

**MANUAL TÉCNICO**

**DAVID ALEJANDRO ANZOLA CAICEDO**

**FRANKLIN VILLALBA**

**DAVID SANTIAGO BELTRAN PEDRAZA**

**PROYECTO        NO. 9**

**“SISTEMA DE MONITOREO DE CALIDAD DEL AIRE EN INVERNADEROS”**

**SERVICIO NACIONAL DE APRENDIZAJE “S.E.N.A.”**

**ANÁLISIS Y DESARROLLO DE SOFTWARE**

**FICHA - 3203084**

**MOSQUERA, CUNDINAMARCA, AGOSTO 2025**

# **CONTENIDO**

## **1. Introducción**

- 1.1. Objetivo del manual
- 1.2. Alcance del sistema
- 1.3. Metodología de desarrollo

## **2. Análisis del Sistema**

- 2.1. Análisis de requerimientos del sistema
- 2.2. Análisis de la Arquitectura del Sistema
- 2.3. Análisis de Viabilidad

## **3. Diagramas de Arquitectura de Software (UML)**

- 3.1. Diagrama de Componentes
- 3.2. Diagrama de Interacción (Secuencia)
- 3.3. Diagrama de Despliegue

## **4. Modelo de Datos**

- 4.1. Diagrama Entidad–Relación (ERD)
- 4.2. Detalle de entidades y atributos
- 4.3. Justificación de entidades y relaciones

## **5. Casos de Uso**

- 5.1. Actores del sistema
- 5.2. Diagramas de casos de uso
- 5.3. Descripción casos de uso

## **6. Componentes del Sistema**

- 6.1 Componentes principales
- 6.2. Relaciones entre componentes
- 6.3. Beneficios del Diagrama de Componentes

## **7. Arquitectura del Sistema**

- 7.1. Distribución de servicios
- 7.2. Flujo de interacción entre componentes
- 7.3. Justificación de la arquitectura adoptada

## **8. Descripción de la Implementación**

- 8.1. Tecnologías utilizadas
- 8.2. Lenguajes y frameworks

## **9. Conclusiones**

- 9.1. Beneficios del sistema
- 9.2. Limitaciones actuales
- 9.3. Recomendaciones futuras

## **Introducción**

El presente manual técnico documenta el diseño, desarrollo e implementación de un sistema de monitoreo de calidad del aire en invernaderos, concebido como una herramienta tecnológica que facilita el control de variables ambientales críticas para la productividad agrícola.

El documento organiza de manera sistemática la información necesaria para comprender la arquitectura del sistema, su funcionamiento y los procedimientos técnicos que permiten su correcta utilización.

## **Objetivo**

El objetivo principal de este manual es proporcionar una guía clara, precisa y estructurada sobre el uso y funcionamiento del sistema de monitoreo. De manera específica, busca:

Describir la arquitectura y componentes principales del sistema.

Explicar el modelo de datos y su implementación.

Establecer las pautas de instalación, configuración y operación.

Ofrecer lineamientos para el mantenimiento preventivo y correctivo.

## **Alcance**

Este manual está dirigido a técnicos, desarrolladores, investigadores y usuarios finales que interactúan con el sistema. El contenido abarca desde la fundamentación conceptual hasta los aspectos prácticos de implementación. No obstante, se excluyen temas avanzados de programación o personalización profunda del software, los cuales requieren documentación especializada.

El alcance se centra en:

La integración de sensores para la medición de temperatura, humedad y niveles de CO<sub>2</sub>.

La conexión con una base de datos centralizada.

El uso de una interfaz gráfica de usuario para la visualización y análisis de datos.

Procedimientos básicos de instalación y mantenimiento.

## **Metodología**

Para la construcción del sistema y elaboración de este manual se siguieron las siguientes etapas metodológicas:

1. Levantamiento de requerimientos: identificación de las necesidades del entorno agrícola y definición de las variables críticas a monitorear.
2. Diseño del sistema: modelado de la arquitectura, selección de componentes y definición del modelo de datos.
3. Implementación: desarrollo de la base de datos, integración de sensores y construcción de la interfaz de usuario.
4. Pruebas y validación: ejecución de casos de uso, verificación de funcionamiento y ajustes correspondientes.
5. Documentación: elaboración de este manual como guía técnica de referencia para usuarios e implementadores.

## **2. ANÁLISIS DEL SISTEMA**

### **2.1 Análisis de Requerimientos:**

El análisis de requerimientos identifica las necesidades principales que el sistema debe cumplir, tanto a nivel funcional como técnico.

Requerimientos funcionales: el sistema debe capturar y procesar datos en tiempo real (ejemplo: temperatura, humedad, niveles de CO<sub>2</sub>).

Requerimientos no funcionales: debe garantizar bajo consumo energético, estabilidad, seguridad de la información y escalabilidad.

Usuarios objetivo: profesionales, técnicos y personal operativo que requieren información precisa y confiable para la toma de decisiones.

### **2.2 Análisis de la Arquitectura del Sistema:**

La arquitectura define la estructura del sistema y la interacción de sus componentes principales.

Sensores y dispositivos: recogen datos del entorno de manera continua y precisa.

Servidor o base de datos: gestiona, organiza y almacena los registros recolectados.

Interfaz de usuario: muestra la información de forma visual y amigable, permitiendo reportes, consultas y análisis de tendencias.

## 2.3 Análisis de Viabilidad:

Este análisis determina si el sistema es factible de implementar considerando aspectos técnicos, económicos y operativos.

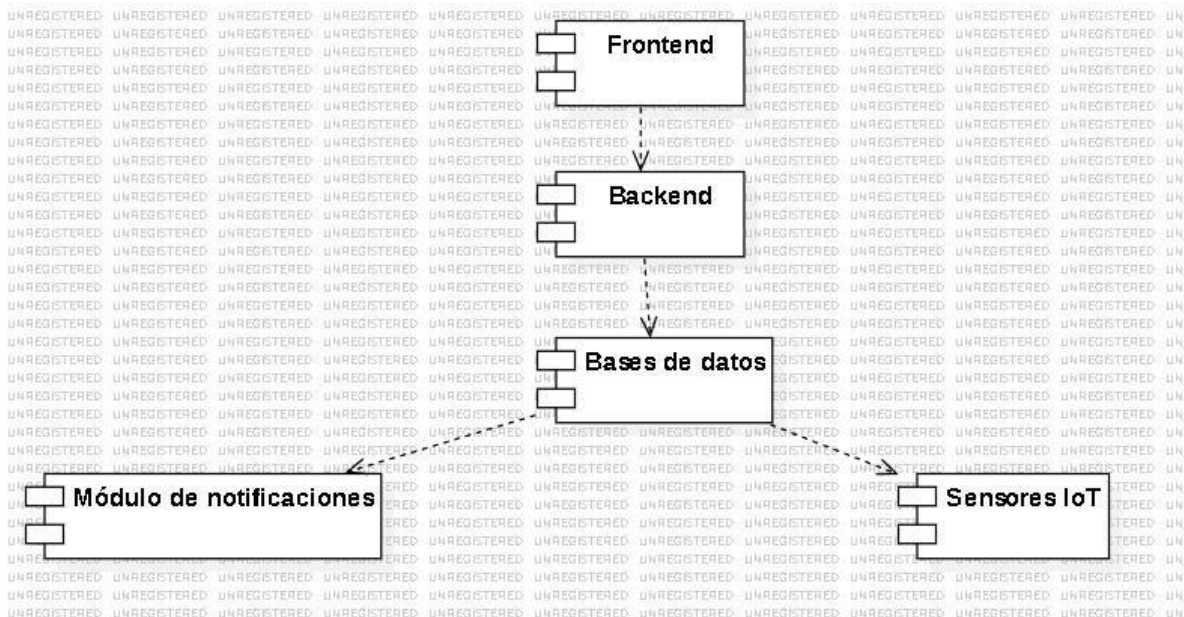
**Viabilidad técnica:** los componentes y tecnologías requeridas están disponibles en el mercado y son compatibles entre sí.

**Viabilidad económica:** el costo de desarrollo, implementación y mantenimiento es acorde al presupuesto proyectado.

**Viabilidad operativa:** el sistema puede integrarse fácilmente en los procesos actuales y el personal puede adaptarse con mínima capacitación.

## 3.0 DIAGRAMA DE ARQUITECTURA DE SOFTWARE (UML)

### 3.1 Diagrama de Componentes:



¿POR QUÉ ESOS COMPONENTES?

Frontend: Básicamente el frontend es lo que le mostrará al usuario o cliente los datos, reportes o acciones que necesite hacer en la plataforma, para esto se usaría un framework como Bootstrap que nos facilitaría el diseño CSS de la página. Backend: El backend básicamente es la parte lógica y funcional desde la cual básicamente hará que funcione correctamente la página y se gestionen bien los datos y se manipule bien la información de las bases de datos, y que a través del frontend se muestren esos datos. Para esto usaremos un lenguaje de programación sencillo y fácil de usar como lo es Python y un framework como Django el cual facilita el desarrollo de las aplicaciones web.

**Bases de Datos:** Aquí básicamente se guardarán todos los datos tanto de usuarios, administradores, sensores IoT, invernaderos, entre más datos. Esto permitiéndonos acceder a la información que se necesita de manera rápida y almacenándola para organizar toda esa información.

En este caso se usaría MySQL que nos permite crear, almacenar, organizar, gestionar o eliminar las bases de datos.

**Sensores IoT:** Estos son necesarios ya que con estos sensores se determinará las variables en la calidad del aire en los invernaderos que pide el cliente y se guardaran en la base de datos, para que después se envíen los datos al backend en el momento que el usuario necesite ver variables en la calidad del aire y se generen los reportes en la plataforma.

**Servicio de Notificaciones:** En esta se utilizará una API externa que el backend utilizará para generar las alertas o notificaciones sobre reportes en la calidad del aire en los invernaderos y avisar al usuario de manera efectiva.

Al igual se usa un tipo de arquitectura Modular distribuida (estilo microservicios ligeros) que es adecuada en este proyecto para lograr escalabilidad, flexibilidad y fácil mantenimiento.

Básicamente estos componentes son adecuados para nuestro proyecto sobre un sistema de monitoreo de calidad de aire pues gracias a estos los reportes en los invernaderos que haya permiten definir bien las condiciones que hay, haciendo que el cliente pueda tomar decisiones y ver problemáticas en sus invernaderos favoreciendo la toma de acciones rápida en los invernaderos, y aprovechando estos datos para la calidad y salud de los cultivos.



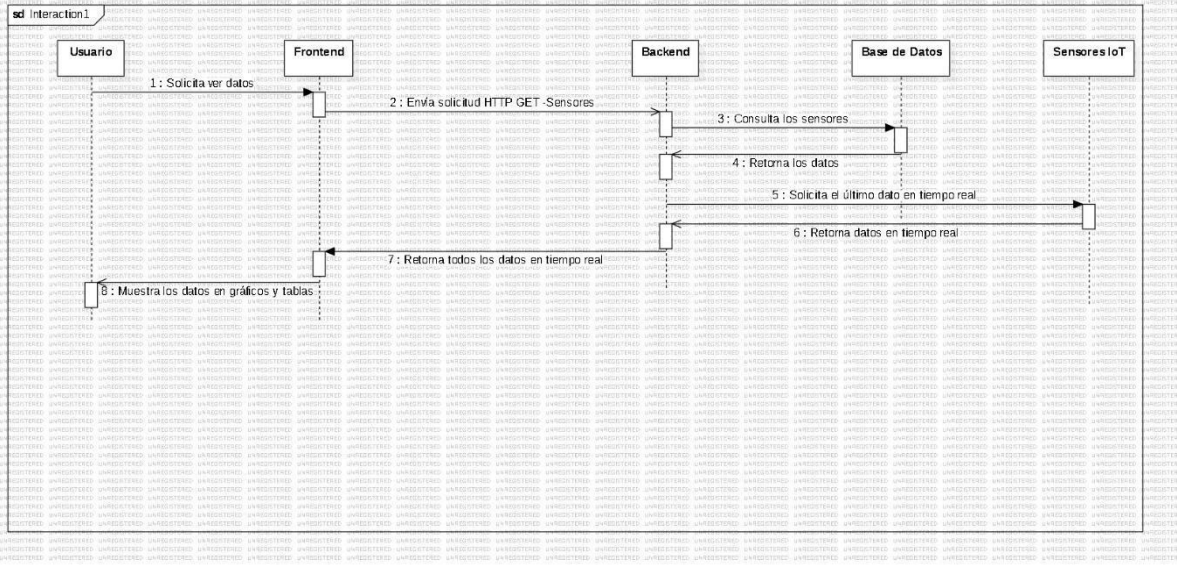
Característica	Caso de Uso UC008: Arquitectura de Componentes del Sistema
<b>**Nombre**</b>	Diagrama de Componentes del Sistema de Monitoreo de Calidad del Aire
<b>**Descripción**</b>	Muestra la estructura interna del sistema, organizando el software en componentes lógicos y físicos (módulos, librerías, clases, servicios) y sus dependencias o interfaces. Detalla cómo interactúan las partes de software entre sí.
<b>**Caso de Uso**</b>	UC008: Arquitectura de Componentes del Sistema

<b>**Autor**</b>	Equipo de Arquitectura de Software
<b>**Fecha**</b>	28/08/2025
<b>**Descripción del Proyecto**</b>	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.
<b>**Actores**</b>	Módulo de Adquisición de Datos, Módulo de Procesamiento de Datos, Módulo de Almacenamiento de Datos (Base de Datos), Módulo de Interfaz de Usuario (UI), Módulo de Notificaciones, Módulo de Configuración de Sensores, Módulo de Generación de Reportes, Componente de Autenticación/Autorización, APIs de Terceros (ej. Servicios de SMS/Email).
<b>**Precondiciones**</b>	Los requisitos de software están definidos, las interfaces entre módulos están especificadas, el entorno de desarrollo está configurado.

<b>**Flujo Normal (Acción del Actor)**</b>	1. El Módulo de Adquisición de Datos recibe información de los sensores.   2. El Módulo de Adquisición de Datos envía los datos crudos al Módulo de Procesamiento de Datos.   3. El Módulo de Procesamiento de Datos valida y transforma los datos.   4. El Módulo de Procesamiento de Datos envía los datos procesados al Módulo de Almacenamiento de Datos.   5. El Módulo de Interfaz de Usuario solicita datos al Módulo de Procesamiento de Datos/Almacenamiento.   6. El Módulo de Notificaciones recibe solicitudes de alerta del Módulo de Procesamiento y las envía a través de APIs de Terceros.   7. El Módulo de Configuración de Sensores envía y recibe configuraciones.
<b>**Flujo Normal (Respuesta del Sistema)**</b>	1. El Módulo de Procesamiento de Datos ejecuta reglas de negocio y lógica de umbrales.   2. El Módulo de Almacenamiento de Datos guarda los datos de forma persistente.   3. El Módulo de Interfaz de Usuario renderiza los datos en gráficos y tablas.   4. El Módulo de Notificaciones gestiona la entrega de alertas.   5. El Módulo de Configuración de Sensores actualiza los parámetros en la base de datos y/o en los dispositivos.   6. El Módulo de Generación de Reportes compila información de la Base de Datos.   7. El Componente de Autenticación/Autorización valida el acceso de los otros módulos.

<b>**Flujo Alternativo**</b>	<p>\- <b>**Fallo en un Componente**</b>: Un componente notifica a un módulo de monitoreo interno sobre su fallo, y el sistema puede intentar reiniciar o aislar el componente. &lt;br&gt; - <b>**Corrupción de Datos**</b>: El Módulo de Procesamiento de Datos detecta inconsistencias y rechaza/corrigie los datos antes de almacenarlos. &lt;br&gt; - <b>**Error de Comunicación entre Componentes**</b>: El sistema implementa reintentos o circuitos de interrupción (circuit breakers) para manejar fallos temporales. &lt;br&gt; - <b>**Base de Datos Inaccesible**</b>: Los módulos que dependen de ella almacenan datos en caché temporalmente o fallan graciosamente.</p>
<b>**Postcondiciones**</b>	<p>Los componentes del sistema interactúan de manera efectiva para capturar, procesar, almacenar, visualizar y alertar sobre la calidad del aire. La arquitectura del software es modular, permitiendo escalabilidad y mantenimiento independiente.</p>
<b>**Frecuencia**</b>	<p>Continua (operación de los componentes), o a demanda (ejecución de ciertos módulos de generación de reportes).</p>
<b>**Tiempo de Respuesta**</b>	<p>Interacciones internas entre componentes: Sub-milisegundos a milisegundos. Latencia de procesamiento de datos: Menos de 1 segundo.</p>

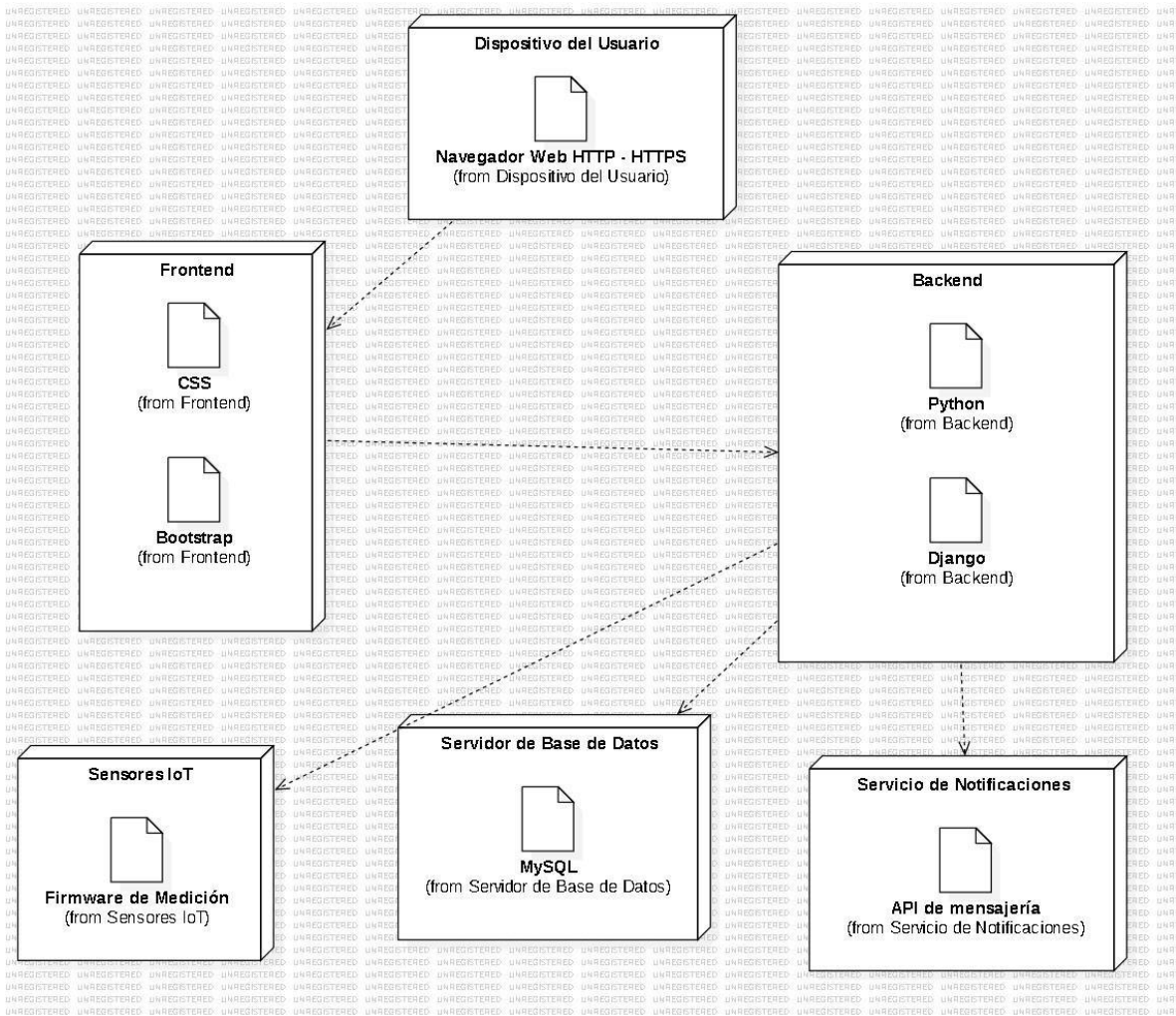
3.2 Diagrama de Secuencia:



En el diagrama de secuencia se muestra el flujo de comunicación entre los diferentes componentes y el usuario:

1. El usuario accede a la plataforma desde su dispositivo mediante un navegador web.
2. El frontend recibe la solicitud y la transmite al backend usando protocolos HTTP/HTTPS.
3. El backend procesa la solicitud y consulta la información en la base de datos.
4. Los sensores IoT alimentan continuamente la base de datos con nuevos registros de medición.
5. El sistema puede enviar notificaciones automáticas al usuario si se cumplen determinadas condiciones (ejemplo: exceso de CO<sub>2</sub>)
6. Finalmente, el frontend muestra al usuario la respuesta en forma de reportes o gráficos.

### **3.3 Diagrama de Despliegue:**



Característica	Caso de Estudio: Diagrama de Despliegue del Sistema
**Nombre**	Diagrama de Despliegue del Sistema de Monitoreo de Calidad del Aire
**Descripción**	Representa la arquitectura física del sistema, mostrando los nodos de hardware, los componentes de software desplegados en ellos y las conexiones entre estos. Detalla cómo se distribuyen las partes del sistema en el entorno real (invernaderos, centro de datos, nube).
**Caso de Uso**	UC006: Despliegue de la Infraestructura del Sistema
**Autor**	Equipo de Arquitectura de Soluciones
**Fecha**	28/08/2025
**Descripción del Proyecto**	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.
**Actores**	Nodos de Sensores (en invernaderos), Concentrador de Datos (Gateway/Servidor local), Servidor de Aplicaciones (Nube/Centro de Datos), Base de Datos (Nube/Centro de Datos), Servidor de Notificaciones, Dispositivos Móviles/PCs de Usuarios.
**Precondiciones**	La infraestructura de red de los invernaderos está establecida (Wi-Fi/LoRaWAN), el entorno de servidor (local o en la nube) está provisionado y configurado, los sensores están físicamente instalados en los invernaderos.



<b>**Postcondiciones**</b>	Los datos de calidad del aire son capturados, transmitidos, procesados y almacenados continuamente. Los usuarios pueden visualizar los datos y recibir alertas basadas en los umbrales configurados. La infraestructura de despliegue está operativa y robusta.
<b>**Frecuencia**</b>	Operación continua del sistema (24/7).
<b>**Tiempo de Respuesta**</b>	Latencia de extremo a extremo (sensor a visualización/alerta): Menos de 10 segundos para datos en tiempo real; menos de 2 segundos para la entrega de alertas críticas.
<b>**Flujo Normal (Acción del Actor)**</b>	1. Los Nodos de Sensores capturan datos ambientales.   2. Los Nodos de Sensores envían datos al Concentrador de Datos.   3. El Concentrador de Datos reenvía datos al Servidor de Aplicaciones.   4. El Servidor de Aplicaciones procesa los datos y los almacena en la Base de Datos.   5. El Servidor de Aplicaciones envía solicitudes de visualización a los Dispositivos de Usuarios.   6. El Servidor de Aplicaciones envía alertas al Servidor de Notificaciones, que las distribuye a los Dispositivos de Usuarios.
<b>**Flujo Normal (Respuesta del Sistema)**</b>	1. El Concentrador de Datos recibe y valida los datos de los sensores.   2. El Concentrador de Datos establece conexión con el Servidor de Aplicaciones.   3. El Servidor de Aplicaciones ingiere, transforma y guarda los datos.   4. El Servidor de Aplicaciones responde a las solicitudes de visualización con datos procesados.   5. El Servidor de Aplicaciones evalúa umbrales y activa el Servidor de Notificaciones.   6. El Servidor de Notificaciones entrega las alertas a los canales configurados (email, SMS, app push).
<b>**Flujo Alternativo**</b>	\- <b>**Fallo de Conectividad del Sensor**</b> : Los Nodos de Sensores intentan reconectar o almacenar datos localmente si es posible.   - <b>**Saturación del Concentrador**</b> : El Concentrador de Datos gestiona la cola de mensajes o rechaza nuevas conexiones temporalmente.   - <b>**Caída del Servidor de Aplicaciones**</b> : El sistema activa un mecanismo de failover o redundancia para mantener el servicio.   - <b>**Error en Base de Datos**</b> : La base de datos replica datos o revierte transacciones para mantener la integridad.   - <b>**Fallo de Servicio de Notificaciones**</b> : El sistema reintenta el envío de notificaciones o registra el fallo para un reenvío posterior.

El diagrama de despliegue muestra la infraestructura física y lógica donde se ejecuta el sistema, detallando los nodos de hardware y software que lo soportan. Su objetivo es representar cómo los componentes descritos en el diagrama anterior se distribuyen en servidores, dispositivos y redes.

En este caso, el sistema está desplegado de la siguiente manera:

### **Servidor de Aplicaciones:**

Aloja el backend desarrollado en Django.

Expone servicios mediante protocolos HTTP/HTTPS.

Contiene el motor de notificaciones para alertas al usuario.

Servidor de Base de Datos: Implementado en MySQL.

Resguarda la información histórica y en tiempo real que envían los sensores IoT.

Puede estar alojado en la misma máquina que el backend o en un servidor dedicado, dependiendo de la escalabilidad requerida.

Sensores IoT (nodos de captura):

Dispositivos físicos distribuidos en el invernadero.

Se comunican con el backend a través de la red local o internet, enviando datos de temperatura, humedad y CO<sub>2</sub>.

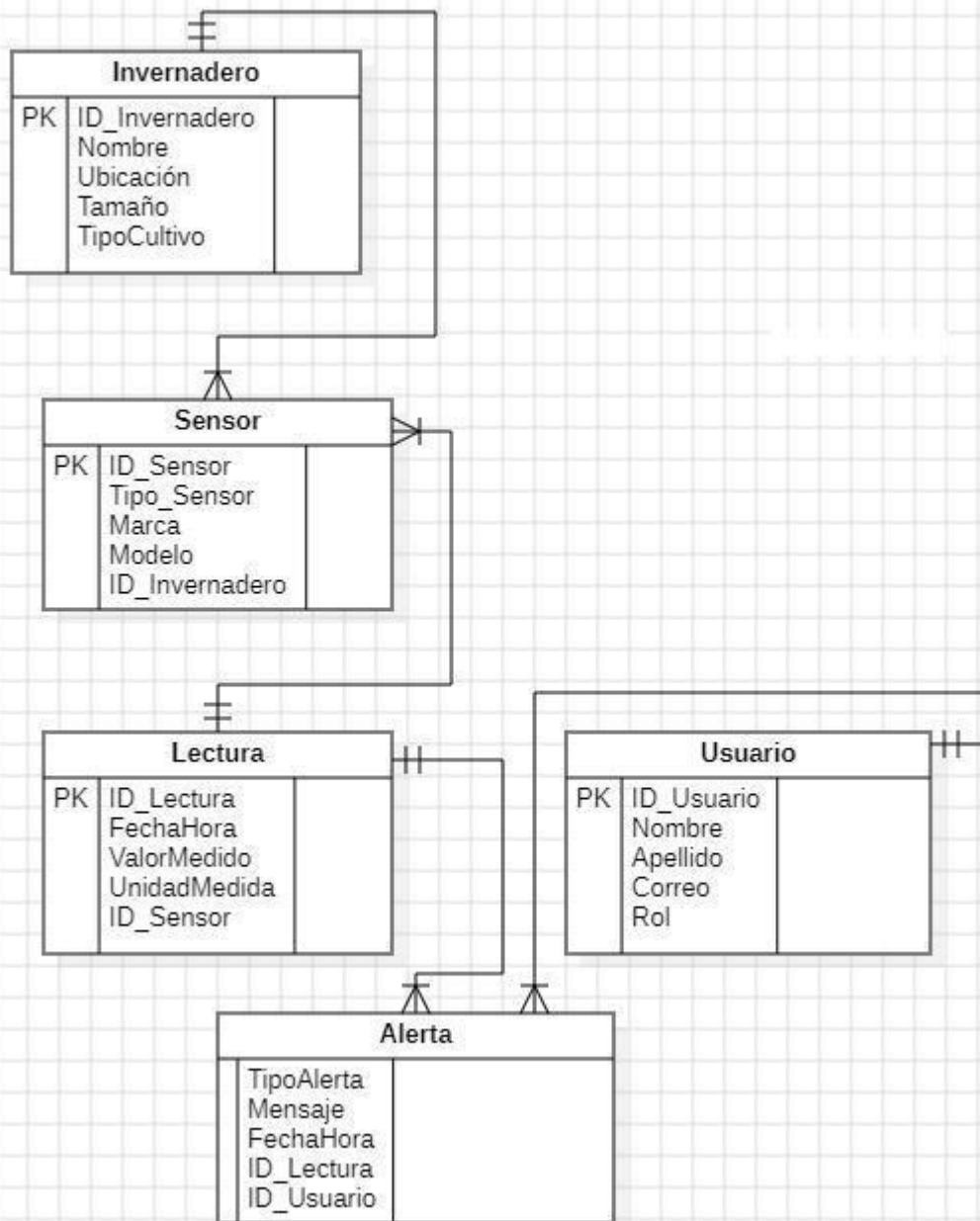
Cliente/Usuario Final:

Accede al sistema mediante un navegador web desde cualquier dispositivo (PC, tablet o smartphone).

Recibe reportes, alertas y notificaciones.

#### **4. MODELADO DE DATOS:**

##### **4.1 Diagrama de Entidad-Relación (ERD):**



Característica	Caso de Uso UC005: Interacción General del Sistema
<b>**Nombre**</b>	Interacción General del Sistema
<b>**Descripción**</b>	Describe cómo el usuario principal interactúa con los diferentes módulos y funcionalidades del sistema de monitoreo para gestionar la calidad del aire en los invernaderos, incluyendo visualización, configuración y respuesta a alertas.
<b>**Caso de Uso**</b>	UC005: Interacción del Sistema
<b>**Autor**</b>	Equipo de Desarrollo
<b>**Fecha**</b>	28/08/2025
<b>**Descripción del Proyecto**</b>	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.
<b>**Actores**</b>	Gerente de Invernadero, Agrónomo, Técnico de Invernadero
<b>**Precondiciones**</b>	El sistema está completamente operativo, los sensores están enviando datos, y el usuario ha iniciado sesión con los permisos correspondientes para todas las funcionalidades.
<b>**Flujo Normal (Acción del Actor)**</b>	1. El actor accede al panel principal del sistema.   2. El actor navega entre las secciones (ej. datos en tiempo real, configuración, alertas, reportes).   3. El actor interactúa con los controles (ej. selecciona un invernadero, ajusta un umbral, genera un reporte).   4. El actor cierra sesión.
<b>**Flujo Normal (Respuesta del Sistema)**</b>	1. El sistema muestra el panel de control general.   2. El sistema carga y muestra la información solicitada en cada sección.   3. El sistema procesa las interacciones del actor y aplica los cambios o genera los resultados.   4. El sistema finaliza la sesión del actor.
<b>**Flujo Alternativo**</b>	\- <b>**Funcionalidad No Disponible**</b> : El sistema informa que una característica seleccionada no está disponible o requiere permisos adicionales. \<br\> - <b>**Error en la Interacción**</b> : El sistema detecta un error en la entrada del usuario y solicita corrección o proporciona ayuda.
<b>**Postcondiciones**</b>	El usuario ha interactuado exitosamente con las diferentes funcionalidades del sistema y ha realizado las operaciones deseadas, o la sesión ha finalizado.
<b>**Frecuencia**</b>	Constante durante las horas de operación del usuario.
<b>**Tiempo de Respuesta**</b>	Menos de 2 segundos para la mayoría de las interacciones, hasta 15 segundos para operaciones complejas como la generación de reportes extensos.

## 4.2 Descripción de Entidades:

Invernadero: almacena información del espacio físico a monitorear (nombre, ubicación, tamaño y tipo de cultivo).

Sensor: identifica los dispositivos IoT instalados en cada invernadero, indicando tipo y modelo.

Lectura: registra los valores obtenidos por los sensores con fecha, hora y unidad de medida.



Usuario: gestiona la información de los clientes o administradores que acceden al sistema.

Alerta: permite notificar eventos críticos (ej. valores fuera de rango), vinculando la lectura y el usuario.

#### **4.3 Justificación del Modelo:**

Este conjunto de entidades refleja la necesidad del sistema de:

Representar el ambiente físico (invernadero).

Gestionar los sensores de medición.

Registrar datos históricos y en tiempo real (lecturas).

Notificar eventos críticos (alertas).

Permitir el acceso de usuarios para la consulta de información y toma de decisiones.

### **5.CASOS DE USO**

**Los casos de uso describen cómo los distintos actores interactúan con el sistema para cumplir objetivos específicos.**

**A continuación, se presentan los principales casos de uso del sistema de monitoreo de invernaderos:**

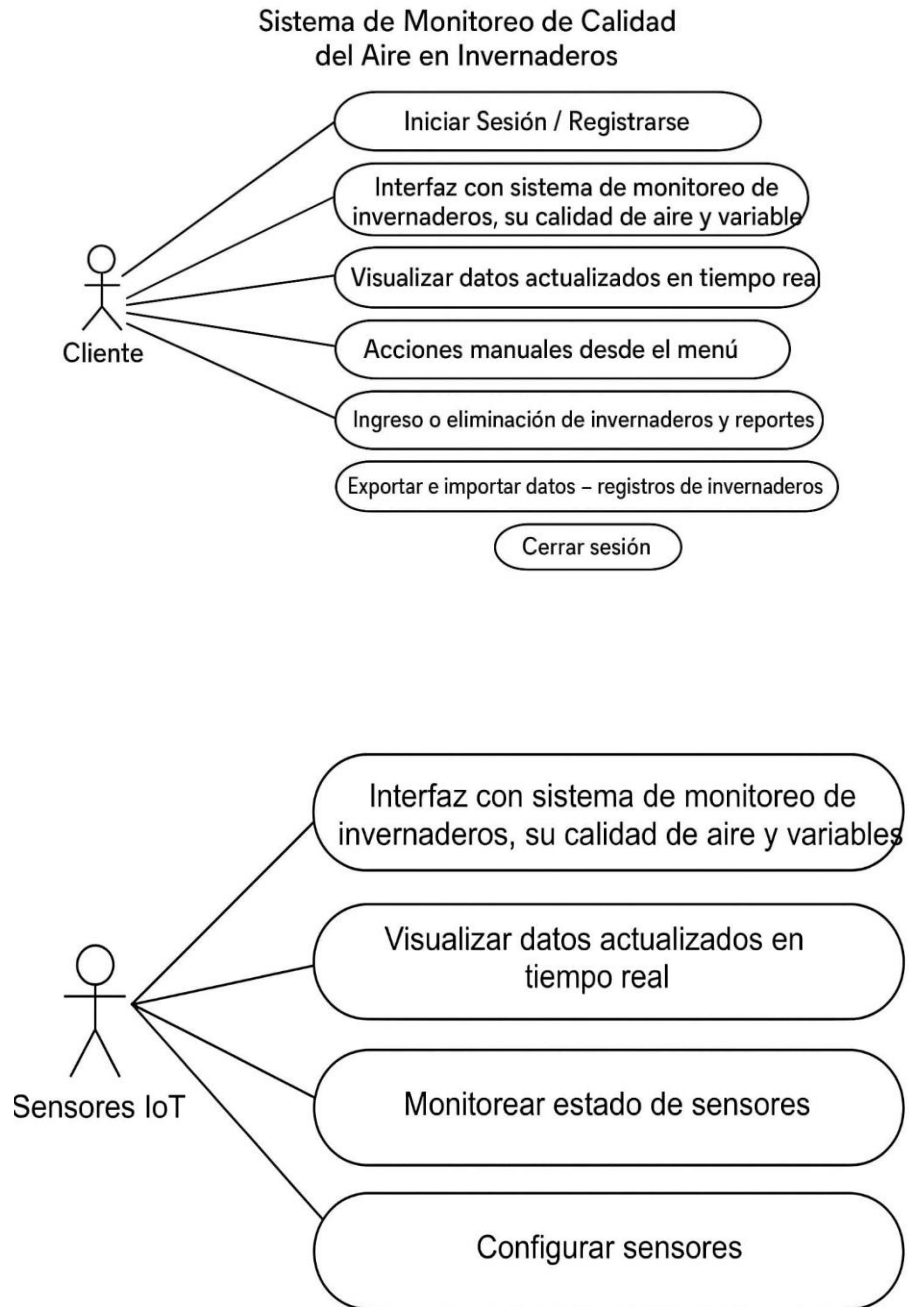
#### **5.1 Actores del Sistema:**

**Administrador:** gestiona usuarios, sensores e invernaderos.

**Usuario:** consulta información de lecturas y alertas.

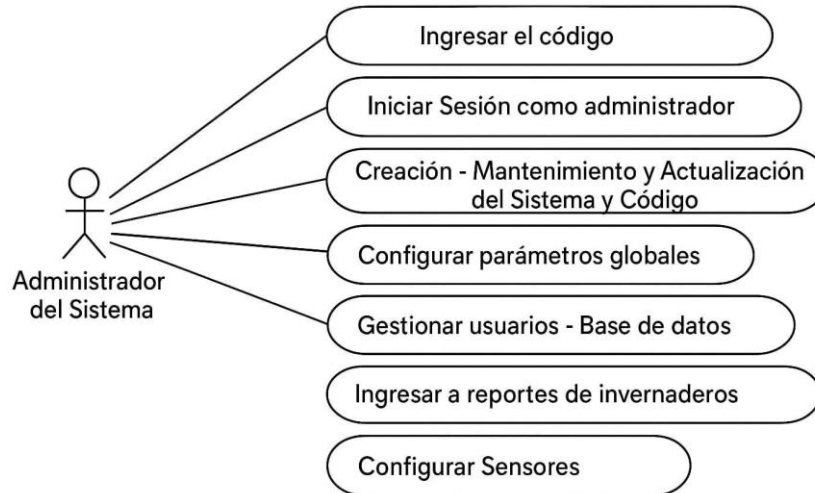
**Sistema IoT (sensores): envía datos al sistema central.**

## 5.2 Diagrama General de Casos de Uso:



Característica	Caso de Uso UC007: Administración del Sistema
<b>**Nombre**</b>	Administración del Sistema
<b>**Descripción**</b>	Permite al administrador gestionar usuarios, roles, permisos, realizar mantenimiento del sistema, configurar copias de seguridad y revisar los registros (logs) para asegurar el correcto funcionamiento y seguridad de la plataforma.
<b>**Caso de Uso**</b>	UC007: Administración del Sistema
<b>**Autor**</b>	Equipo de Operaciones y Mantenimiento
<b>**Fecha**</b>	28/08/2025
<b>**Descripción del Proyecto**</b>	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.
<b>**Actores**</b>	Administrador de Sistema, Personal de TI
<b>**Precondiciones**</b>	El sistema está operativo y el administrador ha iniciado sesión con credenciales de nivel superusuario.
<b>**Flujo Normal (Acción del Actor)**</b>	1. El actor selecciona la opción "Administración del Sistema".   2. El actor elige una tarea de administración (ej. "Gestión de Usuarios", "Mantenimiento del Sistema", "Configuración de Backup", "Revisar Logs").   3. Si es gestión de usuarios: El actor crea, modifica o elimina un usuario, o asigna/revoca roles y permisos.   4. Si es mantenimiento: El actor inicia un proceso de optimización o reinicio de un servicio.   5. Si es backup: El actor configura la frecuencia y destino de las copias de seguridad.   6. Si es logs: El actor filtra y visualiza los registros del sistema.   7. El actor confirma la operación.
<b>**Flujo Normal (Respuesta del Sistema)**</b>	1. El sistema muestra el panel de administración con las opciones disponibles.   2. El sistema presenta la interfaz correspondiente a la tarea seleccionada.   3. Si es gestión de usuarios: El sistema valida los datos y actualiza la base de datos de usuarios/permisos.   4. Si es mantenimiento: El sistema ejecuta la rutina solicitada y muestra el estado.   5. Si es backup: El sistema guarda la configuración del backup y programa las tareas.   6. Si es logs: El sistema recupera y presenta los registros según los filtros.   7. El sistema confirma la finalización exitosa de la operación.
<b>**Flujo Alternativo**</b>	\- <b>**Permisos Insuficientes**</b> : El sistema deniega la acción y notifica al actor que no tiene los permisos necesarios. \  - <b>**Datos Inválidos**</b> : El sistema rechaza la entrada del actor y solicita corrección (ej. formato de correo inválido para un usuario). \  - <b>**Fallo de Operación**</b> : El sistema reporta un error en la ejecución de una tarea (ej. fallo al crear un usuario, error en la copia de seguridad) y registra el incidente en los logs. \  - <b>**Recurso No Disponible**</b> : El sistema informa que un recurso necesario para la operación (ej. espacio de almacenamiento para backup) no está disponible.
<b>**Postcondiciones**</b>	Los ajustes de administración se han aplicado, la tarea seleccionada se ha completado, o se ha generado una notificación de error con el registro correspondiente. El sistema mantiene su integridad y seguridad.
<b>**Frecuencia**</b>	Esporádica, según las necesidades de gestión, mantenimiento o resolución de problemas (ej. diario para revisión de logs, semanal para backups, ocasional para gestión de usuarios).
<b>**Tiempo de Respuesta**</b>	Menos de 3 segundos para operaciones de gestión de usuarios o visualización de logs. Hasta 30 segundos para tareas de mantenimiento o configuración de backups que impliquen procesos internos complejos.

## Administración del Sistema



Característica	Caso de Uso UC001: Configurar Sensores de Calidad del Aire	Caso de Uso UC002: Visualizar Datos de Calidad del Aire	Caso de Uso UC003: Recibir Alertas de Calidad del Aire	Caso de Uso UC004: Generar Reportes de Calidad del Aire
<b>**Nombre**</b>	Configurar Sensores de Calidad del Aire	Visualizar Datos de Calidad del Aire	Recibir Alertas de Calidad del Aire	Generar Reportes de Calidad del Aire
<b>**Descripción**</b>	Permite al administrador o técnico configurar los parámetros de los sensores de calidad del aire instalados en los invernaderos.	Permite a los usuarios visualizar en tiempo real y datos históricos de los niveles de gases (CO2, O2, humedad, temperatura) dentro de los invernaderos.	El sistema notifica a los usuarios designados cuando los niveles de calidad del aire exceden los umbrales predefinidos.	Permite a los usuarios generar informes detallados sobre la calidad del aire en un periodo de tiempo específico para análisis y cumplimiento.
<b>**Caso de Uso**</b>	UC001: Configuración de Sensores	UC002: Visualización de Datos	UC003: Notificación de Alertas	UC004: Generación de Reportes
<b>**Autor**</b>	Equipo de Desarrollo	Equipo de Desarrollo	Equipo de Desarrollo	Equipo de Desarrollo
<b>**Fecha**</b>	28/08/2025	28/08/2025	28/08/2025	28/08/2025
<b>**Descripción del Proyecto**</b>	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.	Sistema automatizado para monitorear y gestionar la calidad del aire en invernaderos.
<b>**Actores**</b>	Administrador de Sistema, Técnico de Invernadero	Gerente de Invernadero, Agrónomo, Técnico de Invernadero	Gerente de Invernadero, Agrónomo, Técnico de Invernadero	Gerente de Invernadero, Agrónomo
<b>**Precondiciones**</b>	El sistema está operativo y el usuario ha iniciado sesión con los permisos adecuados.	El sistema está recibiendo datos de los sensores y el usuario ha iniciado sesión.	Los umbrales de alerta están configurados y los sensores están activos.	Existen datos históricos suficientes en el sistema.

<p><b>**Flujo Normal (Acción del Actor)**</b></p>	<p>1. El actor selecciona la opción "Configurar Sensores". &lt;br&gt; 2. El actor elige un sensor específico o un tipo de sensor. &lt;br&gt; 3. El actor introduce los parámetros deseados (ej. tipo de gas, frecuencia de muestreo, umbrales). &lt;br&gt; 4. El actor confirma la configuración.</p>	<p>1. El actor selecciona la opción "Ver Datos de Calidad del Aire". &lt;br&gt; 2. El actor elige el invernadero y el período de tiempo a visualizar. &lt;br&gt; 3. El actor selecciona los parámetros específicos (ej. CO2, humedad) que desea ver.</p>	<p>1. El sistema detecta que un parámetro de calidad del aire excede un umbral. &lt;br&gt; 2. El sistema genera una alerta. &lt;br&gt; 3. El actor recibe la notificación. &lt;br&gt; 4. (Opcional) El actor accede a la alerta para ver detalles.</p>	<p>1. El actor selecciona la opción "Generar Reporte". &lt;br&gt; 2. El actor especifica el tipo de reporte, el período y los parámetros. &lt;br&gt; 3. El actor selecciona el formato de salida (PDF, CSV). &lt;br&gt; 4. El actor inicia la generación del reporte.</p>
<p><b>**Flujo Normal (Respuesta del Sistema)**</b></p>	<p>1. El sistema muestra la lista de sensores disponibles. &lt;br&gt; 2. El sistema muestra los parámetros actuales del sensor seleccionado. &lt;br&gt; 3. El sistema valida los datos introducidos. &lt;br&gt; 4. El sistema guarda la nueva configuración y aplica los cambios al sensor.</p>	<p>1. El sistema muestra un panel de control con los datos en tiempo real. &lt;br&gt; 2. El sistema carga y muestra los datos históricos en gráficos o tablas. &lt;br&gt; 3. El sistema permite filtrar y organizar la información.</p>	<p>1. El sistema verifica los datos de los sensores continuamente. &lt;br&gt; 2. El sistema envía una alerta por el canal configurado (correo, SMS, notificación push). &lt;br&gt; 3. El sistema registra la alerta en el historial. &lt;br&gt; 4. El sistema muestra los detalles de la alerta al actor.</p>	<p>1. El sistema muestra las opciones de reporte disponibles. &lt;br&gt; 2. El sistema valida los parámetros de entrada. &lt;br&gt; 3. El sistema compila los datos y genera el reporte. &lt;br&gt; 4. El sistema ofrece el reporte para descarga o lo envía por correo.</p>
<p><b>**Flujo Alternativo**</b></p>	<p>\- <b>**Datos Inválidos**</b>: El sistema rechaza la configuración y solicita al actor corregir los errores. &lt;br&gt; - <b>**Error de Comunicación con Sensor**</b>: El sistema notifica al actor que no se pudo aplicar la configuración al sensor.</p>	<p>\- <b>**No hay Datos Disponibles**</b>: El sistema muestra un mensaje indicando la ausencia de datos para el período seleccionado. &lt;br&gt; - <b>**Fallo en la Carga de Datos**</b>: El sistema notifica un error y sugiere reintentar o contactar soporte.</p>	<p>\- <b>**Fallo en Envío de Notificación**</b>: El sistema reintenta el envío y registra el fallo en el log. &lt;br&gt; - <b>**Sensor Fuera de Línea**</b>: El sistema genera una alerta sobre la indisponibilidad del sensor en lugar de una alerta de umbral.</p>	<p>\- <b>**Datos Insuficientes**</b>: El sistema notifica al actor que no hay datos suficientes para el período o tipo de reporte solicitado. &lt;br&gt; - <b>**Error en Formato de Salida**</b>: El sistema informa un error y sugiere</p>

### 5.3 Descripción de Casos de Uso:

CU01 – Registrar Usuario

Actor: Administrador

Descripción: Permite crear un nuevo usuario con datos de acceso y permisos.

Entradas: Nombre, correo, contraseña, rol.

Salidas: Confirmación de registro.

Relaciones: Extiende a "Gestionar Usuarios".

## 6. COMPONENTES DEL SISTEMA

El diagrama de componentes describe la arquitectura física del sistema, mostrando cómo se organizan y relacionan los distintos módulos de software y hardware. Representa la estructura estática del sistema y la interacción entre sus partes, facilitando la comprensión de su implementación.

### 6.1 Componentes Principales:

**Interfaz de Usuario (Frontend):** Encargada de la interacción directa con el usuario mediante pantallas y formularios.

**Lógica de Negocio (Backend):** Procesa la información y aplica las reglas del sistema.

**Base de Datos:** Gestiona el almacenamiento y recuperación de la información.

### 6.2 Relaciones entre Componentes:

La Interfaz de Usuario se comunica con la Lógica de Negocio mediante servicios o APIs.

La Lógica de Negocio consulta y actualiza la Base de Datos.

Los módulos externos (sensores, sistemas de terceros) se conectan al backend.

### **6.3 Beneficios del Diagrama de Componentes:**

Facilita la comprensión de la arquitectura general del sistema.

Permite identificar dependencias y puntos de integración.

Sirve como referencia para la implementación y el mantenimiento.

## **7. ARQUITECTURA DEL SISTEMA**

### **7.1 Distribución de servicios:**

El sistema se organiza bajo una arquitectura modular distribuida, en la cual cada servicio cumple una función específica y se comunica de manera eficiente con los demás componentes. Los sensores IoT recolectan los datos ambientales (temperatura, humedad y CO<sub>2</sub>) y los envían al backend, alojado en un servidor de aplicaciones. Dicho backend procesa la información, la registra en la base de datos MySQL y, de ser necesario, activa el servicio de notificaciones mediante una API externa. Por su parte, la interfaz de usuario (frontend) accede a los datos a través de servicios web seguros, ofreciendo al usuario reportes, gráficos y alertas en tiempo real.

### **7.2 Flujo de interacción entre componentes:**

El flujo inicia con la captura de datos por parte de los sensores distribuidos en el invernadero. Estos envían la información al backend, el cual valida y almacena los registros en la base de datos. El frontend consulta al backend mediante solicitudes HTTP/HTTPS

para presentar los resultados al usuario en forma de dashboards interactivos. Cuando se detectan valores fuera de rango, el motor de notificaciones genera alertas que llegan directamente al usuario final por la interfaz o por canales externos.

### **7.3 Justificación de la arquitectura adoptada**

La arquitectura distribuida se seleccionó por su escalabilidad, flexibilidad y facilidad de mantenimiento. Este enfoque modular permite agregar nuevos sensores o servicios sin afectar la operatividad del sistema. Asimismo, el uso de microservicios ligeros garantiza un manejo eficiente de los recursos y facilita la integración con tecnologías futuras, asegurando un sistema robusto y adaptable a diferentes entornos agrícolas.

## **8. Descripción de la Implementación**

### **8.1 Tecnologías utilizadas:**

- **Sensores IoT:** dispositivos de medición de temperatura, humedad y CO<sub>2</sub>, con conectividad inalámbrica.
- **Servidor de aplicaciones:** alojado en un entorno compatible con Django.
- **Base de datos:** MySQL para el almacenamiento de datos históricos y en tiempo real.
- **API de notificaciones:** integración con servicios externos para envío de alertas inmediatas.
- **Frontend web:** diseñado con HTML5, CSS3 y Bootstrap para mejorar la experiencia de usuario.



## 8.2 Lenguajes y frameworks:

- **Python:** lenguaje base para el backend por su simplicidad, potencia y extensa comunidad de soporte.
- **Django:** framework web que facilita la organización modular y el manejo de datos.
- **JavaScript:** utilizado en la capa cliente para dotar de dinamismo e interactividad a la interfaz.
- **Bootstrap:** framework CSS que proporciona un diseño responsivo y estandarizado.
- **SQL:** lenguaje de consultas para la gestión y manipulación de la base de datos MySQL.

## 9. Conclusiones

### 9.1 Beneficios del sistema:

El sistema contribuye a optimizar la gestión de invernaderos mediante el monitoreo constante de variables ambientales críticas. Su implementación facilita la detección temprana de riesgos, mejora la toma de decisiones y promueve un uso eficiente de los recursos agrícolas. Además, al ser modular, permite escalar y adaptarse a diferentes necesidades productivas.

### 9.2 Limitaciones actuales:

Entre las limitaciones identificadas se encuentran la dependencia de la conectividad a internet para la transmisión de datos, posibles costos adicionales asociados a la adquisición de sensores IoT y la necesidad de personal capacitado para la correcta instalación y mantenimiento del sistema.

### **9.3 Recomendaciones futuras:**

Se recomienda integrar técnicas de inteligencia artificial y aprendizaje automático para predecir condiciones críticas y generar recomendaciones automáticas. Asimismo, se sugiere ampliar la compatibilidad con otros tipos de sensores (luminosidad, pH del suelo) y explorar la implementación de aplicaciones móviles que permitan una gestión más versátil y accesible desde cualquier lugar.