



INSTITUTO TECNOLÓGICO DE PABELLÓN DE ARTEAGA



**“DESARROLLO DE BASE DE DATOS Y DISEÑO DE INTERFAZ
PARA UN SISTEMA DE CONTROL DE VARIABLES
AMBIENTALES EN UNA GRANJA VERTICAL PARA EL
CULTIVO DE SETAS”**

Titulación integral de tesis

Para obtener el grado de:

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES

Presenta:

Valeria Carolina Campos Hernández

Director:

Dra. S.B. Pamela Romo Rodríguez

Co directores:

Ing. Oscar Lenin Espinoza Alvarez

Dr. Adan Jair Serna Reyes

Pabellón de Arteaga, Aguascalientes, junio del 2024



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Pabellón de Arteaga
Subdirección Académica
Ingenierías

Pabellón de Arteaga, Aguascalientes, 21/octubre/2024

Oficio No. DIN-0153/2024

Asunto: Tesis-titulación

VALERIA CAROLINA CAMPOS HERNÁNDEZ
ESTUDIANTE DE LA CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
NO. DE CONTROL 191050170

PRESENTE

Por medio de este documento me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los(las) revisores(as) de su trabajo de Tesis titulado: **"DESARROLLO DE BASE DE DATOS Y DISEÑO DE INTERFAZ PARA UN SISTEMA DE CONTROL DE VARIABLES AMBIENTALES EN UNA GRANJA VERTICAL PARA EL CULTIVO DE SETAS"** realizado en el **"Laboratorio de Biotecnología Fúngica del ITPA"**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su acto protocolario de titulación integral por Tesis, así mismo fungiendo como **Presidenta S.B. Pamela Romo Rodríguez, Secretario Oscar Lenin Espinoza Álvarez y Vocal Adán Jair Serna Reyes.**

Sin otro particular, aprovecho la ocasión para enviarle un cordial saludo, quedo de Usted.

ATENTAMENTE

Excelencia en Educación Tecnológica
"Tierra Siempre Fértil"

EDINGUER VÁZQUEZ AYALA
JEFE DEL DEPARTAMENTO DE INGENIERÍAS



Pabellón de Arteaga, Aguascalientes, **21/octubre/2024**
Oficio No. DIN-0154/2024
Asunto: Tesis-Licenciatura

EDINGUER VÁZQUEZ AYALA
JEFE DEL DEPARTAMENTO DE INGENIERÍAS

PRESENTE

Por medio del presente doy el visto bueno a la Tesis de Licenciatura titulada **“DESARROLLO DE BASE DE DATOS Y DISEÑO DE INTERFAZ PARA UN SISTEMA DE CONTROL DE VARIABLES AMBIENTALES EN UNA GRANJA VERTICAL PARA EL CULTIVO DE SETAS”** realizado en el **“Laboratorio de Biotecnología Fúngica del ITPA”** de la estudiante **VALERIA CAROLINA CAMPOS HERNÁNDEZ** con numero de control 191050170 de la carrera Ingeniería en Tecnologías de la Información. Dicho trabajo ya fue revisado por cada uno de los miembros del comité titular y el estudiante ya realizó los cambios sugeridos, por lo que autorizamos su impresión.

Sin otro particular, aprovecho la ocasión para enviarle un cordial saludo, quedo de Usted.

ATENTAMENTE
Excelencia en Educación Tecnológica-
“Tierra Siempre Fértil”®

S.B. PAMELA ROMO RODRÍGUEZ
MIEMBRO DE COMITÉ TUTORIAL

OSCAR LENIN ESPINOZA ÁLVAREZ
MIEMBRO DE COMITÉ TUTORIAL

ADÁN JAIR SERNA REYES
MIEMBRO DE COMITÉ TUTORIAL

Agradecimientos

Quiero agradecer a mi misma, recordar que, gracias a mi fuerza y perseverancia puedo estar escribiendo mi tesis, gracias por no rendirme en ningún momento y por siempre tener en mente que lo que hago es por mí, gracias por dudar y confiar ya que de eso aprendí de muchas personas y situaciones que me pusieron donde estoy.

Agradezco a mi mamá, es mi fuente de vida, mi ejemplo a seguir y en la que pienso todos los días, gracias por darme el sustento para mis estudios y por nunca perder la esperanza en mí, termino este paso en mi vida y comienza uno más en donde espero poder ayudarte más; gracias por darme paz, confianza, enojos y corajes ya que es por eso que conocemos más de nosotras.

Gracias César, por simplemente estar a mi lado cuanto lo necesito, espero que puedas quedarte mucho tiempo más, fuiste tan vital en este proceso que agradecerte es poco para mí, me ayudaste a no darme por vencida y a darme cuenta de que mis esfuerzos valen de verdad.

Doy gracias por tener a mi familia y amigos, son todos parte de mi proceso en la carrera, gracias Abuela Berta, Dana, Vanessa, Rominna, Martin, Astrid, todos mis tíos y tías, Karla, Alejandra, Ian, Anna Paula, Leo, Victor, Diego, Omar, Armando, Fany.

También agradezco a todos mis profesores por el esfuerzo que pusieron en enseñarme lo que saben, a la doctora Pamela por darme la oportunidad de contribuir en su laboratorio y asesorarme con mi tesis de forma paciente y considerada; desde profesores de primer semestre como Diana que me convenció de que el cálculo si te puede gustar y es tan fácil como separar manzanas verdes de las manzanas rojas, hasta Lenin que me mostró que se puede ser profesional y exigente sin comprometer la propia salud; todos me mostraron los caminos que hay para poder desenvolverme en el área que más me gusta y también en lo que no me gusta.

Simplemente gracias a todas las personas que conozco por formar parte de mi vida dentro del instituto y sí, también a las personas que me hicieron la vida imposible, por eso sé cómo no debo tratar a los demás y que puedo ser mejor persona.

Gracias a todos por existir y hacerme ver que todos llevamos nuestro ritmo en cada etapa, no todos pasamos por lo mismo y que el ritmo que llevo está bien, porque el pasado ya no está, el futuro es incierto y el presente es un regalo que debo disfrutar día a día.

Resumen

El presente proyecto detalla el proceso de desarrollo, la programación de la base de datos y la interfaz de control y monitoreo para una granja vertical, que conforma la integración entre el frontend y el backend, mediante el desarrollo de una Api RESTful que gestiona las peticiones del usuario para obtener los datos que se muestran en la interfaz de monitoreo y para enviar los datos del cambio de variables a la granja vertical desde la interfaz de control mismas que se guardarán en la base de datos.

Además, se demuestra el proceso para montar el sistema en una Raspberry Pi que se instalará directamente en la granja vertical. Aquí se documenta el curso que toma de desarrollo e implementación del sistema propuesto, presentando los resultados obtenidos y las lecciones aprendidas, con el objetivo de contribuir al avance y el control climático en cultivos especializados como el de setas.

Índice

Agradecimientos.....	2
Resumen.....	3
1. Introducción.....	5
2. Planteamiento del problema.....	9
3. Objetivos.....	11
3.1 OBJETIVO GENERAL.....	11
3.2 OBJETIVOS ESPECÍFICOS.....	11
4. Justificación.....	12
5. Hipótesis.....	13
6. Alcances y limitaciones.....	13
6.1 ALCANCES.....	13
6.2 LIMITACIONES.....	14
7. Marco teórico.....	15
7.1 BASE DE DATOS.....	15
7.1.1 Base de datos.....	15
7.1.2 Una base de datos relacional.....	16
7.1.3 SQL.....	16
7.1.4 MySQL.....	16
7.2 INTERFAZ.....	17
7.2.1 Apache.....	17
7.2.2. PHP.....	17
7.2.3 Laravel.....	17
7.2.4 HTML.....	18
7.2.5 JavaScript.....	18
7.2.6 React	18
7.2.7 TypeScript.....	19
7.2.8 Node.js.....	19
7.2.10 Composer.....	20
7.2.11 Axios.....	20
7.2.12 CORS.....	20
7.2.13 JSON.....	20
7.2.14 Python.....	21
8. Metodología.....	22
8.1 Implementación de procesos.....	22
8.2 Identificación y documentación de requerimientos del sistema mediante la metodología Scrum.....	22
8.2.1 Niveles de Requerimientos.....	22
8.3.2 Diagramas de Flujo de Datos (DFD).....	23
8.3.3 Diagramas de Casos de Uso.....	23
8.3.4 Modelado de la Base de Datos.....	23
9. Resultados.....	25
9.2 Identificación de objetivos y Recopilación de requisitos.....	25

9.2.1 Análisis de Stakeholders.....	25
9.2.2 Entrevistas y encuestas.....	26
9.2.3 Requerimientos Funcionales y no funcionales.....	28
9.3 Modelado de Procesos.....	34
9.3.1 Contexto.....	34
9.3.2 Diagramas de casos de uso.....	35
9.4 DESARROLLO DE UNA BASE DE DATOS.....	37
9.4.1 CONFIGURACIÓN DEL ENTORNO DE DESARROLLO.....	38
9.4.2 DESARROLLO DEL BACKEND.....	38
9.5 CREACIÓN DE UNA INTERFAZ DE USUARIO.....	39
9.5.1 INTEGRACIÓN DEL BACKEND Y EL FRONTEND.....	41
9.5.2 MONITOREO Y CONTROL.....	41
9.5.3 DESPLIEGUE EN LA RASPBERRY PI.....	42
10. Conclusiones.....	47
11. Referencias.....	48

1. Introducción

Las bases de datos son sistemas organizados para la gestión, almacenamiento y recuperación de información estructurada. Utilizadas en una amplia variedad de aplicaciones, las bases de datos permiten manejar grandes volúmenes de datos de manera eficiente y segura. Desde registros financieros hasta inventarios de productos, su capacidad para organizar datos de forma coherente y accesible es fundamental para el funcionamiento de casi cualquier sistema informático moderno. Sirven para varias funciones críticas. En primer lugar, proporcionan un método estructurado para almacenar información, lo que facilita la recuperación y el análisis de datos. Además, permiten la integración de datos de diferentes fuentes, asegurando que la información esté centralizada. Las bases de datos también soportan múltiples usuarios y aplicaciones simultáneamente, garantizando la integridad y consistencia de los datos a través de mecanismos de control de acceso y gestión de transacciones (Gomez, 2012).

En el contexto de la agricultura moderna, las granjas verticales representan una innovación significativa, permitiendo la producción agrícola eficiente en espacios urbanos limitados mediante el uso de tecnología avanzada. La implementación de sistemas de bases de datos en estas granjas es crucial para su óptimo funcionamiento. Estos sistemas facilitan la gestión de grandes cantidades de datos provenientes de sensores, cámaras y otros dispositivos IoT (Internet de las Cosas), que monitorean continuamente el ambiente de cultivo.

Las interfaces de usuario para granjas verticales se benefician enormemente del uso de bases de datos bien estructuradas. Estas interfaces, que pueden incluir paneles de control, aplicaciones móviles y sistemas de alerta, dependen de la disponibilidad de datos precisos y en tiempo real para la toma de decisiones. Las bases de datos permiten la recopilación y análisis de datos sobre variables críticas como la temperatura, humedad y luz, entre otros. Esto no solo mejora la eficiencia operativa, sino que también optimiza el rendimiento de los cultivos, reduce el desperdicio y permite una mejor gestión de la información requerida por el usuario.

El diseño de interfaces es una disciplina esencial en el desarrollo de sistemas interactivos, que se centra en crear puntos de contacto entre el usuario y la tecnología de manera efectiva y eficiente. Un buen diseño de interfaz no solo facilita la usabilidad, sino que también mejora la experiencia del usuario al hacer que las interacciones sean intuitivas, atractivas y productivas. En un mundo cada vez más digitalizado, donde las aplicaciones y sistemas informáticos están presentes en todos los

aspectos de la vida, desde la educación hasta el entretenimiento y la gestión de negocios, el diseño de interfaces se ha convertido en un elemento crucial para el éxito de cualquier producto tecnológico (Wood, 2022).

La gestión eficiente de sistemas complejos como las granjas verticales requiere interfaces de usuario que faciliten el uso de la información que estas emiten.

En particular, para el diseño de interfaces en sistemas de granjas verticales tiene un impacto significativo en varios aspectos:

- Usabilidad y Eficiencia: Las interfaces intuitivas permiten a los operadores gestionar y ajustar rápidamente los parámetros del entorno de cultivo, reduciendo la posibilidad de errores y mejorando la productividad.
- Monitoreo en Tiempo Real: Interfaces bien diseñadas ofrecen visualizaciones claras y comprensibles de datos en tiempo real, permitiendo una respuesta rápida a cualquier cambio o problema en las condiciones del cultivo.
- Automatización y Control: Las interfaces que integran sistemas de automatización permiten la programación y ajuste de procesos sin intervención manual constante, optimizando el uso de recursos y garantizando la consistencia en la producción.
- Accesibilidad Remota: Con el avance de la tecnología, las interfaces móviles permiten a los operadores monitorear y controlar las granjas verticales desde cualquier lugar, aumentando la flexibilidad y la capacidad de respuesta.
- Integración de Datos y Análisis: Las interfaces que incorporan análisis de datos ayudan a los agricultores a entender mejor los patrones y resultados de sus cultivos, facilitando la toma de decisiones informadas y mejorando los rendimientos a largo plazo.

Los sistemas de control son una parte integral de la ingeniería y la tecnología moderna, diseñados para gestionar y regular el comportamiento de otros sistemas mediante el uso de bucles de retroalimentación. Estos sistemas pueden ser tan simples como un termostato en una casa o tan complejos como los sistemas de navegación de una nave espacial. Su principal objetivo es mantener un equilibrio en los procesos mediante el ajuste automático de variables clave, asegurando un funcionamiento eficiente y seguro. (Kuo, Sistemas de control digital, 2009)

La utilidad de los sistemas de control radica en su capacidad para optimizar el rendimiento y la eficiencia de diversos procesos. En la industria, se emplean para controlar la temperatura, la

presión, el flujo de fluidos, y otros parámetros críticos, mejorando la calidad del producto y reduciendo los costos operativos. En la vida cotidiana, se encuentran en una amplia gama de dispositivos, desde electrodomésticos hasta sistemas de climatización, ofreciendo comodidad y seguridad a los usuarios. (Kuo, Sistemas de control Automatico , 1996)

En el contexto de las granjas verticales, los sistemas de control juegan un papel crucial en la creación de un ambiente de cultivo óptimo. Las granjas verticales, que permiten el cultivo de plantas en capas apiladas verticalmente, requieren un monitoreo y ajuste preciso de factores como la luz, la humedad, la temperatura y los nutrientes para maximizar la producción y la calidad de las cosechas. Los sistemas de control automatizados permiten gestionar estos factores de manera eficiente, reduciendo la necesidad de intervención manual y permitiendo una agricultura más sostenible y productiva.

Además, la integración de sistemas de control con interfaces de usuario avanzadas es fundamental para la operación efectiva de las granjas verticales. Estas interfaces proporcionan a los agricultores datos en tiempo real y herramientas para supervisar y ajustar las condiciones de cultivo. La facilidad de uso y la accesibilidad de estas interfaces aseguran que los usuarios puedan tomar decisiones informadas y rápidas, mejorando la gestión de la granja y la productividad general. En última instancia, los sistemas de control, combinados con interfaces intuitivas, permiten que las granjas verticales funcionen con mayor precisión y eficiencia, contribuyendo significativamente al futuro de la agricultura urbana y sostenible.

Las granjas verticales representan una innovadora solución agrícola que permite cultivar plantas en capas apiladas verticalmente, optimizando el uso del espacio y los recursos. Esta modalidad de agricultura utiliza técnicas avanzadas como la hidroponía, la aeroponía y la acuaponía, y suele implementarse en entornos urbanos o espacios interiores controlados. Las granjas verticales se diseñan para maximizar la eficiencia en el uso del agua y los nutrientes, además de minimizar la necesidad de pesticidas y herbicidas. (Diaz, 2017)

Se tienen múltiples propósitos y beneficios. En primer lugar, ofrecen una solución sostenible para la producción de alimentos en áreas urbanas densamente pobladas, donde el espacio para la agricultura tradicional es limitado. Además, permiten la producción de cultivos durante todo el año, independientemente de las condiciones climáticas externas, lo que contribuye a la seguridad alimentaria. Estas granjas también reducen la distancia de transporte de los alimentos, lo que disminuye la huella de carbono asociada a la logística de distribución.

La implementación de granjas verticales ha impulsado el desarrollo y la integración de diversas tecnologías avanzadas. Sistemas de iluminación LED de espectro controlado, sensores ambientales, y software de gestión de cultivos son algunos de los avances tecnológicos que han sido potenciados por esta práctica agrícola. La automatización permite optimizar las condiciones de crecimiento de las plantas, asegurando una producción eficiente y sostenible. Además, estas tecnologías facilitan la recopilación y el análisis de datos en tiempo real, permitiendo ajustes precisos para maximizar el rendimiento y la calidad de los cultivos. (Gonzalias, 2015)

Las granjas verticales no solo representan una solución innovadora para la producción de alimentos en espacios limitados, sino que también impulsan el avance tecnológico, promoviendo una agricultura más inteligente y sostenible.

2. Planteamiento del problema

El problema radica en la necesidad de monitorear y controlar de manera precisa y eficiente las variables ambientales cruciales para el cultivo de setas en una granja vertical. Las setas requieren condiciones específicas de humedad, temperatura, y luz para crecer adecuadamente. Sin un sistema automatizado y bien diseñado, es difícil mantener estas condiciones óptimas de manera constante, lo que puede resultar en bajas tasas de éxito en el cultivo y pérdidas económicas.

El problema afecta varios aspectos clave:

- **Producción:** Una gestión deficiente de las variables ambientales puede reducir la cantidad y calidad de las setas producidas.
- **Eficiencia de recursos:** Sin un control adecuado, se puede malgastar agua, energía y otros recursos.
- **Rentabilidad:** Las pérdidas en la producción y el aumento de los costos operativos pueden afectar significativamente la rentabilidad de la granja.
- **Sostenibilidad:** Una gestión ineficiente puede aumentar la huella ambiental de la producción, contrarrestando los beneficios de las granjas verticales.

Es fundamental resolver este problema para asegurar una producción de setas constante y de alta calidad. Al controlar las variables ambientales de manera precisa, se pueden evitar problemas como crecimiento irregular y baja productividad. Esto no solo mejora la rentabilidad del cultivo, sino que también contribuye a la sostenibilidad y eficiencia de la granja vertical, aprovechando al máximo los recursos disponibles.

Parte de la solución, es desarrollar un sistema de control de variables ambientales que sea preciso y fácil de usar. Este sistema incluirá una base de datos que almacene información en tiempo real sobre las condiciones ambientales y un diseño de interfaz para facilitar el monitoreo y la gestión por parte de los operadores. Con este sistema, se pretende:

- **Optimizar las condiciones de cultivo:** Asegurando que las setas crezcan en un ambiente ideal en todo momento.
- **Aumentar la productividad:** Maximizar el rendimiento y la calidad de las setas producidas.
- **Mejorar la eficiencia de recursos:** Reducir el uso de agua, energía y otros insumos mediante un control preciso.
- **Incrementar la sostenibilidad:** Minimizar el impacto ambiental y promover prácticas agrícolas sostenibles.

- Facilitar la toma de decisiones: Proveer datos y análisis en tiempo real para tomar decisiones informadas y rápidas.

3. Objetivos

3.1 OBJETIVO GENERAL

Desarrollar una base de datos y diseñar una interfaz para un sistema de control de variables ambientales en una granja vertical para el cultivo de setas, con el fin de proporcionar a los usuarios un acceso rápido, organizado y fiable a esta información.

3.2 OBJETIVOS ESPECÍFICOS

- Identificar y documentar los requisitos del sistema mediante la metodología Scrum, enfocándose en las necesidades específicas de monitoreo y control de variables ambientales en una granja vertical para el cultivo de setas.
- Identificar el flujo de información de la interfaz representando gráficamente los procesos de desarrollo utilizando diagramas de flujo y diagramas de clases
- Desarrollar una base de datos con herramientas como Apache para MySQL, Laravel y lenguaje en php para almacenar y gestionar información relevante sobre variables ambientales en una granja vertical para el cultivo de setas.
- Crear una interfaz de usuario usando TypeScript, React y Node.js, que permita a los usuarios visualizar, gestionar y controlar los datos recopilados sobre las variables ambientales dentro de la granja vertical.

4. Justificación

El cultivo de *Pleurotus ostreatus* en granjas verticales enfrenta desafíos significativos relacionados con la gestión y el control de las condiciones climáticas. Actualmente, existe carencia de infraestructura adecuada que permita la monitorización precisa y el control eficiente de los parámetros ambientales en estas instalaciones especializadas.

La falta de un sistema integrado de almacenamiento de datos y una interfaz de usuario intuitiva dificulta la capacidad de los productores para gestionar exitosamente las variables críticas del entorno, como la temperatura y la humedad. Esta deficiencia puede tener consecuencias negativas en la calidad y el rendimiento del cultivo en cuestión.

Por tanto, la implementación de un sistema de almacenamiento y control climático diseñado para granjas verticales de cultivo de setas es una necesidad clara. Este proyecto tiene como objetivo abordar esta problemática desarrollando una solución integral que permita la recopilación, almacenamiento y gestión de datos sobre variables ambientales clave.

Al desarrollar una base de datos y una interfaz de usuario eficiente, los productores podrán acceder de manera rápida y organizada a la información sobre las condiciones climáticas dentro de la granja vertical. Esto no solo facilitará la toma de decisiones en tiempo real, sino que también optimizará las condiciones del cultivo, mejorando la calidad y el rendimiento.

La implementación exitosa de esta solución no solo beneficiará a los productores, si no también sentará las bases para futuras investigaciones y prácticas en contextos educativos.

En resumen, el desarrollo de este sistema es fundamental para optimizar las condiciones del cultivo, mejorar la productividad y garantizar condiciones idóneas en el futuro.

5. Hipótesis

Si se desarrolla e implementa un sistema de almacenamiento eficiente y una interfaz de usuario adecuada para el control de variables ambientales dentro de la granja vertical de cultivo de setas, entonces se observará una mejora significativa en la gestión y el monitoreo de las condiciones ambientales, lo que resultará en un aumento de la calidad y el rendimiento del cultivo de setas.

6. Alcances y limitaciones

6.1 ALCANCES

- Desarrollar y completar el código de los controles climáticos para la granja vertical.
- Realizar pruebas exhaustivas de los controles climáticos para garantizar su funcionalidad y precisión.
- Programar la conexión de la interfaz de usuario con el código de los controles climáticos.
- Diseñar y desarrollar la interfaz de usuario para permitir el monitoreo y control de los parámetros climáticos de la granja vertical.
- Realizar pruebas de integración para asegurar que la interfaz de usuario funcione correctamente con los controles climáticos y el sistema de monitoreo.
- Capacitar a los usuarios finales sobre el uso adecuado de la interfaz de usuario y el sistema de control climático.

6.2 LIMITACIONES

- Retraso en el desarrollo del código de los controles: El retraso en la disponibilidad del código de los controles climáticos afectará el progreso del desarrollo, ya que la integración de estos controles es fundamental para el funcionamiento del sistema. Esto puede requerir ajustes en el plan de implementación y la priorización de otras tareas mientras se espera la disponibilidad del código.
- Pruebas prolongadas de los controles: Dado que las pruebas de los controles climáticos pueden llevar más tiempo del previsto, existe el riesgo de que se retrase aún más el desarrollo y la integración del sistema en su conjunto.
- Dificultades en la programación de la conexión de la interfaz: La programación de la conexión entre la interfaz de usuario y el código de los controles climáticos puede resultar compleja y requerir un tiempo considerable. Se deben asignar recursos adicionales y expertos en programación para abordar estos desafíos de manera efectiva y minimizar el impacto en el cronograma general del proyecto.
- Falta de desarrollo de la interfaz y la codificación del monitoreo: la falta de desarrollo de la interfaz de usuario y la codificación del monitoreo representa una limitación significativa, ya que estas son partes clave del sistema que requieren tiempo y recursos adicionales para su implementación. Se debe priorizar estas tareas y asignar recursos adecuados para complementarlas dentro del plazo establecido.

- Compatibilidad con diferentes dispositivos, al estar en un equipo propio del laboratorio como es una raspberry pi puede variar la compatibilidad al querer utilizar la interfaz con otro dispositivo que no comparta las características del equipo disponible.

7. Marco teórico

7.1 BASE DE DATOS

7.1.1 Base de datos

Una base de datos es un conjunto de datos organizados de manera estructurada que permite el almacenamiento, recuperación y manipulación eficiente de la información. Estas estructuras pueden contener datos de diferentes tipos, como texto, números, imágenes, y otros formatos, organizados de tal manera que se facilita su uso y manejo por parte de los usuarios y aplicaciones (Marqués, 2011).

Esta definición se integrará en el proyecto para gestionar grandes volúmenes de información que se recopilan en tiempo real, facilitando su posterior análisis y control.

7.1.2 Una base de datos relacional

Es un conjunto de información registrada, organizada y estructurada dentro de tablas relacionadas entre sí de manera específica para que su contenido pueda ser tratado y analizado de forma rápida y sencilla. Estas bases de datos en ocasiones se utilizan para almacenar y asegurar la integridad de los datos, ya sea de una empresa o sistema electrónico. Existen varios tipos de bases de datos, como las estáticas, que son de solo lectura y se utilizan para análisis, y las dinámicas, que permiten realizar consultas de datos, actualizaciones e incluso eliminar los datos (Cabello, 2010).

Dado que el sistema de control de variables ambientales requiere almacenar datos de forma organizada y relacionada entre sí. Las tablas que contendrán información sobre las variables ambientales, los sensores, y los parámetros de control estarán conectadas mediante relaciones específicas, lo que permitirá una gestión eficiente de los datos. La base de datos relacional facilitará la consulta rápida y precisa de la información, asegurando la integridad de los datos en todo momento.

7.1.3 SQL

Es un lenguaje estándar para acceder y manipular bases de datos, se utiliza para realizar operaciones como consultas, inserciones, actualizaciones y eliminaciones en los datos almacenados, con una estructura simple, permite crear y modificar la estructura de una base de datos con comandos como

CREATE TABLE, ALTER TABLE y DROP TABLE, así como manipular los datos dentro de las tablas con los comandos SELECT, INSERT, UPDATE y DELETE.

SQL se utiliza en sistemas como MySQL para almacenar y recuperar datos en sitios web y aplicaciones (Gomez, 2012).

Esta herramienta será fundamental para el correcto funcionamiento del sistema y garantizará la flexibilidad y eficiencia en la gestión de los datos del sistema en el proyecto.

7.1.4 MySQL

Es un sistema de gestión de bases de datos relacionales y su objetivo principal es almacenar y administrar datos, esta herramienta funciona para crear bases de datos y organizarlos mediante relaciones entre las tablas, recibe solicitudes de los usuarios, estos envían instrucciones SQL a las tablas de la base de datos, las procesa y devuelve la información solicitada a los usuarios. Es ampliamente utilizado en aplicaciones web, sistemas empresariales y más (Santillan, 2014).

MySQL facilitará la organización y administración de grandes cantidades de información ambiental, respondiendo a las solicitudes de los usuarios para visualizar y controlar las variables ambientales en la granja vertical. La implementación de esta tecnología será clave para asegurar la fiabilidad y rendimiento del sistema de control.

7.2 INTERFAZ

7.2.1 Apache

Es un servidor web que responde a las solicitudes de contenido de los usuarios. Funciona como un intermedio entre el navegador del usuario y los archivos o aplicaciones alojadas en el servidor y permite ejecutar aplicaciones escritas en PHP, Python y más. Es fundamental para alojar aplicaciones web de forma local y brindar contenido a los usuarios (Camazón, 2011).

Esto permitirá ejecutar las aplicaciones desarrolladas en PHP, lo que será esencial para garantizar la accesibilidad y la correcta respuesta del sistema.

7.2.2. PHP

(Procesador de Hipertexto) es un lenguaje de programación interpretado diseñado específicamente para el desarrollo web puede incluirse en HTML para hacer las webs dinámicas, su sintaxis se basa

en Java, con algunas características propias. PHP facilita la conexión con bases de datos como MySQL, procesando datos enviados desde formularios web (Cobo, 2005).

Será fundamental para procesar los datos enviados desde los formularios de la interfaz de usuario, permitiendo a los administradores de la granja visualizar y modificar las variables ambientales en tiempo real. Este lenguaje será parte clave del backend del sistema de control, manejando las operaciones críticas de actualización y recuperación de datos.

7.2.3 Laravel

Es un marco de aplicación Web construido en PHP que proporciona una gestión de autenticación de usuarios y permisos, además, permite crear y modificar la estructura de la base de datos de manera controlada (migraciones), ofrece herramientas para validar que los formularios estén completados de forma correcta en los datos de entrada, envía notificaciones y correos electrónicos de manera sencilla y ejecuta tareas programadas en intervalos específicos (Ciceri, 2019).

Su capacidad para gestionar migraciones en la base de datos y programar tareas recurrentes es de real importancia para mantener el sistema actualizado y operando eficientemente, garantizando la integridad de los datos almacenados para el proyecto.

7.2.4 HTML

(Lenguaje de Etiquetas de Hipertexto) no es un lenguaje de programación; es un lenguaje de marcado que define la estructura del contenido de una página web. Se trata de una serie de elementos que contienen diferentes partes de la página web para que se vean o se comporten de una manera determinada (Luna A. C., 2024).

Aunque no es un lenguaje de programación, HTML definirá la estructura del contenido visible en la interfaz, permitiendo que los usuarios interactúen con el sistema de manera clara y organizada. Este lenguaje será combinado con TypeScript y React para crear una interfaz dinámica que facilite el monitoreo y control de las variables ambientales dentro de la granja vertical.

7.2.5 JavaScript

Es un lenguaje de programación interpretado y orientado a objetos que se utiliza principalmente para añadir características interactivas a un sitio web, como ejemplo, para aplicar efectos de estilo;

esto es fundamental para que los sitios web sean interactivos y llamativos. JavaScript es un lenguaje:

- **Interpretado:** No necesitas compilarlo; los navegadores lo ejecutan directamente.
- **Orientado a objetos:** Permite crear objetos y manipularlos.
- **Versátil:** Puede interactuar con HTML para crear experiencias dinámicas en la web.
- **Dinámico:** Algo que cambia o se adapta en tiempo real (Luna F. , 2019).

Además, permitirá actualizar y mostrar en tiempo real los datos recopilados sobre las variables ambientales, brindando una experiencia interactiva y atractiva para los usuarios que controlan los parámetros de la granja. Esto para que el sistema sea fácil de usar y responda dinámicamente a las necesidades del usuario.

7.2.6 React

Es una biblioteca de JavaScript para construir interfaces de usuario. Te permite crear componentes reutilizables y combinarlos para formar aplicaciones web interactivas. React se basa en la idea de componentes. Un componente es una pieza aislada de la interfaz de usuario que puede contener lógica y representación visual.(Adam Boduch, 2020).

En el proyecto, los componentes React se usarán para mostrar información crítica, como gráficos, permitiendo a los usuarios interactuar con ellos de manera fluida. La capacidad de React para gestionar estados y actualizaciones en tiempo real es crucial para mantener la interfaz sincronizada con los datos de la granja.

7.2.7 TypeScript

Es un lenguaje de programación que se basa en JavaScript que detecta errores tempranos en el editor de texto antes de ejecutar el código y se puede ejecutar en navegadores además facilita la comprensión del código por parte de otros desarrolladores (Puciarelli, Angular: TypeScript – Arquitectura – Instalación – Directivas y Bindings – Forms – Ruteo y más, 2020).

La tipificación estática de TypeScript ayudará a reducir posibles errores en el manejo de datos ambientales, haciendo que el sistema sea más confiable y fácil de mantener a largo plazo.

7.2.8 Node.js

Es un entorno de ejecución de JavaScript de código abierto gratuito que permite crear aplicaciones web y scripts, permite ejecutar código fuera del navegador, lo que lo hace ideal para el desarrollo del lado del servidor. Es utilizado para crear servidores web, APIs y aplicaciones en tiempo real y gracias a NPM, es posible tener acceso a una gran cantidad de paquetes y bibliotecas esenciales para la programación, ya que permiten reutilizar código y acelerar el desarrollo de aplicaciones (Puciarelli, Node JS - Vol. 1: Instalación - Arquitectura - node y npm, 2020).

7.2.9 NPM

NPM (Node Package Manager) es el gestor de paquetes oficial para Node.js. Permite a los desarrolladores instalar, compartir y gestionar bibliotecas y dependencias necesarias para sus proyectos de manera eficiente. Con NPM, es posible acceder a una amplia gama de módulos de código abierto que ayudan a acelerar el desarrollo, ya que ofrecen funcionalidades preconstruidas que se pueden integrar en el proyecto del sistema para la granja vertical. Además, facilita la gestión de versiones y actualizaciones de los paquetes, asegurando que todo el software utilizado esté alineado con las últimas mejoras y correcciones de errores (Puciarelli, 2020).

7.2.10 Composer

Es un gestor de dependencias para el PHP que ayuda a administrar las bibliotecas necesarias para proyectos y es muy utilizado en proyectos con Laravel (AGUIRRE, 2024).

Al usar Composer, se garantiza que todas las dependencias estén correctamente instaladas y actualizadas, optimizando así el flujo de trabajo del equipo de desarrollo dentro del proyecto.

7.2.11 Axios

Permite realizar solicitudes y recibir respuestas HTTP desde el navegador o el servidor (Herrera, 2020).

Será empleado para obtener datos en tiempo real sobre las variables ambientales desde el backend, asegurando que la interfaz de usuario esté constantemente actualizada con la información más reciente del sistema.

7.2.12 CORS

(Intercambio de recursos de origen cruzado) es un mecanismo basado en HTTP que permite a un servidor y los distintos dominios o puertos puedan realizar solicitudes seguras entre navegadores; CORS se aplica a API como Fetch (Ruiz Barea, 2023), esto funcionará dentro del proyecto para definir qué dominios tienen permiso para realizar solicitudes HTTP al servidor de control de la granja vertical.

7.2.13 JSON

(Notación de objeto de JavaScript) es un formato de texto sencillo para el intercambio de datos y se utiliza comúnmente para transmitir datos entre el servidor y el cliente en aplicaciones web (Aguirre, 2020). Esto permitirá una comunicación eficiente y estructurada, facilitando el envío de datos sobre las variables ambientales desde el backend a la interfaz. La sencillez y compatibilidad de JSON lo hacen ideal para mantener la información de la granja organizada y fácilmente accesible para los usuarios.

7.2.14 Python

Es un lenguaje de programación versátil y poderoso que te permite trabajar de manera rápida e integrar sistemas de manera más efectiva y tiene una amplia gama de bibliotecas y marcos para diversos propósitos, como desarrollo web (Vahid Mirjalili, 2020).

Además, su amplia gama de bibliotecas lo convierte en una opción atractiva para complementar el desarrollo de herramientas adicionales, como la creación de algoritmos de predicción o control automatizado en la granja.

8. Metodología

8.1 Implementación de procesos

La implementación del proceso Scrum en el proyecto de "Desarrollo de Base de Datos y Diseño de Interfaz para un Sistema de Control de Variables Ambientales en una Granja Vertical para el Cultivo de Setas" sigue una serie de pasos claramente definidos que aseguran la entrega ágil y eficiente del producto final. Esta metodología ágil se basa en iteraciones cortas y entregables frecuentes, lo que permite una constante adaptación a los cambios y necesidades del proyecto.

8.2 Identificación y documentación de requerimientos del sistema mediante la metodología Scrum

La preparación inicial del proyecto implica la definición del Product Backlog, donde el Product Owner elabora una lista priorizada de características y requisitos en este caso por medio de una serie de evaluaciones y formularios para obtener los requerimientos; Para asegurar una base sólida de trabajo, se siguen los siguientes niveles de requerimientos:

8.2.1 Niveles de Requerimientos

1. Requerimientos de Negocio

- **Identificación de Objetivos:** Comprensión clara de los objetivos del proyecto, como mejorar el monitoreo y control de las variables ambientales en la granja vertical.
- **Análisis de Stakeholders:** Identificación de todos los interesados y sus expectativas, técnicos y administradores de la granja.

2. Requerimientos de Usuario

- **Entrevistas y Encuestas:** Realización de entrevistas y encuestas con los usuarios finales para entender sus necesidades y problemas actuales.
- **User Stories:** Creación de historias de usuario que describen las funciones que los usuarios necesitan del sistema, como el seguimiento en tiempo real de la humedad y temperatura.

3. Requerimientos Funcionales y No Funcionales

- **Especificaciones Detalladas:** Descripción detallada de las funcionalidades del sistema, como el registro y visualización de datos ambientales, alertas y notificaciones.
- **Requerimientos No Funcionales:** Incluyen aspectos como la seguridad, rendimiento y usabilidad del sistema.

8.3 IDENTIFICAR EL FLUJO DE INFORMACIÓN DE LA INTERFAZ CON REPRESENTANDO GRÁFICAMENTE CON EL MODELADO DE PROCESOS

El modelado de procesos es esencial para entender y documentar cómo los datos fluyen y cómo se realizan las operaciones dentro del sistema. Este paso incluye:

8.3.2 Diagramas de Flujo de Datos (DFD)

- **Contexto:** Representa el sistema en su totalidad y las interacciones con entidades externas.
- **Niveles de Detalle:** Detalla los procesos internos del sistema, como la recopilación de datos de sensores y la actualización de la base de datos.

8.3.3 Diagramas de Casos de Uso

- **Identificación de Actores:** Define quiénes interactúan con el sistema, como los usuarios y administradores.
- **Escenarios de Uso:** Describe las interacciones entre los actores y el sistema para cumplir con los objetivos, como la consulta de datos históricos de variables ambientales.

8.3.4 Modelado de la Base de Datos

- **Esquema Entidad-Relación (ER)**
- **Identificación de Entidades y Relaciones:** Define las tablas y relaciones necesarias para almacenar los datos del sistema.
- **Normalización:** Asegura que la base de datos sea eficiente y libre de redundancias.

8.4 DESARROLLO DE LA BASE DE DATOS

- Se utiliza Apache para MySQL y Laravel con PHP para el desarrollo de la base de datos.
- Descargar e instalar estas herramientas tomando en cuenta que hay que tener actualizado el sistema operativo del equipo.
- Crear el proyecto con Laravel mediante el Símbolo del sistema del equipo especificando con anterioridad una carpeta pre-creada en el explorador de archivos, específica para la base de datos, después, con el comando necesario que otorga la página oficial de COMPOSER.
- La creación de tablas y relaciones necesarias para almacenar y gestionar la información sobre variables ambientales con PHP.
- Se crea la conexión de la programación con PHP a MySQL.
- La migración de las tablas con código se hace mediante comandos ejecutados en la consola y finalmente se pueden ver las tablas en MySQL.

8.5 DISEÑO DE LA INTERFAZ DE USUARIO

- Usando TypeScript, React y Node.js, se desarrolla la interfaz que permitirá a los usuarios visualizar y gestionar los datos.
- Las herramientas son descargadas y en su respectivo caso, importadas directamente en el proyecto ya creado mediante la consola del editor de texto que se esté utilizando.
- La creación del proyecto se realiza en el Símbolo del sistema del equipo, especificando con anterioridad una carpeta pre-creada en el explorador de archivos, específica para la interfaz, después, con el comando necesario especificado en la página oficial de TypeScript, se crea el proyecto y se abre en el editor de texto deseado.
- Se crean componentes interactivos que faciliten el control de las variables ambientales y que satisfagan los requerimientos visuales que solicitan los usuarios.
- Finalmente se pone en funcionamiento el servicio que abre la interfaz en un navegador para poder utilizarla, esto se hace mediante comandos específicos de Node.js.

9. Resultados

9.2 Identificación de objetivos y Recopilación de requisitos para el proyecto

De acuerdo con los niveles de requerimientos, se realizó una pequeña y concisa lista de objetivos provenientes de los interesados, esto para comprender principalmente lo que se busca resolver para la granja vertical, a continuación, se muestra la lista.

Tabla 1 Requisitos principales

Interfaz de Control para Gestión de Temperatura y Humedad: La interfaz debe permitir a los usuarios ajustar y monitorear los grados de temperatura y los sensores de humedad en tiempo real, asegurando el correcto funcionamiento de las condiciones ambientales en el sistema de cultivo.
Visualización Gráfica del Desempeño de Variables: La interfaz debe presentar gráficos que muestren el estado actual y el comportamiento histórico de las variables ambientales del sistema (como temperatura y humedad), permitiendo identificar fallos y anomalías en su funcionamiento.
Historia de Datos: Las gráficas deben permitir la visualización de las variaciones de las variables ambientales en rangos temporales flexibles, desde 1 día hasta 365 días , para un seguimiento detallado de su comportamiento a largo plazo.
Propuesta de Diseño Centrada en el Usuario: Se deben diseñar y presentar propuestas funcionales y estéticas basadas en las preferencias y necesidades de los usuarios finales, asegurando que la interfaz sea intuitiva y útil para su propósito.
Disponibilidad exclusiva en Computadora: El sistema debe estar disponible únicamente para uso en computadora , garantizando una experiencia optimizada para equipos de computo.

9.2.1 Análisis de Stakeholders

La interfaz de control para una granja vertical considera usuarios que son parte de la comunidad estudiantil dentro del Tecnológico de Pabellón de Arteaga, investigadores y cualquier persona que requiera modificar o analizar de estado de la granja vertical en cierto periodo de tiempo

9.2.2 Entrevistas y encuestas

Para la realización de la entrevista y cuestionario se tomó en cuenta recopilar varias preguntas preliminares antes de redactar la lista final de preguntas para la entrevista a los usuarios de la interfaz.

Tabla 2 Entrevista inicial

¿Cuál es el objetivo principal al analizar imágenes de setas?
¿Qué tipo de información esperas obtener de este análisis?
¿Qué funcionalidades específicas te gustaría que la interfaz proporcionara?
¿Existen características particulares que desearías destacar en el análisis?
¿Qué tipos de setas estarás analizando?
¿Necesitas analizar imágenes de setas en condiciones específicas o entornos particulares?
¿En qué formato se encuentran las imágenes que planeas analizar? (JPEG, PNG, etc.)
¿Cuántos tipos de setas deseas analizar?
¿Hay características específicas que te interesen en cada tipo?
¿Tienes preferencias sobre el diseño y la disposición de la interfaz?
¿Hay elementos visuales o datos específicos que consideres críticos para la presentación de resultados?
¿Cómo te gustaría que se presentarán los resultados del análisis? (gráficos, estadísticas, informes detallados, etc.) gráficas para cada dato, humedad, temperatura, e iluminación durante los 10 días para cada ciclo
¿La interfaz debe ser accesible desde dispositivos específicos? (computadoras, tabletas, dispositivos móviles)
¿Hay requisitos particulares en cuanto a la seguridad y privacidad de los datos? Límite de usuarios y todo será visible para todo el que lo necesite
¿Existen otras consideraciones o requisitos específicos que debamos tener en cuenta para el desarrollo de la interfaz?
¿Cuál es el objetivo principal al analizar imágenes de setas?

Después de revisar esta entrevista y con ayuda del asesor, se tomaron solo algunas preguntas para poder desarrollar la entrevista final al usuario; las preguntas toman en cuenta el tema de preferencias en la interfaz, como colores, formas y contrastes.

Tabla 3 Entrevista Preliminar

¿Podría proporcionar una breve descripción de su interés en el análisis de imágenes de setas? Saber a qué temperatura se encontró la seta durante el ciclo de análisis
¿Cuál es el objetivo principal que espera lograr con este proyecto? Facilitar la forma en que se administra la información y que el usuario haga el menor esfuerzo posible
¿Cuáles son las necesidades específicas que espera cubrir con este análisis de imágenes?
¿Tiene expectativas particulares en términos de resultados o información obtenida?

¿Quiénes serán los usuarios finales de los resultados obtenidos a través de este proyecto?
¿Tiene alguna preferencia en cuanto a la presentación de los resultados para los usuarios finales?
¿Puede proporcionar ejemplos específicos de cómo planea utilizar los resultados del análisis de imágenes de setas?
¿Existen casos de uso particulares que le gustaría destacar?
¿Hay requisitos técnicos específicos que debamos tener en cuenta al desarrollar la interfaz?
¿Tiene preferencias en cuanto a la tecnología utilizada para el análisis de imágenes?
¿Cómo imagina la colaboración entre su equipo y el equipo de desarrollo durante el proyecto?
¿Qué tipo de retroalimentación espera proporcionar a lo largo del desarrollo?
¿Existen tiempos específicos que debamos tener en cuenta para la realización del proyecto?
¿Hay documentación específica que espera recibir en etapas clave del desarrollo?
¿Identifica algún desafío potencial o preocupación que debamos abordar desde el principio?
¿Hay factores externos que podrían afectar el desarrollo del proyecto?
¿Existen requisitos particulares en cuanto a la seguridad y privacidad de los datos recopilados durante el análisis de imágenes?
¿Cómo prevé la gestión de posibles mejoras continuas una vez que la interfaz esté en uso?

En la entrevista preliminar solo se tomaron preguntas que tienen que ver con las preferencias de funcionalidad, lo que al usuario le gustaría hacer y que le muestre la interfaz como, gráficos y datos de los controles.

Tabla 4 Entrevista final

¿Cuál es el propósito principal de este proyecto?
¿Cuáles son los objetivos y metas que esperas lograr con este proyecto?
¿Cuáles son las funciones principales que debe tener el proyecto?
¿Cuáles son las características del equipo que utilizan?
¿Ha utilizado anteriormente alguna herramienta para el análisis de datos?
¿Cuenta ya con la fuente de los datos?
¿Conoce si las fuentes de datos se encuentran limpias y preparadas para ser usadas (proceso ETL)?
¿cómo se accederá a las fuentes de obtención de datos?
¿Ha trabajado antes con conjuntos grandes de datos?
¿Cuáles son las fórmulas que se aplicarían para los cálculos?
¿Es necesario tener cierta forma específica de visualizar los datos (si son tablas, gráficas, etc)?
¿Necesita que sus datos se mantengan en informes mensualmente?
¿Necesita que sus datos estén en actualización acorde a la fuente de obtención?
¿Tiene en mente formas de mantener de forma segura los datos que se van a tratar en este proceso?
¿Cuáles son los requisitos de seguridad y privacidad que deben cumplirse?
¿Hay requisitos legales o normativos que debemos cumplir?

Con base en la entrevista realizada para comenzar la lista de requerimientos, los usuarios proporcionan imágenes de lo que deseaban ver como la interfaz, dando así una idea de lo que esperaban en cuanto al diseño y lo que esperaban de la funcionalidad requerida para el control de la granja vertical.



Figura 1 Idea proporcionada por el Product owner

A partir de esto, se tomó la decisión de aplicar un cuestionario rápido a través de la aplicación Forms de Google para algunos los usuarios que se encuentran dentro del laboratorio. Estas preguntas ayudarán a entender las necesidades y preferencias de los usuarios para diseñar una interfaz que se adapte mejor a sus requerimientos. A continuación, se muestran gráficas con cada pregunta y sus respectivas respuestas.

9.2.3 Requerimientos Funcionales y no funcionales

Tomando en cuenta la entrevista y el formulario realizados tanto al cliente como el usuario podemos tomar decisiones para definir al realizar la lista de los requerimientos que se contempló dividirlos por requerimientos del cliente que en este caso es el laboratorio de biotecnología fúngica y requerimientos de usuario donde se tomaron en cuenta a las personas que usarán la interfaz.

Adicionalmente a cada requerimiento se agregó la gráfica y la pregunta que generó el mismo.

1. Requerimientos del cliente:

La interfaz debe ser capaz de mostrar información en tiempo real sobre temperatura y humedad dentro de la unidad (nombre de la unidad).

Requerimientos del usuario:

El usuario requiere obtener gráficos visuales que representen las tendencias y cambios de los rangos establecidos en cada variable dentro de la unidad (nombre de la unidad).

Requerimientos del sistema

- Requerimientos funcionales:
 - Desarrollo de interfaz con TypeScript, React y JAVASCRIPT.
 - Integración de la librería Chart para la creación de los gráficos que podrá visualizar el usuario.
 - Uso de Node.js para procesos de tiempo real y con esto poder utilizar los datos de lo que pasa dentro de la granja vertical.
- Requerimientos no funcionales:
 - Autenticación y autorización con Laravel para garantizar la seguridad de la aplicación al momento de entrar al sistema con un usuario.

¿Qué funciones de control esperas de la interfaz?

5 respuestas

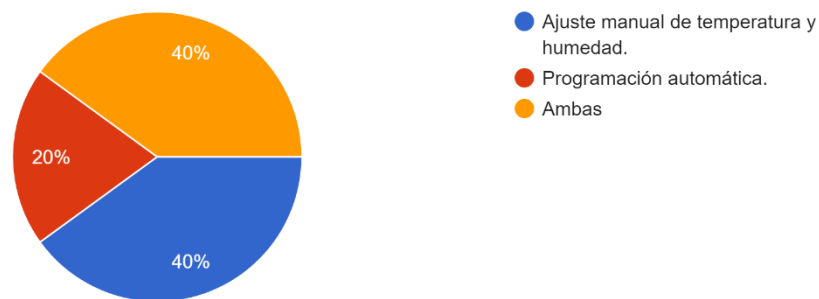


Figura 2 Figura 2 Gráfico de respuestas de formularios. Título de la pregunta: ¿Qué funciones de control esperas de la interfaz?

Respuestas:

Color azul: 40% votó por un ajuste manual de variables.

Color rojo: 20% votó por una programación automática de variables en la granja vertical

Color amarillo: 40% votó por ambas opciones.

Con esta gráfica podemos obtener como resultado el entendimiento de las herramientas que necesitaremos para poder programar las funciones de los controles en la interfaz

2. Requerimientos del cliente:

Debe ser posible ajustar y controlar el estado del ambiente con las variables correspondientes en los rangos que el usuario lo requiera.

Requerimientos del usuario:

Por medio de la interfaz, se desea modificar regularmente el estado de las temperaturas y la humedad dentro de la granja vertical.

Requerimientos del sistema

- Requerimientos funcionales:
 - Desarrollo con JavaScript para activar los eventos de los botones que modifican los rangos de las variables que el usuario desee.
- Requerimientos no funcionales:
 - Desarrollar en visual studio.
 - Prever casos de pruebas.
 - Utilizar estándares para las pruebas de software.

¿Qué rangos de humedad te gustaría monitorear?
5 respuestas

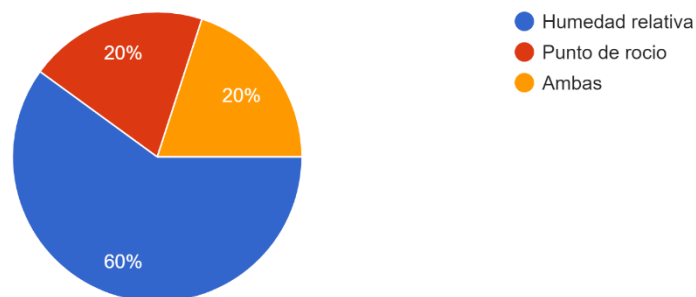


Figura 3 Gráfico de respuestas de formularios. Título de la pregunta: ¿Qué rangos de humedad te gustaría monitorear?.

Respuestas:

Color azul: 60% votó por monitorear la humedad relativa dentro de la granja vertical.

Color rojo: 20% votó por el punto de rocío.

Color amarillo: 20% votó po monitorear ambos rangos de humedad.

En el diagrama se observan los rangos que se desean observar con mayor urgencia en los controles, dando como resultado los requerimientos de desarrollo para los eventos de control y monitoreo en conjunto.

3. Características de la interfaz

Requerimientos del cliente:

Debe haber un esquema de colores intuitivo para representar diferentes rangos de temperatura y humedad.

Requerimientos del usuario:

El usuario espera poder identificar los colores de cada variable en los gráficos y al momento de usar la interfaz para ajustar las variables de la unidad (el nombre de la unidad).

Requerimientos del sistema

- **Requerimientos funcionales:**
 - Delimitar los colores con código hexadecimal para poder usarlos en las gráficas hechas con Chart y también usar los mismos colores en los botones para el manejo y cambio de temperatura y humedad de la granja vertical.
- **Requerimientos no funcionales:**
 - Asegurar la preferencia de las paletas de colores por los usuarios.

¿Qué tipo de gráficos preferirías para el monitoreo visual?

5 respuestas

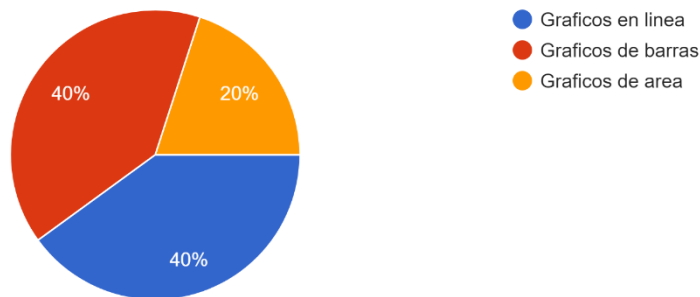


Figura 4 Gráfico de respuestas de formularios. Título de la pregunta: ¿Qué tipo de gráficos preferirías para el monitoreo visual?.

Respuestas:

Color azul: 40% de los votantes prefieren gráficos en línea.

Color rojo: 40% votó por gráficos de barras.

Color amarillo: 20% prefirió gráficos de área para poder monitorear las condiciones de la granja vertical.

En el esquema anterior se identifica un mayor resultado en la preferencia del gráfico que se mostrará al usuario cuando se interprete la información pedida por el mismo.

4. Requerimientos del cliente:

Los usuarios deben tener la opción de personalizar la apariencia de la interfaz considerando las necesidades de accesibilidad, especialmente para usuarios con dificultades visuales.

Requerimientos del usuario:

Debe ser posible que el usuario pueda modificar el brillo de fondo en la interfaz, ya sea de claro a oscuro o bien de oscuro a claro.

Requerimientos del sistema

- **Requerimientos funcionales:**

-Desarrollo de un botón de modo oscuro con JavaScript para permitirle al usuario personalizar la apariencia de la interfaz.

A continuación, se pueden apreciar los resultados de la encuesta realizada al usuario para saber si se desea tener la opción para personalizar la interfaz.

¿Te gustaría poder personalizar la apariencia de la interfaz, como luz o modo oscuro?

5 respuestas

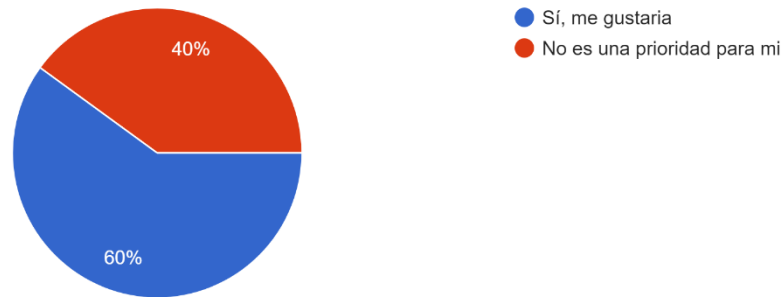


Figura 5 Gráfico de respuestas de formularios. Título de la pregunta: ¿Te gustaría poder personalizar la apariencia de la interfaz, como luz o modo oscuro?.

Respuestas:

Color azul: al 60% le gustó la idea de poder cambiar la interfaz a modo oscuro.

Color rojo: 40% de los votantes no cree necesario contar con esta opción.

9.3 Modelado de Procesos

9.3.1 Contexto

Se realizó un sólo diagrama de flujo para comprender y representar los procesos de la interfaz en general al interactuar con el usuario, desde el inicio al ingresar con el usuario correspondiente, hasta realizar cualquier actividad que desee para interactuar con la granja vertical.

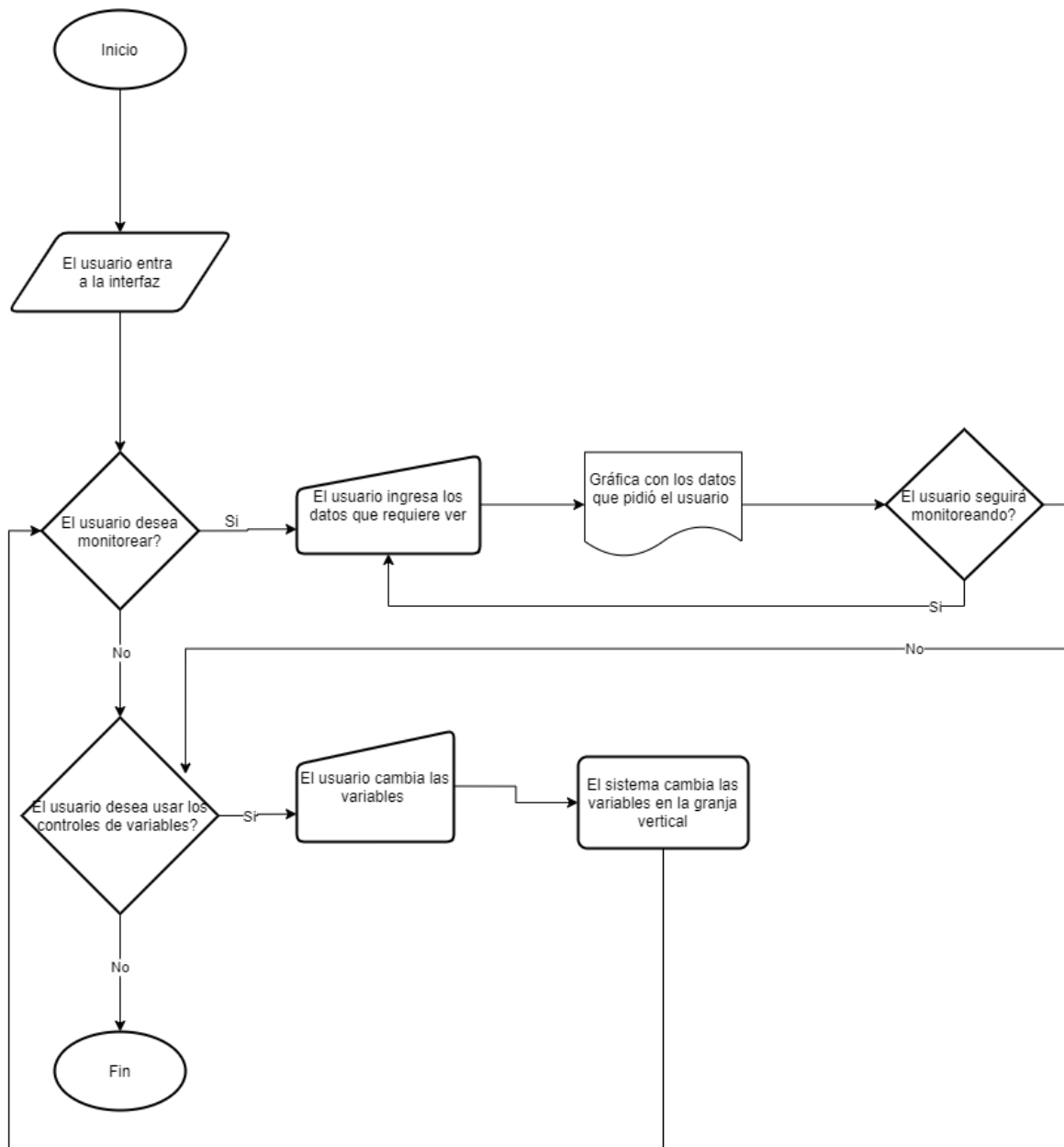


Figura 6 Diagrama de flujo de la interfaz de control y monitoreo para una granja vertical

9.3.2 Diagramas de casos de uso

Identificación de actores y escenarios de uso

Para identificar los actores en la interfaz y saber cómo se cumplirán los objetivos de la misma se efectuaron diagramas de flujo de datos lógicos de niveles del 0 al 2.



Figura 7 Diagrama de flujo de datos lógicos Nivel 0

El DFD Lógico de Nivel 0 es el primer paso para entender cómo interactúa el sistema en su conjunto con el exterior y marca la pauta para los niveles de detalle que se desarrollarán posteriormente.

A partir de la figura anterior podemos identificar inicialmente los usuarios principales que intervendrán en el uso de la interfaz.

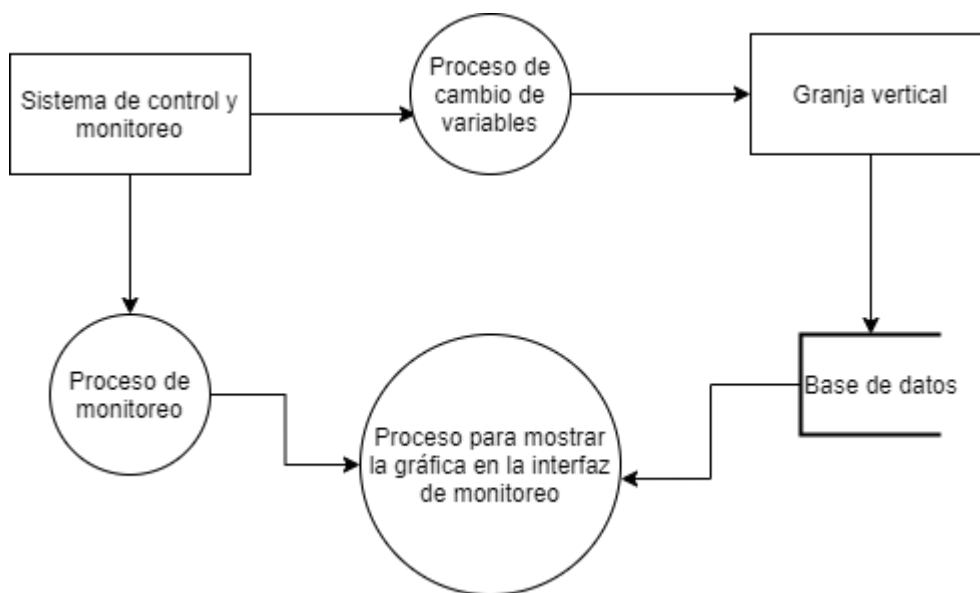


Figura 8 Diagrama de flujo de datos lógicos Nivel 1

En la Figura del diagrama de flujo de datos lógico de nivel 1, además de incluir a los usuarios se puede apreciar el principal funcionamiento de la lógica del proceso que llevará la interfaz.

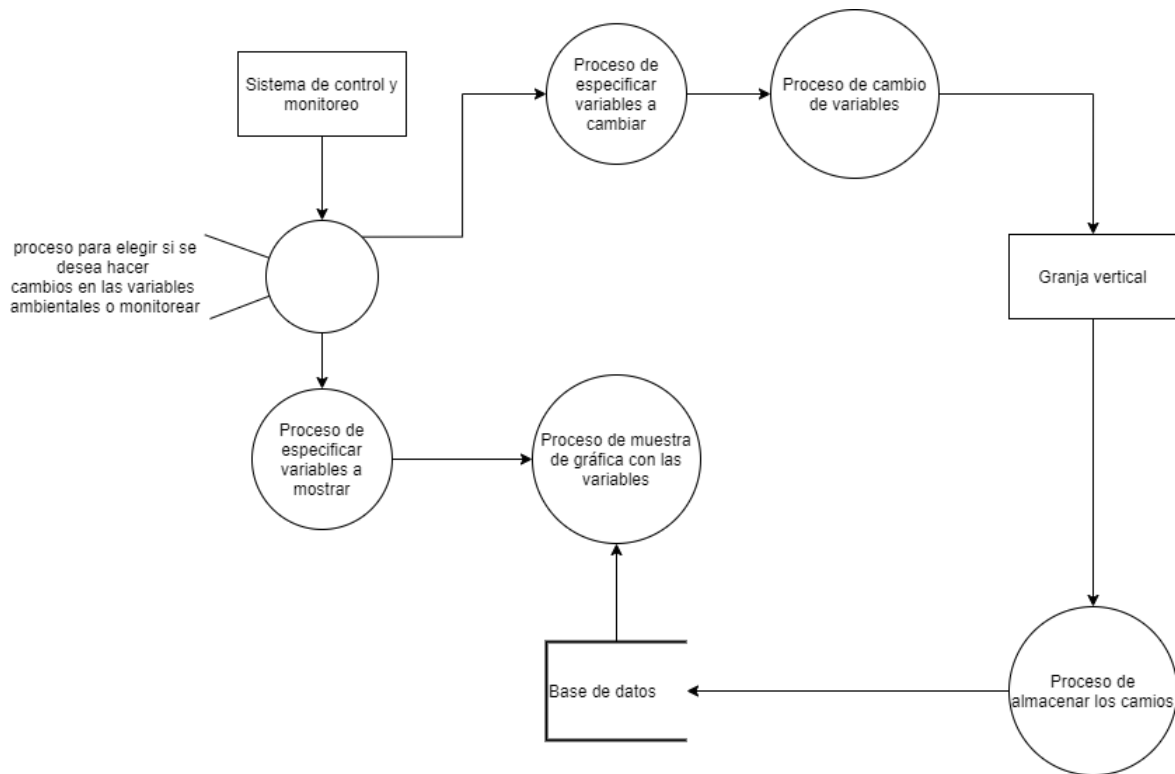


Figura 9 Diagrama de flujo de datos lógicos Nivel 2

Para finalizar con el proceso del flujo de la información que lleva la base de datos y la interfaz en conjunto se puede tener un mejor entendimiento mediante el 2do nivel del diagrama.

Este nivel de detalle es útil para analizar en profundidad el funcionamiento de subprocessos y actividades específicas dentro de cada proceso principal del sistema.

9.4 DESARROLLO DE UNA BASE DE DATOS

De acuerdo a los objetivos específicos, lo primero a realizar para el desempeño adecuado del proyecto es el levantamiento de requerimientos por parte de los usuarios de la interfaz, ya que por medio de una serie de preguntas cuidadosamente formuladas, se llegó a la conclusión de que el sistema sería montado en una microcomputadora Raspberry Pi para ser integrado a la granja vertical por lo que, a continuación se revisaron las posibles tecnologías factibles a implementar para

el correcto funcionamiento del proyecto, dando como resultado la elección de MySQL para gestionar la base de datos y JavaScript con React como lenguaje para el desarrollo de la interfaz.

Teniendo el conocimiento de las herramientas a utilizar, el proceso para iniciar toma lugar con la instalación de estas herramientas desde la terminal de la Raspberry Pi con comandos para asegurarse de que ninguna paquetería fuera extraviada en la instalación.

9.4.1 Configuración del entorno de desarrollo

Se configuró un entorno de desarrollo adecuado en la Raspberry Pi, incluyendo la instalación de apache y MySQL; Composer para gestionar las dependencias de PHP y Laravel.

```
composer create-project --prefer-dist laravel/laravel mi-proyecto
```

Este comando crea el proyecto en la carpeta designada para el backend que contiene nuestra Base de datos, solo se cambia “mi proyecto” por el nombre preferido para el mismo.

9.4.2 Desarrollo del Backend

Laravel se configuró para interactuar con MySQL y este a su vez almacena datos, como las configuraciones de la interfaz de control y los registros del monitoreo de las variables de temperatura y humedad.

A continuación, se muestra el código utilizado para la configuración de la conexión del backend con la base de datos desde el archivo. Env.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=BDIGVCS
DB_USERNAME=labbiofun
DB_PASSWORD=laboratorio
```

Figura 10 Configuración de la base de datos

Estas líneas son editadas según las preferencias del desarrollador al crear la conexión de la base de datos con MySQL ya que se toman en cuenta datos como nombre de la base de datos, el usuario principal y una contraseña si así se desea.

Las migraciones para crear las tablas con sus respectivos campos, además de la programación de lo que realizarán las peticiones de cada tabla y sus rutas.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Parametro extends Model
{
    use HasFactory;
    protected $table = 'Parametro';
    protected $fillable = ['fase_id','dia' , 'temperatura', 'humedad', 'iluminacion'];
    protected $hidden = ['created_at', 'updated_at'];
}
```

Figura 11 Programación de la tabla Parámetro para la base de datos

El código mostrado anteriormente muestra la programación una de las dos tablas creadas para la base de datos con PHP siendo esta la más importante ya que contiene los datos de las variables de la granja vertical, que después se migrará a MySQL para poder manipular los datos desde la interfaz; a continuación, se adjunta un link al repositorio que contiene toda la programación de las tablas para la base de datos:

<https://github.com/ValeriaAMD/BDIGVCS/tree/main/BDIGVCS2/BDIGVCS2BACK>

9.5 Creación de una interfaz de usuario

El frontend se desarrolló usando React JS y TypeScript para asegurar una experiencia de usuario dinámica. React JS permitió crear componentes reutilizables y administrar el estado de la aplicación de manera eficiente.

Después de crear una carpeta específica para la interfaz, se crea el proyecto en el símbolo del sistema, utilizando el siguiente comando:

```
npx create-react-app nombre-del-proyecto --template typescript
```

A partir de este solo queda cambiar el nombre del proyecto por el de la preferencia del desarrollador y ejecutarlo en el editor de texto preferido, en este caso se utilizó Visual studio Code.

Una vez abierto se crean carpetas con la organización necesaria para mantener el orden dentro del proyecto, como los componentes aparte de las rutas y las funciones.

Enseguida se muestran las funciones que hacen posible la funcionalidad de la interfaz de monitoreo.

```

// Constantes que usan las variables para tomar los valores dentro de la interfaz y enviarlas al backend
const Monitoreo: React.FC = () => {
  const [modoOscuro, setModoOscuro] = useState<boolean>(false);
  const [faseSeleccionada, setFaseSeleccionada] = useState<string>('');
  const [fechaInicio, setFechaInicio] = useState<string>('');
  const [fechaFin, setFechaFin] = useState<string>('');

  const handlePlayClick = () => {
    console.log('Fase seleccionada:', faseSeleccionada);
    console.log('Fecha de inicio:', fechaInicio);
    console.log('Fecha de fin:', fechaFin);

    const url = `http://127.0.0.1:8000/api/parametro?fase_id=${faseSeleccionada}&fecha_inicio=${fechaInicio}&fecha_fin=${fechaFin}`;

    fetch(url)
      .then(response => {
        if (!response.ok) {
          throw new Error('Hubo un problema al realizar la solicitud.');

```

Figura 12 programación de las acciones de la interfaz de monitoreo

Este código muestra la programación de los parámetros necesarios para mostrarle al usuario una gráfica con dimensiones correctas y colores que permitan diferenciarse entre sí, además de algunas de las funciones de la interfaz de monitoreo.

Los códigos completos que muestran la programación de los componentes que hacen posible la visualización de la interfaz se encuentran en el repositorio oficial del proyecto, a continuación, se adjunta el link a este:

<https://github.com/ValeriaAMD/BDIGVCS/tree/main/BDIGVCS2/BDIGVCS2FRONT>

9.5.1 Integración del backend y el frontend

Para la integración entre el frontend y el backend, se desarrollaron APIs RESTful con Laravel. Estas APIs permitieron la comunicación entre React JS y Laravel, gestionando las operaciones para obtener y enviar datos de control y monitoreo necesarios para la aplicación.

Se anexan las partes del código que se programaron y configuraron para poder lograr la integración.

```
Route::get('/parametro/{id}', [ParametroController::class, 'show']);
Route::get('/parametro', [ParametroController::class, 'index']);
Route::get('/fase/{id}', [FaseController::class, 'show']);
Route::get('/fase', [FaseController::class, 'show']);
Route::post('/update-settings', [ValuesController::class, 'update']);
```

Figura 13 Rutas para conectar la base de datos y la interfaz

Con la programación de estas rutas se puede lograr la conexión entre la interfaz y la base de datos para poder hacer las peticiones que requiera el usuario ver en la parte de las gráficas en la interfaz de monitoreo.

9.5.2 Monitoreo y control

Se implementan funcionalidades de monitoreo y control en tiempo real utilizando Python para manejar la comunicación con sensores y actuadores en la granja vertical. Estos datos se almacenaron en MySQL para una rápida consulta y procesamiento.

Para la correcta comunicación entre el programa de Python y la interfaz de control, se agregaron líneas de código en Python donde se habilitan los CORS y se abre un puerto determinado que abrirá

paso a las peticiones para cambiar las variables en la granja vertical, a continuación se muestran las líneas de código:

```
from flask import Flask, request, jsonify
from flask_cors import CORS

app = Flask(__name__)
CORS(app)
```

Figura 14 Cors habilitados

9.5.3 Despliegue en la Raspberry Pi

Finalmente, se procedió al despliegue del sistema completo en la Raspberry Pi. Apache sirvió como servidor web, mientras que los servicios de base de datos MySQL y MongoDB se configuraron para ejecutarse de manera eficiente en el hardware limitado de la Raspberry Pi. El proceso incluyó:

- Configuración del servidor Apache para servir la aplicación frontend construida con React JS.
- Despliegue de la aplicación Laravel en el mismo servidor.
- Configuración de Node JS para ejecutar servicios de monitoreo en segundo plano.

Finalmente, los resultados son satisfactorios, ya que el sistema en conjunto fue integrado en la raspberry con éxito y está a su vez fue integrada a la granja vertical.

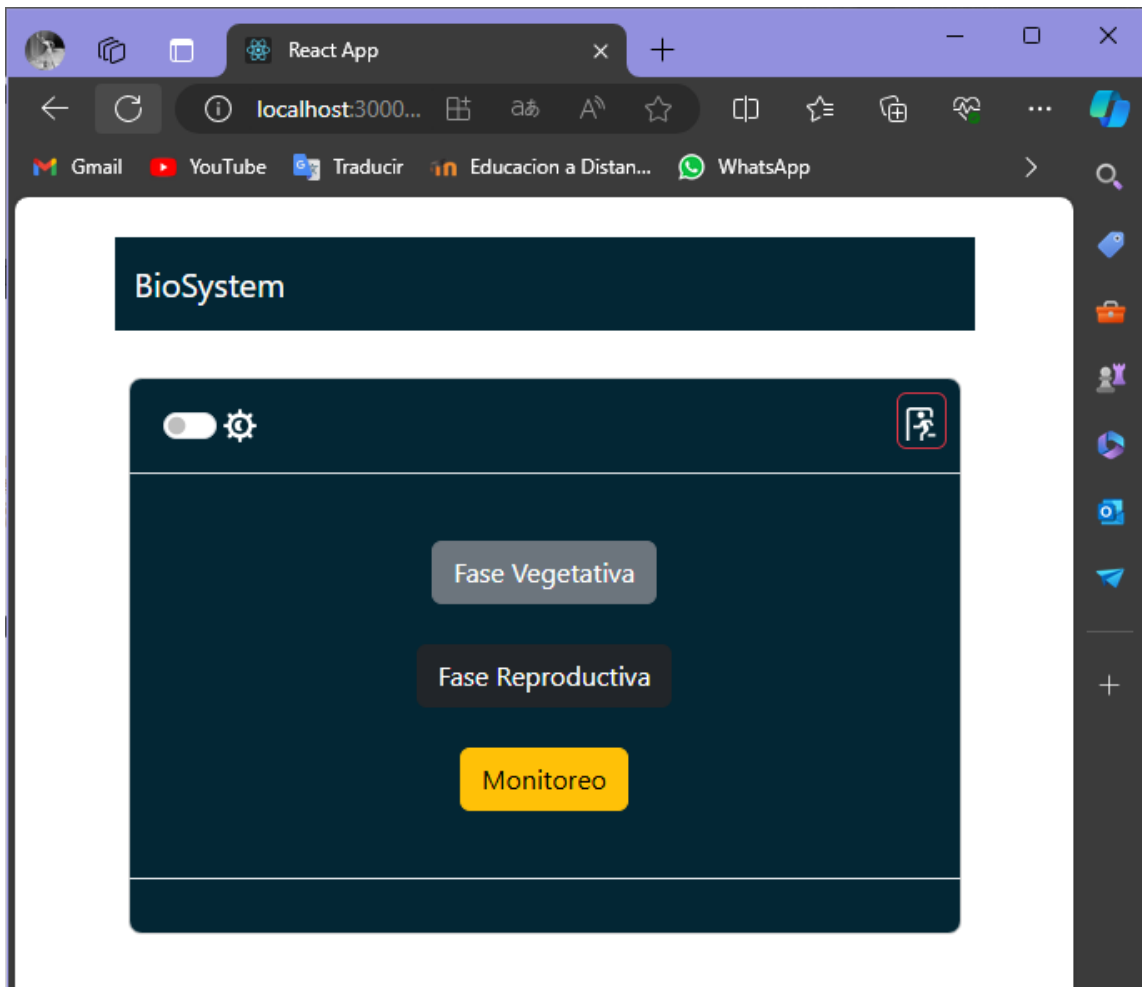


Figura 15 Resultado de la interfaz "Página de opciones"

En esta figura se puede observar la interfaz en la página de las opciones que el usuario puede elegir según la acción que quiera realizar, ya sea modificar las variables de temperatura y humedad de la fase vegetativa dentro de la granja vertical o la reproductiva o bien, pasar a la sección de monitoreo donde podrá visualizar los datos de las fechas que guste organizados en un gráfico.

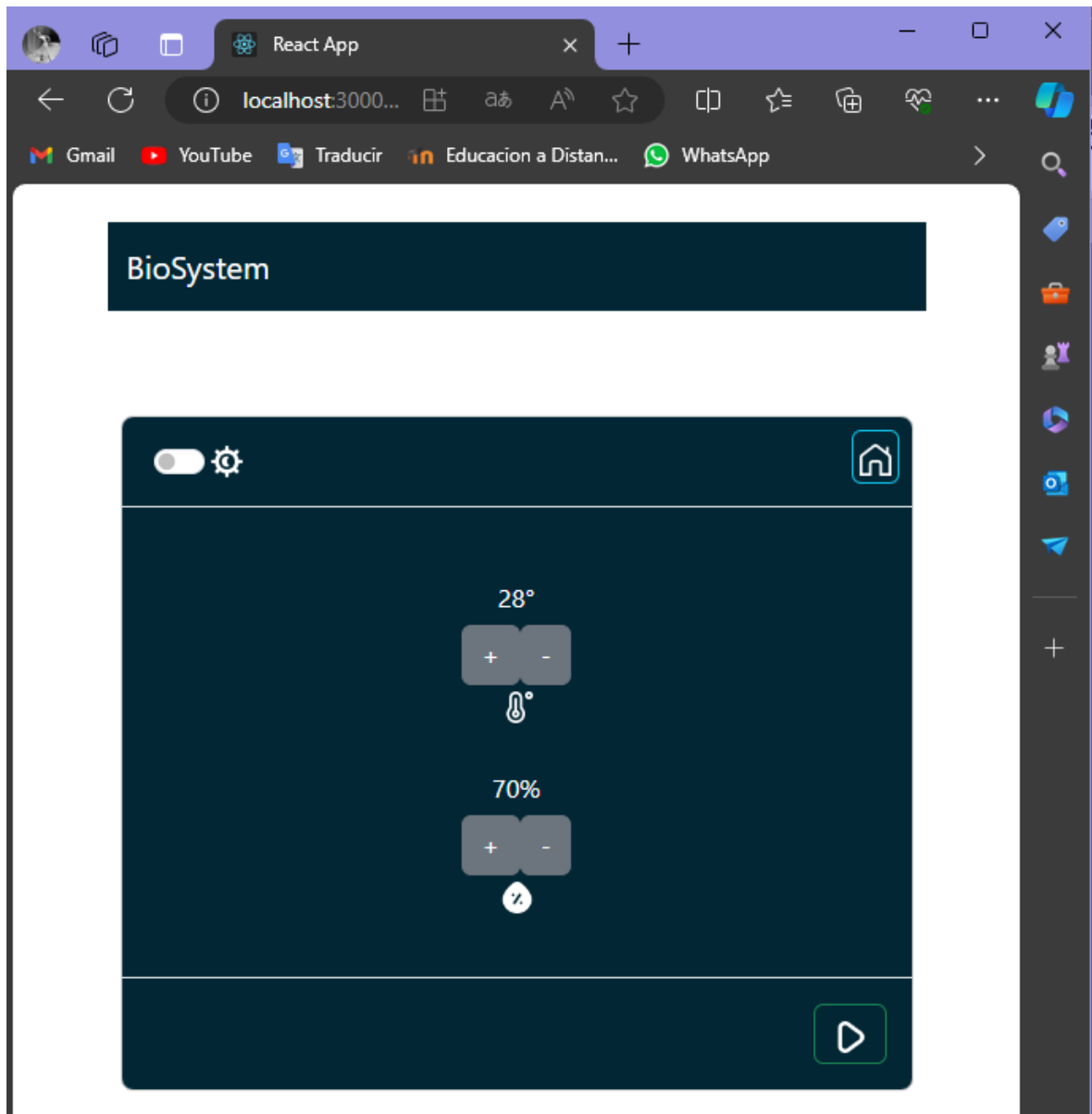


Figura 16 Resultado de la interfaz de control en la fase vegetativa

En la figura 16 la pantalla muestra los controles para la opción de la fase vegetativa que controla la temperatura y la humedad, además es visible el control de modo oscuro en la parte superior izquierda y un botón de regreso a la página de opciones; esta pantalla es igual en la opción de fase reproductiva pero diferenciando los valores iniciales en los que se encuentran las variables.

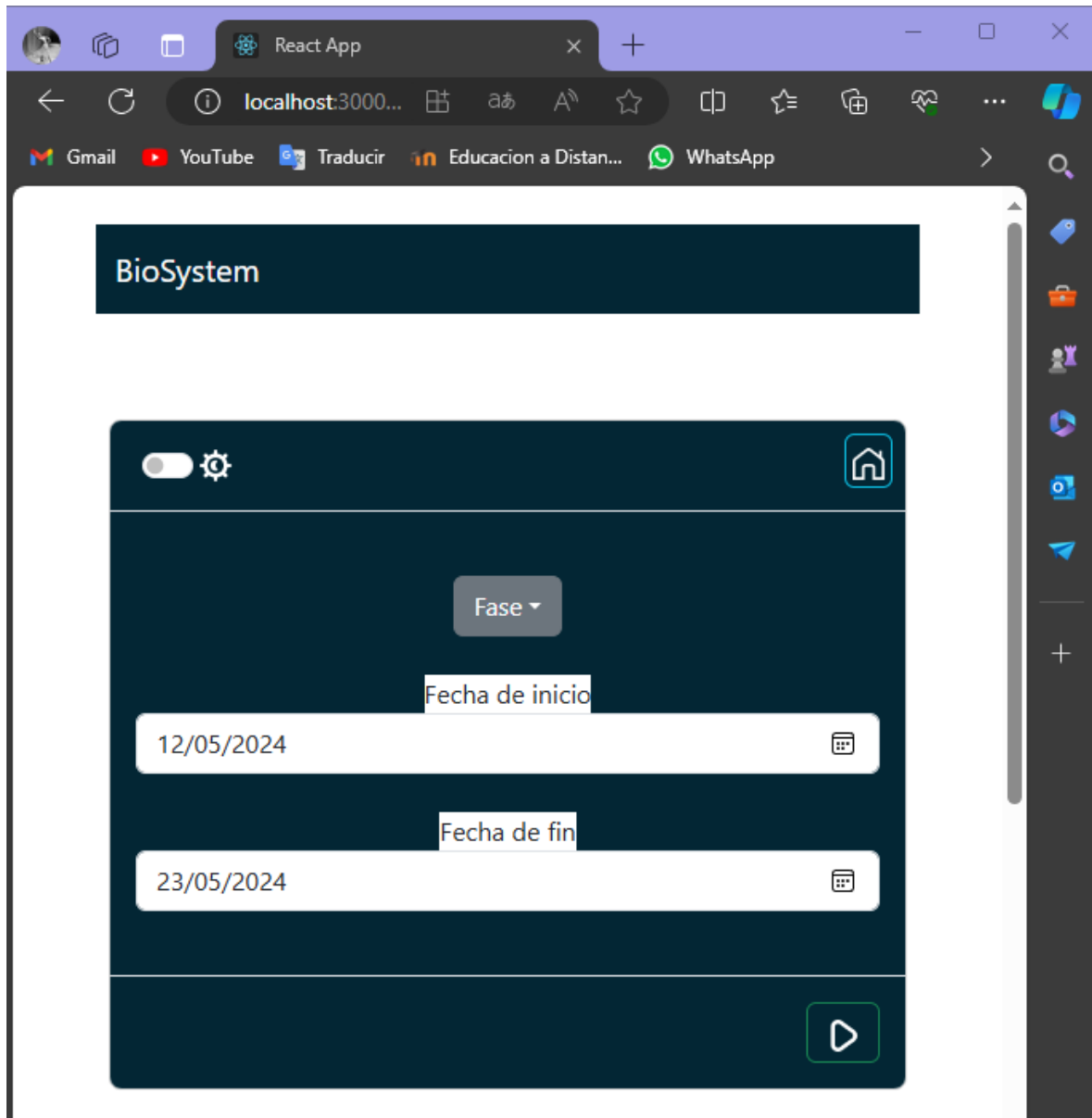


Figura 17 Resultado de la interfaz "Página de monitoreo"

Como se muestra, la interfaz de monitoreo en la fase de especificación, donde el usuario ingresa las fechas que desea consultar y posteriormente se mostraran al presionar el botón de "play" color verde.

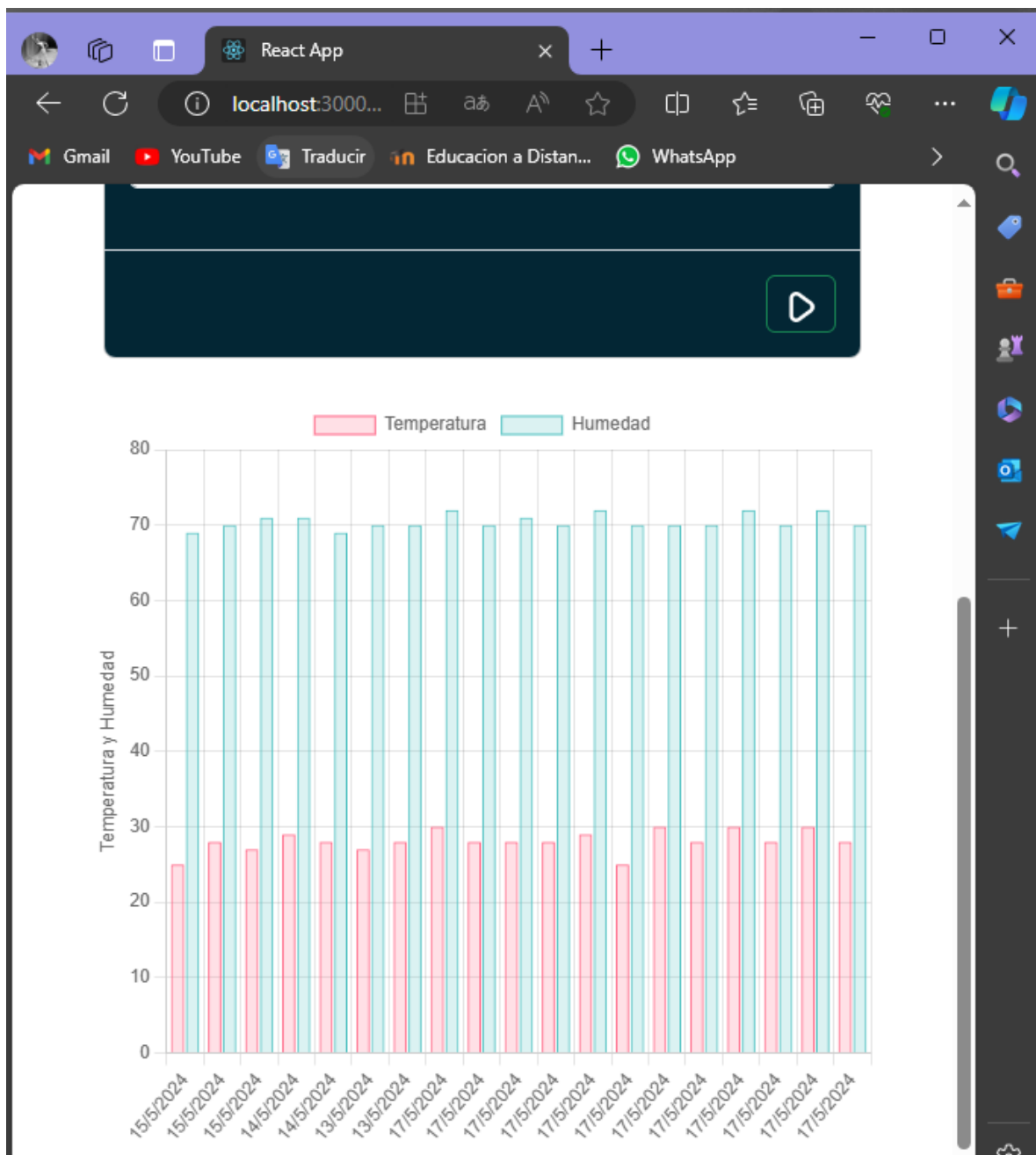


Figura 18 Resultado de la interfaz "Página de monitoreo" gráfica

En la figura 18 se observa un ejemplo de cómo se muestran las fechas y los resultados para el usuario, según los filtros que él mismo haya realizado por medio de las fechas.

10. Conclusiones

- Metodología Scrum: Al implementar la metodología Scrum, no solo se logró una identificación precisa de los requisitos del sistema, sino que también se mejoró la capacidad de respuesta ante cambios y ajustes durante el desarrollo del proyecto. El enfoque iterativo permitió priorizar las necesidades críticas para el control de variables ambientales, lo que impactó directamente en la capacidad de mejorar las condiciones de cultivo de las setas, resultando en una producción más eficiente.
- Diagramas de flujo y clases: La utilización de diagramas de flujo y de clases permitió una representación clara de los procesos de desarrollo y el flujo de información. Esto no solo facilitó la comprensión del diseño de la interfaz y su interconexión con los diferentes componentes, sino que también optimizó la eficiencia del sistema en la recolección y procesamiento de datos ambientales. Esta mejora en el diseño técnico del sistema tuvo un impacto directo en la producción, ya que las decisiones informadas permitieron optimizar las condiciones de crecimiento de las setas.
- Base de datos: La creación de una base de datos eficiente con las herramientas seleccionadas permitió el almacenamiento seguro y el manejo adecuado de los datos ambientales. Esto facilitó un monitoreo en tiempo real y una gestión más eficaz de las variables críticas, como la humedad y la temperatura, lo que mejoró las condiciones de cultivo. Esta infraestructura confiable fue esencial para aumentar la producción de setas al permitir un control más preciso y automatizado de las variables ambientales.
- Interfaz de usuario: El desarrollo de la interfaz con tecnologías modernas permitió a los usuarios controlar y visualizar las variables ambientales de manera más efectiva. Esto resultó en una mejor capacidad para ajustar las condiciones de cultivo en tiempo real, maximizando así la producción de setas. La combinación de usabilidad y rendimiento de la interfaz no solo mejoró la experiencia del usuario, sino que también aseguró un control más riguroso de los factores ambientales que influyen directamente en la producción.

11. Referencias

- Adam Boduch, R. D. (2020). *React y React Native : una guía práctica completa para el desarrollo web y móvil moderno con React.js*. Packt Publishing Ltd.
- Aguirre, S. (2020). *JSON - Vol.1: Primeros pasos - Sintaxis - Tipos de datos*. RedUsers.
- AGUIRRE, S. (2024). *Laravel: Curso completo*. Ediciones de la U.
- Cabello, M. V. (2010). *Introducción a las bases de datos relacionales*. Vision libros .
- Camazón, J. N. (2011). *Servidores de aplicaciones web (aplicaciones web)*. Editex.
- Cíceri, M. (2019). *Introducción a Laravel: Aplicaciones robustas y a gran escala*. RedUsers.
- Cobo, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Díaz de Santos .
- Díaz, I. (2017). *Granjas verticales: una respuesta sostenible al crecimiento urbano*. Universidad Tecnológica de Panamá: Prisma.
- DuBois, P. (2013). *MySQL*. Addison-Wesley Professional.
- Gomez, M. M. (2012). *Bases de datos*. Entrevistador.
- Gonzalías, Y. F. (2015). *Diseño e implementación de un sistema de control para maximizar la capacidad productiva de las plantas en granjas verticales por medio de luz artificial*. Universidad Autónoma de Occidente : BS thesis .
- Herrera, D. J. (2020). *Desarrollo de páginas web utilizando la tecnología React-Ant Design con librería de promesas Axios*. Mazatlán, Sinaloa.
- Kuo, B. (1996). *Sistemas de control automático*. Pearson Educación.
- Kuo, B. (2009). *Sistemas de control digital*. Innovación Educativa.
- Luna, A. C. (2024). *CREACIÓN DE PÁGINAS WEB: HTML*. ICB, S.L.
- Luna, F. (2019). *JavaScript - Aprende a programar en el lenguaje de la web*. RedUsers.
- Puciarelli, L. (2020). *Angular: TypeScript – Arquitectura – Instalación – Directivas y Bindings – Forms – Ruteo y más*. RedUsers.
- Puciarelli, L. (2020). *Node JS - Vol. 1: Instalación - Arquitectura - node y npm*. RedUsers.
- Ruiz Barea, R. (23 de 06 de 2023). *openaccess*. Obtenido de Universitat Oberta de Catalunya: <http://hdl.handle.net/10609/148140>
- Santillan, L. A. (2014). *Bases de datos en MySQL*. Universidad Oberte de Catalunya: academia.edu.
- Vahid Mirjalili, S. R. (2020). *Python Machine Learning*. Marcombo.
- Wood, D. (2022). *Diseño de interfaces: Introducción a la comunicación visual en el diseño de interfaces de usuario*. Parramón Paidotribo.