



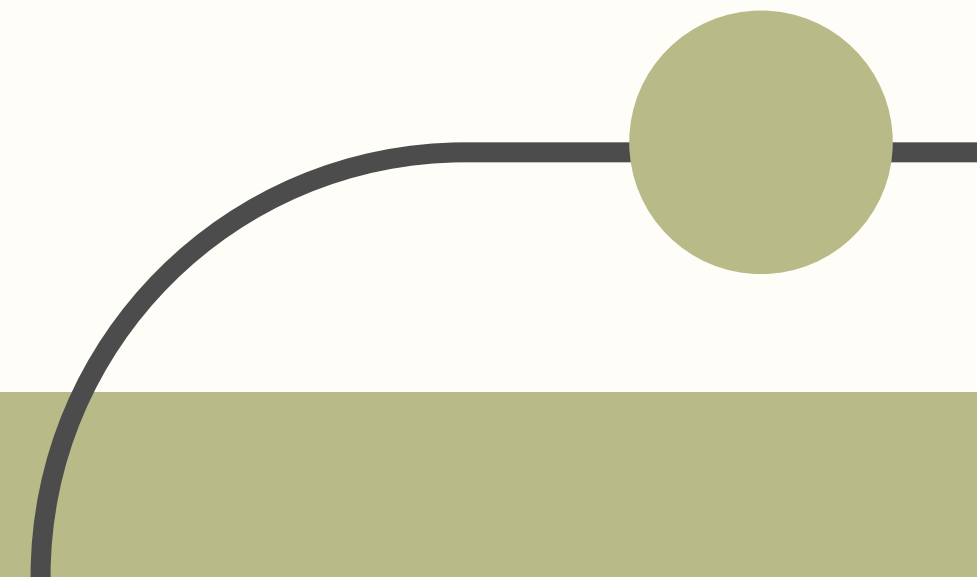
UNIVERSIDAD DE TARAPACÁ
Universidad del Estado

Ingeniería@
Computación e Informática

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

INTRODUCCIÓN



PROBLEMÁTICA

En la actualidad, la gestión manual de las condiciones ambientales en los invernaderos puede ser ineficiente y propensa a errores, esto debido a que está sujeto a la susceptibilidad de la precisión humana, lo que resulta en condiciones subóptimas para el cultivo de hortalizas. Estas condiciones pueden llevar a una disminución de la productividad, una mayor incidencia de enfermedades, y una baja calidad de los frutos.



PANORAMA GENERAL

➔ *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.

OBJETIVOS DEL PROYECTO

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.
- 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.

Actividades de trabajo (Carta Gantt)



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.

Requerimientos Funcionales

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

Escalabilidad:

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

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.

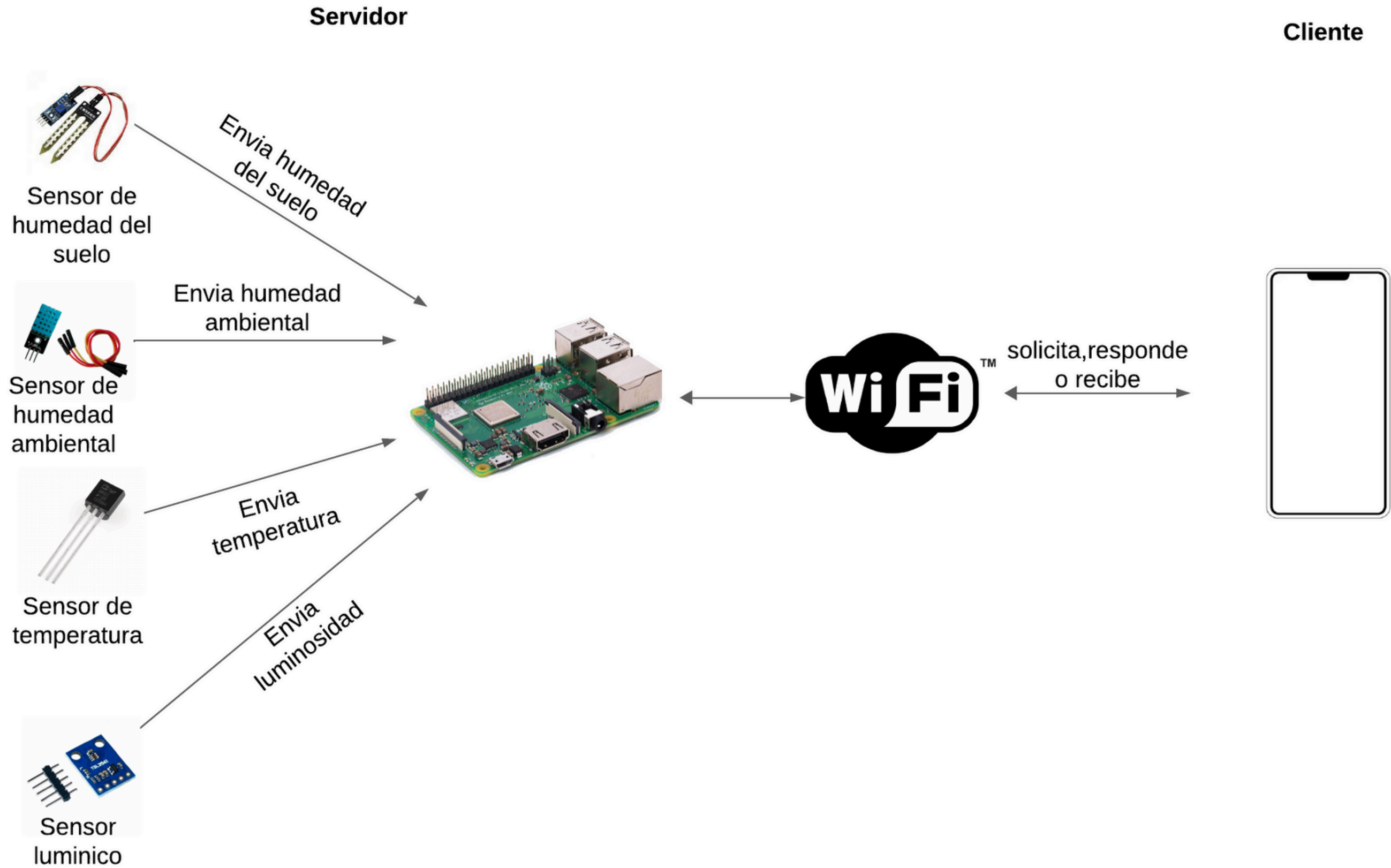
Rendimiento:

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.

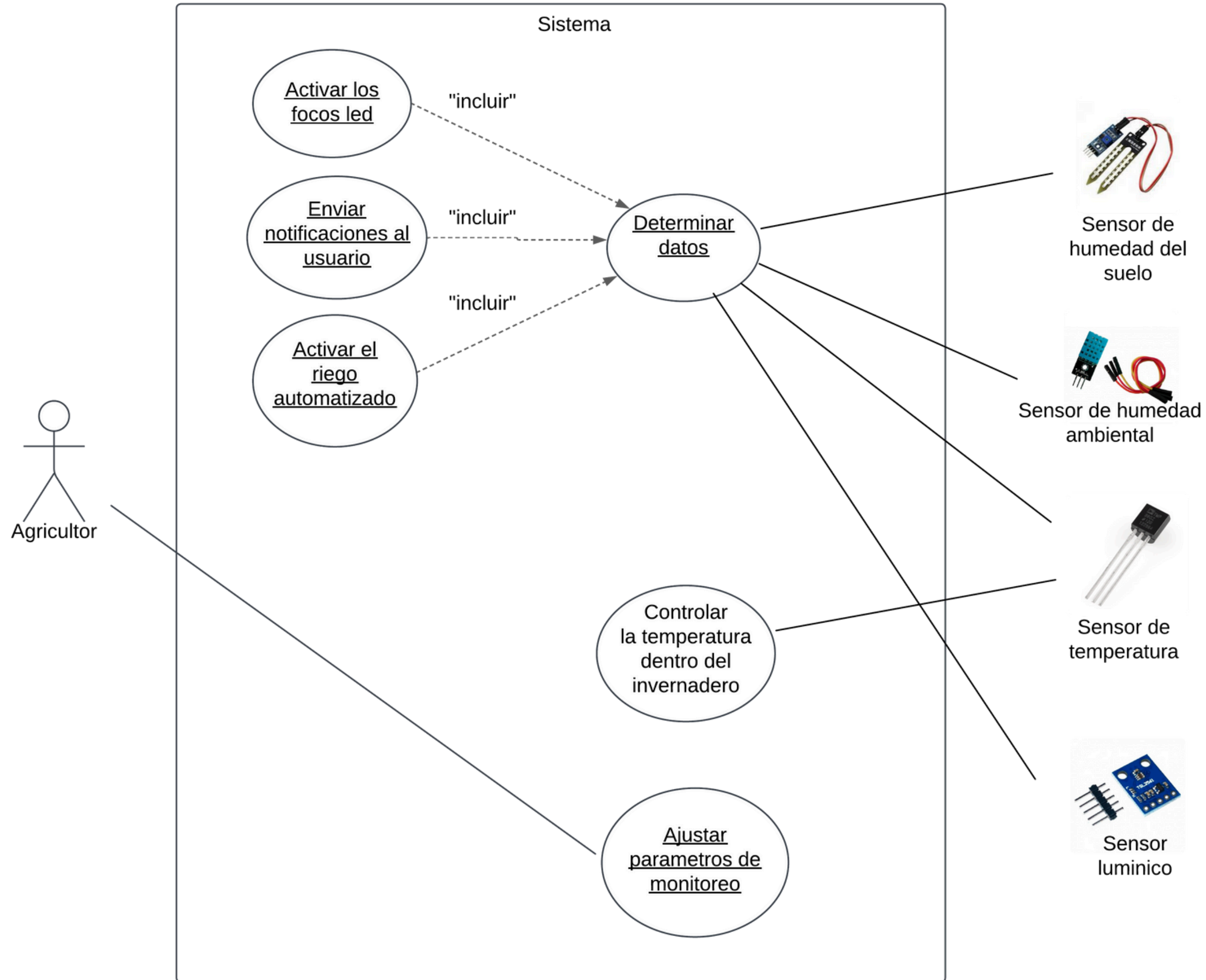
Disponibilidad:

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

Descripción de la arquitectura



Caso de uso general



Herramientas

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

- Python
- Visual Studio Code
- Telegram



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.



Módulos
implementados

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.

```
1 # Funciones de lectura de sensores
2 def read_temp_humidity():
3     try:
4         [temp, humidity] = grovepi.dht(TEMP_HUM_SENSOR_PORT, 0)
5         return temp, humidity
6     except Exception as e:
7         return None, None
8
9 def read_light():
10    try:
11        return grovepi.analogRead(LIGHT_SENSOR_PORT)
12    except Exception as e:
13        return None
14
```


Ajustar parámetros de monitoreo.

```
1 # Comandos para cambiar los límites
2 async def set_temp_H(update: Update, context: ContextTypes.DEFAULT_TYPE):
3     global temp_H
4     try:
5         temp_H = float(context.args[0])
6         await update.message.reply_text(f"Límite superior de temperatura cambiado a {temp_H}°C.")
7     except (IndexError, ValueError):
8         await update.message.reply_text("Por favor, usa el formato correcto: /setTempH <valor>.")
9
10    async def set_temp_L(update: Update, context: ContextTypes.DEFAULT_TYPE):
11        global temp_L
12        try:
13            temp_L = float(context.args[0])
14            await update.message.reply_text(f"Límite inferior de temperatura cambiado a {temp_L}°C.")
15        except (IndexError, ValueError):
16            await update.message.reply_text("Por favor, usa el formato correcto: /setTempL <valor>.")
17
18    async def set_hum_H(update: Update, context: ContextTypes.DEFAULT_TYPE):
19        global hum_H
20        try:
21            hum_H = float(context.args[0])
22            await update.message.reply_text(f"Límite superior de humedad cambiado a {hum_H}%.")
23        except (IndexError, ValueError):
24            await update.message.reply_text("Por favor, usa el formato correcto: /setHumH <valor>.")
25
26    async def set_hum_L(update: Update, context: ContextTypes.DEFAULT_TYPE):
27        global hum_L
28        try:
29            hum_L = float(context.args[0])
30            await update.message.reply_text(f"Límite inferior de humedad cambiado a {hum_L}%.")
31        except (IndexError, ValueError):
32            await update.message.reply_text("Por favor, usa el formato correcto: /setHumL <valor>.")
33
34    async def set_light_L(update: Update, context: ContextTypes.DEFAULT_TYPE):
35        global luz_L
36        try:
37            luz_L = int(context.args[0])
38            await update.message.reply_text(f"Límite inferior de luz cambiado a {luz_L}.")
39        except (IndexError, ValueError):
40            await update.message.reply_text("Por favor, usa el formato correcto: /setLightL <valor>.")
```

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.

```
1  async def enviar_datos(update: Update, context: ContextTypes.DEFAULT_TYPE):
2      temp, humidity = read_temp_humidity()
3      light = read_light()
4      mensaje = ""
5
6      # Datos de temperatura y humedad
7      if temp is not None and humidity is not None:
8          mensaje += f"Temperatura: {temp}°C\nHumedad: {humidity}%\n"
9          if temp > temp_H:
10             mensaje += "Advertencia: Temperatura alta.\n"
11         elif temp < temp_L:
12             mensaje += "Advertencia: Temperatura baja.\n"
13         if humidity > hum_H:
14             mensaje += "Advertencia: Humedad alta.\n"
15         elif humidity < hum_L:
16             mensaje += "Advertencia: Humedad baja.\n"
17     else:
18         mensaje += "Error al leer el sensor de temperatura y humedad.\n"
19
20     # Datos de luz
21     if light is not None:
22         mensaje += f"Intensidad de luz: {light}\n"
23         if light < luz_L:
24             mensaje += "Advertencia: Luminosidad baja.\n"
25     else:
26         mensaje += "Error al leer el sensor de luz.\n"
27
28     await update.message.reply_text(mensaje)
```

Reporte 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.	×		
El usuario puede ajustar los umbrales de temperatura, humedad y luminosidad fácilmente.	×		
Activar automáticamente el ventilador cuando la temperatura supere los valores establecidos.	×		

Requerimiento	cumplido	No cumplido	Comentarios
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.	×		



Demostración

Conclusiones

- El uso de sensores para monitorizar condiciones ambientales resulta fundamental para garantizar un crecimiento saludable de las hortalizas.
- La implementación de sistemas automatizados de riego y control de temperatura permite una respuesta inmediata a las variaciones en el entorno, mejorando así la eficiencia y la productividad del invernadero.
- La integración de un bot de Telegram asegura que el usuario reciba notificaciones inmediatas ante cualquier condición crítica, facilitando una intervención rápida y reduciendo riesgos para las hortalizas.
- El sistema puede ajustarse para diferentes regiones o estaciones, adaptando los parámetros según las necesidades del cultivo.
- La gestión de procesos críticos facilita que los usuarios puedan enfocarse en otras tareas importantes, mejorando la eficiencia en la administración del invernadero.

Referencias

1.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-](https://drive.google.com/drive/u/2/folders/1UuvNJzyck15JT-yyUDPLgOYbxPFKfGEz)

[yyUDPLgOYbxPFKfGEz](https://drive.google.com/drive/u/2/folders/1UuvNJzyck15JT-yyUDPLgOYbxPFKfGEz)

2.«Producción de hortalizas bajo invernadero»

[https://biblioteca.inia.cl/server/api/core/bitstreams/283e8b58-a688-4843-bed9-](https://biblioteca.inia.cl/server/api/core/bitstreams/283e8b58-a688-4843-bed9-1d6892e26ff7/content)

[1d6892e26ff7/content](https://biblioteca.inia.cl/server/api/core/bitstreams/283e8b58-a688-4843-bed9-1d6892e26ff7/content)

3.Raspberry Pi OS Documentation» <https://www.raspberrypi.com/documentation/>

4.«Producción de hortalizas bajo invernadero»

[https://biblioteca.inia.cl/server/api/core/bitstreams/283e8b58-a688-4843-bed9-](https://biblioteca.inia.cl/server/api/core/bitstreams/283e8b58-a688-4843-bed9-1d6892e26ff7/content)

[1d6892e26ff7/content](https://biblioteca.inia.cl/server/api/core/bitstreams/283e8b58-a688-4843-bed9-1d6892e26ff7/content)



Muchas
GRACIAS

