

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Plan de proyecto

Monitoreo y control de un invernadero de hortalizas

Autor(es): José Escalante

Felipe Flores Valencia

Fabian Astorga Castillo

Cristofer Pinto Maita

Asignatura: Proyecto II

Profesor: Diego Aracena Pizarro

ARICA, 04 de Noviembre del 2024

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
04/09/2024	1.0	Versión preliminar del formato	Jose Escalante Felipe Flores Fabian Astorga Cristofer Pinto
10/09/2024	1.1	Revisión y modificación del plan	Jose Escalante Felipe Flores Fabian Astorga
24/09/2024	1.2	Ajustes y refinamiento del informe	Jose Escalante Felipe Flores Cristofer Pinto
04/11/2024	1.4	Versión final del informe 2	Jose Escalante Felipe Flores Fabian Astorga Cristofer Pinto

1. Panorama General.....	4
1.1. Introducción.....	4
1.2. Resumen del Proyecto.....	4
1.2.1. Propósito.....	4
1.2.2. Alcance.....	4
1.2.3. Restricciones.....	4
1.2.4. Objetivos.....	5
1.2.5. Suposiciones y restricciones.....	6
1.2.6. Entregables del Proyecto.....	6
2. Organización del proyecto:.....	7
2.1. Personal.....	7
2.2. Roles y responsabilidades.....	7
2.3. Mecanismos de comunicación:.....	7
3. Planificación de los procesos del proyecto:.....	8
3.1. Planificación inicial del proyecto.....	8
3.1.1. Estimaciones.....	8
3.1.2. Planificación de Recursos Humanos.....	9
3.2. Actividades de trabajo (Carta Gantt).....	10
3.3. Planificación de Riesgos.....	11
4. Planificación de procesos técnicos:.....	12
4.1. Modelos de procesos.....	12
4.1.1. Requerimientos establecido.....	12
4.1.2. Casos de uso.....	15
4.1.3. Diagramas de secuencia.....	22
4.1.4. Boceto Interfaz de usuario.....	27
4.2. Herramientas y técnicas.....	29
5. Implementación.....	32
5.1. Plan de integración.....	32
5.2. Modelo de implementación.....	32
5.3.1. Módulo de Monitoreo de Sensores.....	33
5.3.2. Módulo de Control Automático.....	34
5.3.3. Módulo de Notificaciones.....	37
5.4. Reporte de evaluación.....	40

1. Panorama General

1.1. Introducción

El informe aborda las bases para el proyecto semestral de monitoreo y control de un invernadero de hortalizas mediante el uso de sensores. Se pasará por las etapas del proyecto, desde la concepción de la idea, la problemática, la solución, los costes, restricciones y suposiciones para finalmente llegar a la confección del artefacto y utilización.

1.2. Resumen del Proyecto

1.2.1. Propósito

El proyecto permitirá, mediante el uso de sensores, monitorear las distintas condiciones dentro de un invernadero necesarias para el cuidado de hortalizas, siendo estas la temperatura, la luminosidad, la humedad en el ambiente y el suelo como también activar los sistema de calefacción, ventilación, iluminación y riego por goteo.

1.2.2. Alcance

Sujetos que posean un invernadero o deseen poseer un invernadero de hortalizas y no posean experiencia o desconfíen de las habilidades humanas para el cuidado de uno

1.2.3. Restricciones

Las restricciones del proyecto constan del uso de hardware y software correspondiente a raspberryPi 4, sensores de humedad, luminosidad, ph y temperatura

También una restricción a destacar es la estructura del invernadero donde se instalarán los dispositivos debe ser estudiada previamente con cautela para poder optimizar el espacio de monitoreo de los sensores, para así lograr optimizar el rango de alcance que tienen estos, es decir, es de suma importancia el lugar donde se coloquen los sensores para poder dar una lectura real y fiel del ambiente y no valores erróneos

1.2.4. Objetivos

Objetivo general

Desarrollar un sistema de monitorización y control automatizado que optimice el cuidado y crecimiento de hortalizas en invernaderos, garantizando condiciones ambientales adecuadas mediante la integración de sensores y actuadores.

Objetivos específicos

- Diseñar la solución para la automatización del monitoreo y control de condiciones en un invernadero mediante el análisis de las necesidades del sistema.
- Planificar y construir una maqueta física que represente el escenario de un invernadero, utilizando materiales reciclables para un enfoque sostenible, y establecer el esquema general del sistema de monitoreo y control.
- Desarrollar el sistema de monitoreo y control del invernadero, integrando los sensores y actuadores
- Realizar pruebas para el perfecto funcionamiento del sistema.
- Documentar el desarrollo, resultados y conclusiones del proyecto realizado.

1.2.5. Suposiciones y restricciones

Suposiciones:

- Se debe suponer que hay un sistema de riego ya instalado, una infraestructura de invernadero tradicional (suelo de tierra, paredes y techo de lona y un sistema de ventiladores). Esto debido a que el sistema al hacer de monitoreo y control, utiliza el riego y el sistema de ventilación de manera automática y en el momento necesario según los sensores instalados en el invernadero
- Los integrantes del equipo tendrán el conocimiento necesario para llevar a cabo la realización del proyecto

Restricciones:

- Realizar el proyecto en el tiempo establecido
- Realizar el proyecto con los sensores y recursos proveídos por el departamento. En caso de comprarlos, estos no deben superar el límite propuesto.

1.2.6. Entregables del Proyecto

A medida que se realice el proyecto se entregarán diferentes documentos con el fin de actualizar la información y el avance que se tiene hasta el momento, estos son:

- Informes
- Bitacoras
- Presentaciones
- Maqueta de la solución

2. Organización del proyecto:

2.1. Personal

- Jefe de proyecto: Encargado de gestionar, designar tareas y roles, monitorear y controlar la actividad del grupo de proyecto
- Documentador: Encargado de documentar la actividad del grupo y los avances en el proyecto
- Diseñador de Hardware: Encargado de diseñar y confeccionar los artefactos de hardware necesarios para el proyecto
- Programador: Encargado de diseñar, confeccionar y implementar software necesario para el hardware utilizado
- Tester: Encargado de revisar los artefactos de software, hardware y probar los límites del sistema creado(Fuera del ambiente previsto y dentro del ambiente esperado)

2.2. Roles y responsabilidades

Fabian Astorga Castillo: Diseñador de hardware, programador

Felipe Flores Valencia: Tester, Programador.

Cristofer Pinto Maita: Diseñador, Programador

José Escalante Aduvire: Documentador, programador

2.3. Mecanismos de comunicación:

- Discord
- WhatsApp
- Google drive
- Correo electrónico de Google

3. Planificación de los procesos del proyecto:

3.1. Planificación inicial del proyecto

3.1.1. Estimaciones

Tabla 1: Estimaciones

Herramienta	Cantidad	Costo
Sensor Humedad del suelo	1	\$2000
Sensor Temperatura	1	\$5000
Sensor humedad ambiental	1	\$4200
Sensor lumínico	1	\$1190
Raspberry Pi	1	\$60.000
Notebook	4	\$450.000
Luces led	4	\$3000
Cableado Necesario	4	\$3000
Lampara led UV	1	\$4.500
Ventilador	2	\$7000
Smartphone	1	\$120.000
Total Implementos	21	\$684,890

3.1.2. Planificación de Recursos Humanos

Tabla 2: Planificación de recursos humanos

Rol	Cantidad	Costo
Jefe de grupo	1	\$1.500.000
Documentador	1	\$350.000
Programador	2	\$1.000.000
Diseñador de Hardware	3	\$1.000.000
Tester	1	\$800.000
Total Roles	8	\$7.650.000

3.2. Actividades de trabajo (Carta Gantt)



3.3. Planificación de Riesgos

Tabla 3: Planificación de riesgos

Riesgos	Probabilidad de ocurrencia	Nivel de impacto	Acción Remedial
El equipo deja de funcionar.	30%	2	Tratar de conseguir un equipo nuevo a la brevedad
Se necesitan más equipos de lo que se tenía planeado.	30%	2	Realizar petición formal para solicitar más equipos
Problemas de conectividad.	30%	2	Realizar una petición formal para agendar un técnico y que solucione el problema
Una planta requiere más tiempo para germinar.	60%	4	Extender el uso del sistema de monitoreo y control
Cambios en los requerimientos.	50%	2	Realizar un análisis de impacto de los cambios para decidir si es óptimo aplicarlos.
Ausencia prolongada de un miembro del equipo de trabajo.	30%	3	Reasignar tareas de manera temporal para cumplir con el avance del proyecto
Falta de capacitación y conocimientos técnicos.	40%	3	Realizar capacitaciones, otorgar guía para el manejo de componentes específicos

4. Planificación de procesos técnicos:

4.1. Modelos de procesos

4.1.1. Requerimientos establecido

Requerimientos funcionales y **requerimientos no funcionales** son dos tipos de especificaciones que se utilizan en el desarrollo de software para describir cómo debe comportarse un sistema y las condiciones bajo las cuales debe operar. A continuación se describe cada uno:

Requerimientos funcionales:

1. Determinación de datos sensoriales:

Capacidad de leer y procesar datos de los sensores conectados, incluyendo:

- Sensor de humedad del suelo.
- Sensor de humedad ambiental.
- Sensor de temperatura.
- Sensor lumínico.

2. Activación de focos LED:

Activación de los focos LED en función de los datos obtenidos del sensor lumínico. Por ejemplo, cuando la intensidad de luz esté por debajo de un rango óptimo, el sistema debe encender los LED para suplir la falta de luz.

3. Envío de notificaciones al usuario:

Capacidad de enviar notificaciones al agricultor cuando se detecten condiciones críticas o anormales (por ejemplo, humedad del suelo baja, temperatura elevada, etc.), permitiéndole tomar acciones adicionales si es necesario.

4. Activación del riego automatizado:

Basándose en los datos del sensor de humedad del suelo, el sistema debe activar automáticamente el riego cuando el nivel de humedad esté por debajo de un rango óptimo predeterminado, asegurando así que las plantas reciban la cantidad adecuada de agua.

5. Control de la temperatura dentro del invernadero:

El sistema debe monitorear y controlar la temperatura del invernadero mediante el sensor de temperatura, activando mecanismos de ventilación o calefacción para mantener la temperatura dentro de los rangos óptimos.

6. Ajuste de parámetros de monitoreo:

Debe permitir al agricultor ajustar los parámetros de monitoreo, tales como los rangos óptimos de luz, humedad y temperatura, para que el control y las notificaciones se adapten a las necesidades específicas del cultivo.

Requerimientos no funcionales

1. Escalabilidad:

La solución debe permitir la integración de los distintos sensores y actuadores.

2. Usabilidad:

La interfaz de usuario debe ser intuitiva y fácil de utilizar, para que el agricultor pueda visualizar los datos, recibir notificaciones y ajustar parámetros sin necesidad de conocimientos técnicos avanzados.

3. Confiabilidad:

La recolección y procesamiento de datos de los sensores debe ser confiable, minimizando errores de lectura o fallas en el funcionamiento, y asegurando que las notificaciones se envíen de manera precisa y oportuna.

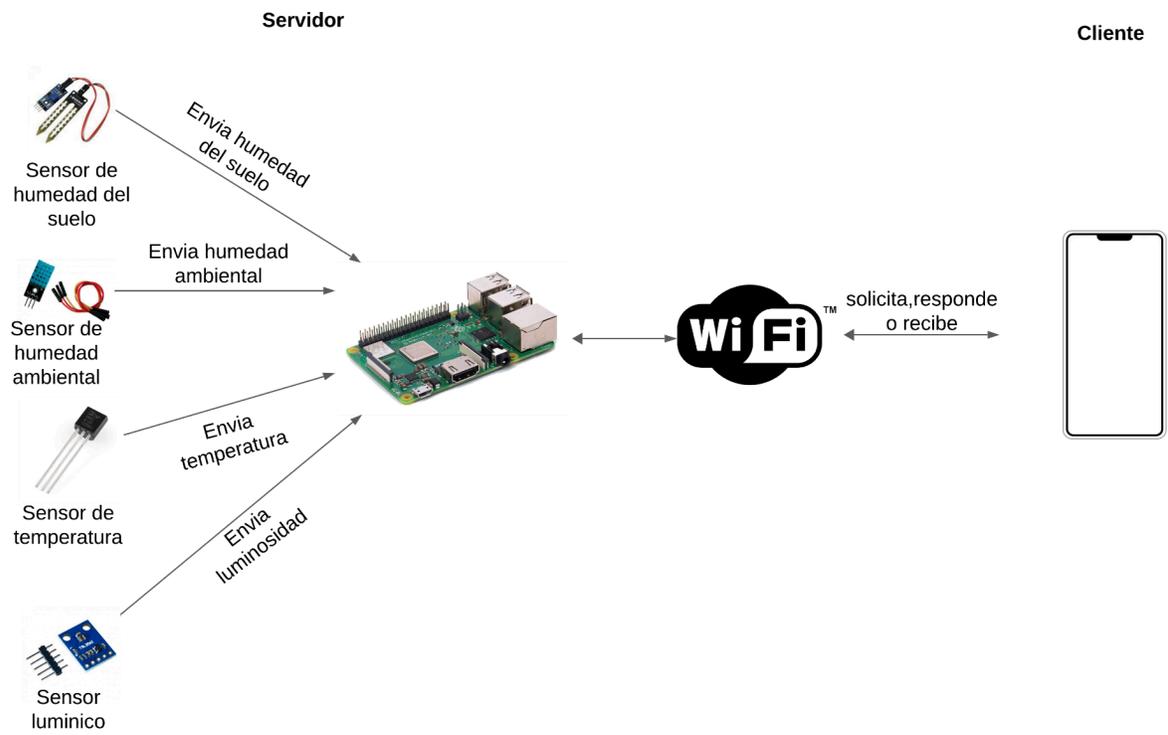
4. Rendimiento:

El procesamiento y actualización de los datos de los sensores deben ser en tiempo real para garantizar una respuesta rápida ante cambios en las condiciones ambientales.

5. Disponibilidad:

El acceso y control remoto deben estar disponibles en todo momento para el agricultor.

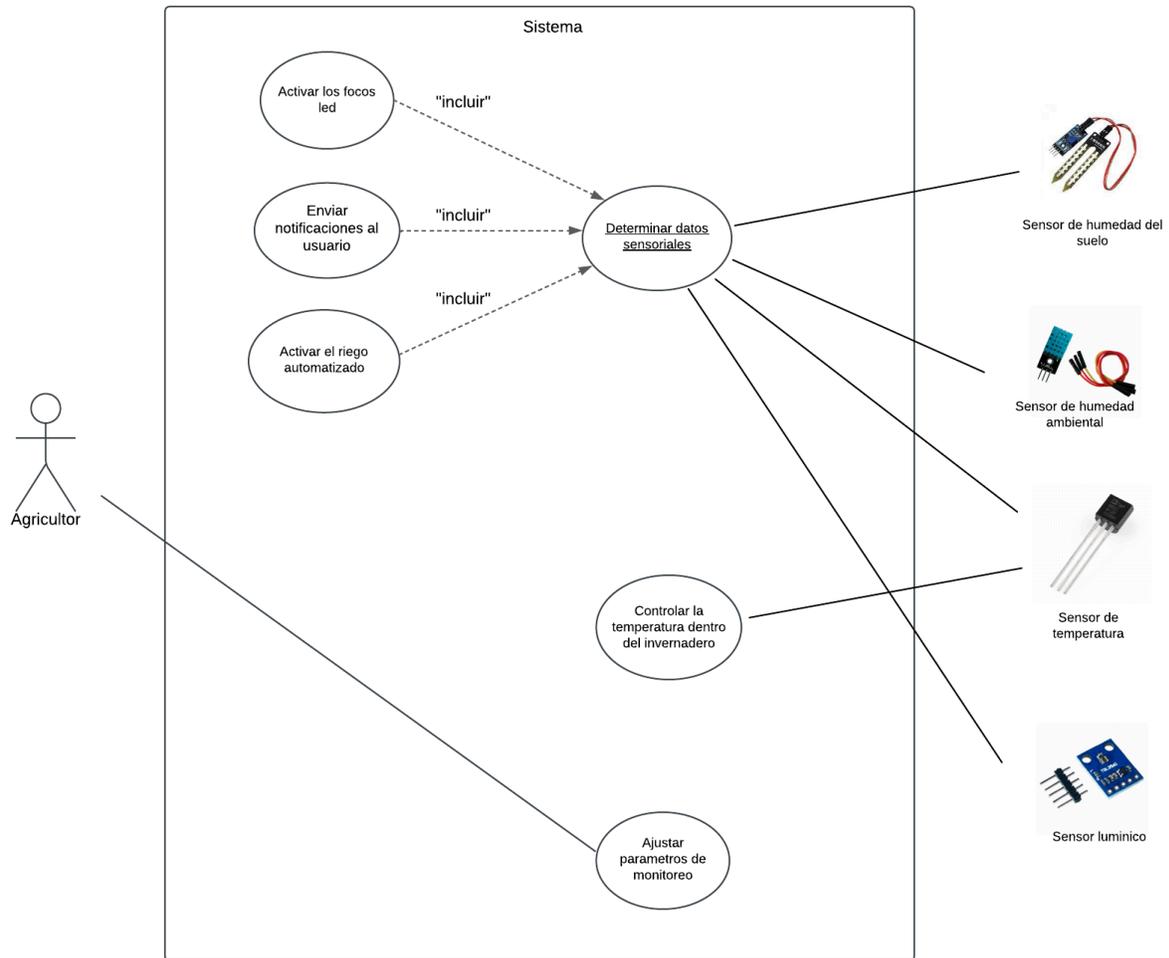
Descripción de arquitectura



4.1.2. Casos de uso

Diagrama de casos de uso base

Figura 1: Casos de uso base



Descripción de los casos de uso:

Nombre CUS: Determinar datos sensoriales	
Descripción: El sistema determina las condiciones ambientales actuales en el invernadero	
Actor: Sensor de humedad del suelo, Sensor de humedad ambiental, Sensor de temperatura, Sensor lumínico	
Precondición: Los rangos óptimos deben estar registrados en el sistema.	
Flujo Principal: Sensores 2.- Los sensores envían las medidas actuales: <ul style="list-style-type: none">● Temperatura● Humedad ambiente● Humedad del suelo● Nivel de luz	Flujo Principal: Sistema 1. El sistema le pide los sensores enviar las mediciones de las condiciones actuales
Postcondiciones: Las medidas entregadas por los sensores quedan registradas en el sistema	

Nombre CUS: Ajustar parámetros de monitoreo	
Descripción: El agricultor ajusta los parámetros de humedad, temperatura y luz en el sistema según las necesidades de las hortalizas.	
Actor: Agricultor	
Precondición:	
Flujo Principal: Agricultor 1. El agricultor ajusta los niveles de referencia para cada sensor (humedad ambiente, humedad del suelo, temperatura, luz)	Flujo Principal: Sistema 2. El sistema guarda los cambios y ajusta el monitoreo en base a los nuevos valores.
Postcondiciones: Los nuevos parámetros quedan registrados en el sistema para monitorear las condiciones del invernadero.	

Nombre CUS: Activar el riego automatizado	
Descripción: El sensor de humedad del suelo gatilla la activación del riego automatizado	
Actor: Sensor de humedad del suelo	
Precondición: Los rangos óptimos deben estar registrados en el sistema.	
Flujo Principal: 1.<<Include>> C.U.S Determinar datos sensoriales	Flujo Principal: Sistema 2. Si los datos sensoriales del sensor de humedad del suelo es menor al rango óptimo se activa el riego automatizado.
Postcondiciones:	

Nombre CUS: Activar focos led	
Descripción:El sistema activa las luces led por la falta de luz de una planta	
Actor: Luces led	
Precondición: Luces led previamente instaladas.	
Flujo Principal: Actor 1. 1. <<Include>> C.U.S Determinar datos sensoriales	Flujo Principal: Sistema 2-El sistema activa las luces durante un periodo de tiempo
Postcondiciones:	

Nombre CUS: Enviar notificaciones al usuario	
Descripción: El sistema envía notificaciones al usuario cuando algún dato sensorial está fuera del rango óptimo.	
Actor: Sensor de humedad del suelo, Sensor de humedad ambiental, Sensor de temperatura, Sensor lumínico	
Precondición: El sistema de notificaciones está habilitado. Los rangos óptimos deben estar registrados en el sistema.	
<p>Flujo Principal: Temperatura</p> <p>1. <<Include>> C.U.S Determinar datos sensoriales</p>	<p>Flujo Principal: Sistema</p> <p>2. Verifica si los datos están dentro de los rangos óptimos registrados</p> <p>3. Envía una notificación al agricultor que incluye:</p> <ul style="list-style-type: none"> ● Dato sensorial que esté fuera del rango óptimo ● hora ● fecha
Postcondiciones: Datos de la notificación registrados en el sistema	

Nombre CUS: Controlar la temperatura dentro del invernadero	
Descripción: El sistema mediante al sensor de temperatura va a cambiar la temperatura dependiendo de la necesidad del invernadero	
Actor: Sensor temperatura	
Precondición: El sensor de temperatura, ventiladores, calefacción instalados y parámetros de la planta registrados en el sistema	
<p>Flujo Principal: Temperatura bajo el umbral establecido</p> <p>1.- El Sensor mide la temperatura actual en el invernadero</p>	<p>Flujo Principal: Sistema</p> <p>2- El sistema recibe los datos proporcionados por el sensor de temperatura</p> <p>3- El Sistema Activa la calefacción del invernadero</p>
<p>Flujo alternativo: Temperatura mayor al umbral establecido</p>	<p>3.1- El Sistema Activa los ventiladores del invernadero (Disminuye la temperatura del invernadero)</p>
Postcondiciones: Se cambia la temperatura del invernadero	

4.1.3. Diagramas de secuencia

Figura 2: Activar el riego automatizado

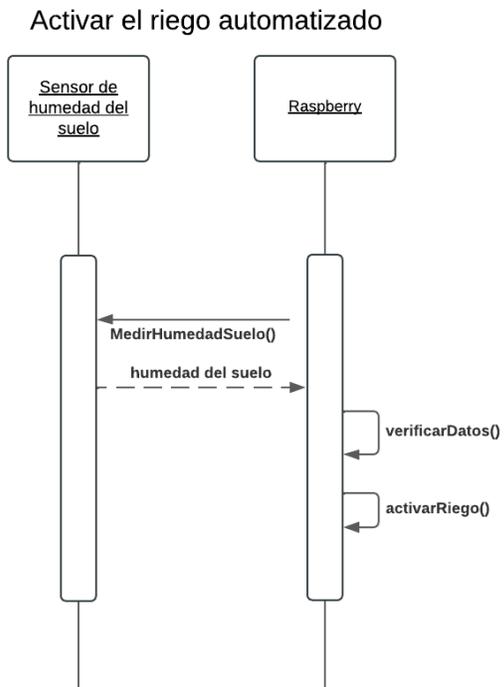


Figura 3: Enviar notificaciones al usuario

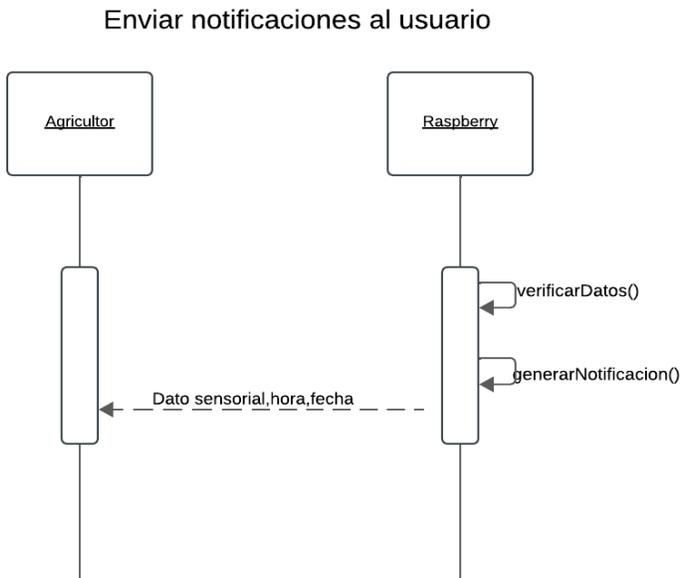


Figura 4: Ajustar parámetros de monitoreo

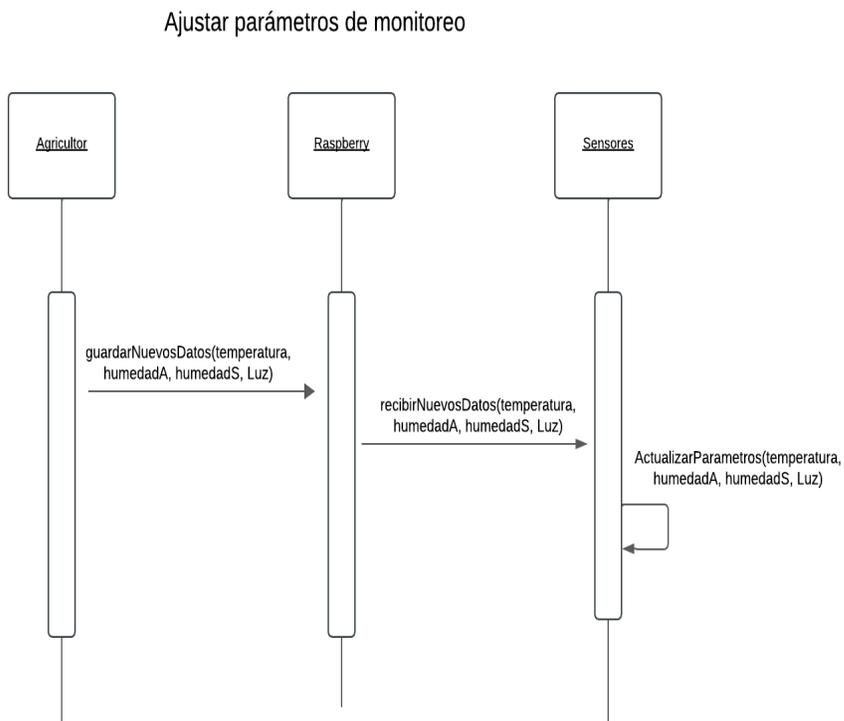


Figura 5: Determinar datos sensoriales

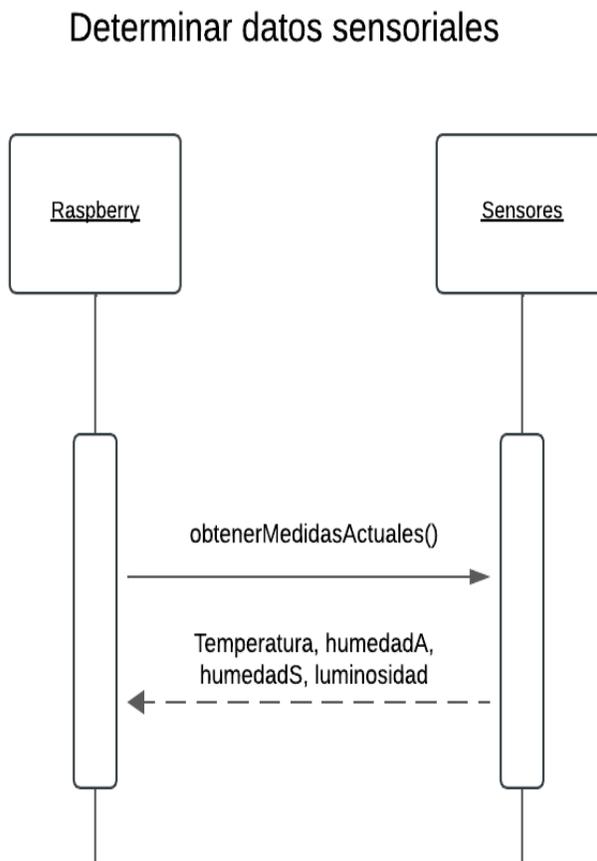


Figura 6: Prender luces led

Prender luces Led

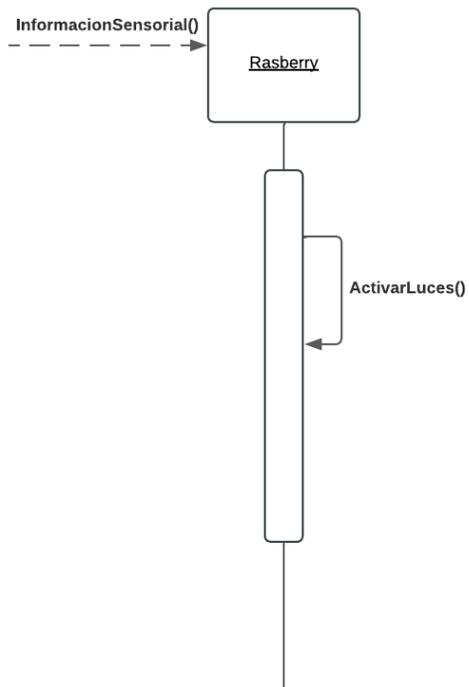
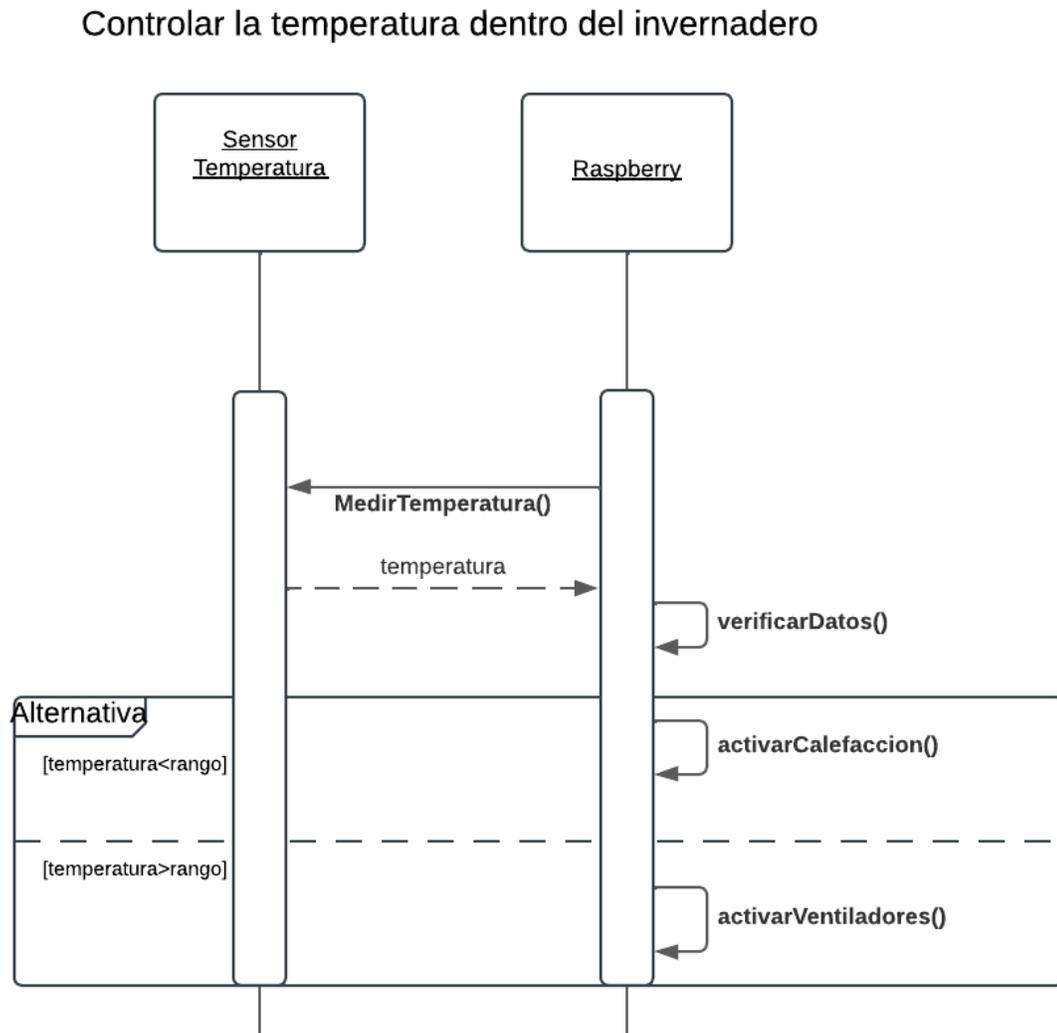


Figura 7: Controlar la temperatura dentro del invernadero



4.1.4. Boceto Interfaz de usuario

Se eligió un diseño minimalista, con un número reducido de botones y menús, para facilitar la navegación. Esta simplicidad no solo mejora la experiencia del usuario, sino que también permite un acceso más rápido y directo a las funcionalidades principales del sistema.

Figura 8: Pantalla principal

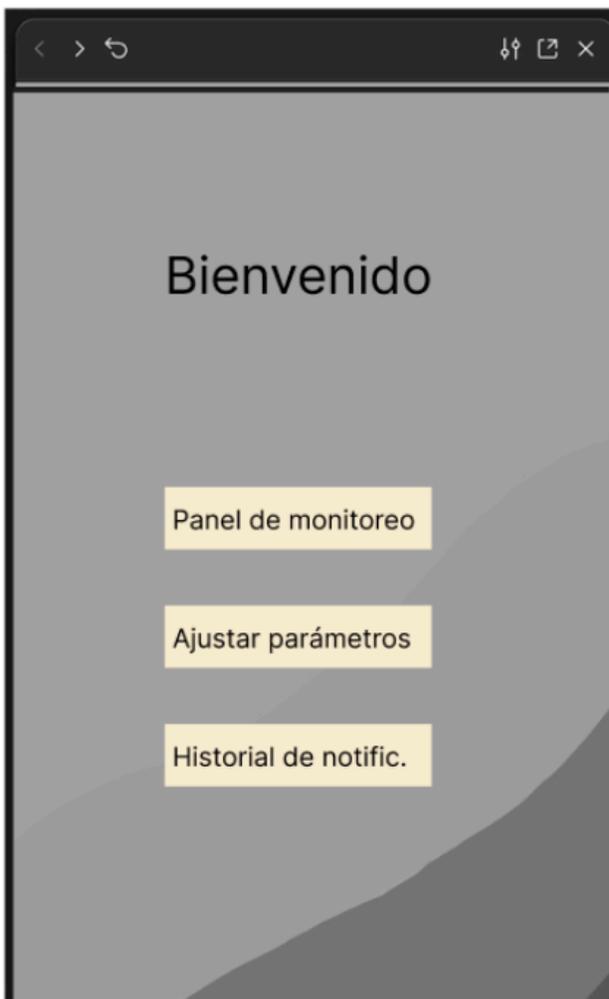


Figura 9: Panel de monitoreo

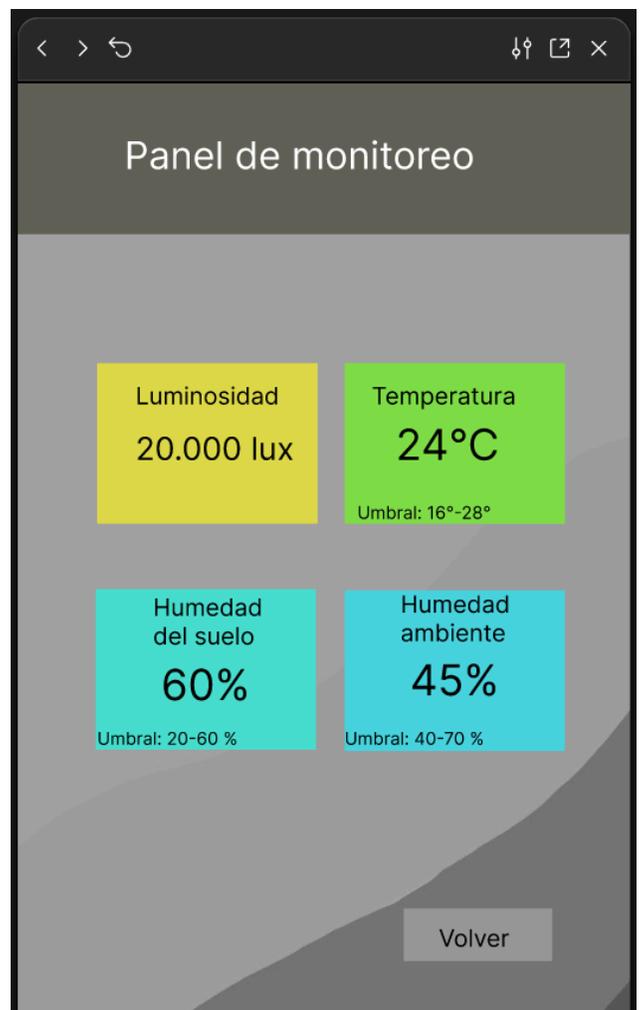


Figura 10: Historial de notificaciones

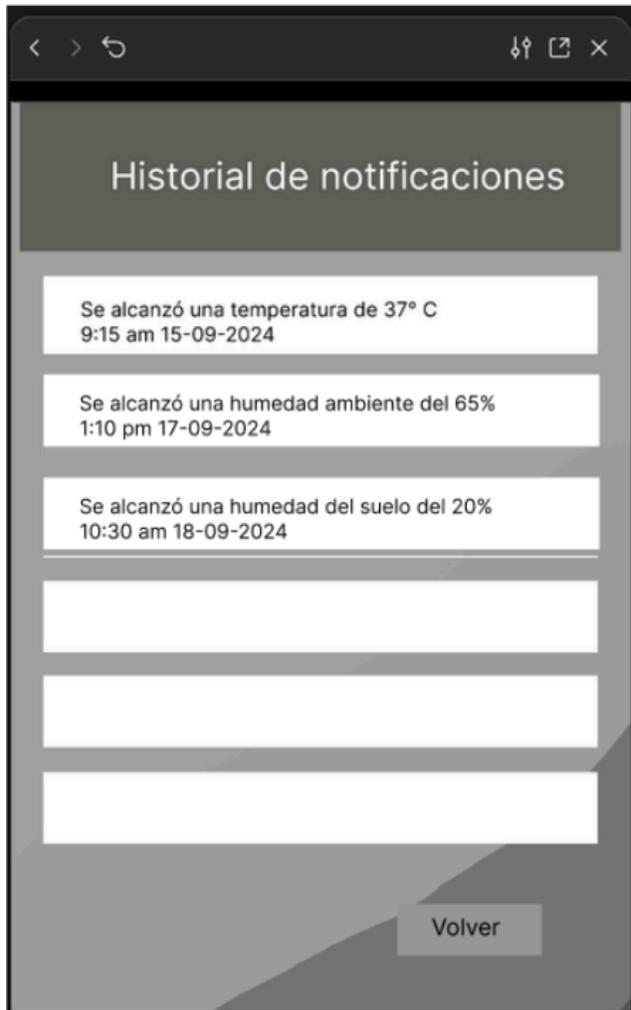
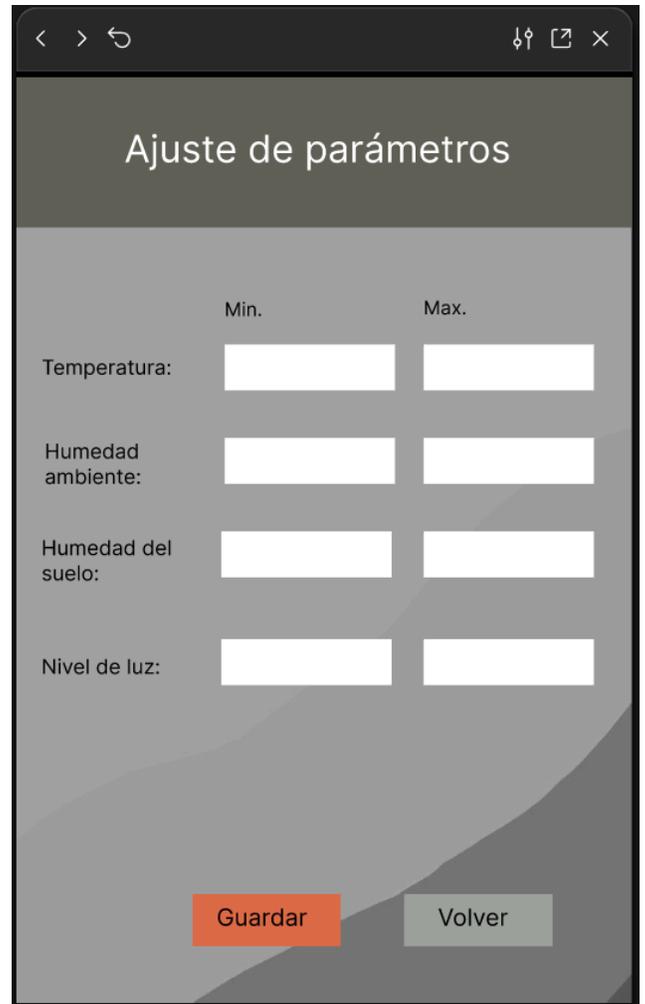


Figura 11: Ajuste de parámetros



4.2. Herramientas y técnicas

Para poder programar en el Raspberry Pi, la lógica y tratamiento de datos utilizaremos:

- Python

- Visual Studio Code/Thonny

-Telegram

Raspberry Pi : La Raspberry Pi es una computadora de bajo costo y con un tamaño compacto, del porte de una tarjeta de crédito, puede ser conectada a un monitor de computador o un TV, y usarse con un mouse y teclado estándar. Es un pequeño computador que es un sistema operativo linux .

En esta nueva versión incorpora una CPU Cortex-A53 de cuatro núcleos, 64bit y 1.2Ghz, el cual mejora su velocidad de respuesta en un 50% en comparación a su versión anterior (RPi v2 modelo B) y diez veces más rápida que la primera Raspberry Pi.

La Raspberry Pi 3 es compatible con todos los accesorios para la Raspberry Pi B+ y V2 modelo B, ya que mantiene el mismo conector GPIO para la conexión de sensores, periféricos y otros.

El procesador no es lo único que se ha mejorado en la nueva Raspberry Pi 3, ya que ahora también incluye conexión Wi-Fi de 802.11n y Bluetooth 4.1, con lo cual ya no dependerás de una conexión tipo Ethernet para acceder a Internet o gracias a la conexión Bluetooth conectar un teclado o mouse inalámbrico para controlar tu Raspberry Pi 3, así como también comunicarte de manera inalámbrica con sensores y otros dispositivos.

Sensor de Humedad Suelo : Este sensor de humedad puede leer la cantidad de humedad presente en el suelo que lo rodea. Es un sensor de baja tecnología, pero es ideal para el seguimiento de un jardín urbano.

Sensor de temperatura :Un sensor de temperatura es un sistema que detecta variaciones en la temperatura del aire o del agua y las transforma en una señal eléctrica que llega hasta un sistema electrónico. Esta señal conlleva determinados cambios en ese sistema electrónico para la regulación de la temperatura.

Sensor Humedad Ambiental: El sensor de humedad es un dispositivo utilizado en espacios de interior con el objetivo de controlar la humedad del aire y en ocasiones también la temperatura ambiente

Sensor Lumínico: Estos dispositivos funcionan midiendo la cantidad de luz que llega a su superficie. Cuanta más luz llegue al sensor, más electricidad generará.

Telegram : Esta aplicación la vamos a utilizar para crear un bot que va a notificar al usuario del proyecto los diferentes datos que va ir arrojando la raspberry pi

Python : Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

Visual Studio Code: Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft, diseñado para ser ligero, rápido y altamente personalizable. A diferencia de otros editores de código, como los IDE completos (Entornos de Desarrollo Integrados), VS Code se enfoca en ofrecer una experiencia ágil para la escritura y edición de código, sin sacrificar características avanzadas.

Lo que lo diferencia es su capacidad de ser extendido con una gran cantidad de extensiones y herramientas, adaptándose a las necesidades específicas de cada desarrollador.

Entre sus características clave, se incluyen la autocompletado inteligente mediante IntelliSense, la depuración integrada, y el control de versiones con Git, lo que convierte a VS Code en una opción versátil tanto para desarrolladores individuales como para equipos.

Thonny : Thonny es un entorno de desarrollo integrado (IDE) diseñado específicamente para programadores Python. Su objetivo principal es hacer que la programación en Python sea más accesible para principiantes y, al mismo tiempo, proporcionar las herramientas necesarias para programadores más experimentados.

Una de las características más destacadas de Thonny es su simplicidad. No tienes que preocuparte por configurar entornos virtuales, instalar paquetes o lidiar con configuraciones complicadas. Thonny se encarga de gran parte de esto de forma automática, lo que te permite concentrarte en escribir código Python de manera efectiva.

5. Implementación

5.1. Plan de integración

En este proyecto nuestro objetivo es que la persona se sienta cómoda usando nuestra aplicación , esto lo vamos a conseguir , teniendo una aplicación intuitiva visualmente , contamos con usos muy simples para el uso diario

5.2. Modelo de implementación

La implementación del sistema se llevará a cabo en fases, comenzando con la configuración del **Módulo de Monitoreo de Sensores**, donde los datos serán recolectados y procesados en la Raspberry Pi mediante GrovePi. Posteriormente, el **Módulo de Control Automático** integrará los actuadores, como ventiladores y luces LED, activados según condiciones predefinidas basadas en los datos de los sensores. Luego, el **Módulo de Notificaciones** será enlazado mediante un bot de Telegram para alertar al usuario sobre valores fuera de rango en tiempo real. Finalmente, se integrará el **Módulo de Interfaz de Usuario**, diseñado en Python con Tkinter, permitiendo la visualización de datos, ajustes de parámetros y consulta del historial. Cada módulo será probado de forma independiente y luego integrado en conjunto, garantizando su correcto funcionamiento antes de su despliegue final.

5.3. Módulos implementados

5.3.1. Módulo de Monitoreo de Sensores

Este módulo se encarga de recolectar datos en tiempo real desde los sensores instalados en el invernadero.

Sensores involucrados:

- Sensor de humedad del suelo.
- Sensor de humedad ambiental.
- Sensor de temperatura.
- Sensor lumínico.

Responsabilidades:

- Leer datos del entorno.
- Filtrar y procesar los datos recolectados.
- Garantizar la confiabilidad de las lecturas.

Tecnologías:

- Raspberry Pi para conexión y procesamiento.
- Programación en Python para el manejo de datos.
- GrovePi: Interfaz de hardware que actúa como un puente entre Raspberry Pi y los sensores del sistema, permitiendo conexiones rápidas y configuración sencilla. Incluye soporte para sensores de temperatura, humedad, luminosidad y más.

```
1 import grovepi
2 import time
3
4 # Configuración de los pines de los sensores
5 sensor_temp_hum = 4 # DHT sensor conectado al puerto D4
6 sensor_luz = 0      # Sensor lumínico conectado al puerto A0
7
8 # Configuración del tipo de sensor DHT
9 dht_sensor_type = 0
10
11 def leer_sensores():
12     try:
13         # Leer temperatura y humedad
14         [temp, hum] = grovepi.dht(sensor_temp_hum, dht_sensor_type)
15         # Leer nivel de luz
16         luz = grovepi.analogRead(sensor_luz)
17
18         # Mostrar datos
19         print(f"Temperatura: {temp:.1f}°C, Humedad: {hum:.1f}%, Luz: {luz}")
20
21         # Retornar los valores
22         return temp, hum, luz
23     except IOError:
24         print("Error al leer los sensores.")
25         return None, None, None
26
27 while True:
28     leer_sensores()
29     time.sleep(2) # Leer cada 2 segundos
30
```

5.3.2. Módulo de Control Automático

Este módulo activa los sistemas del invernadero según los valores obtenidos de los sensores.

Acciones automáticas:

- Activar el sistema de riego cuando la humedad del suelo sea insuficiente.
- Encender las luces LED si la luminosidad es baja.
- Controlar la temperatura mediante ventiladores o calefacción según el caso.

Componentes involucrados:

- Sistema de riego automatizado.
- Luces LED.

- Ventiladores y calefacción.

Responsabilidades:

- Gestionar los actuadores basándose en condiciones predefinidas.
- Asegurar tiempos de respuesta mínimos para evitar daños en los cultivos.

```
1  import grovepi
2  import time
3
4  # Pines de LEDs
5  led_azul = 4   # LED Azul (ventilación) en D4
6  led_rojo = 5  # LED Rojo (calefacción) en D5
7  led_verde = 6 # LED Verde (luminosidad) en D6
8
9  # Umbrales predeterminados
10 TEMP_MAX = 30.0 # Máxima temperatura permitida
11 TEMP_MIN = 18.0 # Mínima temperatura permitida
12 HUM_MIN = 40.0  # Mínima humedad permitida
13 LUZ_MIN = 300   # Mínimo nivel de luz permitido
14
15 def ingresar_umbral(parametro):
16     """Función para ingresar umbrales"""
17     while True:
18         try:
19             valor = float(input(f"Ingrese el valor para {parametro}: "))
20             return valor
21         except ValueError:
22             print("Por favor, ingrese un valor numérico válido.")
23
24 # Solicitar al usuario los umbrales
25 print("Configuración de umbrales:")
26 TEMP_MAX = ingresar_umbral("Temperatura máxima (°C)")
27 TEMP_MIN = ingresar_umbral("Temperatura mínima (°C)")
28 HUM_MIN = ingresar_umbral("Humedad mínima (%)")
29 LUZ_MIN = ingresar_umbral("Luminosidad mínima (valor)")
```

```

30
31 def controlar_sistemas(temp, hum, luz):
32     """Controla los LEDs según las condiciones del ambiente."""
33     try:
34         # Control del riego (indicador: humedad baja)
35         if hum < HUM_MIN:
36             grovepi.digitalWrite(led_verde, 1) # Activar LED Verde
37             print("LED Verde ACTIVADO (Riego indicado).")
38         else:
39             grovepi.digitalWrite(led_verde, 0) # Apagar LED Verde
40
41         # Control de temperatura
42         if temp > TEMP_MAX:
43             grovepi.digitalWrite(led_azul, 1) # Encender LED Azul (ventilación)
44             grovepi.digitalWrite(led_rojo, 0) # Apagar LED Rojo
45             print("LED Azul ACTIVADO (Enfriamiento).")
46         elif temp < TEMP_MIN:
47             grovepi.digitalWrite(led_rojo, 1) # Encender LED Rojo (calefacción)
48             grovepi.digitalWrite(led_azul, 0) # Apagar LED Azul
49             print("LED Rojo ACTIVADO (Calefacción).")
50         else:
51             grovepi.digitalWrite(led_azul, 0) # Apagar LED Azul
52             grovepi.digitalWrite(led_rojo, 0) # Apagar LED Rojo
53
54         # Control de luces (luminosidad baja)
55         if luz < LUZ_MIN:
56             grovepi.digitalWrite(led_verde, 1) # Encender LED Verde
57             print("LED Verde ACTIVADO (Luminosidad).")
58         else:
59             grovepi.digitalWrite(led_verde, 0) # Apagar LED Verde
60
61     except IOError as e:
62         print(f"Error en el control de los sistemas: {e}")
63
64
65 def leer_sensores():
66     return 32.0, 35.0, 250 # Temp, Hum, Luz
67
68 # Bucle de control
69 while True:
70     temp, hum, luz = leer_sensores()
71     if temp is not None and hum is not None and luz is not None:
72         controlar_sistemas(temp, hum, luz)
73     time.sleep(2)
74

```

5.3.3. Módulo de Notificaciones

Este módulo se encarga de enviar alertas al usuario cuando los valores de los sensores están fuera de los rangos óptimos, utilizando un bot de Telegram para garantizar una comunicación rápida y eficiente.

Notificaciones incluyen:

- Tipo de condición crítica detectada (e.g., baja humedad, alta temperatura).
- Valores exactos fuera de rango.
- Hora y fecha del evento.

Canales de comunicación:

- Bot de Telegram como principal medio de interacción.
- Mensajes automáticos y en tiempo real enviados directamente al usuario.

Ventajas del bot de Telegram:

- Soporte multiplataforma (móvil y escritorio).
- Comunicación bidireccional: el usuario puede recibir alertas y enviar comandos para consultar datos o ajustar configuraciones.
- Seguridad y cifrado en las comunicaciones.
- Integración sencilla mediante la API oficial de Telegram.

Tecnologías:

- API de Telegram: Para la creación y gestión del bot.
- Python: Uso de bibliotecas como python-telegram-bot para manejar mensajes, eventos y comandos.
- Raspberry Pi: Procesamiento de datos y envío de notificaciones a través del bot.

Responsabilidades del módulo:

- Detectar condiciones críticas a partir de los datos recolectados por los sensores.
- Formatear mensajes claros y fáciles de interpretar.
- Asegurar que las notificaciones lleguen al instante.

```

1  from telegram import Bot
2  from telegram.ext import Updater, CommandHandler
3  import threading
4  import grovepi
5  import time
6
7  # Configuración del Bot de Telegram
8  TELEGRAM_TOKEN = "123456789:ABCD1234EFGH5678IJKL91011MNOPQR"
9  CHAT_ID = "987654321"
10
11 # Crear el bot de Telegram
12 bot = Bot(token=TELEGRAM_TOKEN)
13
14 # Estado actual de los sensores
15 estado_actual = {"temp": 0, "hum": 0, "luz": 0}
16
17 def enviar_alerta(mensaje):
18     """Envía un mensaje al chat de Telegram."""
19     try:
20         bot.send_message(chat_id=CHAT_ID, text=mensaje)
21         print(f"Notificación enviada: {mensaje}")
22     except Exception as e:
23         print(f"Error enviando notificación: {e}")
24
25 def verificar_alertas():
26     """Verificación de los umbrales."""
27     if estado_actual["temp"] > TEMP_MAX:
28         enviar_alerta(f"⚠️ Alerta de Temperatura Alta: {estado_actual['temp']}°C")
29     if estado_actual["hum"] < HUM_MIN:
30         enviar_alerta(f"⚠️ Alerta de Humedad Baja: {estado_actual['hum']}%")
31     if estado_actual["luz"] < LUZ_MIN:
32         enviar_alerta(f"⚠️ Alerta de Luminosidad Baja: Nivel {estado_actual['luz']}")
33
34 def estado(update, context):
35     """Comando /estado para consultar valores actuales."""
36     mensaje = (f"🌡️ Temperatura: {estado_actual['temp']}°C\n"
37              | f"💧 Humedad: {estado_actual['hum']}%\n"
38              | f"💡 Luminosidad: {estado_actual['luz']}")
39     update.message.reply_text(mensaje)

```

```
40
41 updater = Updater(token=TELEGRAM_TOKEN)
42 dispatcher = updater.dispatcher
43
44 dispatcher.add_handler(CommandHandler("estado", estado))
45
46 # Iniciar el bot
47 threading.Thread(target=updater.start_polling).start()
48
49
50 while True:
51     temp, hum, luz = leer_sensores()
52     if temp is not None and hum is not None and luz is not None:
53         estado_actual.update({"temp": temp, "hum": hum, "luz": luz})
54         verificar_alertas()
55     time.sleep(5)
56
```

5.4. Reporte de evaluación

Requerimiento	Cumplido	No Cumplido	Comentarios
Conexión entre usuario y raspberry por medio de Wifi	✘		
El sistema de riego automático debe activarse cuando los niveles de humedad desciendan por debajo del umbral establecido.		✘	No se implementó debido a que no se pudo conseguir el sensor de humedad del suelo
En caso de condiciones ambientales críticas, debe enviarse una notificación inmediata al usuario.	✘		
La interfaz debe permitir al usuario ajustar los umbrales de temperatura, humedad y luminosidad fácilmente.	✘		
Activar automáticamente el ventilador cuando la temperatura supere los valores establecidos.	✘		
Activar automáticamente los focos led cuando el nivel de luminosidad esté por debajo de los valores establecidos.	✘		
Recopilar y desplegar estadísticas sobre las condiciones ambientales en el invernadero	✘		
El sistema debe apagar automáticamente los dispositivos (como ventiladores o luces) cuando las condiciones vuelvan a estar dentro del rango óptimo, para minimizar el consumo de energía.	✘		

Conclusiones

El desarrollo del sistema de monitoreo y control para un invernadero de hortalizas ha demostrado ser una solución tecnológica viable para optimizar las condiciones ambientales y mejorar la productividad agrícola. A través de la implementación de sensores de temperatura, humedad ambiental y luminosidad, se logra recopilar información en tiempo real, lo que permite a los usuarios tomar decisiones informadas para el manejo eficiente del invernadero.

El uso de herramientas como Raspberry Pi y la programación en Python ha simplificado el procesamiento de datos y la automatización de tareas críticas, como la activación de ventiladores y luces. Además, la integración de un bot de Telegram garantiza una comunicación efectiva, alertando al usuario sobre condiciones críticas y permitiendo ajustes rápidos a través de una interfaz accesible y fácil de usar.

Si bien el proyecto enfrentó algunas limitaciones, se logró cumplir con los objetivos principales, destacando la capacidad del sistema para adaptarse a diferentes tipos de hortalizas y escenarios de uso. En este sentido, el proyecto sienta las bases para futuras mejoras, como la incorporación de más sensores y actuadores, asegurando un sistema aún más completo y robusto.

Referencias

1. «Raspberry Pi OS – Raspberry Pi», Raspberry Pi. <https://www.raspberrypi.com>
- 2.«Raspberry Pi OS Documentation» <https://www.raspberrypi.com/documentation/>
- 3.«Requisitos y casos de usos extraídos de : H. Gomaa 2017, Ing. Software: Diseño Distribuido y en Tiempo Real Profesor Ing. MSc. y Dr(c) Diego Aracena Pizarro » <https://drive.google.com/drive/u/2/folders/1UuvNJzyck15JT-yyUDPLgOYbxPFKfGEz>
- 4.«Producción de hortalizas bajo invernadero» <https://biblioteca.inia.cl/server/api/core/bitstreams/283e8b58-a688-4843-bed9-1d6892e26ff7/content>