación del proyecto, como capacidad de almacenamiento, ancho de banda y soporte para tecnologías específicas. Se acordó preparar un análisis más detallado de costos y beneficios para la próxima reunión.

Evidencias:

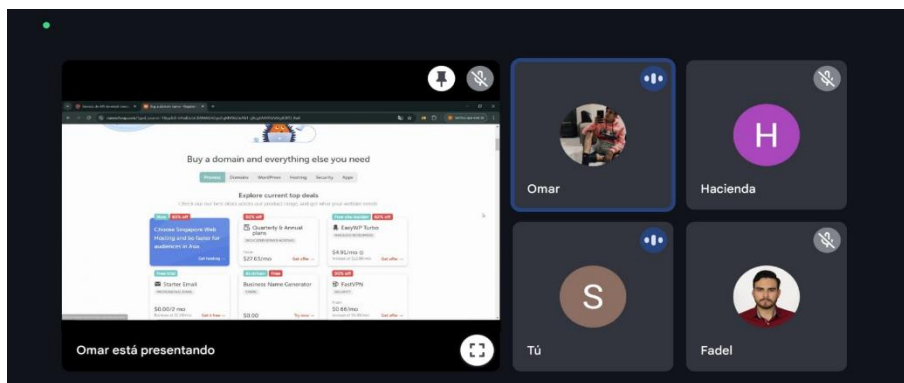
- Se eligió mailgun para el servicio de mail ya que tiene precios más accesibles y tiene una prueba gratuita de un mes que permite enviar hasta 100 mails por día.



- Por recomendación del consultor se consideró usar digital ocean para adquirir el servidor ya que él cuenta con desarrollos que están subidos en servidores de estos proveedores.



- Y previamente también hemos usado esta plataforma para registrar dominios ya que por ciertas temporadas los subdominios tienen descuentos a buenos precios, así mismo para servicios de hosting, por estas razones llegamos a la conclusión de usar Namecheap.



8.1.5. Acta #5

Fecha: 1 de noviembre de 2024

Lugar: Sala virtual

Hora: 20:15 PM

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

- Ing. Víctor Alcívar Rodríguez (Beneficiario - Hacienda Orillas del Sol)

Objetivos de la sesión:

- Informar al beneficiario sobre los costos de las herramientas tecnológicas seleccionadas para el proyecto.
- Presentar el diseño de la página web y sus funcionalidades.
- Discutir el diseño y costo estimado de construcción del carrito hecho con Arduino.

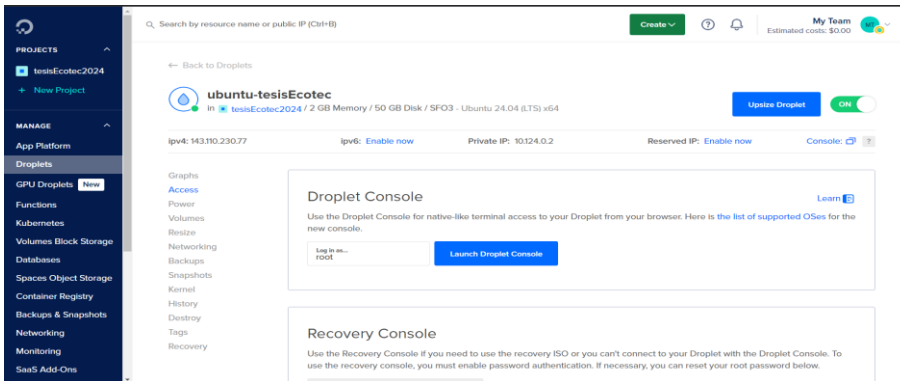
Desarrollo:

- Se notificó al beneficiario sobre la compra de un servidor en Digital Ocean, que tiene un costo de \$12 mensuales y ofrece 2GB de RAM y 50GB de espacio de almacenamiento. Además, de la adquisición del dominio haciendaorillas.live por un precio de \$3.5. Se discutieron las ventajas de tener un dominio propio y cómo esto podría mejorar la visibilidad y el acceso al sitio web del proyecto.

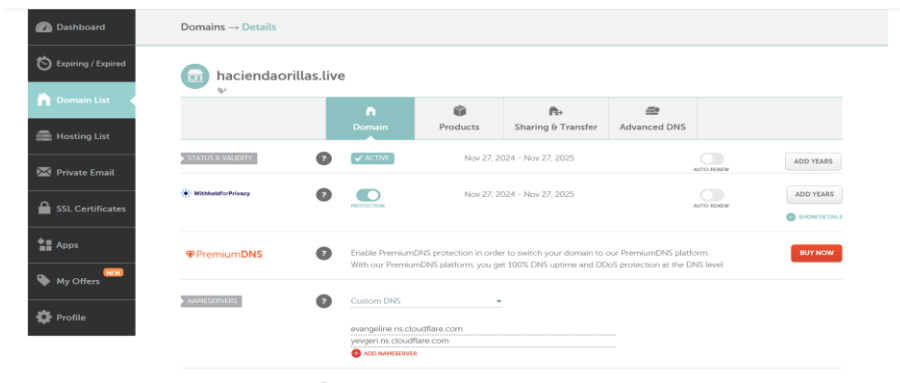
Se demostró el diseño preliminar de la página web, incluyendo una demostración de las funcionalidades principales, se presentó del diseño y el costo estimado para la construcción del carrito hecho con Arduino, diseñado para mejorar la logística interna de la hacienda. Se detallaron los componentes necesarios y se proporcionó un presupuesto preliminar para su construcción y puesta en marcha.

Evidencias:

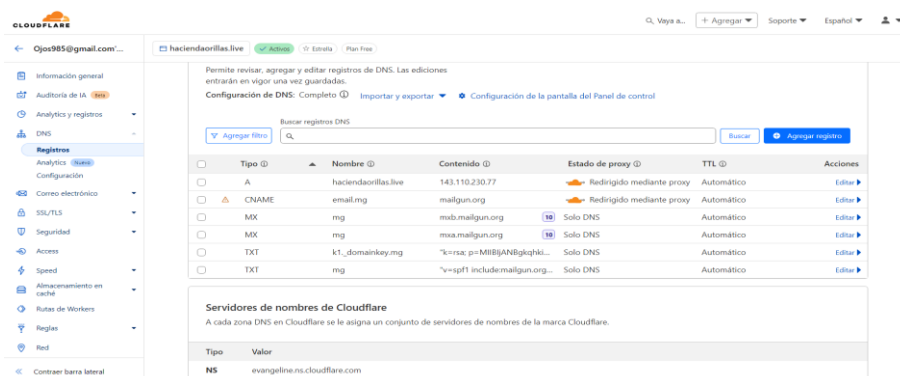
- Compra y configuración del servidor para alojar nuestro proyecto.



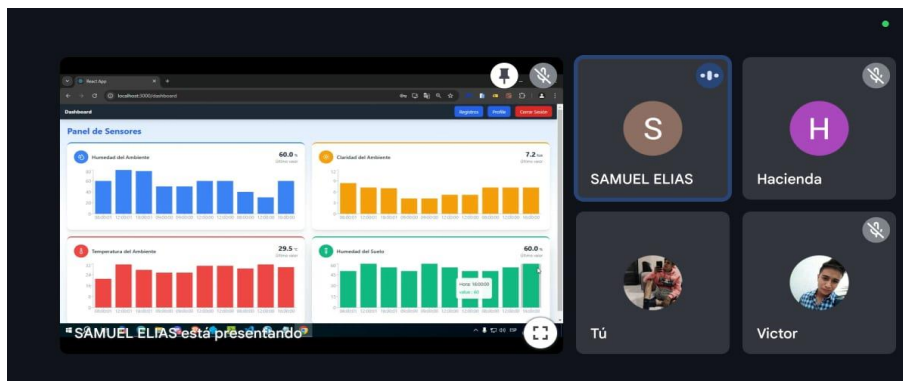
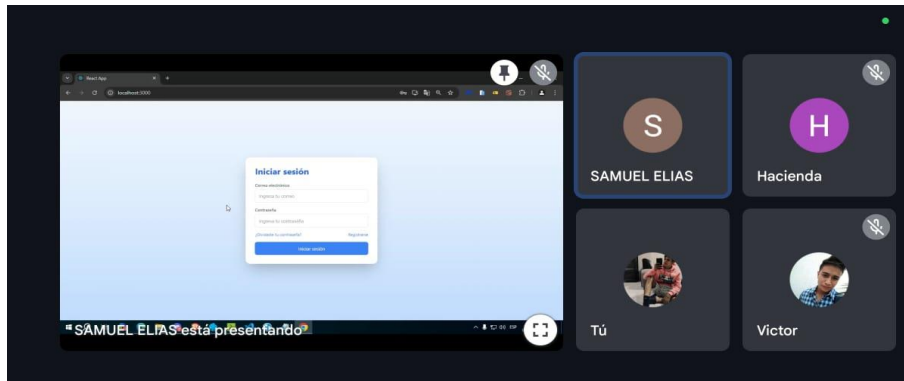
- Compra y configuración del dominio en Namecheap.



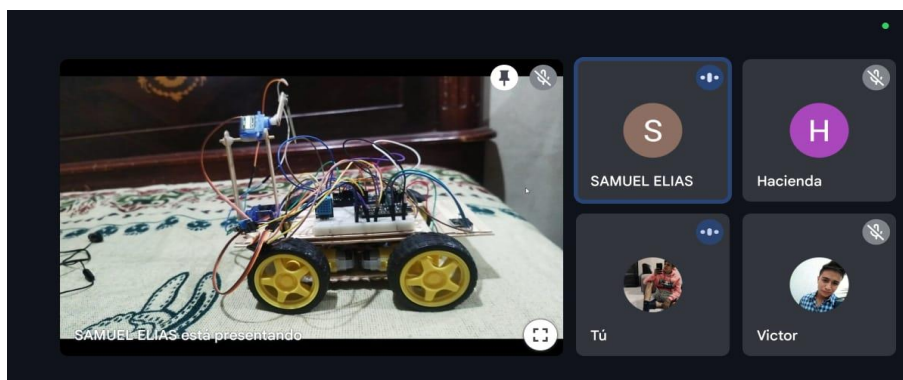
- Una vez comprado el dominio se configuro los records para que sea funcional el dominio con el correo, por recomendación del consultor se agregó el dominio a CloudFlare para mayor seguridad y uso de certificados a la hora de levantar el frontend con otra herramienta llamada Nginx.



- Demostración de la página y sus funcionalidades.



- Demostración del diseño y funcionamiento del carrito hecho con Arduino.



8.1.6. Acta #6

Fecha: 26 de noviembre de 2024

Lugar: Sala virtual

Hora: 20:00 PM

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)
- Ing. Víctor Alcívar Rodríguez (Beneficiario - Hacienda Orillas del Sol)

Objetivos de la sesión:

- Planificar la subida del proyecto al servidor.
- Definir la fecha de entrega posible del producto final.
- Abordar posibles soluciones debido al déficit energético.

Desarrollo:

- Se discutió sobre la subida del proyecto al servidor. Se explicó el proceso y las etapas necesarias para asegurar que el proyecto esté operativo y accesible en línea, considerando la implementación de medidas de seguridad y pruebas de funcionamiento.

El tema principal fue la planificación de la entrega del producto final. Debido a los cortes de energía frecuentes en la hacienda durante la mañana y gran parte de la tarde, se acordó que la entrega del proyecto se realizará de forma virtual. Esto incluirá una demostración en un entorno simulado con características similares a las de la hacienda para garantizar que el beneficiario pueda familiarizarse con el sistema y su funcionamiento sin interrupciones.

Se estableció que la posible entrega del producto final sería antes del 20 de diciembre de 2024, dependiendo de la estabilización de la situación energética en el país. Se acordó que una vez la energía sea confiable y no haya cortes, se procederá a implementar el proyecto de manera física en la hacienda.

8.2. Sección #2 Evidencias de Avance Técnico del Proyecto

8.2.1. Avance #1

Fecha: 20 de septiembre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos del Avance:

Selección y configuración de sensores, configuración de Arduino y MongoDB.

Descripción de Actividades:

N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Investigación y selección de sensores	Se investigaron y seleccionaron los sensores adecuados para medir temperatura, humedad del ambiente, humedad del suelo y luminosidad.	20/09/2024	22/09/2024
2	Configuración de Arduino	Se hizo una investigación de las librerías y todos los componentes necesarios para que los sensores funcionen con el IDE.	20/09/2024	21/09/2024
3	Configuración de MongoDB	Se creo y levanto tanto como el servidor de MongoDB como su	20/09/2024	20/09/2024

administrador de bases
MongoDB Compass.

Resultados:

Se investigaron y seleccionaron sensores para medir temperatura, humedad, humedad del suelo y luminosidad. Luego, se configuró Arduino con las librerías necesarias para los sensores. Finalmente, se instaló y configuró el servidor MongoDB junto con su administrador MongoDB Compass para almacenar datos del sistema.

Evidencias:

- Sensores y componentes iniciales adquiridos.



- Instalación de dependencias y creación del código.

```

sketch_oct09a
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <ESP8266HTTPClient.h>

#define DHTPIN 2
#define DHTTYPE DHT11
#define PH_PIN A0

DHT dht(DHTPIN, DHTTYPE);
const char* ssid = "PlantaRejalesEnriquez_Xtrix";
const char* password = "M19c1915";
const char* serverName = "http://hacienda-orilla.com/api/plantation";

void setup() {
  Serial.begin(115200);
  dht.begin();
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando a Wi-Fi...");
  }

  Serial.println("Conectado a Wi-Fi");
}

void loop() {
  float humedad = dht.readHumidity();
  float temperatura = dht.readTemperature();
  int pHValue = analogRead(PH_PIN);

  if (!isnan(humedad) || !isnan(temperatura)) {
    Serial.println("Fallo al leer del sensor DHT");
    return;
  }
}

```

- Levantamiento y creación de requerimientos para MongoDB.

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' panel lists several databases, with 'dbTest2024' selected. Under 'dbTest2024', the 'Usuarios' collection is highlighted. The main area displays the 'Usuarios' collection with 0 documents. A search bar is present with the placeholder text 'Type a query: { field: 'value' } or Generate query'. Below the search bar are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. At the bottom, a message states 'This collection has no data' and provides an 'Import Data' button.

```

C:\Users\LT Technology>mongod --dbpath="C:\data\db"
{"t":{"$date":"2024-11-10T18:22:58.123-05:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-11-10T18:22:58.124-05:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2024-11-10T18:22:58.125-05:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related parameters: [\"tcpFastOpenServer\", \"tcpFastOpenClient\", \"tcpFastOpenQueueSize\"]}
{"t":{"$date":"2024-11-10T18:22:58.127-05:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":25},"isInternalClient":true}}}
{"t":{"$date":"2024-11-10T18:22:58.127-05:00"},"s":"I", "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
{"t":{"$date":"2024-11-10T18:22:58.128-05:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting", "attr":{"pid":8544, "port":27017, "dbPath":"C:/data/db", "architecture":"64-bit", "host":"DESKTOP-VQJK48U"}}
{"t":{"$date":"2024-11-10T18:22:58.130-05:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":"Target operating system minimum version", "attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2024-11-10T18:22:58.130-05:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":"Build Info", "attr":{"buildInfo":{"version":"8.0.1", "gitVersion":"fcbe67d668ffff5a370e2d87b9b1f74bc11bb7b94", "modules":[], "allocator":"tcmalloc-gperf", "environment":{"distmod":"windows", "distarch":"x86_64", "target_arch":"x86_64"}}}}
{"t":{"$date":"2024-11-10T18:22:58.131-05:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Operating System", "attr":{"os":{"name":"Microsoft Windows 10", "version":"10.0 (build 19045)}}}}
{"t":{"$date":"2024-11-10T18:22:58.131-05:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten","msg":"Options set by command line", "attr":{"options":{"storage":{"dbPath":"C:\\data\\db"}}}}
{"t":{"$date":"2024-11-10T18:22:58.137-05:00"},"s":"I", "c":"STORAGE", "id":22270, "ctx":"initandlisten","msg":"Storage engine to use detected by data files", "attr":{"dbpath":"C:/data/db", "storageEngine":"wiredTiger"}}
{"t":{"$date":"2024-11-10T18:22:58.138-05:00"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"initandlisten","msg":"Opening WiredTiger", "attr":{"config":"create,cache_size=7598M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,remove=true,path=journal,compressor=snappy).builtin_extension_config=(z

```

8.2.2. Avance #2

Fecha: 2 de octubre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos del Avance:

Creación de proyecto en React y pruebas iniciales.

Descripción de Actividades:

N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Creación y levantamiento del proyecto en React	Se levanto el frontend usando laragon ya que este viene con NodeJS que es necesario en un proyecto de React.	24/09/2024	27/09/2024
2	Configuración de dependencias de React	Se necesito instalar Axios, Date-Fns, Lucide-React y ReCharts. Son dependencias necesarias para crear graficas con React.	24/09/2024	26/09/2024
3	Pruebas iniciales de sensores	Se realizaron pruebas dentro de Arduino para asegurar el funcionamiento de los sensores.	28/09/2024	02/10/2024

Resultados:

Se levantó el frontend del proyecto en React usando Laragon y NodeJS. Se instalaron dependencias clave como Axios, Date-Fns, Lucide-React y ReCharts para gestionar datos y crear gráficas. Además, se realizaron pruebas iniciales en Arduino para verificar el funcionamiento de los sensores.

Evidencias:

- Creación y levantamiento del proyecto en React.

```

EXPLORER
BACKEND_TESIS2024
> CONTROLADOR
  frontend
    node_modules
    public
    src
      components
        SensorCard.jsx
      hooks
        useSensorData.js
        # App.css
        # App.js
        # AppTest.js
        # index.css
        # index.js
        logo.svg
        reportWebVitals.js
        setupTests.js
    .env
    .gitignore
    package-lock.json
    package.json
    postcss.config.js
    README.md
    tailwind.config.js

index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6
7 const root = ReactDOM.createRoot(document.getElementById('root'));
8 root.render(
9   <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
13
14 // If you want to start measuring performance in your app, pass a function
15 // to log results (for example: reportWebVitals(console.log))
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17 reportWebVitals();
18

TERMINAL
PS C:\laragon\www\tesis\BackEnd_Tesis2024> git branch
* SamuelPeña
master

```

```

C:\laragon\www
λ cd tesis\

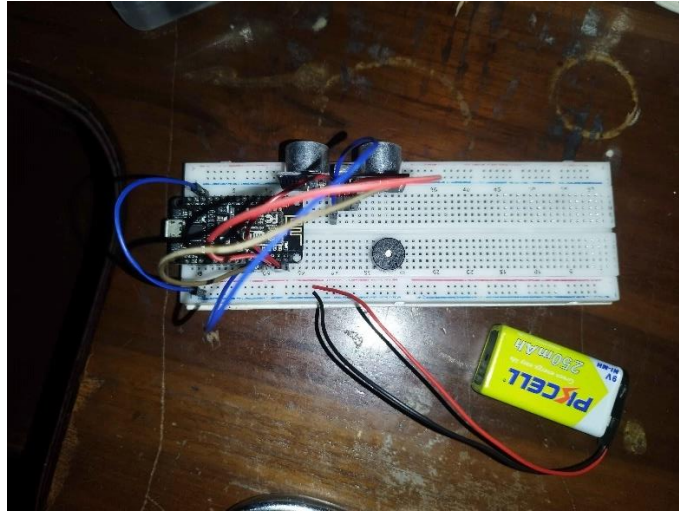
C:\laragon\www\tesis
λ cd BackEnd_Tesis2024\

C:\laragon\www\tesis\BackEnd_Tesis2024(SamuelPeña -> origin)
λ cd frontend\

C:\laragon\www\tesis\BackEnd_Tesis2024\frontend(SamuelPeña -> origin)
λ npm start
|

```

- Pruebas iniciales de sensores.



8.2.3. Avance #3

Fecha: 06 de octubre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos del Avance:

Conexión del módulo con nuestra red wifi, creación del backend con .Net Core, creación de repositorio en GitHub para alojar cambios en el proyecto e instalación de dependencias para usar MongoDB con nuestro proyecto.

Descripción de Actividades:

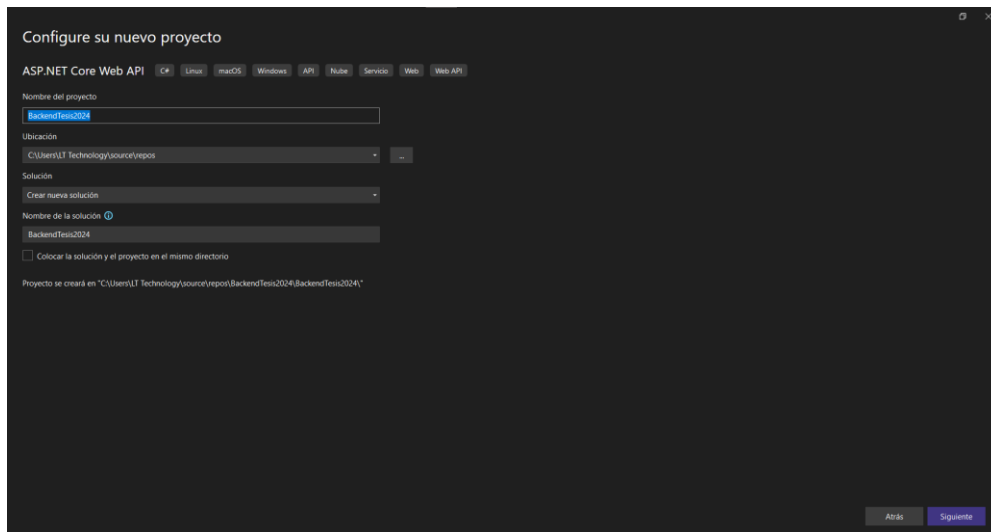
N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Creación proyecto .Net Core	Se creó el proyecto .Net Core usando Visual Studio para obtener una plantilla con todo lo básico.	03/10/2024	05/10/2024
2	Creación del repositorio	Se creó un repositorio en GitHub desde Visual Studio para manejo de versiones.	03/10/2024	04/10/2024
3	Instalación de dependencias y pruebas de conectividad	Se necesitó instalar Mongo Driver para usar la base de datos en nuestro proyecto además de otras para asegurar el levantamiento de nuestro backend.	04/10/2024	06/10/2024

Resultados:

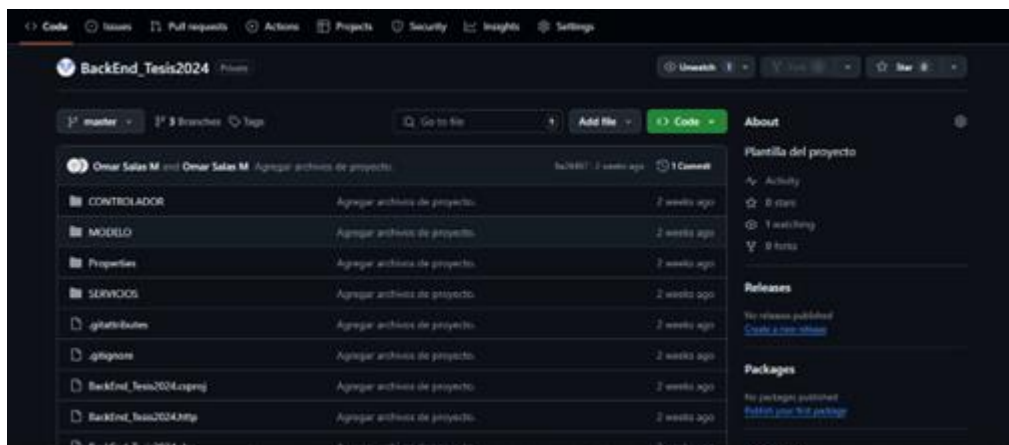
Se creó el proyecto en .NET Core utilizando Visual Studio, estableciendo una plantilla básica. Luego, se configuró un repositorio en GitHub para el control de versiones. Finalmente, se instalaron dependencias, incluyendo Mongo Driver, y se realizaron pruebas de conectividad para asegurar el correcto funcionamiento del backend.

Evidencias:

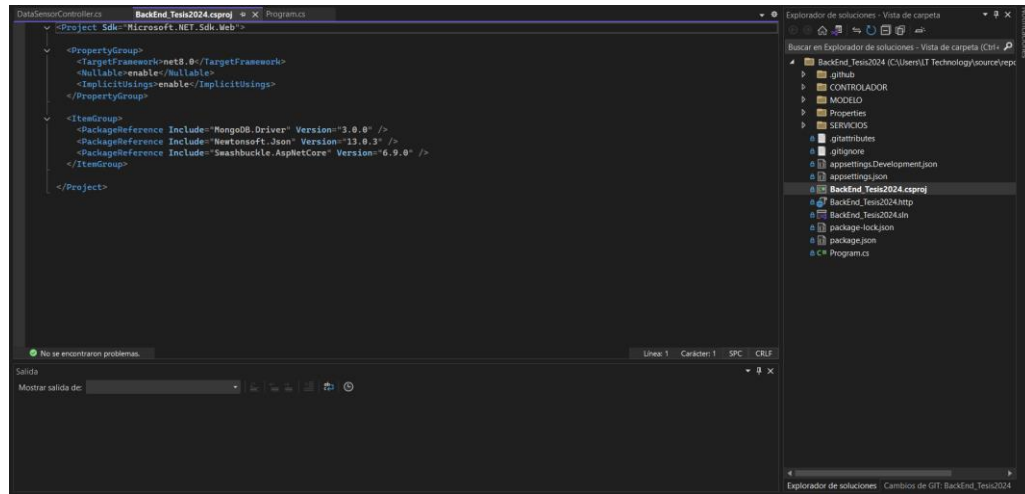
- Creación del proyecto en Visual Studio.



- Repositorio de GitHub creado.



- Dependencias instaladas



8.2.4. Avance #4

Fecha: 12 de octubre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)
- Fadel Ibáñez (Ingeniero en Sistemas - Consultor)

Objetivos del Avance:

Sincronizar el proyecto junto con la base de datos, pruebas de intercambio de datos, optimización de comunicación.

Descripción de Actividades:

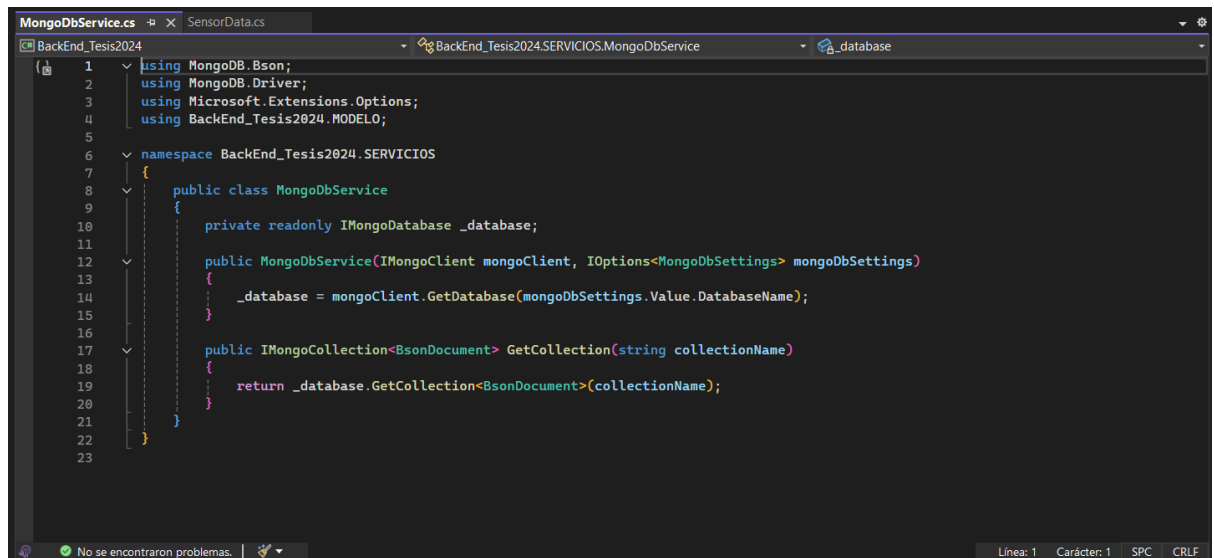
N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Sincronización entre la base y el proyecto .Net Core	Se creo un modelo para usar una cadena de conexión y conectarnos a la base gracias a MongoDB Driver.	06/10/2024	07/10/2024
2	Pruebas de intercambio de datos	Se creo un formulario sencillo con la consola donde se insertaban datos quemados dentro de la base desde el backend.	07/10/2024	09/10/2024
3	Optimización de comunicación	Se utilizo todo tipo de mejoras para evitar retrasos dentro de las respuestas del backend hacia la base de datos.	10/10/2024	12/10/2024

Resultados:

Se sincronizó la base de datos con el proyecto .NET Core mediante MongoDB Driver. Se realizaron pruebas de intercambio de datos con un formulario en consola para verificar la inserción en la base. Finalmente, se optimizó la comunicación entre el backend y la base de datos para evitar retrasos en las respuestas.

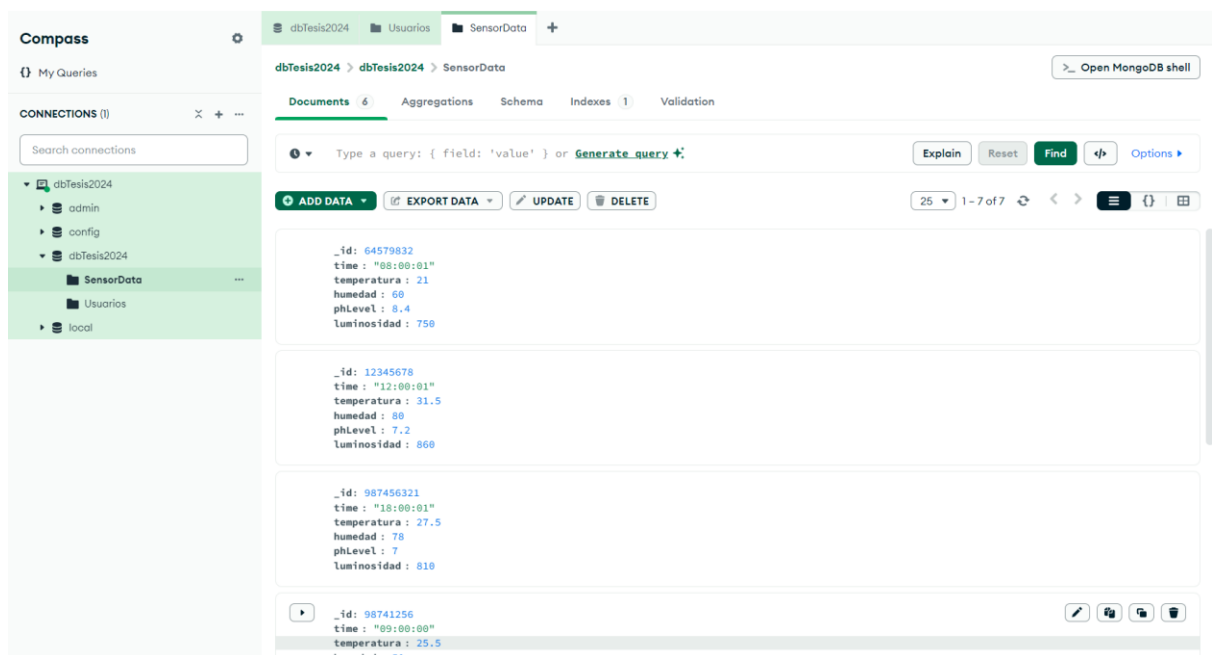
Evidencias:

- Sincronización de la base de datos con el backend.



```
1 using MongoDB.Bson;
2 using MongoDB.Driver;
3 using Microsoft.Extensions.Options;
4 using BackEnd_Tesis2024.MODELO;
5
6 namespace BackEnd_Tesis2024.SERVICIOS
7 {
8     public class MongoDBService
9     {
10         private readonly IMongoDatabase _database;
11
12         public MongoDBService(IMongoClient mongoClient, IOption<MongoDbSettings> mongoDbSettings)
13         {
14             _database = mongoClient.GetDatabase(mongoDbSettings.Value.DatabaseName);
15         }
16
17         public IMongoCollection<BsonDocument> GetCollection(string collectionName)
18         {
19             return _database.GetCollection<BsonDocument>(collectionName);
20         }
21     }
22 }
23
```

- Intercambio de datos entre la base y el proyecto.



_id	time	temperatura	humedad	phLevel	luminosidad
64579832	"08:00:01"	21	60	8.4	750
12345678	"12:00:01"	31.5	80	7.2	860
987456321	"18:00:01"	27.5	78	7	810
98741256	"09:00:00"	25.5	50		

8.2.5. Avance #5

Fecha: 12 de octubre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)
- Fadel Ibáñez (Ingeniero en Sistemas - Consultor)

Objetivos del Avance:

Configuración de los servicios, controladores y modelos, creación de componentes, hooks y métodos GET y POST para el manejo de datos y configuración de zapier para Google Sheets.

Descripción de Actividades:

N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Configuración de componentes principales de MVC	Se crearon modelos para alojar los datos, controladores para manejar los métodos POST y GET de nuestro proyecto, y los servicios para reciclar los métodos.	11/10/2024	25/10/2024
2	Creación de métodos GET y POST con JS dentro de nuestro frontend	Se crearon estos métodos para manejar la inserción y lectura de datos en conjunto con nuestro backend.	15/10/2024	21/10/2024
3	Configuración de Zapier para llevar registros	Se creo trigger para captar el JSON con los datos tomados de las plantaciones, con un correo que se creó previamente se inicia sesión dentro de Zapier para poder asignar una hoja de	15/10/2024	19/10/2024

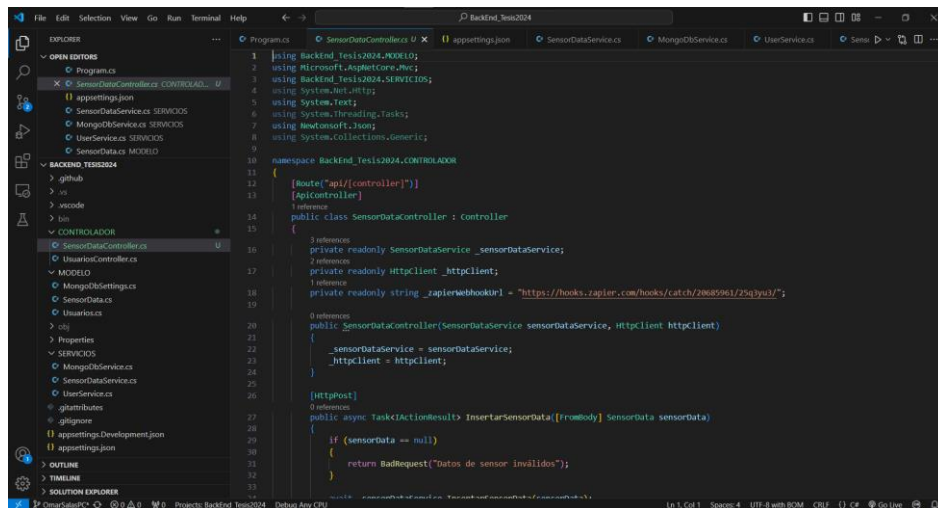
cálculo a la que le lleguen los datos.

Resultados:

Se configuraron los componentes principales de MVC, incluyendo modelos, controladores y servicios para manejar métodos POST y GET. Luego, se implementaron estos métodos en el frontend para facilitar el intercambio de datos con el backend. Además, se configuró Zapier para registrar datos de las plantaciones, activando un trigger que envía información a una hoja de cálculo.

Evidencias:

- Configuración de componentes principales.



```

1  using Backend_Tesis2024.MODELO;
2  using Microsoft.AspNetCore.Mvc;
3  using Backend_Tesis2024.SERVICIOS;
4  using System.Net;
5  using System.Text;
6  using System.Threading.Tasks;
7  using Newtonsoft.Json;
8  using System.Collections.Generic;
9
10 namespace Backend_Tesis2024.CONTROLADOR
11 {
12     [Route("api/[controller]")]
13     [ApiController]
14     public class SensorDataController : Controller
15     {
16         // 3 references
17         private readonly SensorDataService _sensorDataService;
18         // 2 references
19         private readonly HttpClient _httpClient;
20         // 1 reference
21         private readonly string _zapierWebhookUrl = "https://hooks.zapier.com/hooks/catch/20685961/23q3yu3/";
22
23         // 0 references
24         public SensorDataController(SensorDataService sensorDataService, HttpClient httpClient)
25         {
26             _sensorDataService = sensorDataService;
27             _httpClient = httpClient;
28         }
29
30         // 0 references
31         [HttpPost]
32         public async Task InsertarSensorData([FromBody] SensorData sensorData)
33         {
34             if (sensorData == null)
35             {
36                 return BadRequest("Datos de sensor inválidos");
37             }
38         }
39     }
40 }

```

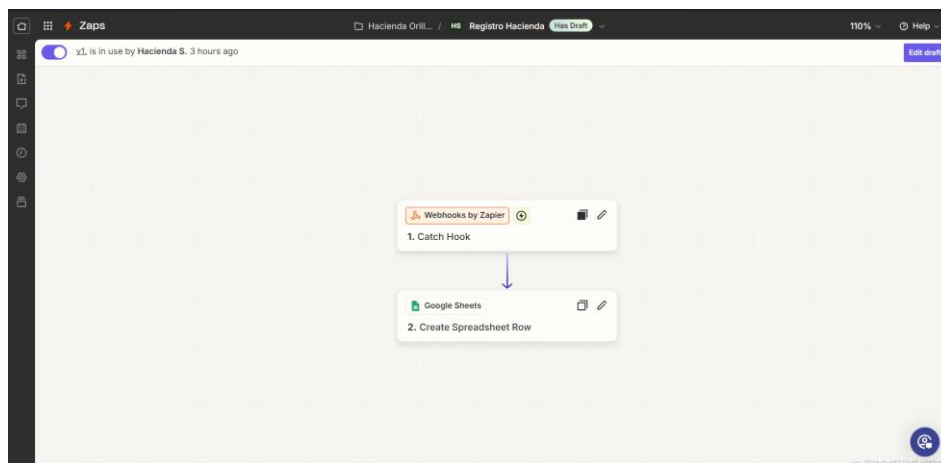
- Js para manejo de métodos post y get.

```

1  import React from 'react';
2  import { BarChart, Bar, ResponsiveContainer, XAxis, YAxis, Tooltip } from 'recharts';
3  import { Droplets, Sun, Thermometer, TestTube } from 'lucide-react';
4
5  const iconMap = {
6    Droplets,
7    Sun,
8    Thermometer,
9    TestTube,
10 };
11
12 export function SensorCard({ sensor }) {
13   const Icon = iconMap[sensor.icon];
14
15   return (
16     <div className="bg-white rounded-xl shadow-lg p-6 hover:shadow-xl transition-shadow">
17       <div className="flex items-center justify-between mb-4">
18         <div className="flex items-center gap-3">
19           <div className="p-2 rounded-lg ${sensor.color}">
20             <Icon className="w-6 h-6 text-white" />
21           </div>
22           <h3 className="text-lg font-semibold text-gray-800">{sensor.name}</h3>
23         </div>
24         <div className="text-right">
25           <p className="text-2xl font-bold text-gray-900">
26             {(sensor.data[sensor.data.length - 1].value ?? 0).toFixed(1)}
27           <span className="text-sm ml-1 text-gray-500">{sensor.unit}</span>
28           </p>
29           <p className="text-sm text-gray-500">Latest Value</p>
30         </div>
31       </div>
32
33       <div className="h-[200px] mt-4">

```

- Configuración de Zapier.



8.2.6. Avance #6

Fecha: 29 de octubre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos del Avance:

Pruebas de estabilidad del sistema, resolución de errores de transmisión y optimización de flujo de datos.

Descripción de Actividades:

N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Pruebas de estabilidad del sistema	Se realizaron pruebas exhaustivas para garantizar la estabilidad del sistema bajo diferentes condiciones de carga.	22/10/2024	24/10/2024
2	Resolución de errores de transmisión	Se identificaron y corrigieron errores en la transmisión de datos entre el frontend y el backend.	21/10/2024	25/10/2024
3	Optimización de flujo de datos	Se optimizó el flujo de datos para mejorar la velocidad y reducir el tiempo de respuesta del sistema.	25/10/2024	29/10/2024

Resultados:

Se realizaron pruebas de estabilidad para asegurar el rendimiento del sistema bajo distintas cargas. Se identificaron y solucionaron errores en la transmisión de datos entre frontend y

backend, mejorando la comunicación. Finalmente, se optimizó el flujo de datos, reduciendo el tiempo de respuesta y aumentando la eficiencia del sistema.

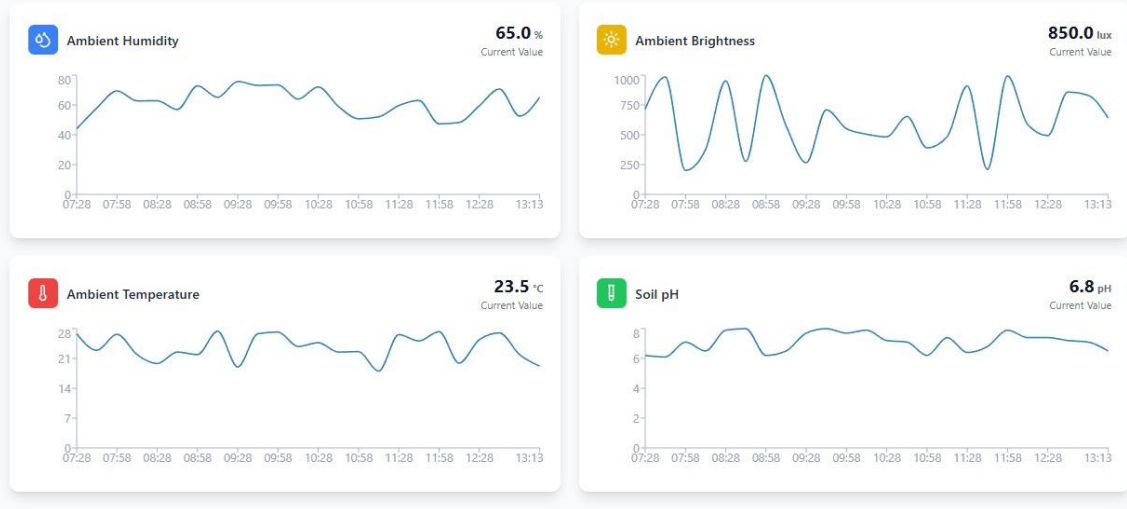
Evidencias:

- Pruebas de estabilidad de sistema.

```
export function SensorCard({ sensor }) {
  const Icon = iconMap[sensor.icon];

  return (
    <div className="bg-white rounded-xl shadow-lg p-6 hover:shadow-xl transition-shad
      <div className="flex items-center justify-between mb-4">
        <div className="flex items-center gap-3">
          <div className={`p-2 rounded-lg ${sensor.color}`}>
            <Icon className="w-6 h-6 text-white" />
          </div>
          <h3 className="text-lg font-semibold text-gray-800">{sensor.name}</h3>
        </div>
        <div className="text-right">
          <p className="text-2xl font-bold text-gray-900">
            {sensor.value.toFixed(1)}
            <span className="text-sm ml-1 text-gray-500">{sensor.unit}</span>
          </p>
          <p className="text-sm text-gray-500">Current Value</p>
        </div>
      </div>
      <div className="h-[200px] mt-4">
        <ResponsiveContainer width="100%" height="100%">
          <LineChart data={sensor.data}>
            <XAxis
              dataKey="timestamp"
              tickFormatter={(time) => format(new Date(time), 'HH:mm')}
              stroke="#94a3b8"
            />
            <YAxis stroke="#94a3b8" />
            <Tooltip
```


Dashboard de Monitoreo Ambiental



- Corrección de errores de transmisión y optimización de flujo de datos.

```

1  using BackEnd_Tesis2024.MODELO;
2  using BackEnd_Tesis2024.SERVICIOS;
3  using Microsoft.Extensions.Options;
4  using MongoDB.Driver;
5
6  var builder = WebApplication.CreateBuilder(new WebApplicationOptions
7  {
8      ContentRootPath = AppContext.BaseDirectory
9  });
10
11 builder.WebHost.UseUrls("http://25.45.246.188:5000");
12
13 builder.Services.AddCors(options =>
14 {
15     options.AddPolicy("PermitirReact",
16         builder => builder.AllowAnyOrigin()
17             .AllowAnyMethod()
18             .AllowAnyHeader());
19 });
20
21 builder.Services.AddControllers();
22
23 builder.Services.Configure<MongoDbSettings>(builder.Configuration.GetSection("MongoDbSettings"));
24 builder.Services.AddSingleton<IMongoClient, MongoClient>(sp =>
25 {
26     var settings = sp.GetRequiredService<Options<MongoDbSettings>>().Value;
27     return new MongoClient(settings.ConnectionString);
28 });
29 builder.Services.AddScoped<UserService>();
30 builder.Services.AddScoped<SensorDataService>();
31 builder.Services.AddHttpClient();
32 var app = builder.Build();
33
34 app.UseCors("PermitirReact");
35 app.MapControllers();
36 app.Run();
  
```

8.2.7. Avance #7

Fecha: 2 de noviembre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos del Avance:

Implementación de pruebas unitarias es *validar el correcto funcionamiento* de cada componente clave del sistema frontend.

Descripción de Actividades:

Para garantizar el funcionamiento adecuado de los componentes desarrollados en React, se implementaron pruebas unitarias utilizando la herramienta Jest, una librería especializada en la validación de aplicaciones JavaScript. Las pruebas se centraron en los siguientes módulos críticos:

1. **SensorCard.test.js:** Validación de la correcta recepción y visualización de los datos de los sensores.
2. **LoginForm.test.js:** Prueba del formulario de inicio de sesión, verificando la funcionalidad de validación.
3. **Dashboard.test.js:** Comprobación de la visualización en tiempo real de los parámetros ambientales.

4. **useSensorData.test.js**: Validación del hook personalizado para el manejo de datos provenientes del backend.
5. **App.test.js**: Prueba global que asegura la correcta integración y funcionamiento del sistema completo.

Resultados:

La implementación de estas pruebas unitarias confirma que los componentes clave del frontend desarrollados en React cumplen con los requerimientos funcionales, permitiendo una integración adecuada con el backend en .NET Core y asegurando una visualización en tiempo real de los datos recopilados. Estas validaciones contribuyen a la *estabilidad del sistema*, facilitando la entrega de un producto funcional y listo para la fase de pruebas finales en condiciones reales.

Evidencias:

Las pruebas realizadas obtuvieron resultados satisfactorios, con un total de 5 suites ejecutadas y 7 pruebas aprobadas. A continuación, se presentan las evidencias de los resultados obtenidos:

1. Resultados de Ejecución de Pruebas Unitarias

```
C:\laragon\www\tesis_frontend\frontend(SamuelPeña -> origin)
λ npx jest
PASS src/components/__tests__/SensorCard.test.js
PASS src/components/__tests__/LoginForm.test.js
PASS src/components/__tests__/Dashboard.test.js
PASS src/hooks/__tests__/useSensorData.test.js
```

```
PASS src/App.test.js
Test Suites: 5 passed, 5 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        6.215 s
Ran all test suites.
```

8.2.8. Avance #8

Fecha: 10 de noviembre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos del Avance:

Optimización del código, pruebas finales en condiciones reales, evaluación y análisis de datos, y documentación de resultados.

Descripción de Actividades:

N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Optimización del código	Limpiar y optimizar el código para mejorar rendimiento y eficiencia.	30/10/2024	03/11/2024

2	Pruebas finales en condiciones reales	Probar el sistema en un entorno de plantación para verificar su rendimiento.	31/10/2024	06/11/2024
3	Evaluación y análisis de datos	Realizar análisis de tendencias de datos recopilados para evaluación.	01/11/2024	07/11/2024
4	Pruebas unitarias	Validar la funcionalidad de los componentes.	02/11/2024	02/11/2024
5	Documentación de resultados	Documentar hallazgos, recomendaciones y resultados finales del proyecto.	05/11/2024	09/11/2024

Resultados:

Se realizaron pruebas de estabilidad para asegurar el rendimiento del sistema bajo distintas cargas. Se identificaron y solucionaron errores en la transmisión de datos entre frontend y backend, mejorando la comunicación. Finalmente, se optimizó el flujo de datos, reduciendo el tiempo de respuesta y aumentando la eficiencia del sistema.

Evidencias:

- Optimización de código y evaluación de datos.

```

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
}

    _sensorDataService = sensorDataService;
    _httpClient = httpClient;

    [HttpPost]
    public async Task<ActionResult> InsertarSensorData([FromBody] SensorData sensorData)
    {
        if (sensorData == null)
        {
            return BadRequest("Datos de sensor inválidos");
        }

        await _sensorDataService.InsertarSensorData(sensorData);

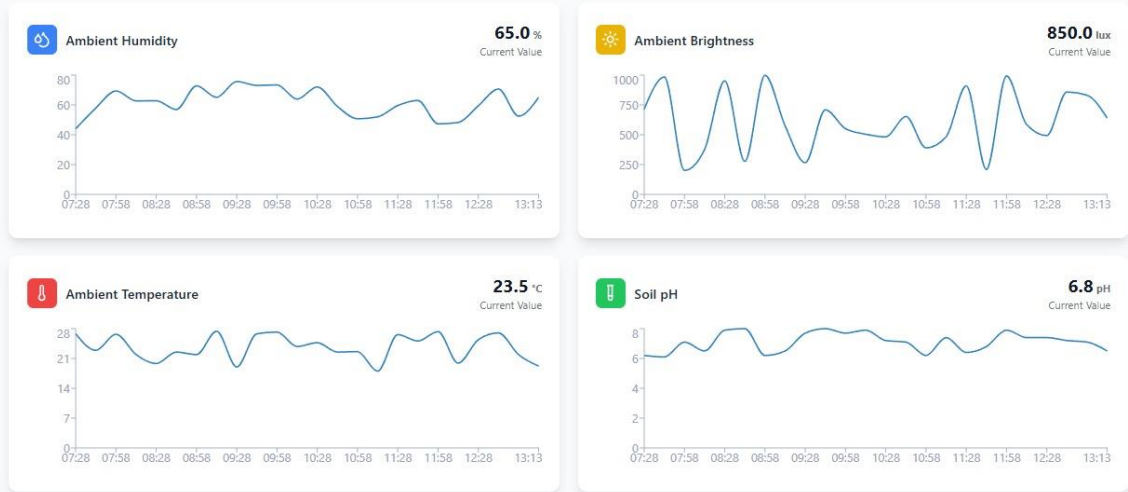
        var contenidoJson = new StringContent(JsonConvert.SerializeObject(sensorData), Encoding.UTF8, "application/json");
        var response = await _httpClient.PostAsync(_zapperWebhookUrl, contenidoJson);

        if (!response.IsSuccessStatusCode)
        {
            return StatusCode((int)response.StatusCode, "Error al enviar datos a Zapper");
        }

        return Ok("Datos del sensor registrados exitosamente y enviados a Zapper");
    }

    [HttpGet]
    public async Task<ActionResult<List<SensorData>>> ObtenerTodosLosDatos()
    {
        var datos = await _sensorDataService.ObtenerTodosLosDatos();
        return Ok(datos);
    }
}
    
```

Dashboard de Monitoreo Ambiental



8.2.9. Avance #9

Fecha: 21 de noviembre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)

Objetivos del Avance:

Revisar la creación de credenciales para las herramientas que van a ser utilizadas para mejorar el flujo del proyecto.

Descripción de Actividades:

N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Namecheap	Creación de cuenta de Namecheap y búsqueda de un dominio accesible para su futura utilización.	10/11/2024	11/11/2024
2	Creación y configuración de mailgun	Se configuro mailgun para utilizar el dominio comprado en Namecheap para servicio de mail y posteriormente ser usado para acceder a nuestra página web.	11/11/2024	13/11/2024
3	Digital Ocean	Compra del servidor que va a servir para albergar nuestro frontend y backend.	16/11/2024	16/11/2024

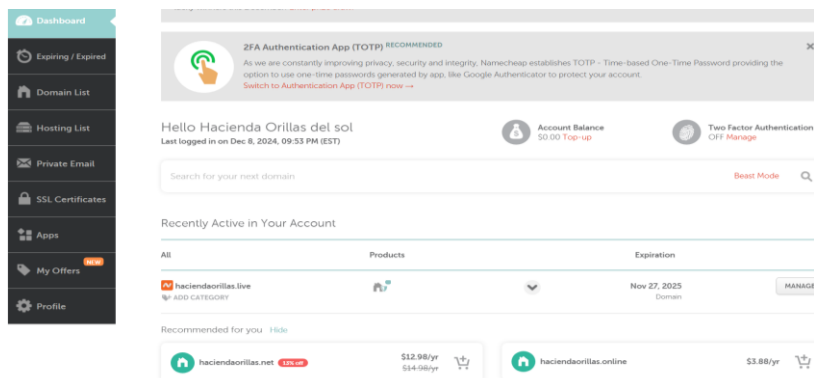
4	CloudFlare	Utilización de CloudFlare para generar certificados y garantizar la seguridad de la página.	17/11/2024	21/11/2024
---	------------	---	------------	------------

Resultados:

Los resultados obtenidos de las actividades recientes incluyen la exitosa creación de una cuenta en Namecheap y la adquisición de un dominio accesible, configuración efectiva de Mailgun usando este dominio para el envío seguro de correos electrónicos, y la adquisición y configuración de un servidor en Digital Ocean que alojará tanto el frontend como el backend del proyecto, asegurando un rendimiento óptimo. Además, se implementó CloudFlare para generar certificados SSL y garantizar la seguridad del sitio web, mejorando la confianza y protección de la información transmitida. Estos avances establecen una base sólida para la continuidad y expansión del proyecto.

Evidencias:

- Dominio configurado y listo para su uso.



- CloudFlare conectado con nuestro dominio

Permite revisar, agregar y editar registros de DNS. Las ediciones entrarán en vigor una vez guardadas.

Configuración de DNS: Completo [Importar y exportar](#) [Configuración de la pantalla del Panel de control](#)

Buscar registros DNS

Tipo	Nombre	Contenido	Estado de proxy	TTL	Acciones
A	haciendaorillas.live	143.110.230.77	Redirigido mediante proxy	Automático	Editar
CNAME	email.mg	mailgun.org	Redirigido mediante proxy	Automático	Editar
MX	mg	mail.mailgun.org	Solo DNS	Automático	Editar
MX	mg	mx.mailgun.org	Solo DNS	Automático	Editar
TXT	k1_domainkey.mg	"k1=rsa: p=MIIBJANBgkqhki...	Solo DNS	Automático	Editar
TXT	mg	"v=spf1 include:mailgun.org...	Solo DNS	Automático	Editar

Servidores de nombres de Cloudflare

A cada zona DNS en Cloudflare se le asigna un conjunto de servidores de nombres de la marca Cloudflare.

Tipo	Valor
NS	evangelina.ns.cloudflare.com

- Servidor de digital ocean levantado y funcional

Search by resource name or public IP (Ctrl+B) [Create](#) [My Team](#) Estimated costs: \$0.00

← Back to Droplets

ubuntu-tesisEcotec in [tesisEcotec2024](#) / 2 GB Memory / 50 GB Disk / SFO3 - Ubuntu 24.04 (LTS) x64 [Upsize Droplet](#) [ON](#)

IPv4: 143.110.230.77 [IPv6: Enable now](#) [Private IP: 10.124.0.2](#) [Reserved IP: Enable now](#) [Console: 1](#)

Graphs
Access
Power
Volumes
Resize
Networking
Backups
Snapshots
Kernel
History
Destroy
Tags
Recovery

Droplet Console [Learn](#)

Use the Droplet Console for native-like terminal access to your Droplet from your browser. Here is the list of supported OSes for the new console.

Log in as... root [Launch Droplet Console](#)

Recovery Console

Use the Recovery Console if you need to use the recovery ISO or you can't connect to your Droplet with the Droplet Console. To use the recovery console, you must enable password authentication. If necessary, you can reset your root password below.

8.2.10. Avance #10

Fecha: 28 de noviembre de 2024

Participantes:

- Omar Salas (Estudiante - Desarrollador del Proyecto)
- Samuel Peña (Estudiante - Desarrollador del Proyecto)
- Fadel Ibáñez (Ingeniero en Sistemas - Consultor)

Objetivos del Avance:

Revisar el proceso de subida del frontend y backend al servidor, configuración del servidor web con nginx y añadir las credenciales SSL en el servidor.

Descripción de Actividades:

N	Actividad	Descripción detallada	Fecha de inicio	Fecha de finalización
1	Subida al servidor	Se subio una imagen Docker del frontend y backend para su futura configuración	22/11/2024	22/11/2024
2	Instalación de Nginx	Se instalo Nginx en el servidor para su configuración cuando se añadan los certificados SSL.	22/11/2024	23/11/2024
3	Creación de los certificados SSL	Se crearon los certificados dentro de las rutas especificas en Ubuntu.	23/11/2024	23/11/2024
4	Creación del archivo de configuración de Nginx	Utilización de los certificados para crear la configuración que sirva a nuestro frontend usando Nginx.	23/11/2024	25/11/2024

Resultados:

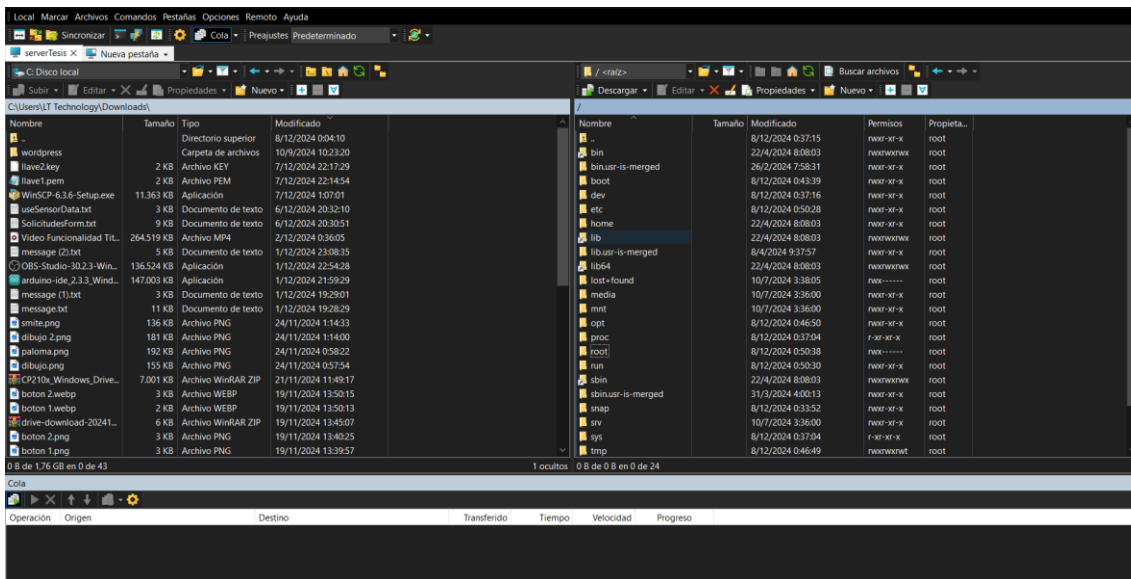
Creación de las rutas y asignación de cada una con su llave ya que las llaves se movieron a la carpeta root, para moverlas a las carpetas especificas se usaron los comandos.

```
sudo mkdir -p /etc/ssl/certs
```

```
sudo mkdir -p /etc/ssl/private
```

```
sudo cp llave1.pem /etc/ssl/certs/llave1.pem
```

```
sudo cp llave2.key /etc/ssl/private/llave2.key
```



- Creación del archivo de configuración de Nginx.

```

GNU nano 7.2 /etc/nginx/
server {
    listen 80;
    server_name haciendaorillas.live www.haciendaorillas.live;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name haciendaorillas.live www.haciendaorillas.live;

    ssl_certificate /etc/ssl/certs/llave1.pem;
    ssl_certificate_key /etc/ssl/private/llave2.key;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    root /var/www/react-frontend;
    index index.html;

    location / {
        try_files $uri $uri/ /index.html;
    }

    location /api/ {
        proxy_pass http://localhost:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    access_log /var/log/nginx/haciendaorillas.access.log;
    error_log /var/log/nginx/haciendaorillas.error.log;
}

```

8.3. Sección #3: Otros

8.3.1. Materiales:

Componente	Cantidad	Descripción
ESP8266 o ESP32	1	Placa microcontroladora con conectividad Wi-Fi.

Sensor DHT11 o DHT22	1	Sensor de temperatura y humedad.
Módulo de humedad del suelo	1	Sensor YL-69.
Cables Dupont	Según sea necesario	Para las conexiones entre sensores y la placa.
Protoboard	1	Opcional para organizar las conexiones de forma más limpia.
Fuente de alimentación (USB)	1	Cable USB o adaptador de corriente para la placa.
Resistencias (10kΩ, opcional)	Según sea necesario	Para conexiones seguras y precisas (dependiendo del sensor).

8.3.2. *Manual de Usuario*



Manual de Usuario
para el Frontend.do