

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El presente trabajo se enfoca en el desarrollo de un sistema de monitoreo y control automatizado para tres invernaderos en una granja. El objetivo principal es optimizar las condiciones ambientales en los invernaderos para mejorar la producción agrícola. Los invernaderos son: un invernadero dedicado al cultivo de hongos, y dos invernaderos dedicados al cultivo de tomates y otras plantas.

Para el invernadero de hongos, se implementará un sistema de monitoreo que medirá los niveles de CO<sub>2</sub>, temperatura, humedad y el estado de un ventilador (activado o desactivado). Este monitoreo es crucial para mantener las condiciones óptimas necesarias para el crecimiento de los hongos ostra. El sistema automatizado encenderá el ventilador cuando los niveles de CO<sub>2</sub> alcancen un umbral predefinido, asegurando que el ambiente se mantenga propicio para el cultivo de los hongos. En los dos invernaderos dedicados a los tomates, el sistema se enfocará en medir la temperatura y la humedad del ambiente. Estos datos permitirán a los agricultores visualizar en tiempo real las condiciones dentro de los invernaderos y tomar decisiones informadas sobre la necesidad de riego.

La arquitectura del sistema se puede dividir en varias capas:

- **Sensores:** Cada invernadero estará equipado con sensores específicos para las métricas que se desean monitorizar:
  - Invernadero de hongos: Sensores de CO<sub>2</sub>, temperatura, humedad, y estado del ventilador.
  - Invernadero 1: Sensores de temperatura y humedad.
  - Invernadero 2: Sensores de temperatura y humedad.
- **Conectividad:** Los sensores del invernadero de hongos y del invernadero 1 se conectarán mediante WiFi y enviarán datos a un servidor MQTT. Los sensores del invernadero 2 utilizarán un módulo LoRaWAN para comunicarse con un gateway, que luego enviará los datos al servidor MQTT.
- **Procesamiento de Datos:** Un servidor back-end recogerá todos los datos enviados al servidor MQTT y los almacenará en una base de datos no relacional (MongoDB). Se desarrollará una API REST para que el front-end pueda acceder y visualizar los datos en tiempo real.
- **Visualización:** La aplicación web permitirá a los agricultores visualizar los datos actuales y pasados mediante gráficos y tablas. También podrán recibir alertas en tiempo real si se detectan condiciones anómalas (por ejemplo, niveles altos de CO<sub>2</sub>).

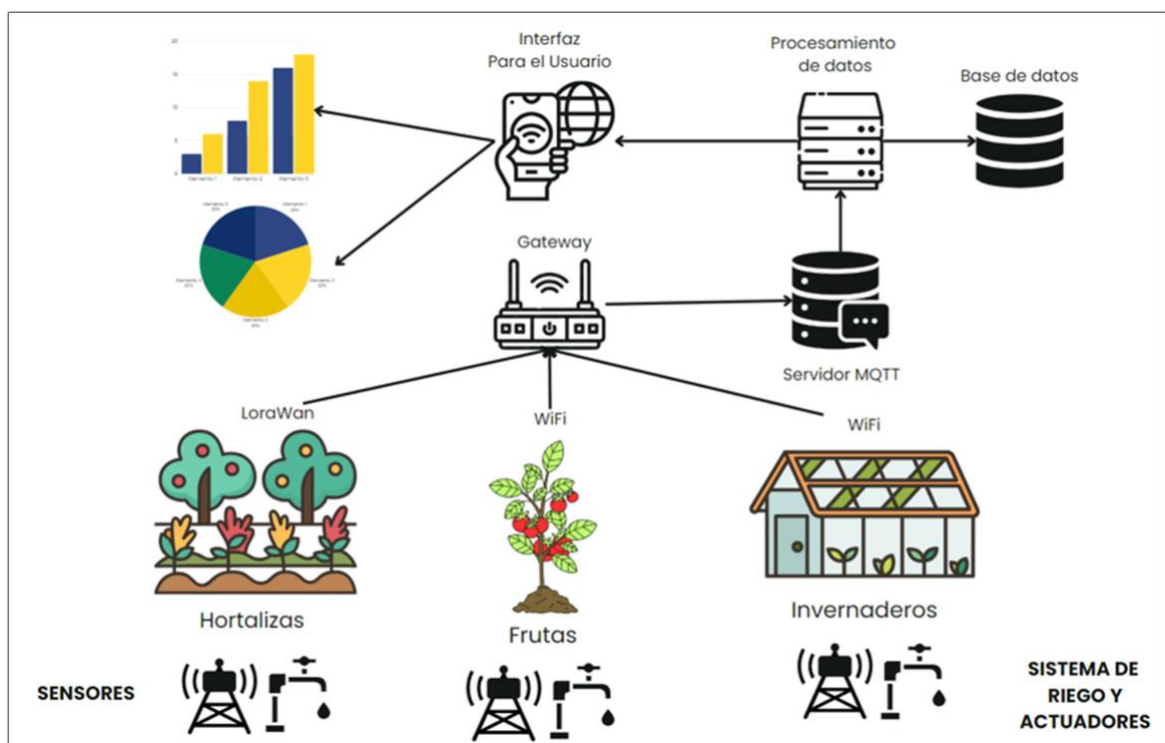
Lo explicado anteriormente se puede resumir mediante la **Figura 1**, donde se muestra la arquitectura sobre la cual se fundamenta el desarrollo del componente.

Para el diseño de las interfaces, se utilizará la herramienta Figma para crear mockups que muestren cómo se presentarán los datos y las funcionalidades del sistema. Estos mockups ayudarán a asegurar que la interfaz sea intuitiva y fácil de usar para los agricultores.

El desarrollo del sistema se llevará a cabo utilizando la metodología ágil SCRUM. Se utilizarán diversas tecnologías y lenguajes de desarrollo web, incluyendo HTML, CSS, JavaScript para el front-end, y Node.js y MongoDB para el back-end. La aplicación web tendrá varias pestañas con diferentes funcionalidades, tales como:

- Visualización en tiempo real de los datos más recientes de cada invernadero.
- Gráficas históricas que permitan seleccionar rangos de fechas y exportar datos en formato EXCEL.
- Alertas y notificaciones basadas en las condiciones monitorizadas.

Se realizarán pruebas de usabilidad para medir la efectividad de la aplicación web y obtener retroalimentación de los usuarios. Con los resultados de estas pruebas, se llevará a cabo un refinamiento del prototipo para mejorar la experiencia del usuario. La **figura 1** ilustra la arquitectura del sistema desarrollado, mostrando cómo se interconectan los diferentes componentes.



**Figura 1.** Arquitectura del sistema

## **1.1 Objetivo general**

Desarrollar una aplicación web que permita la visualización en tiempo real de las condiciones ambientales de los invernaderos, mediante el uso de tecnologías IoT y microservicios.

## **1.2 Objetivos específicos**

1. Crear y desarrollar interfaces de usuario con Figma, HTML, CSS y JavaScript para una experiencia fluida y accesible.
2. Configurar la comunicación con la API REST para mostrar datos ambientales en tiempo real en la interfaz de usuario.
3. Implementar gráficos que muestren los datos ambientales.
4. Ejecutar pruebas con usuarios finales para mejorar la usabilidad y funcionalidad de la aplicación web.

## **1.3 Alcance**

El alcance de este proyecto incluye el diseño e implementación de la capa front-end de una aplicación web para monitorizar en tiempo real las condiciones ambientales de los invernaderos y recibir alertas sobre parámetros críticos. Esto implica crear interfaces utilizando Figma, HTML, CSS, JavaScript y React; integrar la API REST para visualizar datos en tiempo real; implementar gráficos que muestren los datos ambientales; y desarrollar notificaciones visuales para alertas. El desarrollo seguirá la metodología ágil SCRUM, asegurando entregas periódicas y ajustes continuos basados en la retroalimentación de los usuarios.

## **1.4 Marco teórico**

Esta sección proporciona los fundamentos necesarios para comprender y contextualizar el desarrollo del Front-end del sistema de monitoreo ambiental para los invernaderos de la granja, permitiendo así la creación de una solución frente a las necesidades específicas de los agricultores. En este sentido, las siguientes subsecciones abordan los constructos, conceptos teóricos, herramientas y metodologías de trabajo involucradas en el desarrollo del front-end.

### **Desarrollo de Aplicaciones Web: el Front-end**

Las aplicaciones web han transformado nuestra interacción con la información y los servicios en línea. En el desarrollo del sistema de monitoreo ambiental para invernaderos, es esencial entender los principios de las aplicaciones web para crear interfaces de usuario que satisfagan las necesidades de los agricultores.

Una aplicación web (web app) es un software que se ejecuta en un servidor al que se accede generalmente desde un navegador [1]. Las aplicaciones web proporcionan interfaces de usuario interactivas, permitiendo a los usuarios realizar tareas como visualizar y manipular información, realizar transacciones y colaborar en línea. Además, una web app puede integrarse con otras aplicaciones [2] mediante servicios web, que son servicios remotos que proveen información o funcionalidades consumidas por otras aplicaciones [3]. A diferencia de las aplicaciones de escritorio, las aplicaciones web se usan directamente en los navegadores, sin necesidad de instalación ni actualización en el dispositivo del usuario. En el desarrollo de aplicaciones web, es fundamental comprender términos como HTTP, HTML, CSS y JavaScript.

HTTP, o protocolo de transferencia de hipertexto, es un protocolo sin estado utilizado para la comunicación entre un cliente y un servidor web a través de Internet. Este protocolo es esencial para la comunicación en la World Wide Web, permitiendo a los navegadores realizar solicitudes y recibir páginas web, contenido multimedia, anuncios y otros recursos desde un servidor web. La característica de ser sin estado de HTTP, donde cada conjunto de solicitud y respuesta es independiente, lo hace ideal para el desarrollo web en el front-end, ya que permite la creación de aplicaciones web dinámicas que responden a la interacción del usuario en tiempo real [4].

HTTP también puede utilizarse para la comunicación entre servidores, la exposición de servicios o el balanceo de carga entre servidores [5].

HTML es la base para estructurar el contenido de una página web. Es un lenguaje de marcado utilizado para crear páginas web que pueden ser visualizadas en los navegadores.

Por otro lado, las hojas de estilo en cascada (CSS) permiten gestionar la apariencia visual de los elementos HTML, controlando aspectos como colores, tamaños y formas [6].

JavaScript, un lenguaje de programación, permite junto con HTML y CSS, añadir interactividad en tiempo real desde el lado del cliente, incluyendo características como modales, wizards, validaciones y efectos dinámicos [7].

#### **1.4.1 Implementación del front-end con JavaScript**

El desarrollo de front-end con JavaScript implica la creación de interfaces gráficas que mejoran la experiencia del usuario. JavaScript es esencial para el desarrollo front-end, pero sus frameworks y Bibliotecas son los que realmente aceleran, simplifican y optimizan este proceso [8]. Entre los frameworks más utilizados hasta 2022 se encuentran Vue.js, Angular y React. React se considera más una Biblioteca con frameworks propios como Vite, que será analizado más adelante en este marco teórico [8].

Los frameworks de JavaScript ofrecen una variedad de características valiosas según el proyecto. Cada uno tiene un enfoque y herramientas especializadas que permiten a los desarrolladores abordar diferentes desafíos. Angular es ideal para aplicaciones web de gran escala y complejidad, gracias a su conjunto robusto de características y estructura definida. React se destaca por construir interfaces de usuario interactivas y reactivas, con componentes reutilizables y una capacidad eficiente para manejar grandes cantidades de datos. Vue.js, por su parte, combina simplicidad y flexibilidad, siendo una excelente opción para proyectos más pequeños o aquellos que requieren una curva de aprendizaje más suave.

Cada framework tiene sus propias fortalezas y debilidades, por lo que es crucial evaluar las características específicas de cada uno en relación con los requisitos del proyecto para tomar una decisión informada. La **tabla 1** muestra los resultados de pruebas de velocidad realizadas sobre Angular, React y Vue.js, con el objetivo de identificar posibles frameworks candidatos para el desarrollo del front-end del presente proyecto.

**Table 1.** Resultados de las pruebas de velocidad entre Angular, React y Vue [9].

Metric	Angular	React	Vue
First Contentful Paint	1.1s	0.8s	1.2s
Total Blocking Time	200ms	0ms	0ms
Speed Index	1.2s	0.8s	1.7s
Largest Contentful Paint	2.3s	2.3s	2.4s
Cumulative Layout Shift	Same	Same	Same
Overall Performance Rating	79	82	81

Bundle Size (gzipped)	62.3kb	44.5kb	34.7kb
--------------------------	--------	--------	--------

Donde:

- **First Contentful Paint:** Mide el tiempo que tarda el navegador en renderizar el primer bit de contenido del DOM.
- **Total Blocking Time:** Mide el tiempo total en el que la página está bloqueada y no responde a las interacciones del usuario.
- **Speed Index:** Mide la rapidez con la que el contenido de una página es visualmente mostrado durante la carga.
- **Largest Contentful Paint:** Mide el tiempo que tarda en renderizarse el contenido más grande visible en la ventana gráfica.
- **Cumulative Layout Shift:** Mide la estabilidad visual de una página al evaluar cuánto se mueven los elementos en la pantalla durante la carga.
- **Overall Performance Rating:** s una puntuación general dada por herramientas como PageSpeed Insights que considera varias métricas de rendimiento.
- **Bundle Size (gzipped):** Mide el tamaño del archivo comprimido (gzip) que se envía desde el servidor al navegador.

La comparación de los resultados de las pruebas y el enfoque en la creación de interfaces de usuario interactivas y eficientes mediante el uso de componentes reutilizables es crucial para seleccionar el framework adecuado para el desarrollo del Front-end del proyecto. Cada framework tiene fortalezas en distintas métricas de rendimiento y ofrece diversas técnicas de optimización que pueden mejorar estas métricas.

### 1.4.2 JSON

JavaScript Object Notation (JSON) es un formato estructurado para el intercambio de datos, ampliamente utilizado debido a su simplicidad y facilidad de uso tanto para desarrolladores como para sistemas. En el desarrollo de front-end, JSON es especialmente útil para la comunicación entre cliente y servidor y para el consumo de servicios web [10] [11]. En JavaScript, se utilizan dos métodos principales para manejar JSON: "json.parse()" que convierte una cadena JSON en un objeto JavaScript, y "json.stringify()" que convierte un objeto JavaScript en una cadena JSON. La **Figura 2** y la **Figura 3** ilustran ejemplos de estos métodos, respectivamente.



```
1 let jsonString = '{"code":200,"status":"success","details":{"item":"example"}}';
2 let parsedObject = JSON.parse(jsonString);
3 console.log(parsedObject.details.item); // example
```

**Figura 2.** Conversión de JSON a objeto JavaScript.



```
1 let objectData = {item: 'example'};
2 let jsonStringified = JSON.stringify(objectData);
3 console.log(jsonStringified); // {"item":"example"}
```

**Figura 3.** Conversión de objeto JavaScript a JSON.

JSON es una opción ideal para realizar las peticiones al servidor web del back-end en este proyecto debido a su facilidad de uso y alta compatibilidad.

### 1.4.3 TypeScript en el desarrollo front-end

TypeScript (TS) es un lenguaje de programación creado por Microsoft que amplía las capacidades de JavaScript. Diseñado para uso empresarial, TS no solo se basa en JavaScript, sino que también extiende sus funcionalidades. Algunas de las características que lo hacen ideal para el desarrollo front-end incluyen:

- **Tipado estático:** TypeScript agrega tipado a JavaScript, permitiendo saber qué tipo de datos se espera de una variable o método. Esto ayuda a detectar errores durante la compilación en lugar de en tiempo de ejecución, mejorando así la calidad del código y reduciendo el tiempo dedicado a la corrección de errores [12].
- **Mejor soporte de herramientas:** TypeScript proporciona un mejor soporte para herramientas de completado de código, corrección de errores y refactorización en comparación con JavaScript, gracias a su sistema de tipos [12].
- **Documentación mejorada:** Al permitir la definición de tipos, TypeScript facilita la creación de documentación, con Bibliotecas como "TypeDocs" que se integran casi automáticamente [12].

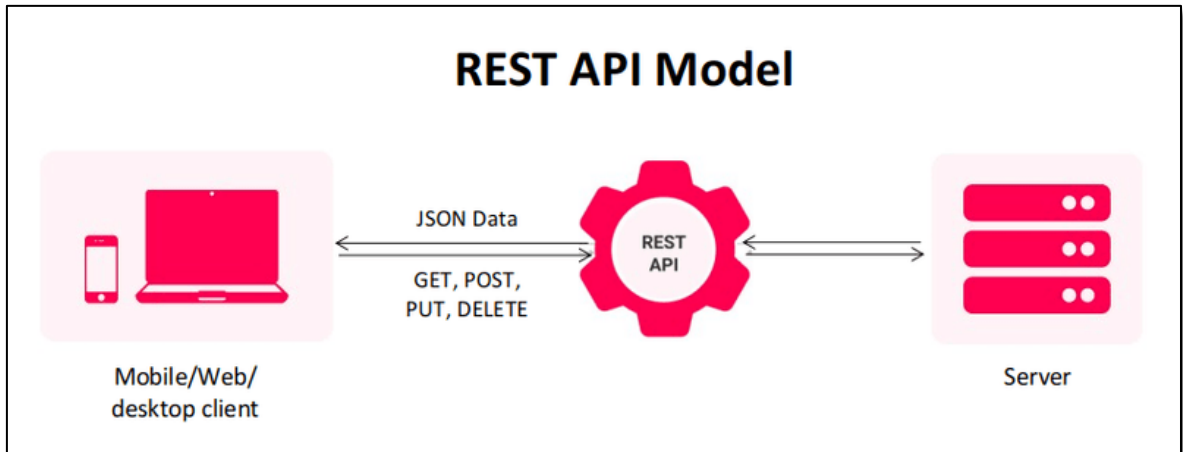
TypeScript se ha vuelto esencial en el desarrollo front-end, ya que mejora la calidad, mantenibilidad y productividad, especialmente en aplicaciones web complejas. Su sistema de tipos estáticos permite identificar y prevenir errores antes de que ocurran, resultando en un código más robusto y confiable. Adoptar TypeScript en proyectos de front-end es una decisión inteligente que conduce a un desarrollo más eficiente y exitoso [12].

#### **1.4.4 API REST para la recolección de datos**

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de protocolos, rutinas y herramientas diseñadas para el desarrollo de software, definiendo cómo los componentes de software deben comunicarse entre sí [13]. Las APIs permiten acceder a información o funcionalidades de una aplicación o servicio, o bien exponerlas para ser utilizadas por otras aplicaciones o servicios. Estas pueden ser accedidas desde diferentes lenguajes y plataformas, utilizando diversos protocolos de comunicación como REST, SOAP o GraphQL [13].

El Estado Representacional de Transferencia (REST) es un estilo arquitectónico que establece una serie de reglas y lineamientos para la creación de servicios web [18]. Una API REST es un tipo de servicio web que utiliza HTTP para realizar operaciones sobre recursos entre dos aplicaciones de software, incluyendo las operaciones GET, POST, PUT y DELETE. El propósito de estas operaciones es facilitar la comunicación entre un cliente, que solicita y recibe los recursos, y un servidor REST, que contiene y expone los recursos. Las APIs REST utilizan Identificadores Uniformes de Recursos (URIs) para identificar recursos específicos y devolver una representación de esos recursos en formatos como JSON y XML [13].





**Figura 4.** Ciclo de vida de una solicitud API REST [14].

Las API REST se utilizan ampliamente en servicios en la nube y aplicaciones web debido a su facilidad para permitir la comunicación entre dos aplicaciones o servicios de software. Además, la capacidad de comunicación asíncrona que ofrecen las API REST ha sido un factor decisivo para seleccionarlas en el presente proyecto. La **Figura 4** ilustra el ciclo de vida de una solicitud API REST, proporcionando una visión detallada del proceso de comunicación.

#### 1.4.5 Vite

Vite es un framework moderno diseñado para el desarrollo rápido de aplicaciones web, ofreciendo una configuración mínima y una experiencia de desarrollo altamente eficiente. Su arquitectura se basa en un servidor de desarrollo ultrarrápido y una optimización de construcción que aprovecha las capacidades modernas del navegador, mejorando significativamente la velocidad de inicio y la recarga en caliente [15].

Una característica clave de Vite es su uso de módulos ES (ESM), que permite servir cada módulo directamente al navegador en lugar de agrupar todo el código en un solo archivo. Esto resulta en tiempos de compilación casi instantáneos, lo cual es especialmente útil en proyectos de gran escala donde la eficiencia es crucial. Vite permite a los desarrolladores integrar fácilmente frameworks populares como React, Vue o Svelte, beneficiándose de su entorno de desarrollo optimizado [16].

Entre las características principales de Vite se encuentran:

- **Recarga en caliente ultrarrápida (Hot Module Replacement, HMR):** Permite actualizaciones instantáneas sin recargar la página completa [16].

- **Compilación eficiente:** Usa Rollup para generar activos estáticos altamente optimizados para producción [17].
- **Interfaces de usuario completamente tipadas:** Facilita el desarrollo mediante soporte de APIs completamente tipadas [17].
- **Extensible:** Mediante su API de plugins, que permite agregar funcionalidades adicionales según las necesidades del proyecto [15].

Vite se ha establecido como una herramienta clave en el desarrollo front-end, facilitando un desarrollo más rápido y eficiente que incrementa la productividad y mejora el código [15] [16].

En el desarrollo del componente de este proyecto, Vite se utiliza para la implementación de interfaces de usuario. Su arquitectura basada en módulos ES y su capacidad para una recarga en caliente rápida permiten crear una experiencia de usuario fluida e interactiva. La combinación de Vite con frameworks como React proporciona una base sólida para desarrollar un front-end robusto y escalable.

#### 1.4.6 Pruebas unitarias con JEST

Jest es un framework de pruebas creado para proyectos web basados en JavaScript, que se adapta perfectamente al diseño basado en componentes de frameworks modernos como Vite. Esta compatibilidad convierte a Jest en una excelente opción para realizar pruebas en proyectos que usan Vite. A diferencia de otras herramientas, Jest incluye pruebas de interfaces de usuario dentro de sus pruebas unitarias, ofreciendo una integración rápida y eficiente [18].

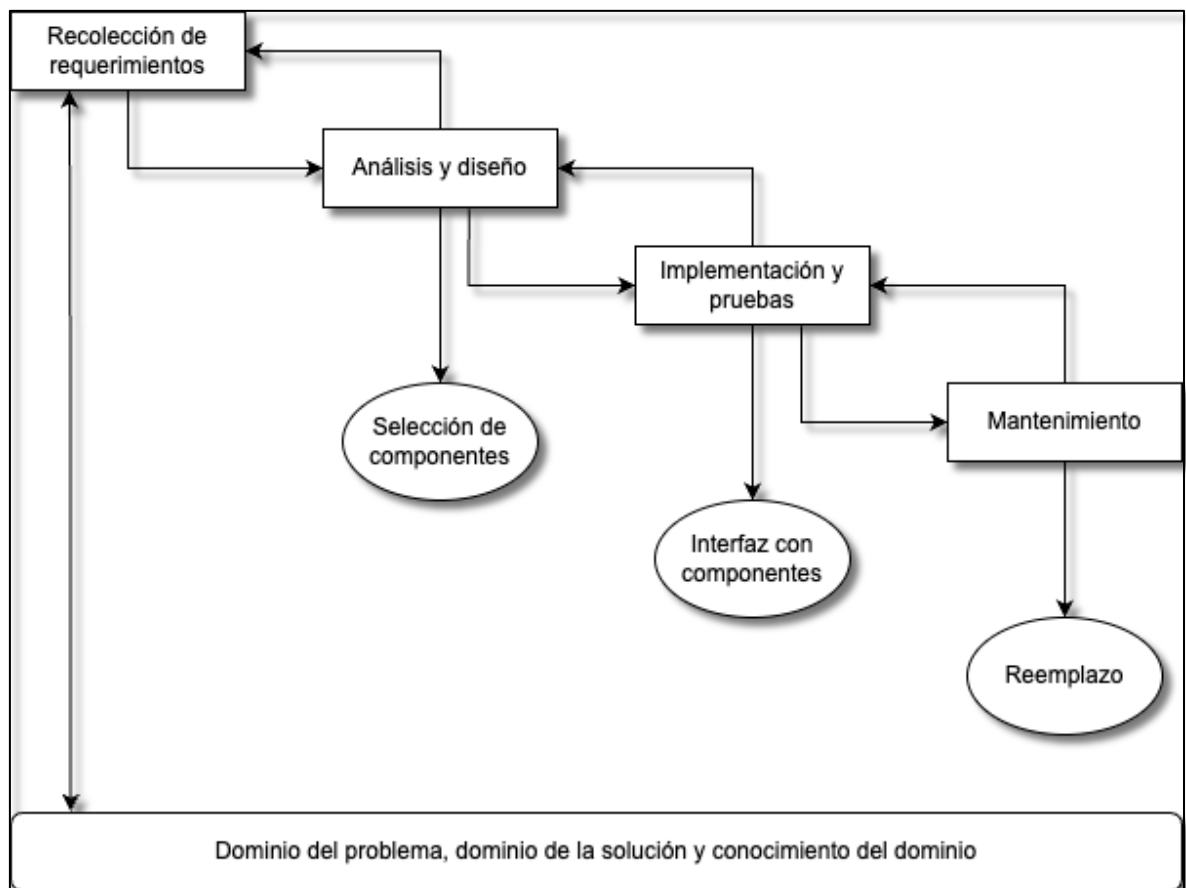
A partir de la versión 12.0, Jest ha mejorado significativamente su rendimiento gracias a su compilador basado en Rust, que acelera los tiempos de generación de builds hasta cinco veces. Además, Jest incorpora nativamente características específicas para proyectos modernos, tales como:

- **Generación de mocks para Bibliotecas y archivos:** Facilita la simulación de módulos externos para pruebas [18].
- **Renderizado de componentes:** Permite realizar pruebas de renderizado de componentes, algo que no es posible de forma predeterminada en muchos frameworks de prueba [18].

- Configuración optimizada: Permite evitar pruebas innecesarias y automatiza las transformaciones de archivos para mejorar la eficiencia del proceso de pruebas [18].

#### 1.4.7 Desarrollo orientado a componentes

El desarrollo de aplicaciones en React y Vite sigue un enfoque orientado a componentes [19]. Los componentes se basan en funciones de JavaScript, creando código independiente y reutilizable que retorna HTML en lugar de JavaScript [20]. Este enfoque fomenta la creación de componentes reutilizables, simples, flexibles y fáciles de mantener a lo largo de todo el ciclo de desarrollo [21]. La **Figura 5** ilustra cómo se aplica este enfoque a lo largo del proyecto.



**Figura 5.** Ciclo de vida del desarrollo basado en componentes

La imagen muestra el ciclo de vida del desarrollo basado en componentes, que incluye las etapas de recolección de requisitos, análisis y diseño, selección de componentes, implementación y pruebas, integración con componentes, mantenimiento y reemplazo de

componentes. Este enfoque asegura que los componentes sean seleccionados y diseñados desde el principio, lo que facilita su reutilización y mantenimiento continuo.

La popularidad de este enfoque ha impulsado la adopción de herramientas como Jest, que permite realizar pruebas unitarias directamente sobre los componentes, incluyendo las interfaces de usuario que estos exponen. Esto subraya la importancia y el impacto positivo del desarrollo orientado a componentes en la creación de aplicaciones [22].

#### **1.4.8 Git**

Git es una herramienta que permite a los desarrolladores llevar un registro detallado de los cambios realizados en el código de sus proyectos a lo largo del tiempo [23]. Funciona tomando instantáneas del estado del código en diferentes momentos, lo que facilita el seguimiento y la gestión de las modificaciones.

Git opera bajo un modelo distribuido, donde cada miembro del equipo de desarrollo tiene una copia completa del repositorio en su máquina local [23]. Esta estructura permite a los desarrolladores trabajar de manera remota y sin conexión, y luego integrar sus cambios a la rama principal cuando lo consideren oportuno [23].

Además de gestionar versiones, Git ofrece herramientas como la creación de ramas, lo que permite desarrollar nuevas características y corregir errores de manera aislada antes de fusionarlos con la rama principal [23]. También permite revisar los cambios realizados (commits), ver cuándo se hicieron y revertirlos si es necesario [23].

### **Diseño y prototipado**

#### **1.4.9 Diseño centrado en el usuario**

El diseño centrado en el usuario (DCU) es un enfoque para crear productos y servicios que se basan en comprender las metas, necesidades y tareas de los usuarios objetivo [24]. Aplicar DCU en el desarrollo del front-end implica seguir varios pasos clave:

1. **Identificar a los usuarios objetivo:** Es fundamental determinar quiénes son los usuarios objetivo y realizar una investigación exhaustiva sobre sus necesidades, preferencias y metas. Esto se logra mediante entrevistas, encuestas y pruebas de usabilidad para entender mejor las tareas que deben realizar y las necesidades que buscan satisfacer.

2. **Crear personas:** Basándose en la investigación, se definen perfiles de personas que representan a grupos de usuarios finales reales con necesidades similares [24].
3. **Desarrollar escenarios para los usuarios:** Se crean escenarios que describen cómo los usuarios interactuarán con el sistema, ayudando a anticipar su comportamiento y necesidades.
4. **Crear prototipos:** Se desarrollan prototipos interactivos del sistema para probarlos con los usuarios [24].
5. **Probar con usuarios:** Los prototipos se prueban con usuarios reales para obtener retroalimentación sobre el diseño e identificar áreas de mejora.
6. **Iterar y refinar:** La retroalimentación obtenida se utiliza para mejorar el prototipo, que luego se vuelve a probar con los usuarios para asegurar que se ajusta a sus necesidades.

Al seguir estos pasos, se espera que el software desarrollado esté alineado con las necesidades y preferencias de los usuarios finales, lo que aumenta la satisfacción y mejora la experiencia de usuario [24].

#### **1.4.10 Herramienta de prototipado (Figma)**

Figma es una herramienta de diseño colaborativa basada en la nube, utilizada para crear interfaces de usuario, prototipos y gráficos. Permite a los diseñadores trabajar juntos en tiempo real, compartir proyectos y recibir retroalimentación instantánea. Figma soporta la creación de componentes reutilizables, diseño vectorial y tiene integración con otras herramientas y plataformas, lo que facilita un flujo de trabajo eficiente en proyectos de diseño digital [25].

#### **1.4.11 Scrum**

Scrum es un framework ágil utilizado para gestionar proyectos de software complejos [26]. Facilita la organización y asignación de responsabilidades, dividiéndolas en sprints, que son períodos de tiempo de una a cuatro semanas [26]. En cada sprint, el equipo de desarrollo debe entregar un incremento funcional del producto al cliente [26].

Scrum define roles, eventos y artefactos específicos. Los principales roles son [26]:

- **Product Owner:** Responsable de definir y priorizar el product backlog, que es la lista de requerimientos del producto.
- **Scrum Master:** Encargado de asegurar que se sigan las pautas de Scrum y de mantener al equipo trabajando de manera ordenada y eficiente.
- **Equipo de Desarrollo:** Grupo de personas que desarrollan y entregan un incremento funcional al final de cada sprint. Este equipo debe tener habilidades diversas, siendo expertos en su área pero también apoyando en otras tareas del equipo.

Los eventos principales en Scrum son [27]:

- **Sprint Planning:** Reunión donde el equipo planea el trabajo para el próximo sprint.
- **Daily Scrum:** Reuniones diarias de aproximadamente 15 minutos para discutir el progreso, planes y obstáculos.
- **Sprint Review:** Reunión para demostrar e inspeccionar el incremento funcional y recibir retroalimentación.
- **Sprint Retrospective:** Reunión al final del sprint para reflexionar sobre el sprint pasado y encontrar formas de mejorar.

Los principales artefactos de Scrum son [26]:

1. **Product Backlog:** Lista de todos los requerimientos del proyecto, priorizados por el product owner.
2. **Sprint Backlog:** Lista de tareas para completar durante el sprint, creada por el equipo de desarrollo con un objetivo claro.
3. **Incremento:** Suma de todo lo completado del product backlog al final de un sprint.

Estos roles, eventos y artefactos se pueden aplicar al desarrollo del front-end siguiendo estos pasos [26]:

1. El product owner crea y define claramente el product backlog.
2. El equipo de desarrollo y el product owner planifican el sprint, asignando tareas específicas.
3. Se ejecuta el sprint y al final se revisa el trabajo completado para identificar mejoras para el próximo sprint.

4. Este ciclo se repite hasta completar todo el product backlog.

## 2 METODOLOGÍA

Esta sección detalla las acciones realizadas durante las iteraciones o Sprints llevadas a cabo para alcanzar el producto final. Con el fin de proporcionar al lector una visión global y estructurada de las actividades realizadas en cada Sprint, cada uno de ellos se describe siguiendo una estructura de cuatro partes:

1. **Introducción:** Explica el objetivo perseguido en el sprint.
2. **Planificación:** Detalla las historias de usuario (HU) relacionadas con el front-end que se abordaron en el sprint. Cada HU describe la funcionalidad específica del sistema, sus criterios de aceptación y las tareas asociadas.
3. **Implementación:** Describe el proceso de construcción desde la perspectiva de cada historia de usuario (HU), incluyendo los resultados visibles y los requisitos necesarios para su desarrollo.
4. **Retrospectiva:** Refleja la evaluación del equipo sobre los obstáculos y las oportunidades surgidas durante el desarrollo del Sprint.
5. **Evaluación de usabilidad y satisfacción del usuario (Solo en el Sprint final):**

Esta estructura de cuatro partes se aplica a todos los sprints, excepto al SPRINT 0. En particular, el SPRINT 0 aborda los aspectos iniciales y preparativos que son la base para los siguientes sprints. Además, es necesario mencionar que los elementos propios de scrum serán incluidos entre las 4 etapas mencionadas.

### 2.1 SPRINT 0: Configuración inicial y Preparación del Proyecto

#### Objetivo del Sprint 0:

El SPRINT 0 se centró en establecer las bases para el desarrollo del sistema de monitoreo para invernaderos, con un enfoque en la configuración y preparación del front-end utilizando React con Vite.

#### Actividades Clave del SPRINT 0:

##### 2.1.1 Definición de requisitos

Durante este sprint, se realizaron reuniones con los stakeholders para documentar los requisitos del sistema.

### **2.1.2 Recopilación de información, requerimientos de los usuarios y entrevistas con los stakeholders**

Para el desarrollo de nuestro sistema de monitoreo de invernaderos, realizamos un proceso de recopilación de información con los usuarios finales para entender sus necesidades y desafíos actuales. En esta sección, describimos los hallazgos de nuestras entrevistas y encuestas con los agricultores involucrados en el proyecto.

Durante nuestras entrevistas iniciales, descubrimos que los agricultores enfrentan varios problemas relacionados con la monitorización de las métricas de sus invernaderos en tiempo real. Un ejemplo concreto es el caso del encargado de los hongos, quien accidentalmente desconectó un enchufe crucial que regulaba el riego automático basado en niveles de CO<sub>2</sub>. Esta falta de información en tiempo real llevó a problemas en el cultivo. Los agricultores expresaron la necesidad de tener acceso a métricas actualizadas en una página web que presente gráficas de líneas claras y fáciles de interpretar. Este acceso permitiría una gestión más eficiente y reduciría errores humanos.

#### **Resultados de la Encuesta**

Realizamos una encuesta a los agricultores para profundizar en sus necesidades específicas. A continuación, se presentan las preguntas y respuestas clave recopiladas:

##### **1. Información General sobre el Proceso de Cultivo:**

- Pregunta: ¿Podrías describir el proceso típico de cultivo de los hongos ostra y las variables más críticas que afectan su crecimiento?
  - Respuesta: Es necesario sacar la semilla del hongo, germinarla, robustecerla en frascos y luego poner la mezcla en los baldes. Mantener una humedad superior al 65% es crucial. Aunque el CO<sub>2</sub> no es tan crítico, sería útil implementar un ventilador que se encienda al alcanzar ciertos niveles.

##### **2. Uso del Dashboard:**

- Pregunta: ¿Qué información específica sobre el ambiente del invernadero necesitas ver en el dashboard?
  - Respuesta: Humedad del ambiente y CO<sub>2</sub> dentro del invernadero.
- Pregunta: ¿Con qué frecuencia necesitas actualizar los datos mostrados en el dashboard?
  - Respuesta: Sería útil actualizar los datos cada 4 horas, entre las 9 y las 12. Es importante poder modificar estos intervalos según sea necesario.

##### **3. Visualización de Datos:**



- Pregunta: ¿Prefieres algún tipo específico de visualización para los datos, como gráficos de barras, líneas, o indicadores?
    - Respuesta: Las gráficas de líneas serían útiles para mostrar los valores actuales de la humedad y del CO2.
  - Pregunta: ¿Hay algún color o diseño que prefieras para ayudar a interpretar rápidamente los datos?
    - Respuesta: Sería bueno que las líneas en las gráficas cambien a rojo cuando los valores estén fuera de los rangos deseados, complementadas con colores verde o azul para valores normales.
- 4. Alertas y Notificaciones:**
- Pregunta: ¿Qué tipo de alertas consideras cruciales recibir en el dashboard?
    - Respuesta: Notificaciones cuando los valores de CO2 o humedad alcancen niveles críticos.
  - Pregunta: ¿Cómo prefieres ser notificado sobre condiciones críticas o cambios importantes en el ambiente del invernadero?
    - Respuesta: Notificaciones similares a las que se reciben en Facebook.
- 5. Funcionalidades Específicas:**
- Pregunta: Aunque no podrás controlar los dispositivos directamente desde el dashboard, ¿hay alguna funcionalidad relacionada con el estado del ventilador o otros dispositivos que te gustaría tener?
    - Respuesta: El control del ventilador sería suficiente por el momento.
  - Pregunta: ¿Te gustaría poder ver el historial de datos y eventos anteriores en el dashboard?
    - Respuesta: Sí, me gustaría ver datos de días anteriores y poder hacer comparaciones.
- 6. Personalización y Preferencias:**
- Pregunta: ¿Hay algún ajuste específico o personalización que te gustaría poder hacer en el dashboard?
    - Respuesta: Poder comparar los valores de humedad y CO2 entre diferentes días.
- 7. Limitaciones y Desafíos Actuales:**
- Pregunta: ¿Hay algún problema o limitación con el sistema actual que esperas que el nuevo dashboard solucione?
    - Respuesta: La automatización del ventilador basada en los niveles de CO2 sería una mejora significativa.
- 8. Expectativas y Resultados Deseados:**

- Pregunta: ¿Qué esperas lograr con la implementación del nuevo dashboard?
  - Respuesta: Obtener reportes históricos de las métricas y visualizar datos en intervalos específicos.
- Pregunta: ¿Cómo definirías un éxito para este proyecto?
  - Respuesta: La capacidad de visualizar datos en un plazo de 5 minutos y acceder a un historial de métricas comparable entre diferentes días.

La información recopilada de los agricultores ha sido fundamental para diseñar un sistema de monitoreo que responda a sus necesidades. A partir de estos datos, hemos desarrollado un conjunto de historias de usuario y prototipos iniciales que guiarán el desarrollo de nuestro dashboard. Este enfoque centrado en el usuario garantiza que nuestro sistema será una herramienta valiosa para la gestión eficiente de los invernaderos.

### 2.1.3 Historias de usuario

- **Historia de Usuario 1: Visualización de Parámetros del Invernadero**

**Como** encargado del invernadero,

**Quiero** visualizar en tiempo real la temperatura y humedad en los invernaderos de tomates, y la temperatura, humedad, CO2 y estado del ventilador en el invernadero de hongos ostra,

**Para** asegurarme de que están dentro de los parámetros adecuados para el crecimiento óptimo.

**Criterios de Aceptación:**

1. Los datos deben actualizarse automáticamente en el dashboard cada 4 horas (preferiblemente cada 5 minutos).
2. Los valores actuales deben mostrarse en gráficos de líneas con colores distintivos: rojo para valores críticos y verde o azul para valores normales.

- **Historia de Usuario 2: Alertas de Condiciones Críticas**

**Como** encargado del invernadero,

**Quiero** recibir alertas automáticas en el dashboard cuando los valores de temperatura o humedad en los invernaderos de tomates, y de temperatura, humedad o CO2 en el invernadero de hongos ostra, alcancen niveles críticos,

**Para** tomar acciones correctivas de manera oportuna.

**Criterios de Aceptación:**

1. Las alertas deben ser visibles en el dashboard similar a las notificaciones de Facebook.
2. El sistema debe permitir al usuario configurar los umbrales de alerta para temperatura, humedad y CO2.
3. Las alertas deben incluir detalles específicos sobre el valor alcanzado y la hora de la alerta.

- **Historia de Usuario 3: Historial y Comparación de Datos**

**Como** encargado del invernadero,

**Quiero** consultar y comparar el historial de datos de temperatura y humedad para los invernaderos de tomates, y de temperatura, humedad y CO2 para el invernadero de hongos ostra,

**Para** analizar tendencias y mejorar las prácticas de manejo del invernadero.

**Criterios de Aceptación:**

1. El dashboard debe permitir al usuario visualizar el historial de datos en gráficos y tablas.
2. Debe ser posible comparar los datos de diferentes días para observar tendencias.
3. La interfaz debe permitir seleccionar fácilmente los días para comparación y mostrar los datos comparativos de manera clara y entendible.

- **Historia de Usuario 4: Automatización de Ventilador**

**Como** encargado del invernadero de hongos ostra,

**Quiero** que el sistema active automáticamente el ventilador cuando los niveles de CO2 superen un umbral preestablecido,

**Para** mantener automáticamente un ambiente adecuado para el crecimiento de los hongos.

**Criterios de Aceptación:**

1. El sistema debe monitorear continuamente los niveles de CO2 y mostrar cuando el ventilador esté encendido.

- **Historia de Usuario 5: Personalización del Dashboard**

**Como** encargado del invernadero,

**Quiero** personalizar la visualización del dashboard, incluyendo la selección de temas oscuros o claros y la disposición de las gráficas,

**Para** adaptar la interfaz a mis preferencias y facilitar la interpretación de los datos.

**Criterios de Aceptación:**

1. El dashboard debe ofrecer opciones de personalización, incluyendo un selector de tema oscuro o claro.

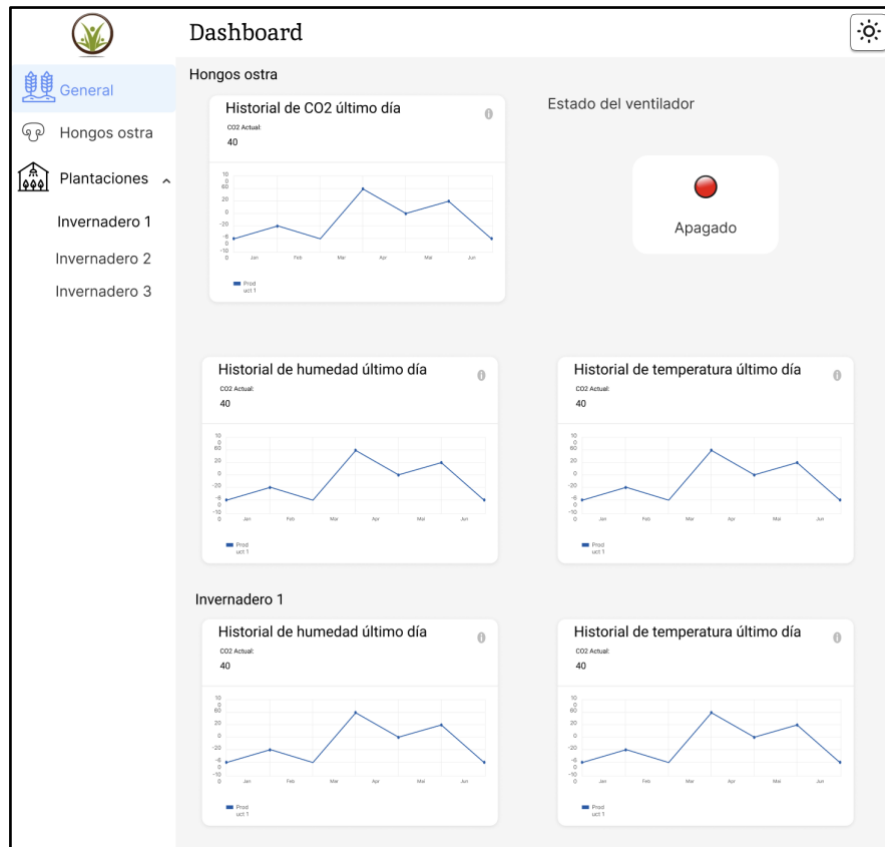
#### **2.1.4 Prototipo inicial**

El proceso de creación de los prototipos iniciales del sistema se llevó a cabo utilizando Figma. Se consideraron como base los perfiles de usuario previamente definidos y la funcionalidad existente identificada en la encuesta actual. Los prototipos de baja fidelidad se centraron en capturar la estructura general de las interfaces y las principales funcionalidades del sistema, sin detallar aspectos visuales o de diseño. El enfoque principal fue garantizar la funcionalidad y la usabilidad del sistema, adaptándolo a las necesidades de los perfiles de usuario definidos.

Una vez finalizados los prototipos de baja fidelidad, se realizó una revisión interna para evaluar la coherencia de la estructura y la funcionalidad del sistema. Se realizaron ajustes y refinamientos necesarios antes del diseño detallado que se llevaría a cabo durante los siguientes sprints. Aunque no se realizaron pruebas de usabilidad formales en esta etapa, se llevaron a cabo iteraciones y mejoras basadas en la retroalimentación personal y la revisión del product owner.

Es importante mencionar que el prototipo inicial se centró exclusivamente en la interfaz del usuario para el sistema de monitoreo de invernaderos. Si bien se consideraron los perfiles de usuario y la funcionalidad existente, se priorizó el diseño y la experiencia del usuario desde la perspectiva del encargado del invernadero. La parte administrativa del sistema, que involucra la configuración y gestión de umbrales, así como el análisis de reportes, aún no ha sido abordada en este prototipo inicial.

El siguiente enlace permite acceder directamente a este prototipo: [Prototipo en Figma](#). En la **Figura 6** ilustra el diseño del prototipo inicial.



**Figura 6.** Prototipo inicial para el dashboard.

## 2.2 Sprint 1

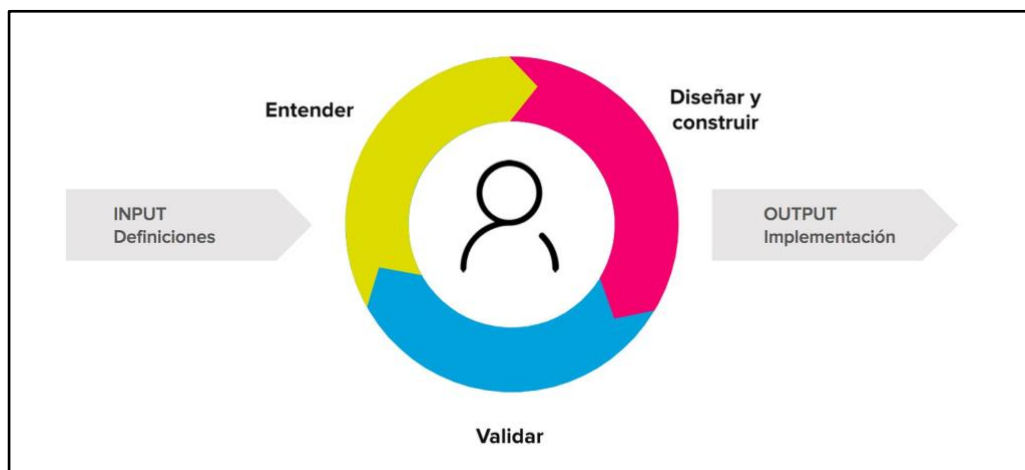
El objetivo del primer sprint es desarrollar dashboards comunes que permitan a los usuarios tener un control claro sobre la pestaña en la que se encuentran. Durante este período de 6 semanas, los esfuerzos se centrarán en diseñar una interfaz intuitiva que muestre gráficos de las métricas de manera separada, resaltando en color rojo los valores que estén por debajo de un cierto umbral. Además, se implementará el modo oscuro. El objetivo es crear un prototipo funcional basado en las historias de usuario que presentan mayor valor, constituyendo así el producto mínimo viable (MVP).

Durante este sprint, se desarrolló un MVP del sistema incluyendo los siguientes puntos como resumen:

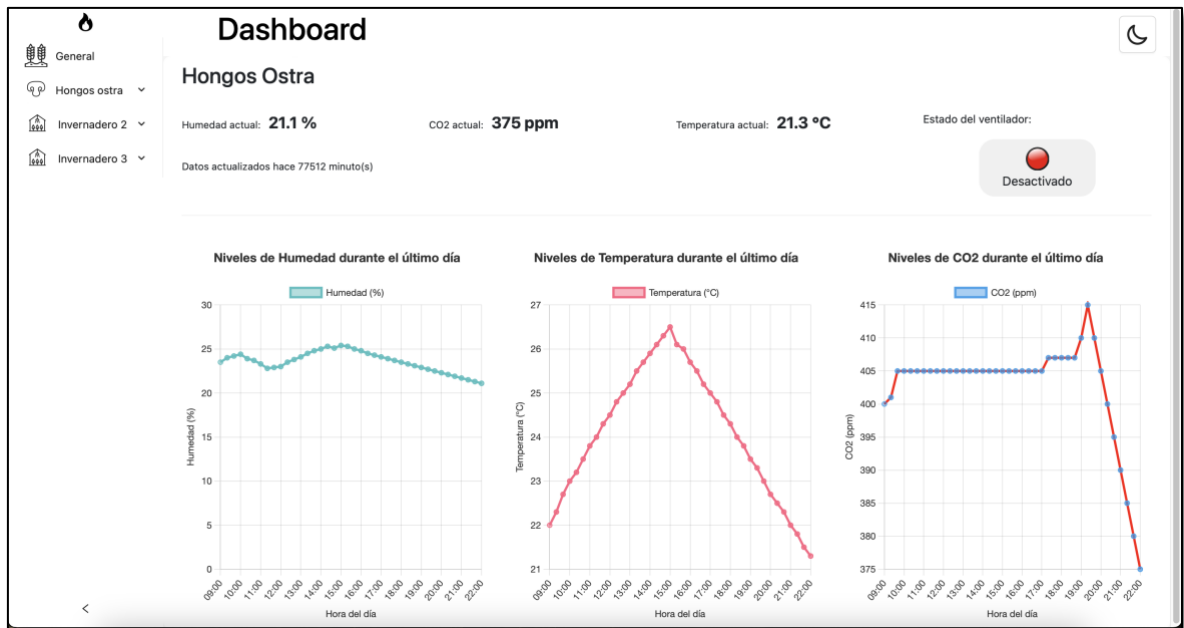
- **Estructuración de la Interfaz:** Se crearon dashboards que optimizan la usabilidad y facilitan la navegación para los usuarios. Se implementó una arquitectura basada en componentes utilizando Figma, lo que permitió una organización eficiente de los elementos y una mayor flexibilidad en el desarrollo.

- **Gráficos de Métricas:** Se desarrollaron gráficos separados para cada métrica (temperatura, humedad y CO2). Además, se resaltó en rojo aquellos que están por debajo del umbral establecido.
- **Modo Oscuro:** Se implementó la opción de modo oscuro para mejorar la experiencia del usuario en diferentes condiciones de iluminación.
- **Diseño Centrado en el Usuario (DCU):** Se adoptó un enfoque de diseño centrado en el usuario para abordar problemas de usabilidad identificados. Este enfoque incluyó las fases clave del DCU: investigación de usuarios, especificación de requisitos, diseño iterativo y evaluación con los usuarios. Las técnicas aplicadas en cada una de estas fases permitieron enfocar el diseño en las necesidades y preferencias de los usuarios, mejorando así la experiencia general del sistema. La **Figura 7** ilustra las fases del proceso de DCU aplicadas durante este sprint, mostrando las etapas de investigación, diseño, evaluación y refinamiento. Estas fases fueron fundamentales para asegurar que el diseño final del sistema se ajustara a las necesidades y expectativas de los usuarios. Como resultado de estos esfuerzos, se logró obtener un diseño mejorado y optimizado que se presenta en la **Figura 8 y Figura 9**.

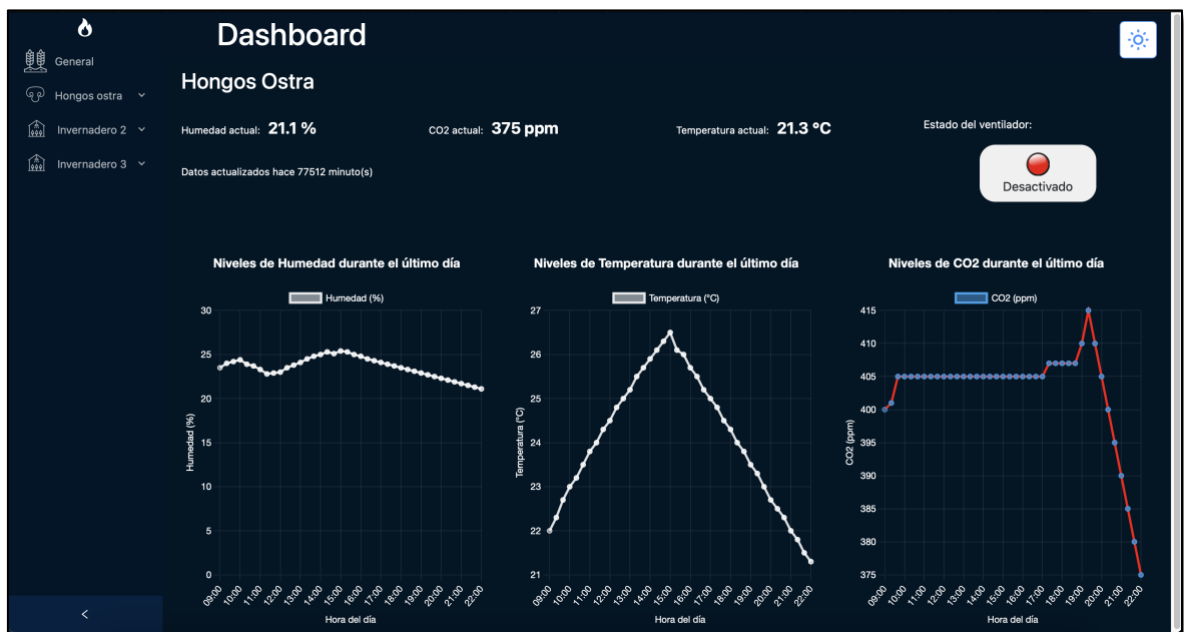
Las mejoras realizadas durante el sprint han permitido avanzar significativamente hacia la creación de un sistema de monitoreo de invernaderos más efectivo y satisfactorio para los usuarios.



**Figura 7.** Fases del proceso de DCU [24].



**Figura 8.** Interfaz del dashboard en tema claro al finalizar el sprint 1.



**Figura 9.** Interfaz del dashboard en tema oscuro al finalizar el sprint 1.

## 2.2.1 Planificación

La **Tabla 1** muestra las siete historias de usuario relacionadas con el front-end que se abordaron en este sprint.

**Tabla 1.** Historias de usuario abarcadas en el Sprint 1.

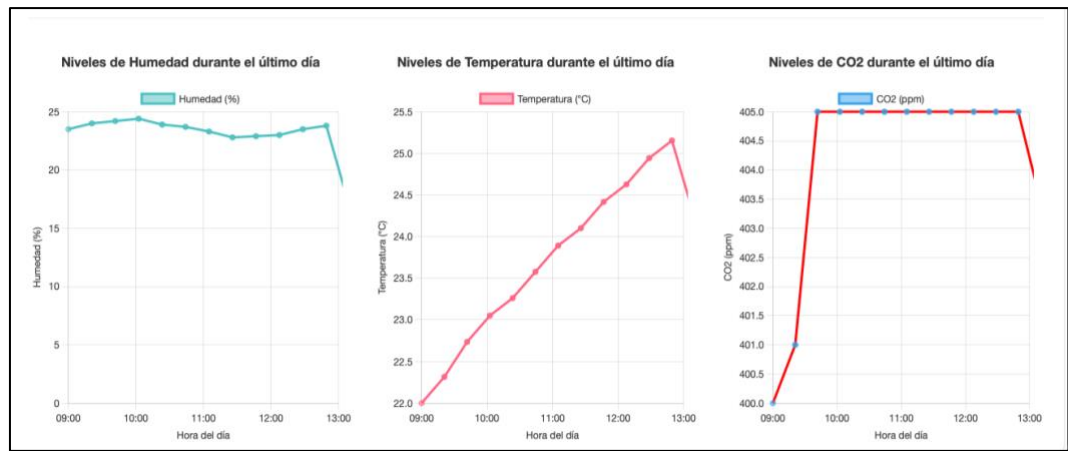
Historia de Usuario 1	Prioridad
Como encargado del invernadero, quiero visualizar en tiempo real la temperatura y humedad en los invernaderos de tomates, y la temperatura, humedad, CO2 y estado del ventilador en el invernadero de hongos ostra, para asegurarme de que están dentro de los parámetros adecuados para el crecimiento óptimo.	1
Criterios de aceptación	Esfuerzo
1. Los datos deben actualizarse automáticamente en el dashboard cada 4 horas (preferiblemente cada 5 minutos). 2. Los valores actuales deben mostrarse en gráficos de líneas con colores distintivos: rojo para valores críticos y verde o azul para valores normales.	8
Historia de Usuario 4	Prioridad
Como encargado del invernadero de hongos ostra, quiero que el sistema active automáticamente el ventilador cuando los niveles de CO2 superen un umbral preestablecido, para mantener automáticamente un ambiente adecuado para el crecimiento de los hongos. Quiero visualizar en tiempo real la temperatura y humedad en los invernaderos de tomates, y la temperatura, humedad, CO2 y estado del ventilador en el invernadero de hongos ostra, para asegurarme de que están dentro de los parámetros adecuados para el crecimiento óptimo.	2
Criterios de aceptación	Esfuerzo
El sistema debe monitorear continuamente los niveles de CO2 y mostrar cuando el ventilador esté encendido.	4
Historia de Usuario 5	Prioridad
Como encargado del invernadero, quiero personalizar la visualización del dashboard, incluyendo la selección de temas oscuros o claros y la disposición de las gráficas, para adaptar la interfaz a mis preferencias y facilitar la interpretación de los datos.	4
Criterios de aceptación	Esfuerzo
El dashboard debe ofrecer opciones de personalización, incluyendo un selector de tema oscuro o claro.	8

### 2.2.2 Implementación

- **HU 1.-** Como encargado del invernadero, quiero visualizar en tiempo real la temperatura y humedad en los invernaderos de tomates, y la temperatura, humedad, CO2 y estado del ventilador en el invernadero de hongos ostra, para



asegurarme de que están dentro de los parámetros adecuados para el crecimiento óptimo.

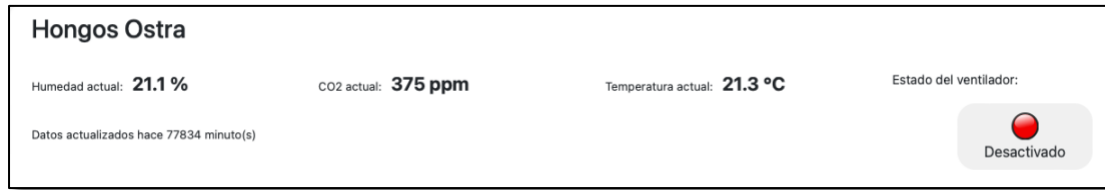


**Figura 10.** Resultado visible HU 1.

La **Figura 10** muestra el resultado visible de la historia de usuario 34, donde cada 5 minutos el usuario puede observar nuevos valores en las gráficas. Además, se actualiza el último valor recibido y se destacan en rojo los valores que están por debajo de un umbral específico. Las métricas por encima de los umbrales establecidos se representan en colores verde, azul y rojo.

Para la implementación de esta historia de usuario fue necesario el uso de:

- Bibliotecas: “chart.js” para la representación gráfica, “react” para la actualización de componentes.
- **HU 4.-** Como encargado del invernadero de hongos ostra, quiero que el sistema active automáticamente el ventilador cuando los niveles de CO2 superen un umbral preestablecido, para mantener automáticamente un ambiente adecuado para el crecimiento de los hongos. Quiero visualizar en tiempo real la temperatura y humedad en los invernaderos de tomates, y la temperatura, humedad, CO2 y estado del ventilador en el invernadero de hongos ostra, para asegurarme de que están dentro de los parámetros adecuados para el crecimiento óptimo.



**Figura 11.** Componente del sistema que indica las métricas y el estado del ventilador.

La **Figura 11** muestra el sistema indicando el estado del ventilador, especificando el momento en que está encendido. Esta funcionalidad permite al usuario verificar visualmente el funcionamiento del ventilador en el invernadero.

Para la implementación de esta funcionalidad fue necesario el uso de:

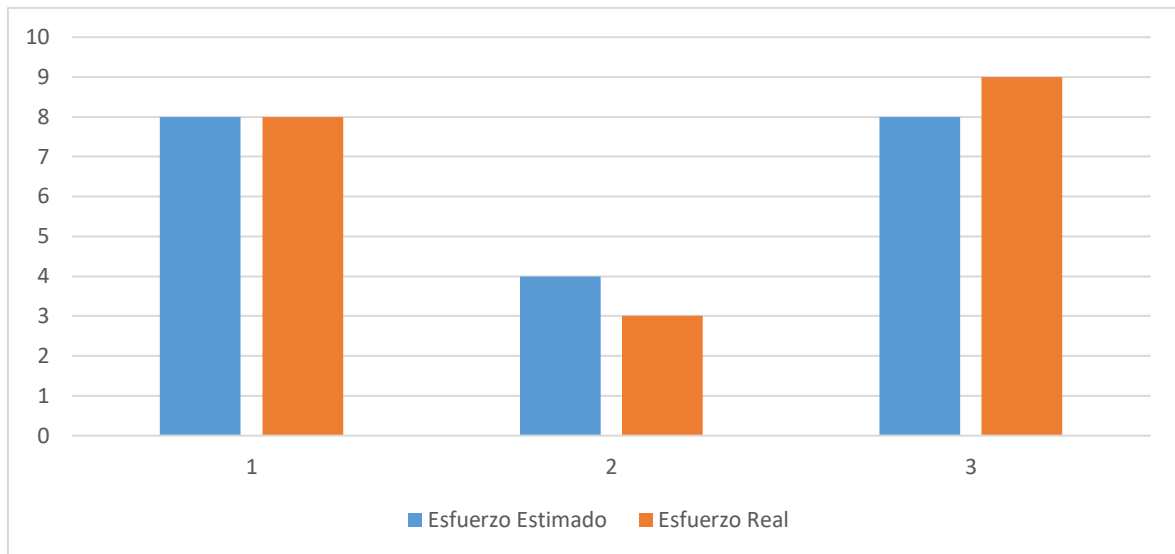
- **Bibliotecas:** “@anttd/icons” para los iconos de estado, “Bootstrap” para el estilo.

### 2.2.3 Review

Durante la Sprint Review, se llevó a cabo un análisis detallado de las historias de usuario completadas durante el sprint, así como una comparación entre el esfuerzo estimado y el esfuerzo real dedicado a su implementación.

**Tabla 2.** Sprint 1 Review.

Historia de Usuario	Esfuerzo Estimado	Esfuerzo Real
HU 1	8	8
HU 3	4	3
HU 5	8	9
Total	20	20



**Figura 12.** Esfuerzo estimado vs esfuerzo real - Sprint 1.

La **Tabla 2**, junto con la representación visual en la **Figura 11**, muestra que el esfuerzo estimado para completar las historias de usuario fue de 20 puntos, mientras que el esfuerzo real totalizó 20 puntos. Sin embargo, se observaron desviaciones individuales en algunas historias de usuario, lo que destaca la necesidad de una mayor precisión en las estimaciones iniciales.

- Historia de Usuario 1 (HU 1): El esfuerzo estimado fue de 8 puntos y se completó con el mismo esfuerzo de 8 puntos, indicando una estimación precisa y una ejecución eficiente.
- Historia de Usuario 3 (HU 3): El esfuerzo estimado fue de 4 puntos, pero el esfuerzo real fue de 3 puntos, sugiriendo una menor complejidad de la esperada.
- Historia de Usuario 5 (HU 5): El esfuerzo estimado fue de 8 puntos, pero el esfuerzo real fue de 9 puntos, mostrando un pequeño incremento debido a desafíos no previstos.

El análisis detallado revela que, aunque el esfuerzo total se mantuvo dentro de los límites esperados, los problemas específicos en la Historia de Usuario 5, tales como la implementación de gráficos de métricas y el ajuste de colores según los umbrales, presentaron desafíos adicionales.

Este análisis subraya la importancia de una planificación detallada y la capacidad de adaptarse a cambios y desafíos durante el desarrollo del proyecto. A partir de estos

resultados, se pueden tomar acciones correctivas para ajustar las estimaciones futuras y mejorar la asignación de recursos.

#### **2.2.4 Retrospectiva**

Durante la retrospectiva, se destacó positivamente la práctica de dividir las historias de usuario en tareas más pequeñas, lo que permitió un enfoque más claro y efectivo en el desarrollo del sistema de monitoreo de invernaderos. Además, el tiempo dedicado a investigar las tecnologías que se utilizarían, como Figma para el diseño de interfaces y Bibliotecas específicas para gráficos y visualización de datos, fue un acierto, ya que proporcionó una base sólida para el desarrollo del proyecto.

Por otro lado, se identificó la necesidad de mejorar la estimación de las tareas con el objetivo de lograr una planificación más precisa y realista. También se mencionó la importancia de establecer metas más cortas para las tareas, evitando acumular trabajo hacia el final del sprint y mejorando la distribución del esfuerzo a lo largo del tiempo.

En cuanto a mejoras futuras, se propone dedicar más tiempo a las tareas de investigación y realizar pruebas funcionales de las interfaces y tecnologías desde el inicio del sprint. Esto permitirá detectar, de manera temprana, posibles problemas o desafíos en el proceso, facilitando ajustes y mejoras durante el desarrollo. Con estas prácticas en mente, se espera aumentar la eficiencia y efectividad del equipo en futuros sprints.

### **2.3 Sprint 2**

En el segundo sprint, nos enfocaremos en un objetivo: permitir a los usuarios seleccionar y comparar métricas a lo largo del tiempo. Esta funcionalidad proporcionará una herramienta valiosa para analizar y optimizar las condiciones en los invernaderos. Para lograrlo, hemos identificado puntos importantes que guiarán nuestro desarrollo en esta etapa:

- Selección de Métricas a Comparar:
  - El usuario podrá seleccionar la métrica específica que desea comparar, como temperatura, humedad o CO2.
  - Se implementará una interfaz intuitiva que facilite la selección de las métricas de interés.
- Selección de Fechas:

- El usuario podrá escoger las fechas que desea comparar, inicialmente limitado a dos fechas.
- Se diseñarán controles de fecha que permitan una selección fácil y rápida de los periodos de tiempo a comparar.
- Visualización Comparativa:
  - Las gráficas de las fechas seleccionadas se mostrarán una sobre otra para facilitar la comparación visual.
  - Se utilizarán colores distintivos para representar cada fecha, asegurando que las diferencias y tendencias sean fácilmente identificables.
- Reflejar las Preferencias y Expectativas de los Usuarios:
  - Considerando el enfoque de Diseño Centrado en el Usuario (DCU) y conscientes de la importancia de la retroalimentación de los usuarios, adaptaremos los estilos usados en el sistema para reflejar las preferencias y expectativas de los usuarios. Este enfoque estará alineado con el objetivo específico del proyecto de "Diseñar interfaces web ajustadas a las necesidades de los usuarios".
- Uso de Bibliotecas:
  - Continuaremos utilizando la biblioteca chart.js para la visualización de gráficos.
  - Se emplearán Bootstrap y CSS para el diseño de la interfaz.

### **2.3.1 Planificación**

La planificación de este sprint se construyó sobre lo aprendido durante el primer sprint, aprovechando los conocimientos adquiridos para abordar este sprint de manera más efectiva. En este sprint, se seleccionó una sola historia de usuario de las que se habían identificado anteriormente, enfocada en permitir la selección y comparación de métricas a lo largo del tiempo. Además, se agregó una nueva historia de usuario para enriquecer la funcionalidad del sistema (HU 6).

Las lecciones aprendidas en el sprint anterior permitieron mejorar la estructura y el enfoque de desarrollo del front-end del sistema de monitoreo de invernaderos. A través de una planificación minuciosa y un enfoque basado en la retroalimentación de los usuarios, se busca avanzar de manera eficiente hacia los objetivos y brindar una herramienta útil para los encargados del invernadero.

La implementación de estas funcionalidades permitirá a los usuarios seleccionar métricas específicas, escoger fechas para comparación y visualizar las gráficas de estos días de manera superpuesta, facilitando así el análisis de tendencias y diferencias en los valores medidos. Utilizando chart.js para las visualizaciones y Bootstrap junto con CSS para el diseño de la interfaz, se espera ofrecer una experiencia de usuario optimizada y adaptable a diversas necesidades.

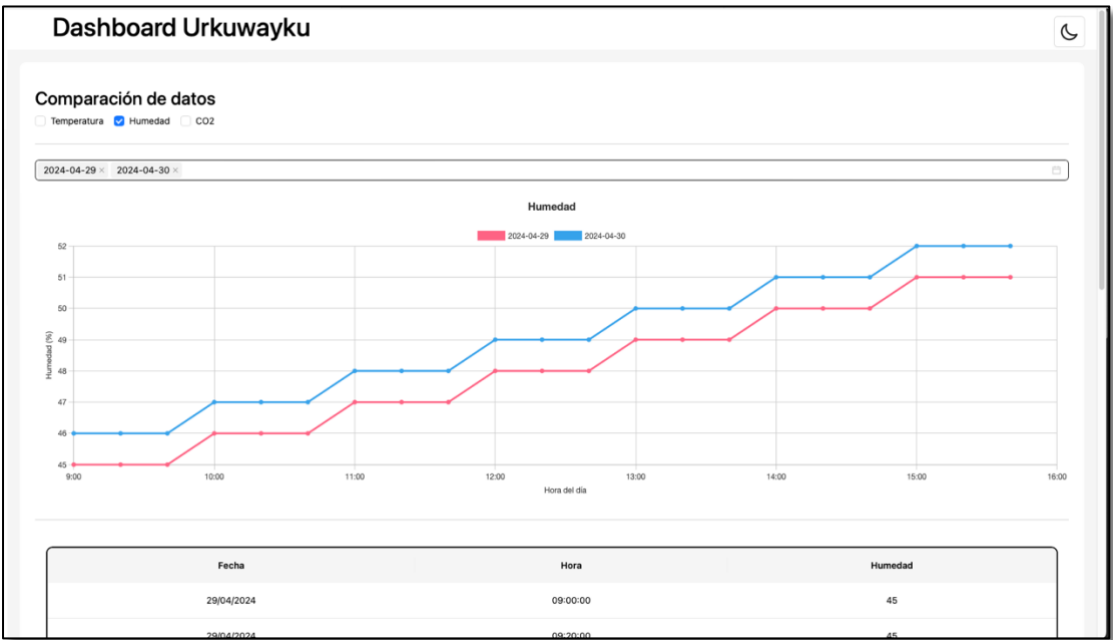
**Tabla 3.** Historias de usuario para el Sprint 2

Historia de Usuario 3	Prioridad
Como encargado del invernadero, quiero consultar y comparar el historial de datos de temperatura y humedad para los invernaderos de tomates, y de temperatura, humedad y CO2 para el invernadero de hongos ostra, para analizar tendencias y mejorar las prácticas de manejo del invernadero.	2
Criterios de aceptación	Esfuerzo
1. El dashboard debe permitir al usuario visualizar el historial de datos en gráficos y tablas. 2. Debe ser posible comparar los datos de diferentes días para observar tendencias. 3. La interfaz debe permitir seleccionar fácilmente los días para comparación y mostrar los datos comparativos de manera clara y entendible.	6
Historia de Usuario 6	Prioridad
Como encargado del invernadero, quiero tener un botón en la parte inferior de la pantalla de comparación de métricas, para poder descargar los datos presentados en las tablas en formato Excel y analizarlos fuera del sistema.	4
Criterios de aceptación	Esfuerzo
Debe haber un botón claramente visible en la parte inferior de la pantalla de comparación de métricas. Al hacer clic en el botón, los datos de las métricas comparadas de los días seleccionados deben descargarse en un archivo Excel. El archivo Excel debe incluir todas las métricas mostradas en las tablas, con una estructura clara y organizada. Los datos en el archivo Excel deben estar correctamente formateados, incluyendo las fechas de las métricas y los valores correspondientes. El botón debe estar disponible en todas las vistas de comparación donde se presenten datos tabulados.	8

### 2.3.2 Implementación

A continuación, se detallan los aspectos de implementación por cada historia de usuario (HU) abordada en el segundo sprint.

- **HU 3:** Como encargado del invernadero, quiero consultar y comparar el historial de datos de temperatura y humedad para los invernaderos de tomates, y de temperatura, humedad y CO2 para el invernadero de hongos ostra, para analizar tendencias y mejorar las prácticas de manejo del invernadero.



**Figura 13.** Interfaz de comparación de métricas para HU 3.

Fecha	Hora	Humedad
29/04/2024	09:00:00	45
29/04/2024	09:20:00	45
29/04/2024	09:40:00	45
29/04/2024	10:00:00	46
29/04/2024	10:20:00	46
29/04/2024	10:40:00	46
29/04/2024	11:00:00	47
29/04/2024	11:20:00	47
29/04/2024	11:40:00	47
29/04/2024	12:00:00	48
29/04/2024	12:20:00	48
29/04/2024	12:40:00	48
29/04/2024	13:00:00	49
29/04/2024	13:20:00	49
29/04/2024	13:40:00	49
29/04/2024	14:00:00	50

**Figura 14.** Interfaz de comparación de métricas para HU 3.

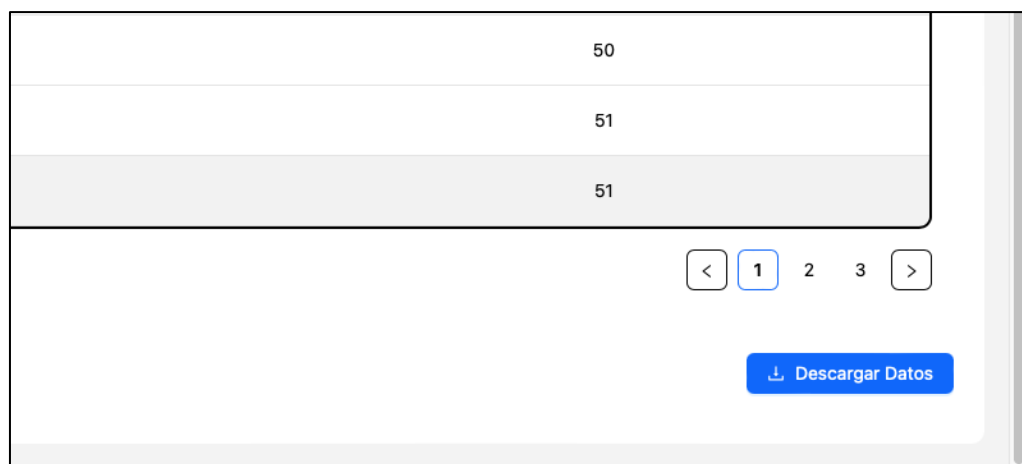
La **Figura 12** y **Figura 13** muestra la implementación de una interfaz que permite seleccionar fácilmente las fechas para comparar los datos de diferentes días. Las gráficas se presentan una sobre otra para facilitar la comparación visual de las métricas seleccionadas.

Fue necesario para la implementación de esta HU el uso de:

- **Bibliotecas:** chart.js para la visualización de los gráficos de comparación, react-datepicker para la selección de fechas.



- **HU 6:** Como encargado del invernadero, quiero tener un botón en la parte inferior de la pantalla de comparación de métricas, para poder descargar los datos presentados en las tablas en formato Excel y analizarlos fuera del sistema.



**Figura 15.** Botón de descarga de datos en Excel para HU 6.

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	date	time	humidity			
2	29/04/2024	09:00:00	45			
3	29/04/2024	09:20:00	45			
4	29/04/2024	09:40:00	45			
5	29/04/2024	10:00:00	46			
6	29/04/2024	10:20:00	46			
7	29/04/2024	10:40:00	46			
8	29/04/2024	11:00:00	47			
9	29/04/2024	11:20:00	47			
10	29/04/2024	11:40:00	47			
11	29/04/2024	12:00:00	48			
12	29/04/2024	12:20:00	48			
13	29/04/2024	12:40:00	48			
14	29/04/2024	13:00:00	49			
15	29/04/2024	13:20:00	49			
16	29/04/2024	13:40:00	49			

**Figura 16.** Botón de descarga de datos en Excel para HU 6.

La **Figura 15** y **Figura 16** muestra la implementación de un botón claramente visible en la parte inferior de la pantalla de comparación de métricas. Al hacer clic

en el botón, se descargan los datos comparados de las fechas seleccionadas en un archivo Excel.

Fue necesario para la implementación de esta HU el uso de:

- **Bibliotecas:** xlsx para la generación y descarga de archivos Excel, Bootstrap y css para el diseño y estilo del botón.

### 2.3.3 Review

Durante el Sprint Review del segundo sprint, evaluamos detalladamente el progreso y los resultados obtenidos. Este sprint se centró en la implementación de dos historias de usuario clave, con un enfoque en la visualización de métricas y la comparación de datos.

**Tabla 4.** Sprint 2 review

Historia de Usuario	Esfuerzo Estimado	Esfuerzo Real
HU 4	8	8
HU 6	4	3
Total	12	11

El análisis del sprint, reflejado en la tabla, muestra que se han cumplido las expectativas y se han logrado los objetivos planteados con una ligera desviación en el esfuerzo real respecto al estimado. La **Tabla 4** ilustra que el esfuerzo estimado total fue de 12 puntos, mientras que el esfuerzo real fue de 11 puntos. Esta discrepancia menor indica una planificación y ejecución efectivas, aunque siempre hay margen para mejorar en términos de precisión en la estimación de esfuerzos.

La implementación de estas funcionalidades permite a los usuarios monitorear métricas en tiempo real y comparar datos históricos, mejorando así las capacidades de gestión de los invernaderos. Este resultado subraya la importancia de una planificación precisa y una ejecución eficiente, lo que nos permitirá optimizar nuestros futuros sprints y alcanzar nuestros objetivos con mayor eficiencia y calidad.

### 2.3.4 Retrospectiva

Durante la retrospectiva del segundo sprint, hemos evaluado el rendimiento y los resultados obtenidos con el objetivo de identificar áreas de mejora y fortalezas en nuestro proceso de desarrollo. En este sprint, hemos logrado una comunicación efectiva con el stakeholder, lo cual ha sido primordial para asegurarnos de que el proyecto está encaminado de buena

manera. Esta comunicación nos ha permitido comprender mejor las expectativas y necesidades del stakeholder, alineando nuestro desarrollo con sus objetivos.

Una de las principales consideraciones técnicas que hemos identificado es la necesidad de escoger una biblioteca que pueda satisfacer las necesidades de la UI solicitadas por el stakeholder. La selección cuidadosa de esta biblioteca garantizará que podamos cumplir con los requisitos visuales y funcionales del proyecto, evitando cambios innecesarios en el futuro y asegurando una mayor coherencia y flexibilidad en el sistema.

Sin embargo, se identificó la necesidad de mejorar la comunicación entre los demás integrantes del proyecto para establecer una arquitectura armónica entre todos. Una comunicación más efectiva entre los miembros del equipo permitirá una mejor coordinación y cohesión en la implementación de las distintas partes del sistema, asegurando que todos los componentes funcionen de manera integrada y eficiente.

Además, esta retrospectiva nos ha recordado la importancia de la colaboración dentro del equipo. Continuaremos fomentando una comunicación abierta y efectiva para resolver cualquier desafío de manera oportuna y unificada. También reconocemos la necesidad de dedicar tiempo a la revisión antes de llegar al sprint review.

## 2.4 Sprint 3

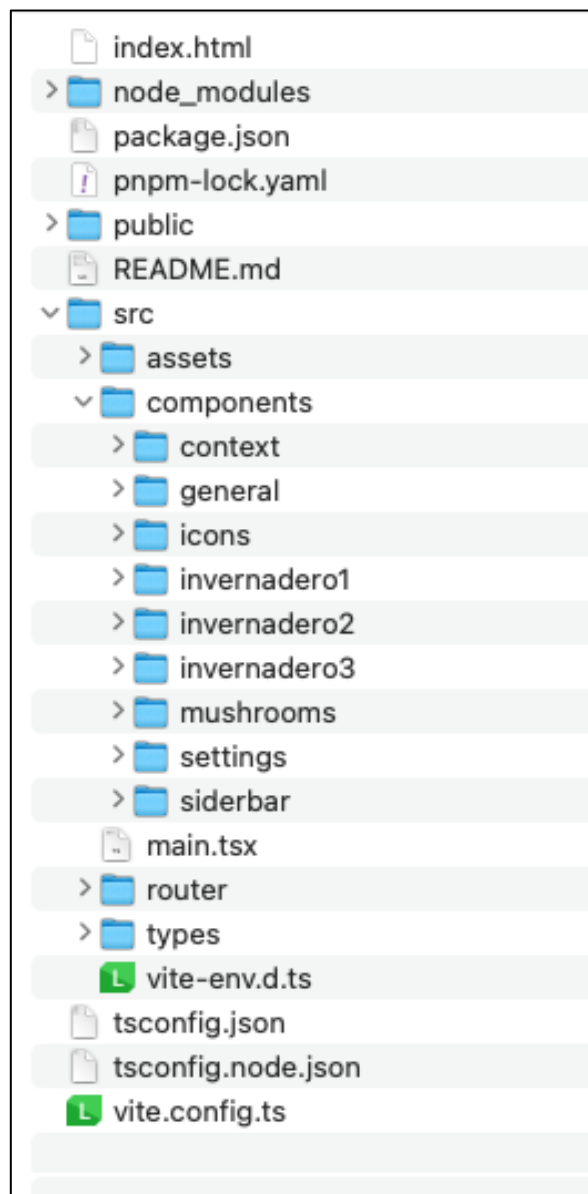
El tercer sprint se centró en: "Organizar y mejorar el front-end del sistema de monitoreo de invernaderos". Este objetivo tenía como propósito reestructurar y optimizar la organización de los archivos del front-end, así como implementar nuevas funcionalidades de configuración y mejorar la experiencia de usuario con el modo oscuro.

Para alcanzar dicho objetivo, se identificaron puntos importantes que guiaron el trabajo en esta etapa:

- **Organización de archivos del Front-end:** Se llevó a cabo una reestructuración total de los archivos del front-end, organizándolos de manera más eficiente para facilitar el mantenimiento y el desarrollo futuro. La modificación se muestra en la **Figura 17**.
- **Implementación de la Pestaña de Configuración:** Se creó una nueva pestaña de configuración donde los usuarios pueden establecer los umbrales para las distintas métricas de los invernaderos. Esta funcionalidad permite personalizar los límites

críticos para la temperatura, humedad y CO2, adaptándose a las necesidades específicas de cada invernadero.

- **Modificación para el Modo Oscuro:** Se realizaron las respectivas modificaciones en las pantallas para soportar el modo oscuro, mejorando así la experiencia de usuario en condiciones de baja iluminación.



**Figura 17.** Reestructuración basándose en organización por componentes.

Además, la creación de la pestaña de configuración es un paso para el cumplimiento de los objetivos específicos del proyecto. Esta funcionalidad permite a los encargados de los

invernaderos gestionar de manera intuitiva y funcional los parámetros críticos de monitoreo, garantizando un ambiente óptimo para el crecimiento de los cultivos.

### 2.4.1 Planificación

Para mejorar la efectividad y el enfoque de la planificación de este tercer sprint, se consideraron las experiencias y lecciones aprendidas en los dos sprints anteriores. La **Tabla 5** muestra la historia de usuario que se abordó en este sprint.

Las mejoras y ajustes realizados en el desarrollo del front-end, basados en la retroalimentación del product owner, proporcionaron una organización más eficiente de los archivos y la implementación de una pestaña de configuración. Esto permitió establecer umbrales para las métricas de los invernaderos y adaptar las pantallas al modo oscuro.

**Tabla 5.** Historias de usuario para el Sprint 3.

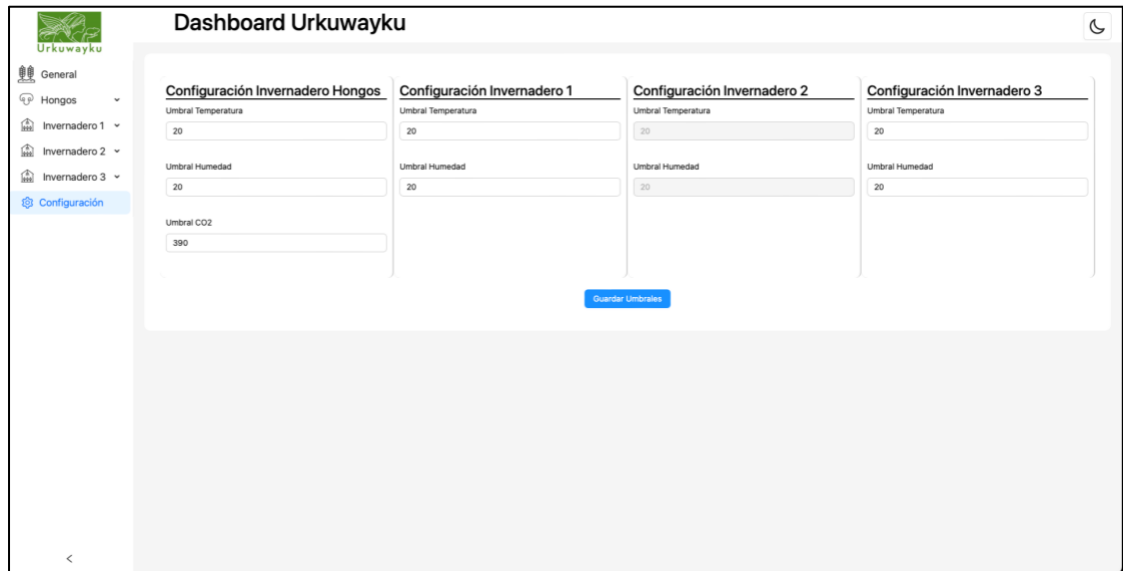
Historia de Usuario 7	Prioridad
Como encargado del invernadero, quiero poder modificar los umbrales para las métricas de temperatura, humedad y CO <sub>2</sub> , para establecer los límites de referencia a partir de los cuales las gráficas se pintarán de color rojo y se enviarán notificaciones en caso de que los valores caigan por debajo de estos umbrales.	2
Criterios de aceptación	Esfuerzo
La interfaz debe permitir al usuario ingresar y modificar los umbrales para cada métrica (temperatura, humedad y CO <sub>2</sub> ) de manera individual. Los umbrales modificados deben reflejarse inmediatamente en las gráficas, mostrando los valores críticos en color rojo. El sistema debe enviar una notificación al encargado del invernadero cuando cualquier métrica caiga por debajo del umbral establecido. Las notificaciones deben incluir detalles específicos sobre la métrica afectada, el valor registrado y el umbral correspondiente. Los umbrales deben ser almacenados de manera persistente y aplicarse a las métricas en tiempo real.	9

### 2.4.2 Implementación

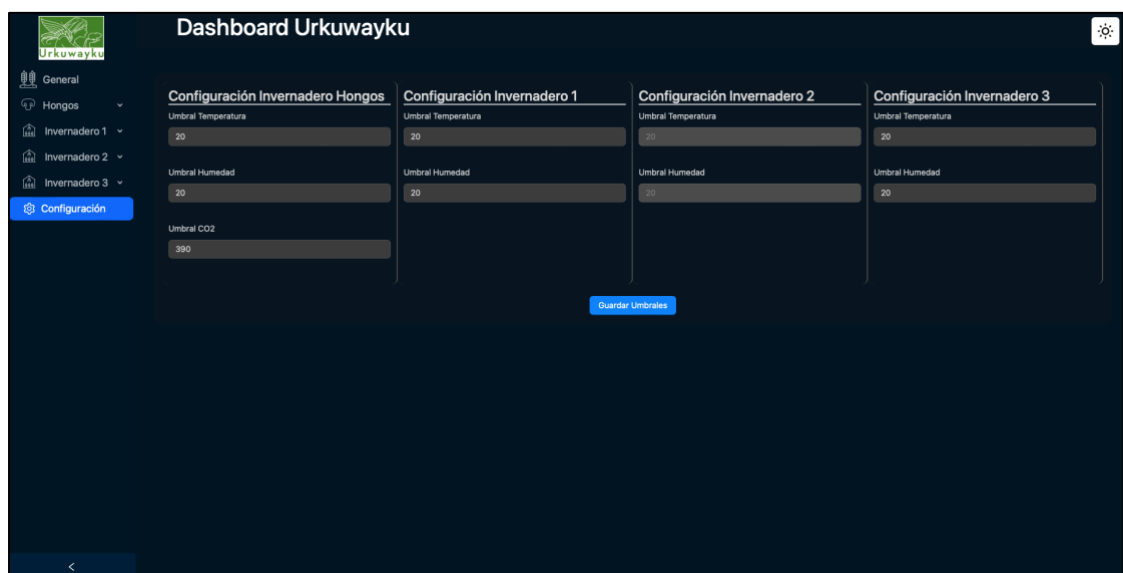
A continuación, se detallan los aspectos de implementación por cada historia de usuario (HU) abordada en el segundo sprint.

- **HU 5:** Como encargado del invernadero, quiero personalizar la visualización del dashboard, incluyendo la selección de temas oscuros o claros y la disposición de

las gráficas, para adaptar la interfaz a mis preferencias y facilitar la interpretación de los datos.



**Figura 18.** Interfaz de la pestaña de Configuración en modo claro.



**Figura 19.** Interfaz de la pestaña de Configuración en modo oscuro.

Las figuras: **Figura 18** y **Figura 19** muestran el resultado visible de la HU 6. A continuación, se explican estos resultados con más detalle:

- **Figura 17:** Muestra la interfaz de configuración en modo claro, donde los usuarios pueden ingresar y modificar los umbrales para las métricas de temperatura, humedad y CO2.
- **Figura 18:** Presenta la misma interfaz de configuración en modo oscuro, destacando las adaptaciones visuales realizadas para mejorar la usabilidad en condiciones de baja iluminación.

Para la implementación de esta historia de usuario se requirió:

- **Bibliotecas:** Se utilizaron varias Bibliotecas existentes en el proyecto. No se introdujeron nuevas bibliotecas, pero se integraron componentes adicionales de las bibliotecas de estilos existentes para lograr una interfaz coherente y funcional.

### 2.4.3 Review

La **Tabla 6** muestra el esfuerzo estimado y el esfuerzo real relacionado con la HU 6 que se abordó en este sprint (Sprint 3).

**Tabla 6.** Sprint 3 review.

Historia de Usuario	Esfuerzo Estimado	Esfuerzo Real
HU 6	8	10

Se evidenció que el esfuerzo estimado fue de 8 puntos, mientras que el esfuerzo real alcanzó los 10 puntos. Este incremento en el esfuerzo se debió a la complejidad adicional al implementar la disponibilidad del umbral en todo el proyecto.

Este retroceso representa un desafío en relación con las mejoras logradas en las estimaciones durante el sprint anterior. Es importante mencionar que la complicación principal fue asegurar que el umbral pudiera ser accedido y modificado desde todas las partes relevantes del sistema, lo cual requirió una integración más profunda de lo anticipado.

Considerando el sprint anterior, en este sprint se volvió a experimentar una discrepancia en la estimación, lo que indica la necesidad de abordar este aspecto de manera más cuidadosa y precisa en futuros sprints. La falta de previsión sobre la complejidad de la integración afectó el progreso del sprint, resultando en un esfuerzo real que superó ligeramente el estimado.

#### **2.4.4 Retrospectiva**

En la retrospectiva del tercer sprint, se reconoció como un acierto la organización de los archivos del front-end y la implementación de una arquitectura basada en componentes. Esto facilitó el mantenimiento y desarrollo futuro del sistema. Sin embargo, se identificó la necesidad de mejorar la planificación, evitando historias de usuario demasiado grandes y dividiéndolas adecuadamente para lograr estimaciones más precisas. Aunque en el sprint 2 se habían mejorado las estimaciones, en este sprint nuevamente se presentaron discrepancias significativas entre el esfuerzo estimado y el esfuerzo real.

Se sugiere comenzar a dedicar más tiempo al análisis detallado de las historias de usuario antes de estimarlas, para tener una comprensión clara de las funcionalidades involucradas y realizar estimaciones más acertadas. Además, se debe considerar la importancia de integrar adecuadamente los umbrales en todo el proyecto desde el inicio.

El equipo también discutió la importancia de mantener una comunicación abierta y efectiva entre todos los miembros, lo que ayudará a identificar y resolver problemas de manera temprana. Se reiteró la necesidad de revisar y ajustar constantemente las prácticas de desarrollo para adaptarse a las necesidades cambiantes del proyecto y asegurar un progreso continuo y exitoso.

Con estas consideraciones, se espera mejorar la planificación y estimación de futuros sprints, manteniendo un enfoque centrado en la eficiencia y la calidad del desarrollo del sistema de monitoreo de invernaderos. A través de una continua evaluación de las prácticas y tomando en cuenta las consideraciones técnicas, se busca optimizar el proceso de trabajo para alcanzar con éxito los objetivos del proyecto.

### **2.5 Sprint 4**

El cuarto sprint se centró en: "Mejorar la visualización de métricas y la implementación de notificaciones". Este objetivo tenía como propósito modificar la historia de usuario 1 para consolidar las métricas en una sola gráfica y, en caso de ser necesario, implementar una segunda escala para métricas que operen en diferentes rangos. Además, se buscó implementar las notificaciones de manera que, si una métrica cae por debajo del umbral establecido, se enviara una alerta, conforme a lo establecido en la historia de usuario 2. Adicionalmente, se diseñó y creó una encuesta para realizar pruebas de usabilidad del



sistema con usuarios que manejan invernaderos y agricultura, con el fin de obtener retroalimentación valiosa para futuras mejoras.

Para alcanzar dicho objetivo, se identificaron puntos importantes que guiaron el trabajo en esta etapa:

1. **Consolidación de Métricas en una Gráfica:** Se modificó la visualización de las métricas para que todas se muestren en una misma gráfica. Esto facilita la comparación directa y el análisis de las diferentes métricas en un solo lugar.
2. **Implementación de una Segunda Escala:** Para métricas que operan en diferentes rangos, se añadió una segunda escala en la gráfica, asegurando una representación precisa y comprensible de todas las métricas.
3. **Notificaciones por Umbrales:** Se implementó el sistema de notificaciones para que, si una métrica está por debajo del umbral definido, se enviara una notificación al encargado del invernadero, siguiendo las especificaciones de la historia de usuario 2.

Estas mejoras permiten una visualización más clara y eficiente de las métricas, además de una respuesta inmediata ante condiciones críticas, contribuyendo a una mejor gestión y monitoreo de los invernaderos. La encuesta de usabilidad ayudará a identificar áreas de mejora y a asegurar que el sistema cumpla con las expectativas y necesidades de los usuarios finales.

### 2.5.1 Planificación

Para la planificación del cuarto sprint, se tomaron en cuenta las historias de usuario que necesitaban mejoras y nuevas implementaciones para cumplir con los objetivos definidos. En este sprint, se abordaron y mejoraron las siguientes historias de usuario, como se muestra en la **Tabla 7**.

**Tabla 7.** Historias de usuario para el Sprint 3.

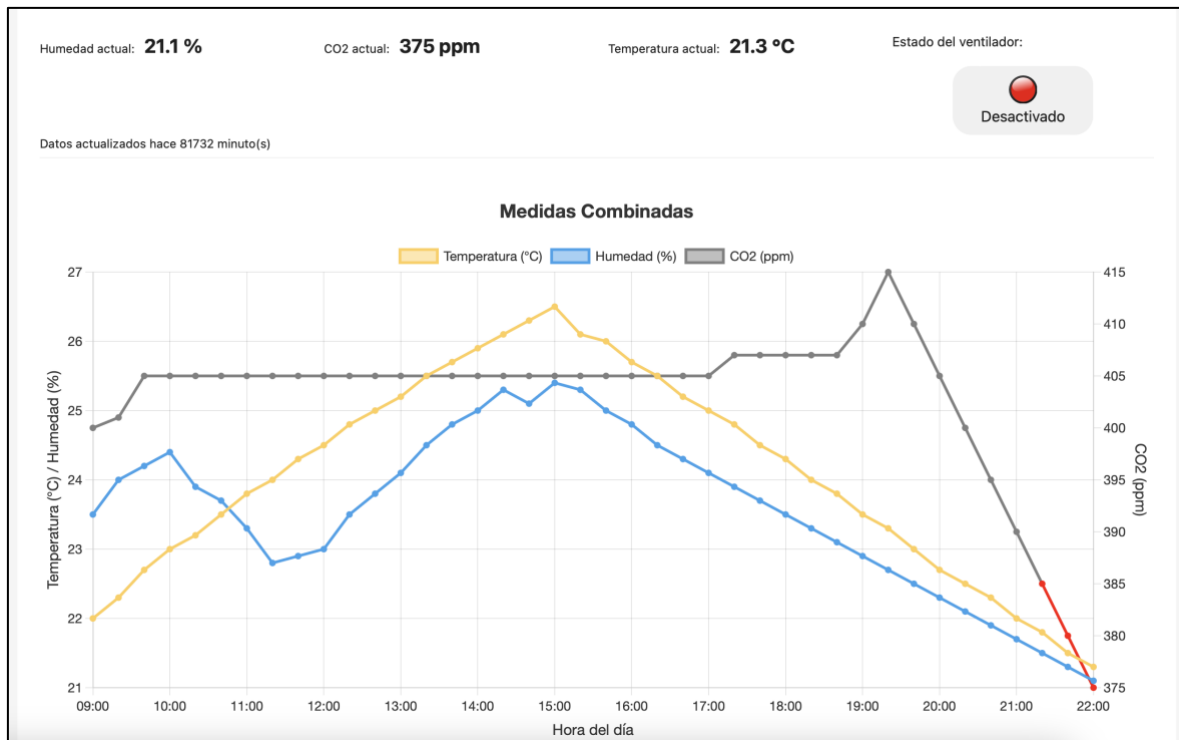
Historia de Usuario 1	Prioridad
Como encargado del invernadero, quiero visualizar en tiempo real la temperatura y humedad en los invernaderos de tomates, y la temperatura, humedad, CO2 y estado del ventilador en el invernadero de hongos ostra, para asegurarme de que están dentro de los parámetros adecuados para el crecimiento óptimo.	1
Criterios de aceptación	Esfuerzo

<p>Los datos deben actualizarse automáticamente en el dashboard cada 4 horas (preferiblemente cada 5 minutos).</p> <p>Los valores actuales deben mostrarse en gráficos de líneas con colores distintivos: rojo para valores críticos y verde o azul para valores normales.</p> <p>Las métricas deben unificarse en una sola gráfica.</p> <p>En caso de que las métricas operen en diferentes escalas, se debe implementar una segunda escala en la gráfica. En caso de que las métricas operen en diferentes escalas, se debe implementar una segunda escala en la gráfica.</p>	8
Historia de Usuario 2	Prioridad
<p>Como encargado del invernadero, quiero recibir alertas automáticas en el dashboard cuando los valores de temperatura o humedad en los invernaderos de tomates, y de temperatura, humedad o CO2 en el invernadero de hongos ostra, alcancen niveles críticos, para tomar acciones correctivas de manera oportuna.</p>	3
Criterios de aceptación	Esfuerzo
<p>Las alertas deben ser visibles en el dashboard similar a las notificaciones de Facebook.</p> <p>El sistema debe permitir al usuario configurar los umbrales de alerta para temperatura, humedad y CO2.</p> <p>Las alertas deben incluir detalles específicos sobre el valor alcanzado, la hora de la alerta, el invernadero correspondiente y la métrica que bajó del umbral.</p> <p>Cada notificación debe mostrar un botón que permita limpiar todas las notificaciones del dashboard.</p>	7

### 2.5.2 Implementación

A continuación, se detallan los aspectos de implementación por cada historia de usuario (HU) abordada en el segundo sprint.

- **HU 1:** Como encargado del invernadero, quiero visualizar en tiempo real la temperatura y humedad en los invernaderos de tomates, y la temperatura, humedad, CO2 y estado del ventilador en el invernadero de hongos ostra, para asegurarme de que están dentro de los parámetros adecuados para el crecimiento óptimo.



**Figura 20.** Gráfico que muestra las métricas juntas.

La **Figura 20** muestra el resultado visible de la historia de usuario 2. A continuación, se explican estos resultados con más detalle:

- **Figura 20:** Presenta el gráfico donde se visualizan en una sola gráfica las tres métricas: temperatura, humedad y CO2, con una segunda escala implementada para las métricas que operan en diferentes rangos.

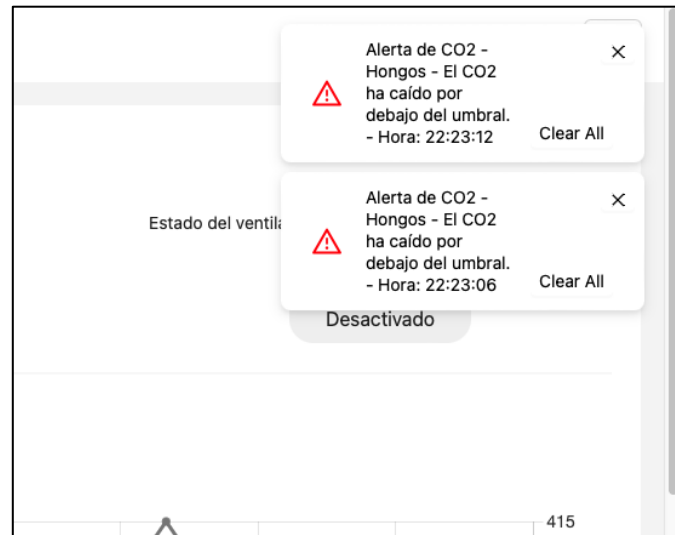
Para la implementación de esta historia de usuario se requirió:

- **Bibliotecas:** Se utilizaron varias bibliotecas existentes en el proyecto. No se introdujeron nuevas Bibliotecas, pero se integraron componentes adicionales de las Bibliotecas de estilos existentes para lograr una interfaz coherente y funcional.

Esta implementación mejoró la capacidad de los encargados de los invernaderos para gestionar y monitorear las condiciones del ambiente de cultivo, asegurando que se mantengan dentro de los parámetros óptimos definidos.

- **HU 2:** Como encargado del invernadero, quiero recibir alertas automáticas en el dashboard cuando los valores de temperatura o humedad en los invernaderos de

tomates, y de temperatura, humedad o CO2 en el invernadero de hongos ostra, alcancen niveles críticos, para tomar acciones correctivas de manera oportuna.



**Figura 21.** Notificaciones de la página.

La **Figura 21** muestra el resultado visible de la historia de usuario "Alertas de Condiciones Críticas". A continuación, se explican estos resultados con más detalle:

- **Figura 21:** Presenta el sistema de notificaciones en el dashboard, donde se visualizan las alertas con detalles específicos sobre el valor alcanzado, la hora de la alerta, el invernadero correspondiente y la métrica que bajó del umbral. Además, cada notificación incluye un botón para limpiar todas las notificaciones del dashboard.

Para la implementación de esta historia de usuario se requirió:

- **Bibliotecas:** Se utilizaron varias Bibliotecas existentes en el proyecto. No se introdujeron nuevas bibliotecas, pero se integraron componentes adicionales de las bibliotecas de estilos existentes para lograr una interfaz coherente y funcional.

Esta implementación mejoró la capacidad de los encargados de los invernaderos para reaccionar rápidamente a condiciones críticas, asegurando que las medidas correctivas se tomen de manera oportuna y efectiva.

### 2.5.3 Review

Durante el sprint review del cuarto sprint, se observó que se realizaron las historias de usuario detalladas en la Tabla 14, con un esfuerzo estimado de 14 puntos y un esfuerzo real de 14 puntos. Se evidenció una mejora en las estimaciones, ya que estuvieron exactas al esfuerzo real, lo cual representa un avance significativo en el proceso de planificación por parte del equipo.

**Tabla 8.** Sprint 4 Review.

Historia de Usuario	Esfuerzo Estimado	Esfuerzo Real
HU 1	8	8
HU 2	7	7
Total	15	15

Es importante destacar que la comunicación y la división de tareas mejoraron notablemente en este sprint, lo que contribuyó a una mayor eficiencia en el desarrollo del sistema de monitoreo de invernaderos. La experiencia acumulada en los sprints anteriores permitió una mejor comprensión de los requerimientos de los usuarios y una planificación más precisa para cumplir con sus necesidades.

Con cada sprint, se puede notar un aprendizaje de las lecciones anteriores y aplicando ese conocimiento para abordar los desafíos en cada etapa del desarrollo. La continua búsqueda de la excelencia y la mejora constante en el enfoque de desarrollo han llevado a un desarrollo más efectivo y eficiente del sistema de monitoreo de invernaderos.

### 2.5.4 Retrospectiva

Durante la retrospectiva del cuarto sprint, se identificó que el enfoque en la consolidación de métricas en una sola gráfica y la implementación de notificaciones mejoró significativamente la usabilidad del sistema. Se conversó con los stakeholders, quienes dieron su aprobación y aceptación de los resultados del sprint.

Se reconoció la necesidad de mejorar ciertos aspectos de la planificación y la estimación de las historias de usuario, así como definir claramente el alcance con el Product Owner. A pesar de que las estimaciones en este sprint fueron precisas, se identificaron áreas de mejora para futuros sprints.

Ideas para mejorar incluyen:

- **Considerar futuros aspectos técnicos**, como la implementación de medidas de seguridad para la consulta de los API.
- **Mejorar la comunicación** entre los miembros del equipo para asegurar una comprensión clara y precisa de los requerimientos y objetivos de cada historia de usuario.
- **Continuar refinando las técnicas de planificación y estimación** para mantener la precisión lograda en este sprint y aplicarla en futuros desarrollos.

La continua evaluación de las prácticas y la implementación de mejoras basadas en la retroalimentación permitirán optimizar el proceso de trabajo y asegurar que el sistema de monitoreo de invernaderos cumpla con las expectativas y necesidades de los usuarios.

### 2.5.5 Evaluación de usabilidad y satisfacción del usuario

En este apartado se procedió a crear las preguntas de la prueba de funcionalidad y satisfacción de los usuarios. Las primeras preguntas buscan recopilar información personal de los encuestados. Posteriormente, se realizaron preguntas para evaluar la facilidad de navegación de la aplicación web. Además, se hicieron preguntas acerca del diseño y la usabilidad del sitio. Finalmente, se incluyeron preguntas para comprobar que la información presentada en la aplicación web es relevante para los usuarios.

Las preguntas utilizadas en la encuesta son las siguientes:

1. ¿La aplicación es fácil de usar?
2. ¿Encontré fácilmente la información que buscaba?
3. ¿Los gráficos y tablas son claros y comprensibles?
4. ¿La navegación dentro de la aplicación es intuitiva?
5. ¿La configuración de umbrales es fácil de entender y usar?
6. ¿La opción de cambiar el tema de oscuro a claro (y viceversa) es útil?
7. ¿La aplicación responde rápidamente a mis acciones?
8. ¿Me siento satisfecho con la funcionalidad general de la aplicación?
9. ¿Las notificaciones funcionan correctamente y son útiles?
10. ¿Recomendaría esta aplicación a otros usuarios?
11. Recomendaciones y Comentarios:
  - Por favor, escribe cualquier recomendación o comentario que tengas para mejorar la aplicación.

Las respuestas para las preguntas del 1 al 5 se califican de la siguiente manera:

- 1: Muy Insatisfecho
- 2: Insatisfecho
- 3: Neutral
- 4: Satisfecho
- 5: Muy Satisfecho

Estas preguntas ayudarán a obtener una comprensión de la experiencia del usuario y a identificar áreas específicas de mejora para asegurar que la aplicación web cumpla con las expectativas de los usuarios y brinde una experiencia óptima. Las mejoras identificadas a partir de estas evaluaciones se podrían implementar en futuras entregas o sprints, permitiendo iterar y perfeccionar continuamente la aplicación para satisfacer mejor las necesidades y expectativas de los usuarios.

Además, esta encuesta se realizará solo al personal que maneja invernaderos, asegurando que las respuestas y sugerencias provengan de usuarios con experiencia directa en la gestión y monitoreo de las condiciones ambientales de los invernaderos. Esto permitirá obtener retroalimentación precisa y relevante para el contexto específico en el que se utilizará la aplicación.

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

#### 3.1 Resultados

Una vez concluida la fase de pruebas de usabilidad, se procedió a realizar el análisis de los resultados obtenidos. Para el análisis, se comenzó por evaluar las respuestas de los usuarios candidatos a utilizar el sistema, quienes tienen experiencia en el manejo de invernaderos y plantaciones.

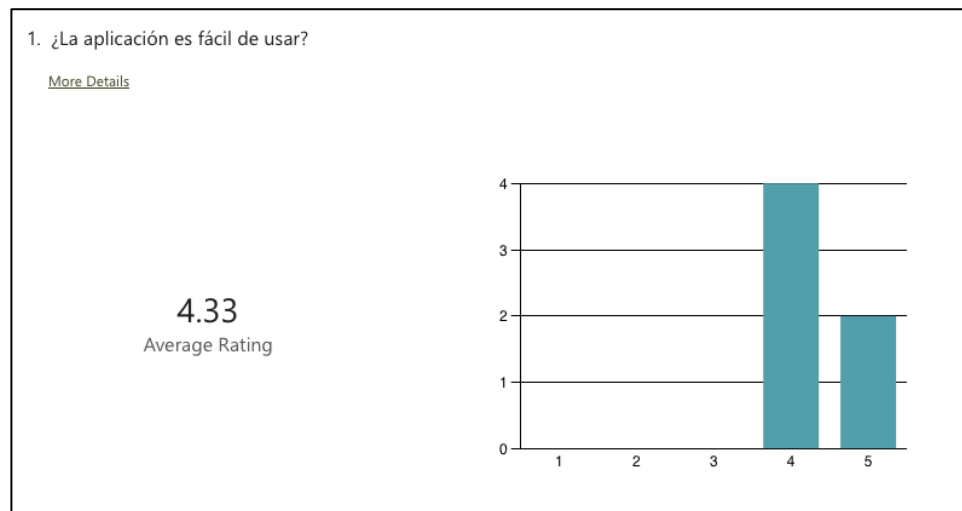
La encuesta incluyó preguntas calificadas del 1 al 5, donde 1 representa "Muy Insatisfecho" y 5 "Muy Satisfecho". Se presentan de manera gráfica en las figuras correspondientes para facilitar su interpretación y análisis.

#### **Pregunta 1: ¿La aplicación es fácil de usar?**

En la primera pregunta de la encuesta, los usuarios evaluaron la facilidad de uso de la aplicación. Como se puede observar en la **Figura 22**, la mayoría de los encuestados calificaron la aplicación con un puntaje alto, reflejando una valoración positiva general. El

promedio de calificaciones fue de 4.33, indicando que los usuarios encuentran la aplicación mayormente fácil de usar.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron 4, seguido de 5, lo que sugiere que la aplicación cumple con las expectativas de usabilidad para la mayoría de los usuarios. Este resultado es alentador y sugiere que las mejoras realizadas en la interfaz y la navegación han sido efectivas.



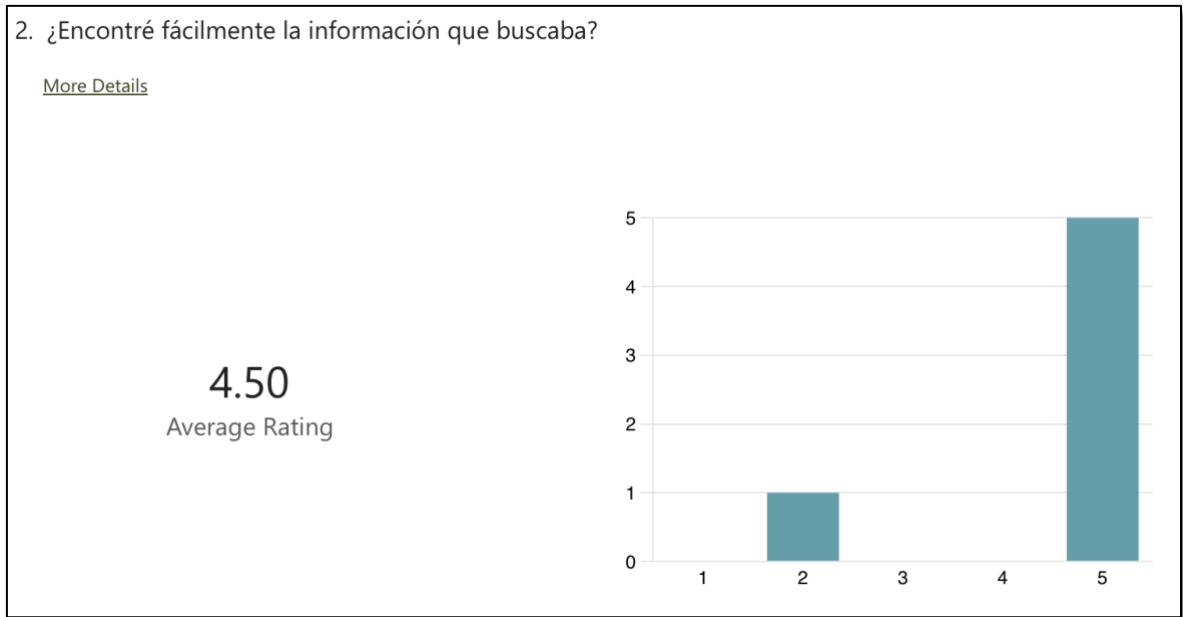
**Figura 22.** Resultados de la Pregunta 1: ¿La aplicación es fácil de usar?

#### **Pregunta 2: ¿Encontró fácilmente la información que buscaba?**

En la segunda pregunta los usuarios evaluaron la facilidad con la que podían encontrar la información dentro de la aplicación. Como se puede observar en la **Figura 23**, la mayoría de los encuestados calificaron esta característica con una puntuación muy alta, obteniendo un promedio de 4.50.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron la calificación más alta (5), seguida de algunos que eligieron 2, lo que sugiere que la aplicación facilita en gran medida la búsqueda de información para la mayoría de los usuarios. Este resultado es positivo y demuestra que la estructura y organización de la información en la aplicación es efectiva para la mayoría de los usuarios.



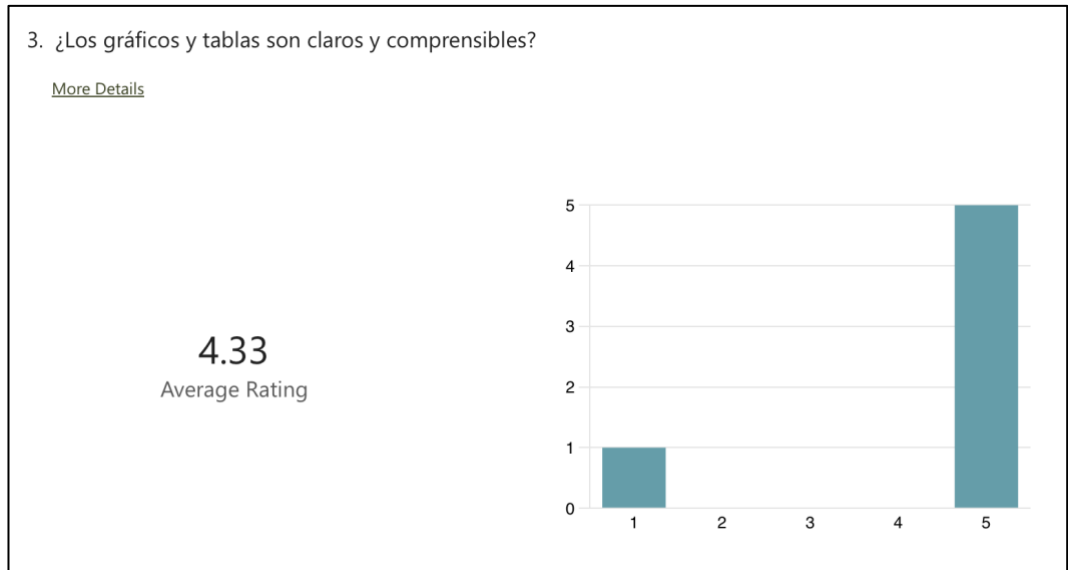


**Figura 23.** Resultados de la Pregunta 2: ¿Encontré fácilmente la información que buscaba?

**Pregunta 3: ¿Los gráficos y tablas son claros y comprensibles?**

En la tercera pregunta de la encuesta, los usuarios evaluaron la claridad y comprensibilidad de los gráficos y tablas dentro de la aplicación. Como se puede observar en la imagen de la **Figura 24**, la mayoría de los encuestados calificaron positivamente la claridad de los gráficos y tablas, obteniendo un promedio de 4.33.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron la puntuación 5, con solo una minoría seleccionando la puntuación 1. Esto sugiere que los gráficos y tablas dentro de la aplicación son considerados claros y comprensibles por la mayoría de los usuarios. Este resultado es muy positivo y demuestra que el diseño de los elementos visuales es efectivo, facilitando la interpretación de la información presentada en la aplicación.

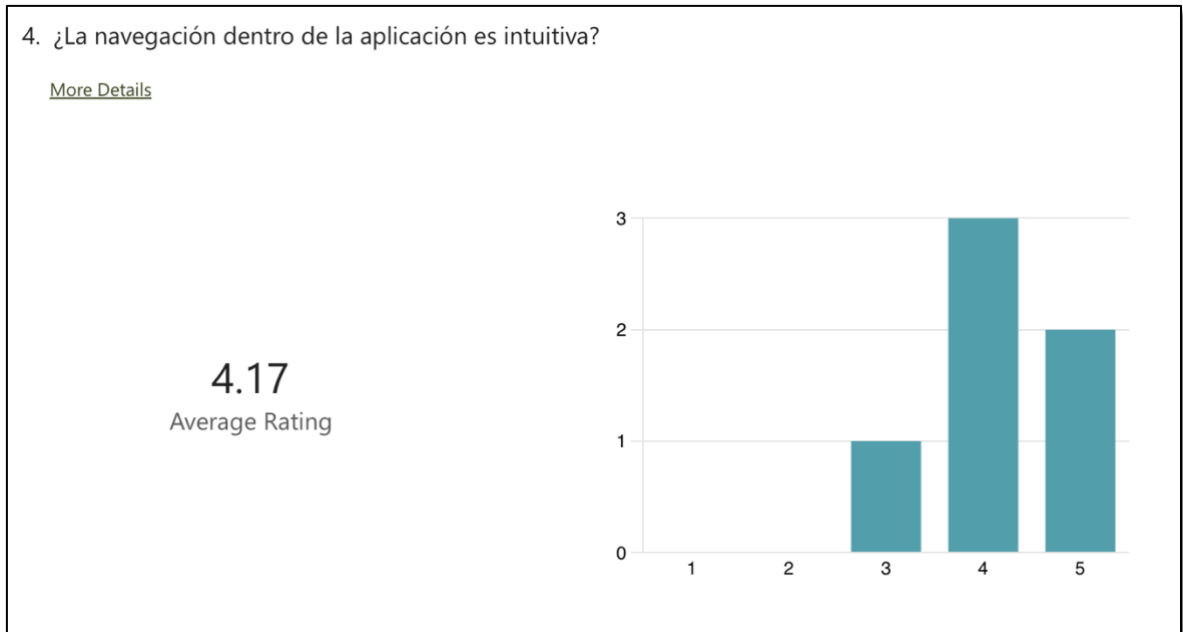


**Figura 24.** Resultados de la Pregunta 3: ¿Encontré fácilmente la información que buscaba?

**Pregunta 4: ¿La navegación dentro de la aplicación es intuitiva?**

En la cuarta pregunta de la encuesta, los usuarios evaluaron la facilidad de navegación dentro de la aplicación. Como se puede observar en la **Figura 25**, la mayoría de los encuestados calificaron la navegación con una puntuación alta, obteniendo un promedio de 4.17.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron 4 y 5, lo que sugiere que la navegación dentro de la aplicación es considerada intuitiva por la mayoría de los usuarios. Este resultado es positivo y demuestra que el diseño de la navegación es efectivo y facilita el uso de la aplicación.

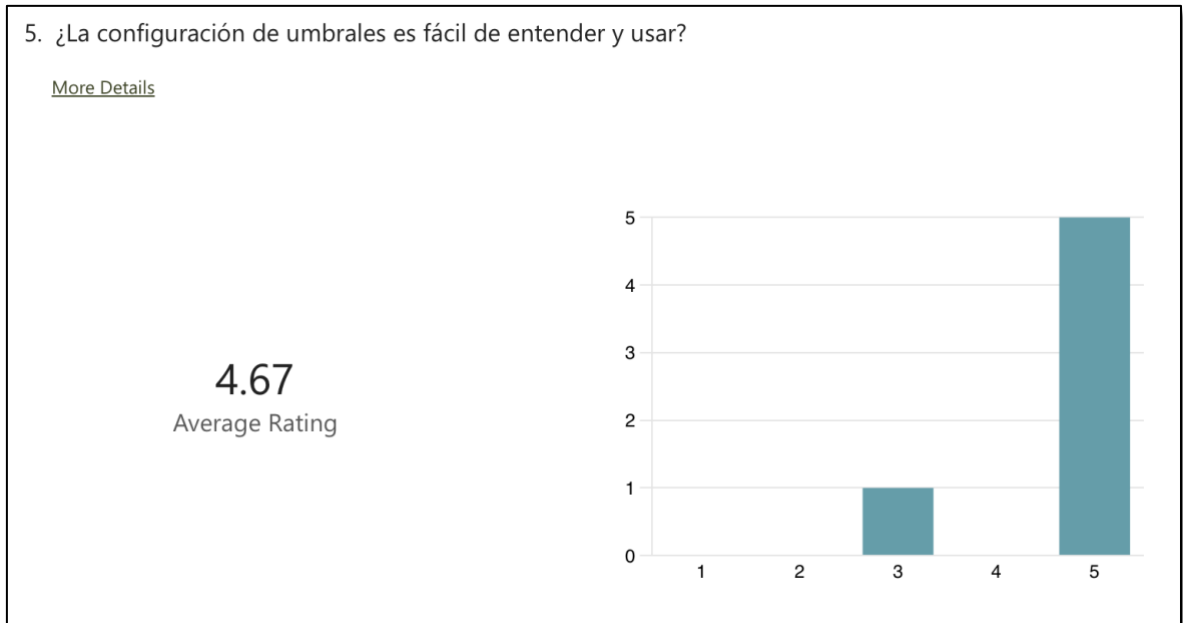


**Figura 25.** Resultados de la Pregunta 4: ¿La navegación dentro de la aplicación es intuitiva?

#### **Pregunta 5: ¿La configuración de umbrales es fácil de entender y usar?**

En la quinta pregunta de la encuesta, los usuarios evaluaron la facilidad de uso de la funcionalidad de configuración de umbrales. Como se puede observar en la **Figura 26**, la mayoría de los encuestados calificaron esta característica con una puntuación muy alta, obteniendo un promedio de 4.67.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron la calificación más alta (5), con una única respuesta en 3, lo que sugiere que la configuración de umbrales es considerada fácil de entender y usar por la mayoría de los usuarios. Este resultado es positivo y refleja que la implementación de esta funcionalidad cumple con las expectativas de usabilidad y facilita la gestión de los parámetros críticos en los invernaderos.

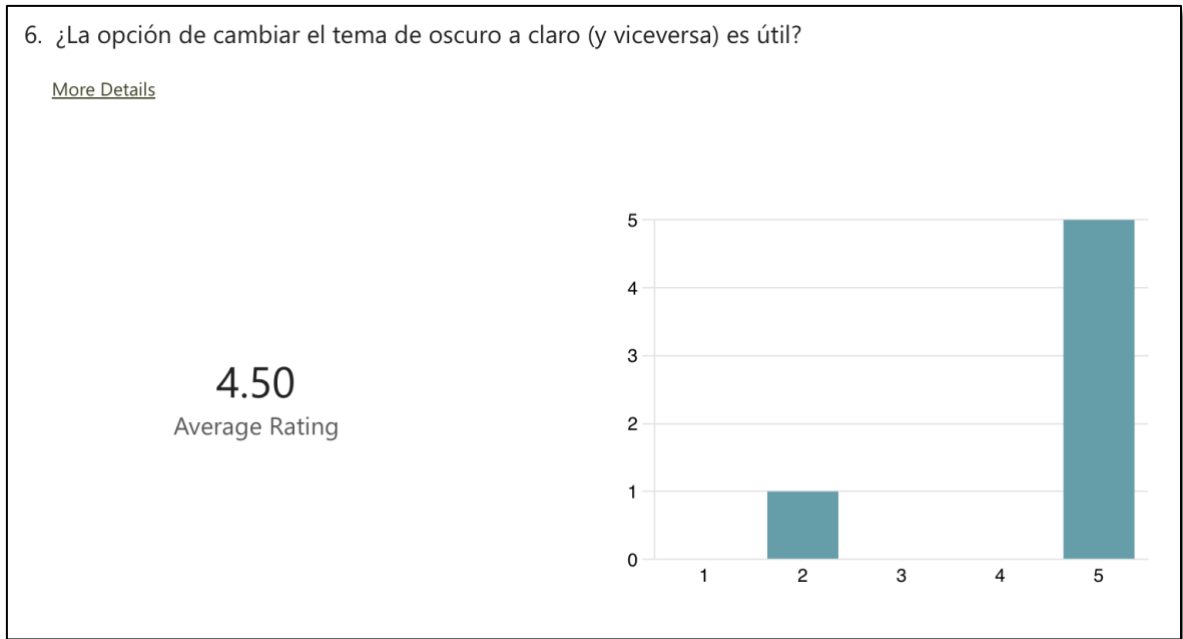


**Figura 26.** Resultados de la Pregunta 5: ¿La configuración de umbrales es fácil de entender y usar?

**Pregunta 6: ¿La opción de cambiar el tema de oscuro a claro (y viceversa) es útil?**

En la sexta pregunta de la encuesta, los usuarios evaluaron la utilidad de la funcionalidad que permite cambiar entre los modos oscuro y claro en la aplicación. Como se puede observar en la **Figura 27**, la mayoría de los encuestados calificaron esta característica con una puntuación muy alta, obteniendo un promedio de 4.50.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron la calificación más alta (5), con una única respuesta en 2, lo que sugiere que la opción de cambiar el tema es considerada útil por la mayoría de los usuarios. Este resultado es positivo y refleja que la implementación de esta funcionalidad cumple con las expectativas de los usuarios, proporcionando una experiencia de uso adaptable a diferentes condiciones de iluminación.

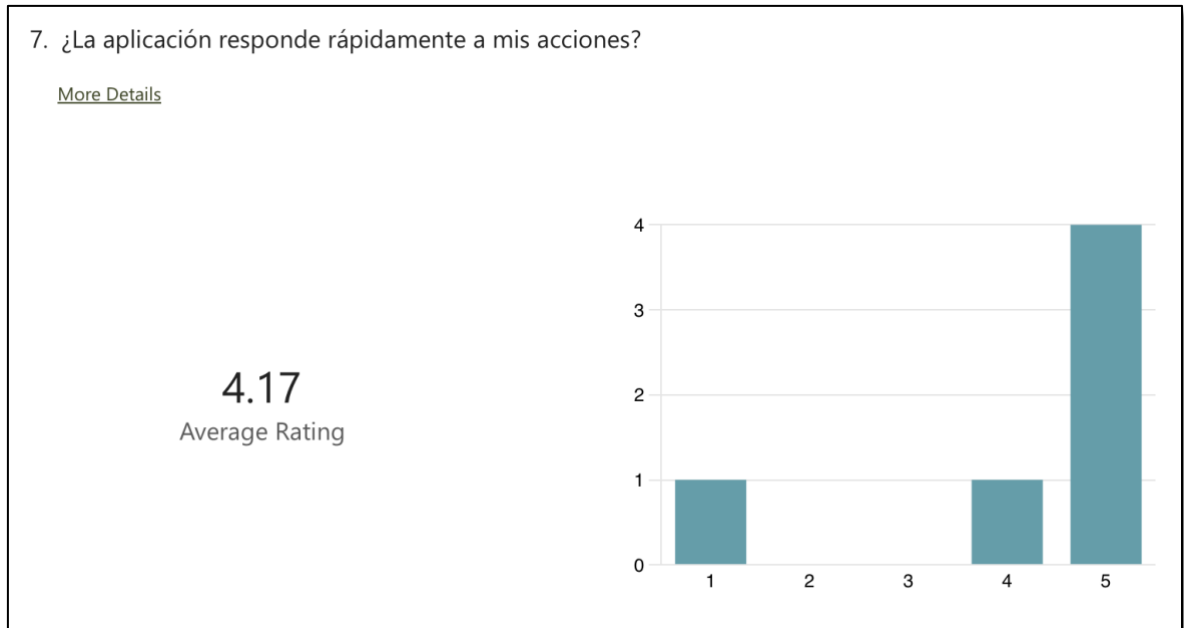


**Figura 27.** Resultados de la Pregunta 6: ¿La opción de cambiar el tema de oscuro a claro (y viceversa) es útil?

**Pregunta 7: ¿La aplicación responde rápidamente a mis acciones?**

En la séptima pregunta de la encuesta, los usuarios evaluaron la capacidad de respuesta de la aplicación. Como se puede observar en la **Figura 28**, la mayoría de los encuestados calificaron esta característica con una puntuación alta, obteniendo un promedio de 4.17.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron la calificación más alta (5), con una única respuesta en 1 y una en 4, lo que sugiere que la aplicación responde rápidamente a las acciones de la mayoría de los usuarios. Este resultado indica que la aplicación tiene un buen desempeño en términos de velocidad y capacidad de respuesta, lo cual es crucial para la satisfacción del usuario.

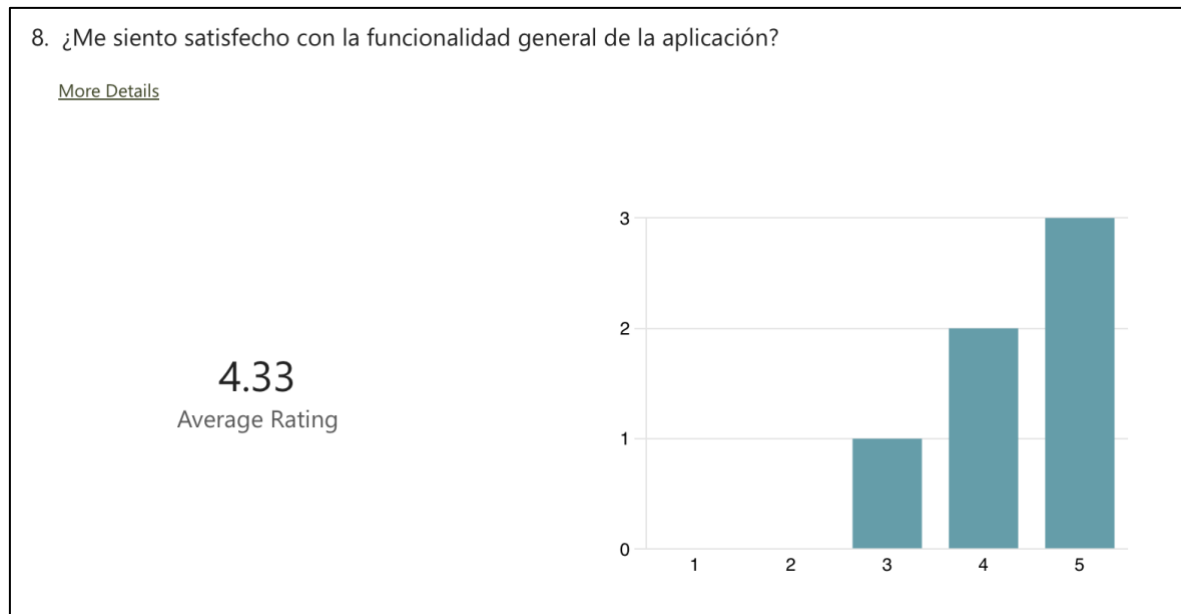


**Figura 28.** Resultados de la Pregunta 7: ¿La aplicación responde rápidamente a mis acciones?

**Pregunta 8: ¿Me siento satisfecho con la funcionalidad general de la aplicación?**

En la octava pregunta de la encuesta, los usuarios evaluaron su nivel de satisfacción con las funcionalidades ofrecidas por la aplicación. Como se puede observar en la **Figura 29**, la mayoría de los encuestados calificaron esta característica con una puntuación alta, obteniendo un promedio de 4.33.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron las calificaciones más altas (4 y 5), con una única respuesta en 3, lo que sugiere que la mayoría de los usuarios están satisfechos con la funcionalidad general de la aplicación. Este resultado es positivo y refleja que las funcionalidades implementadas cumplen con las expectativas y necesidades de los usuarios, proporcionando una experiencia satisfactoria.

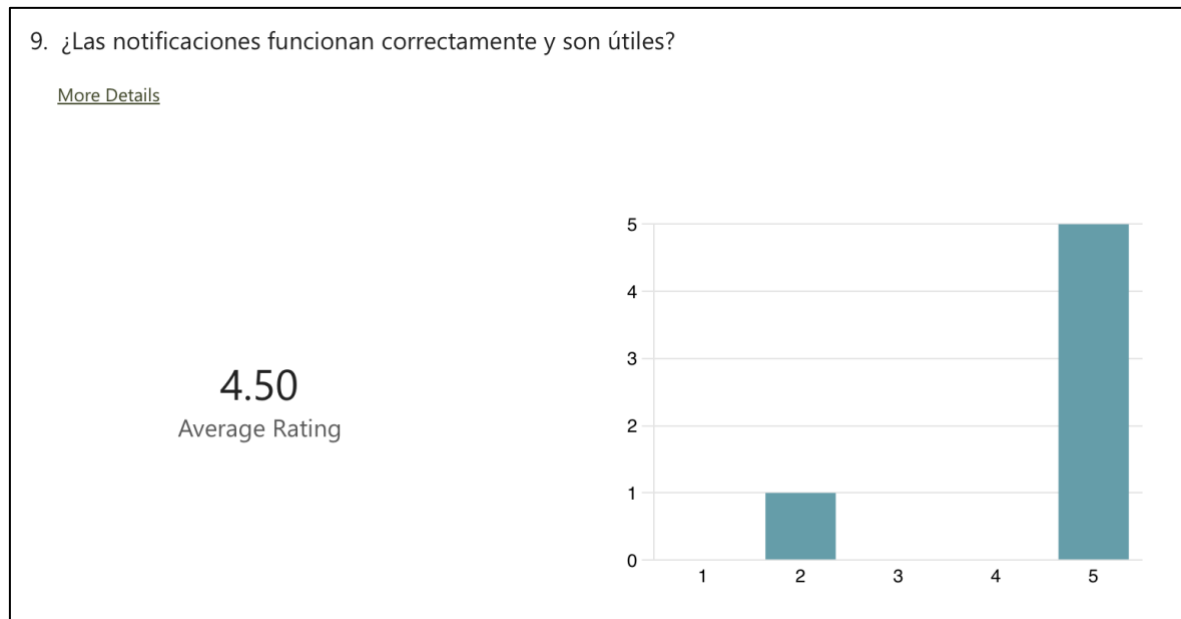


**Figura 29.** Resultados de la Pregunta 8: ¿Me siento satisfecho con la funcionalidad general de la aplicación?

**Pregunta 9: ¿Las notificaciones funcionan correctamente y son útiles?**

En la novena pregunta de la encuesta, "¿Las notificaciones funcionan correctamente y son útiles?", los usuarios evaluaron la efectividad y utilidad de las notificaciones implementadas en la aplicación. Como se puede observar en la **Figura 30**, la mayoría de los encuestados calificaron esta característica con una puntuación alta, obteniendo un promedio de 4.50.

La distribución de respuestas muestra que la mayoría de los usuarios eligieron la calificación más alta (5), con una única respuesta en 2, lo que sugiere que las notificaciones son funcionales y útiles para la mayoría de los usuarios. Este resultado indica que el sistema de notificaciones cumple con su propósito, proporcionando alertas oportunas y relevantes que mejoran la capacidad de respuesta de los usuarios ante situaciones críticas.



**Figura 30.** Resultados de la Pregunta 9: ¿Las notificaciones funcionan correctamente y son útiles?

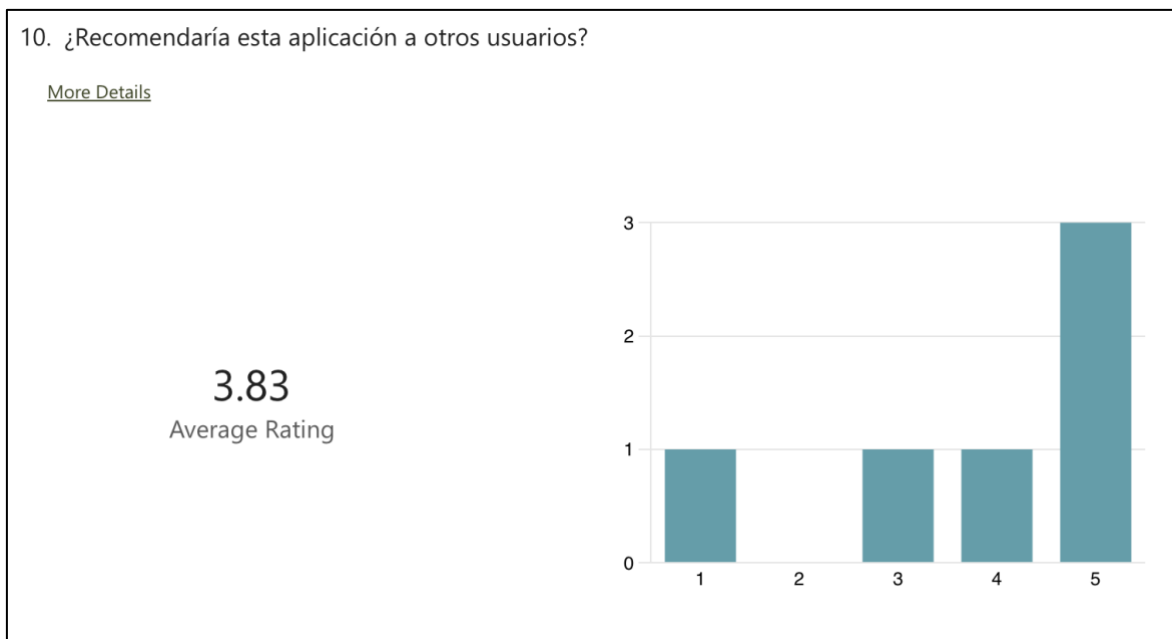
**Pregunta 10: ¿Recomendaría esta aplicación a otros usuarios?**

En la décima pregunta de la encuesta, "¿Recomendaría esta aplicación a otros usuarios?", se evaluó la disposición de los usuarios para recomendar la aplicación. Según la **Figura 31**, la puntuación promedio para esta pregunta fue de 3.83.

La distribución de respuestas muestra una variedad de opiniones. La mayoría de los usuarios seleccionaron la calificación más alta (5), indicando una fuerte disposición a recomendar la aplicación. Sin embargo, también se registraron calificaciones bajas (1), lo que sugiere que algunos usuarios tuvieron experiencias menos positivas.

Este rango de respuestas indica que, aunque la aplicación ha recibido una aceptación positiva en general, hay áreas que podrían beneficiarse de mejoras adicionales para asegurar una recomendación más consistente entre todos los usuarios.





**Figura 31.** Resultados de la Pregunta 10: ¿Recomendaría esta aplicación a otros usuarios?

#### **Pregunta 11: Recomendaciones y Comentarios:**

La última sección de la encuesta solicitó a los usuarios proporcionar recomendaciones y comentarios para mejorar la aplicación. Se recibieron cinco respuestas anónimas, cada una ofreciendo valiosas perspectivas sobre cómo se puede mejorar la aplicación. Las respuestas se presentan en la **Figura 32**.

#### **Respuestas:**

1. **Persona 1:** Implementar CO2 en los invernaderos para medir el cambio entre las plantas pequeñas y las que están produciendo. Poder acceder a los datos desde cualquier dispositivo ayudaría a mantener un mejor control de los invernaderos. Hacer comparaciones de rangos de días como por ejemplo entre semanas. Y para un futuro poder controlar el funcionamiento de los sensores desde la computadora sería estupendo, por ejemplo abrir una válvula de agua o que funcione el ventilador.
2. **Persona 2:** Para nosotros, sería importante integrar la aplicación con nuestro sitio de internet y que haya una forma de operar las válvulas y abanicos desde el dashboard.
3. **Persona 3:** Super interesante porque uno no puede estar ahí siempre pendiente.
4. **Persona 4:** SE ME PARECIÓ UN POCO COMPLICADO.
5. **Persona 5:** Pienso que sería útil tener la información de las 24 horas pero eso me parece que no tiene que ver con la aplicación.

Estos comentarios subrayan la importancia de varias mejoras, como la implementación de CO2, el acceso remoto a los datos, la integración con otros sistemas y la simplificación de la interfaz de usuario. Estas sugerencias proporcionan una guía útil para futuras iteraciones y mejoras del sistema.

11. Recomendaciones y Comentarios: Por favor, escribe cualquier recomendación o comentario que tengas para mejorar la aplicación:		
5 Responses		
ID ↑	Name	Responses
1	anonymous	Implementar CO2 en los invernaderos para medir el cambio entre las plantas pequeñas y las que están produciendo. Poder acceder a los datos desde cualquier dispositivo ayudaría a mantener un mejor control de los invernaderos. Hacer comparaciones de rangos de días como por ejemplo entre semanas. Y para un futuro poder controlar el funcionamiento de los sensores desde la computadora sería estupendo, por ejemplo abrir una válvula de agua o que funcione el ventilador.
2	anonymous	Para nosotros, sería importante integrar la aplicación con nuestro sitio de internet y que hay una forma de operar las valvulas y abanicos desde el dashboard.
3	anonymous	Super interesante porque uno no puede estar ahí siempre pendiente.
4	anonymous	SEMEPARECIO UN POCO COMPLICADO
5	anonymous	Pienso que sería util tener la información de las 24 horas pero eso me parece que no tiene que ver con la aplicación

**Figura 32.** Resultados de la Pregunta 10: Recomendaciones y Comentarios

## 3.2 Conclusiones

El proyecto de desarrollo del sistema de monitoreo y gestión de invernaderos ha logrado avances significativos a través de una serie de sprints planificados y ejecutados. A lo largo de estos sprints, se implementaron funcionalidades que han mejorado la capacidad de los encargados de los invernaderos para monitorear y gestionar las condiciones ambientales de manera eficiente.

Los resultados obtenidos de las pruebas de usabilidad han proporcionado datos valiosos que confirman la eficacia de las funcionalidades implementadas. Las encuestas realizadas a usuarios finales indicaron una alta satisfacción con la facilidad de uso, la claridad de los gráficos y tablas, y la utilidad de las notificaciones y configuraciones de umbrales. Sin

embargo, también se identificaron áreas de mejora que deben ser abordadas en futuros sprints.

Las recomendaciones y comentarios proporcionados por los usuarios subrayan la necesidad de integrar nuevas métricas como el CO<sub>2</sub>, mejorar el acceso remoto a los datos, y simplificar la interfaz de usuario. Estos comentarios representan una guía invaluable para planificar futuras iteraciones del proyecto.

Con base en el feedback recibido, se pueden definir los siguientes pasos para el desarrollo del sistema:

1. **Integración de Nuevas Métricas:** Implementar sensores adicionales, como los de CO<sub>2</sub>, para ofrecer un monitoreo más completo y preciso de las condiciones del invernadero.
2. **Acceso Remoto y Control:** Desarrollar funcionalidades que permitan a los usuarios acceder y controlar el sistema de manera remota, incluyendo la operación de válvulas y ventiladores.
3. **Simplificación de la Interfaz:** Continuar refinando la interfaz de usuario para hacerla más intuitiva y fácil de usar, basándose en los comentarios de los usuarios.
4. **Mejoras en la Visualización de Datos:** Ampliar las capacidades de visualización de datos para permitir comparaciones más detalladas y personalizadas de las métricas.
5. **Optimización del Rendimiento:** Asegurar que la aplicación responda rápidamente a las acciones del usuario y maneje eficientemente grandes volúmenes de datos.

El proyecto ha sentado una base sólida para un sistema robusto y eficiente de monitoreo y gestión de invernaderos. Los datos empíricos recopilados a través de las pruebas de usabilidad han proporcionado una visión clara de las necesidades y expectativas de los usuarios finales, lo que permitirá guiar el desarrollo futuro del sistema. Al adoptar un enfoque iterativo y centrado en el usuario, el equipo de desarrollo puede continuar mejorando la aplicación para satisfacer mejor las demandas del mercado y asegurar el éxito a largo plazo del proyecto.

### 3.3 Recomendaciones

A partir de la evaluación y análisis del proyecto, se presentan las siguientes recomendaciones para guiar el desarrollo futuro del sistema de monitoreo y gestión de invernaderos:

### 1. Incorporar Nuevas Métricas:

- **Implementar Sensores de CO2:** Añadir sensores para medir el CO2 en los invernaderos, lo que permitirá un monitoreo más completo de las condiciones ambientales y ayudará a optimizar el crecimiento de las plantas.
- **Expandir la Variedad de Sensores:** Considerar la inclusión de otros tipos de sensores, como de luz y pH, para ofrecer una visión más integral de las condiciones de cultivo.

### 2. Mejorar el Acceso y Control Remoto:

- **Desarrollar Funcionalidades de Control Remoto:** Permitir que los usuarios puedan operar válvulas, ventiladores y otros dispositivos a través de la aplicación, incluso desde ubicaciones remotas.
- **Integración con Plataformas de IoT:** Considerar la integración del sistema con plataformas de Internet de las Cosas (IoT) para mejorar la conectividad y el control en tiempo real.

### 3. Optimización y Simplificación de la Interfaz:

- **Refinar la Experiencia de Usuario (UX):** Continuar mejorando la interfaz de usuario para hacerla más intuitiva y accesible, reduciendo la complejidad y asegurando que las funcionalidades clave sean fáciles de encontrar y utilizar.
- **Realizar Pruebas de Usabilidad Continuas:** Implementar un ciclo regular de pruebas de usabilidad con usuarios finales para identificar y corregir posibles problemas de diseño.

### 4. Ampliar las Capacidades de Visualización de Datos:

- **Implementar Comparaciones Detalladas:** Desarrollar funcionalidades que permitan a los usuarios comparar métricas de diferentes días o periodos en una sola gráfica, facilitando la identificación de tendencias y patrones.
- **Personalizar la Visualización:** Ofrecer opciones de personalización para que los usuarios puedan ajustar cómo se muestran los datos según sus necesidades específicas.

### 5. Mejorar el Sistema de Notificaciones:

- **Detallar las Alertas:** Asegurarse de que las notificaciones incluyan información completa, como la hora, el invernadero afectado y la métrica que ha cruzado el umbral crítico.
- **Facilitar la Gestión de Notificaciones:** Implementar un botón para limpiar todas las notificaciones y asegurar que los usuarios puedan gestionar fácilmente las alertas recibidas.

#### 6. Optimización del Rendimiento:

- **Asegurar una Respuesta Rápida:** Continuar optimizando el rendimiento de la aplicación para asegurar que responda rápidamente a las acciones del usuario, especialmente en situaciones críticas.
- **Gestión Eficiente de Datos:** Mejorar la gestión de grandes volúmenes de datos para evitar problemas de rendimiento y asegurar que la aplicación pueda manejar eficientemente el monitoreo continuo.

#### 7. Implementar Mecanismos de Seguridad:

- **Proteger la Integridad de los Datos:** Asegurar que los datos recopilados por los sensores sean almacenados y transmitidos de manera segura para evitar pérdidas o manipulaciones.
- **Autenticación y Autorización:** Implementar robustos sistemas de autenticación y autorización para asegurar que solo usuarios autorizados puedan acceder y modificar los datos del sistema.

Estas recomendaciones tienen como objeto no solo mejorar la funcionalidad y usabilidad del sistema actual, sino también preparar el camino para futuras expansiones y mejoras. Además, pretenden que el sistema de monitoreo y gestión de invernaderos continúe evolucionando y satisfaciendo las necesidades de los usuarios de manera efectiva.