

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Plan de proyecto - Sistema de monitoreo de condiciones para el buen crecimiento de plantas

‘InverTrack’

Autores:

- Denis Condori L.
- Alonso Kalise S.
- Alex Muñoz B.
- Kary Tudela H.

Asignatura:

- Proyecto II

Profesor:

- Diego Aracena P.

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
20/12/2025	1.0	Versión preliminar del formato	Alex Muñoz Denis Condori Kary Tudela Alonso Kalise
22/12/2025	1.1	Revisión y modificación del plan	Alonso Kalise Denis Condori Kary Tudela Alex muñoz
24/12/2025	1.2	Ajuste y refinamiento del informe	Alonso Kalise Kary Tudela Alex Muñoz Denis Condori
25/12/2025	1.3	Versión final del informe	Alonso Kalise Kary Tudela Alex Muñoz Denis Condori

Índice

Historial de Cambios.....	2
Índice.....	3
Índice de Tablas y Figuras.....	4
2. Panorama General.....	5
2.1 Resumen del proyecto.....	5
2.1.1 Modelado de maqueta.....	5
2.1.2 Propósito.....	6
2.1.3 Alcance.....	6
2.1.4 Objetivos.....	6
2.1.4.1 Objetivos generales.....	6
2.1.4.2 Objetivos específicos.....	6
2.1.5 Suposiciones.....	7
2.1.6 Restricciones.....	7
2.1.7 Entregables del proyecto.....	7
3. Organización del proyecto.....	8
3.1 Personal y entidades internas.....	8
3.2 Roles y responsabilidades.....	8
3.3 Métodos de comunicación.....	9
4. Planificación de los procesos de gestión.....	10
4.1 Planificación inicial del proyecto.....	10
4.1.1 Planificación de estimaciones.....	10
4.1.2 Planificación de Recursos Humanos.....	12
4.1.3 Costos Totales.....	12
4.2 Lista de Actividades.....	13
4.2.1 Asignación de Trabajo.....	14
4.2.2 Asignación de Tiempo.....	15
4.3 Planificación Gestión de Riesgos.....	16
5. Planificación de los procesos técnicos.....	18
5.1 Modelo de proceso.....	18
5.1.1 Requerimientos establecidos.....	19
5.1.1.1 Requerimientos Funcionales.....	19
5.1.1.2 Requerimientos no Funcionales.....	19
5.1.2 Diagrama de Clases.....	19
5.1.3 Casos de Uso.....	20
5.1.4 Descripción de la Arquitectura.....	30
5.1.5 Diseño de Interface.....	31
5.2 Herramientas y Técnicas.....	33
6. Problemas Encontrados.....	35
7. Implementación.....	36
8. Conclusión.....	39
9. Trabajo Futuro.....	40
10. Referencias.....	41

Índice de Tablas y Figuras

Tablas:

- [Tabla 1 - Entregables del Proyecto Fase II - Página 7](#)
[Tabla 2 - Personal y Entidades internas - Página 8](#)
[Tabla 3 - Roles y Responsabilidades - Página 8](#)
[Tabla 4 - Recursos Hardware - Página 11](#)
[Tabla 5 - Recursos Software - Página 11](#)
[Tabla 6 - Planificación Recursos Humanos - Página 12](#)
[Tabla 7 - Costos Totales - Página 12](#)
[Tabla 8 - Lista de Actividades - Página 13](#)
[Tabla 9 - Asignación de Trabajo - Página 14](#)
[Tabla 10 - Tipos de Riesgos - Página 16](#)
[Tabla 11 - Nivel de Impacto - Página 16](#)
[Tabla 12 - Tabla de Riesgos - Página 17](#)
[Tabla 13 - CU Monitoreo de parámetros - Página 20](#)
[Tabla 14 - CU Control de pH del agua - Página 22](#)
[Tabla 15 - CU Control de temperatura - Página 24](#)
[Tabla 16 - CU Notificación de anomalía en el sistema - Página 26](#)
[Tabla 17 - CU Monitoreo de nutrientes del suelo - Página 28](#)

Figuras:

- [Figura 1 - Vista aérea de la maqueta - Página 5](#)
[Figura 2 - Carta Gantt - Página 15](#)
[Figura 3 - Diagrama de Clases - Página 19](#)
[Figura 4 - DS Monitoreo de parámetros - Página 21](#)
[Figura 5 - DS Control de pH del agua - Página 23](#)
[Figura 6 - DS Control de temperatura - Página 25](#)
[Figura 7 - DS Notificación de anomalía en el sistema - Página 27](#)
[Figura 8 - DS Monitoreo de nutrientes del suelo - Página 29](#)
[Figura 9 - Arquitectura del Sistema - Página 30](#)
[Figura 10 - Diseño menú principal - Página 31](#)
[Figura 11 - Diseño menú pH - Página 32](#)
[Figura 12 - Diseño menú NPK - Página 32](#)
[Figura 13 - Diseño menú Temperatura - Página 33](#)

2. Panorama General

2.1 Resumen del proyecto

La región de Arica y Parinacota se caracteriza por tener suelos fértiles, lo que la convierte en una región ideal para el cultivo. Sin embargo, también enfrenta condiciones climáticas adversas, como la escasez de agua, altas temperaturas, intensa radiación solar y variación en la humedad del suelo. Es imperativo el desarrollo de un sistema que permita al usuario monitorear las condiciones que afectan a sus cultivos.

Para esta problemática se propone el desarrollo de un sistema de monitoreo y control que permita al usuario el fácil monitoreo de sus plantas a través de una interfaz de teléfono. La que desplegará todos los parámetros (en tiempo real) importantes para el buen crecimiento de sus plantas, como por ejemplo: La temperatura del invernadero, el pH del agua de riego, los nutrientes de la tierra de cultivo (NPK), etc.

2.1.1 Modelado de maqueta

Para la fase inicial del proyecto se diseñó una maqueta en realidad virtual que busca reflejar la implementación del sistema planificado.

Para esto se usaron los lentes de realidad virtual Meta Quest 3, junto con software adicional que nos ayudó durante el desarrollo de la maqueta, estos son:

- Blender: Software utilizado para el modelamiento de los objetos que incluye la maqueta.
- Unity: Software que nos ayudó a implementar el modelo del invernadero a un entorno de realidad virtual.

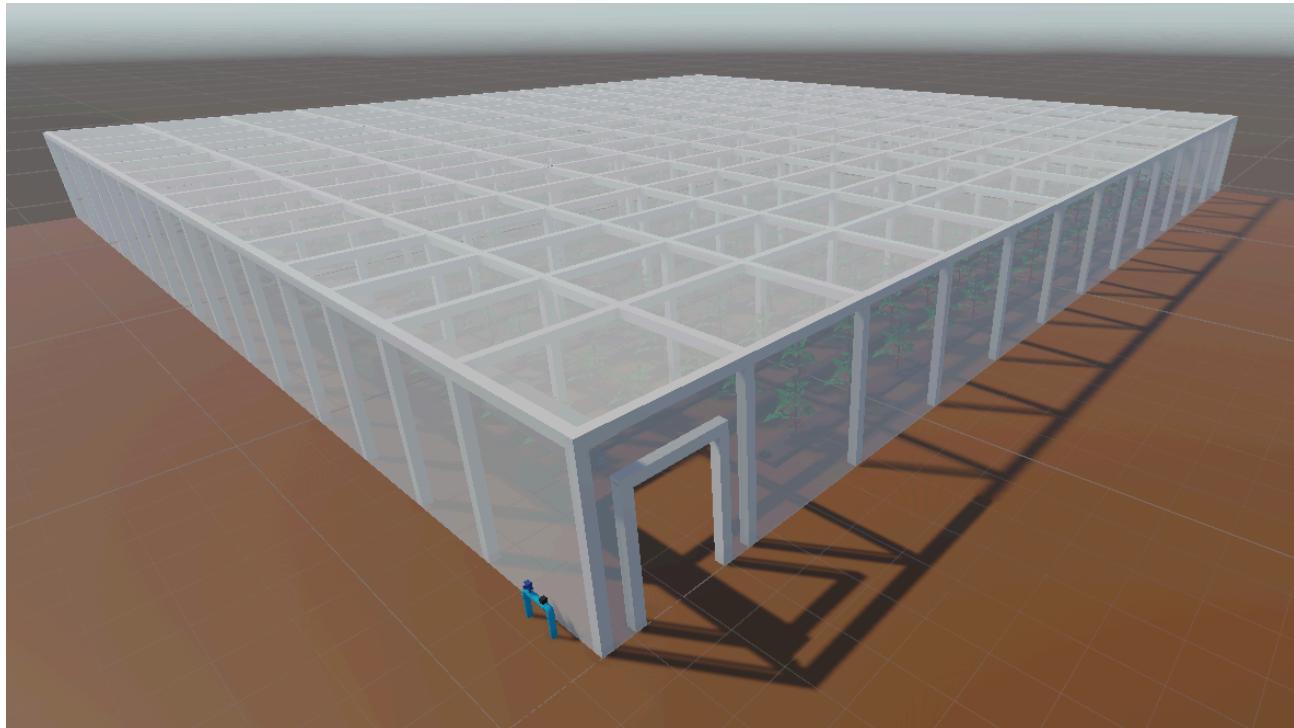


Figura 1 - Vista aérea de la maqueta

2.1.2 Propósito

El propósito de este proyecto es optimizar el manejo de cultivos en la región de Arica y Parinacota, mediante el monitoreo en tiempo real de las variables ambientales y del suelo que influyen directamente en el crecimiento de las plantas.

Donde el proyecto cuenta con un sistema de sensores conectados y programados en un Raspberry PI. Estos mismos están desplegados en distintos puntos del invernadero para realizar la medición correspondiente categorizados según su función, tal como sensores de temperatura, radiación solar, pH, etc.

2.1.3 Alcance

Lo que busca realizar el proyecto, a través, del Raspberry PI y los distintos sensores de medición de temperatura, pH, etc. Son las siguientes funcionalidades:

- Medir en tiempo real la humedad del suelo, el pH, la temperatura ambiente y del suelo, y la radiación solar dentro del invernadero.
- Mostrar los datos recolectados a través de una interfaz accesible para el usuario (celular o computador).
- Notificar al usuario cuando alguna variable se encuentra fuera de los rangos óptimos para el cultivo.
- Entregar recomendaciones básicas al usuario sobre acciones que pueden tomarse para corregir o mejorar las condiciones del cultivo.

2.1.4 Objetivos

2.1.4.1 Objetivos generales

Desarrollar un sistema de monitoreo y control para un invernadero, que permita registrar y analizar variables en tiempo real con el propósito de optimizar el manejo, crecimiento y facilitar las decisiones en función de las condiciones específicas del cultivo.

2.1.4.2 Objetivos específicos

- Investigar y utilizar herramientas como Raspberry PI
- Identificar y desarrollar una problemática haciendo uso de Raspberry Pi 4B.
- Planificar el diseño del proyecto a través de una maqueta en realidad virtual.
- Desarrollar el sistema de monitoreo y control del invernadero, integrando los sensores y actuadores.
- Programar los sensores en base a información sobre los requerimientos agronómicos (humedad, pH, temperatura, radiación solar, entre otros) de distintos cultivos para definir parámetros de referencia en el sistema de monitoreo.
- Desarrollar una interfaz que muestre los datos obtenidos en tiempo real de los sensores.
- Realizar pruebas para el perfecto funcionamiento del sistema.
- Documentar el desarrollo, resultados y conclusiones del proyecto realizado.

2.1.5 Suposiciones

- Se asume que el usuario cuenta con un teléfono inteligente y una red Wifi estable para conectar el equipo Raspberry Pi 4 mediante una conexión cliente-servidor.
- Se asume que la conexión entre los sensores y la Raspberry PI serán estables, permitiendo que la transmisión de datos de los sensores anteriormente mencionados se realicen sin interrupciones.
- Se espera que el equipo de desarrollo tenga conocimientos básicos y experiencia en la integración de hardware y el desarrollo de software, para el desarrollo del proyecto.
- Se asume que los sensores, el sistema Raspberry Pi, y el software desarrollado deberán ser compatibles entre sí, cumpliendo con las funcionalidades esperadas.
- Se asume que para la finalización del proyecto, éste será escalable, o sea, que permitirá más adelante la posibilidad de agregar más sensores y/o expandir a más invernaderos el sistema.
- Se asume que el usuario cuenta con un invernadero sólido, en dónde será implementado el sistema.

2.1.6 Restricciones

- El sistema debe ser desarrollado utilizando Raspberry PI.
- La interfaz y el código correspondiente a cada sensor será programado en Python.
- El sistema debe ser diseñado específicamente para un entorno agricultor.
- Disponibilidad y precisión de sensores: Los sensores pueden tener limitaciones en cuanto a su precisión, vida útil y capacidad para funcionar correctamente bajo condiciones extremas de temperatura y radiación solar intensa propias de la región.
- Limitación en la adaptación a diferentes cultivos: El sistema puede estar inicialmente configurado para un tipo específico de cultivo, por lo que adaptar los parámetros a otros cultivos podría requerir ajustes o actualizaciones posteriores.

2.1.7 Entregables del proyecto

Revisión	Entregables	JP	DO	PR	DI
Fase II	Especificación de requerimientos	X			
Fase II	Bitácoras semanales		X		
Fase II	2do Informe	X	X	X	X
Fase II	Wiki				X
Fase II	Carta Gantt			X	
Fase II	Caso de uso general			X	
Fase II	Descripción de la Arquitectura	X			X
Fase II	Diseño de Interface			X	X
Fase II	Diagrama de Clases		X	X	
Fase II	Diagramas de Secuencia	X	X		

Tabla 1 - Entregables del Proyecto Fase II

3. Organización del proyecto

3.1 Personal y entidades internas

Entidad	Descripción
Jefe de proyecto	Es el encargado de la coordinación, elaboración, cooperación y desarrollo del proyecto, designando al resto de integrantes a sus debidas funciones.
Programador	Es el encargado de establecer la conexión entre los sensores y el Raspberry PI, a su vez se encarga de la asignación de funciones en los sensores y la compatibilidad con el software.
Documentador	Es el encargado de la creación y refinamiento de la documentación del avance del proyecto, responsable de realizar las bitácoras semanales y los informes de avance.
Diseñador	Es el encargado de crear, diseñar y compatibilizar los diseños 3D a través de distintos software para la creación del ambiente y la maqueta del proyecto.

Tabla 2 - Personal y entidades internas

3.2 Roles y responsabilidades

ROL	ENCARGADO	INVOLUCRADO
Jefe de Proyecto	Alex Muñoz	Alex muñoz
Documentador	Kary Tudela	Kary Tudela, Denis Condori
Programador	Denis Condori	Denis Condori, Alonso Kalise, Kary Tudela
Diseñador	Alonso Kalise	Alonso Kalise, Alex muñoz

Tabla 3 - Roles y responsabilidades

3.3 Métodos de comunicación

Reuniones y toma de decisiones (En tiempo real)

Para la comunicación inmediata a la hora de toma de decisiones, distribución, realización, descarte, etc. Todo se hará a través del software **Discord**, en donde se realizarán:

1. Reuniones de coordinación de forma semanal entre los miembros del equipo.
2. Presentación de avances y decisiones importantes que requieran a todos los integrantes del equipo.

Para la comunicación rápida se utilizará la aplicación digital **WhatsApp**, que nos permitirá la:

1. Resolución rápida de dudas respecto al proyecto que surjan durante el desarrollo.
2. Coordinación de reuniones, tareas, todo lo requerido.

Para la recopilación y modificación de archivos, se utilizará la plataforma digital **Google Drive**, que nos permite::

Almacenamiento compartido de archivos referentes al proyecto (Avances, Informes, Bitácoras).
Modificación en tiempo real de los archivos relacionados al proyecto mencionados anteriormente.

4. Planificación de los procesos de gestión

4.1 Planificación inicial del proyecto

4.1.1 Planificación de estimaciones

Recursos Hardware			
Producto	Cantidad	Costo Unidad	Costo Total
Raspberry Pi	1	\$124.990	\$124.990
Kit cables de conexión	11	\$200	\$2.200
Grove Led Rojo	1	\$2.500	\$2.500
Grove Led Verde	1	\$2.500	\$2.500
Grove Led Azul	1	\$2.500	\$2.500
Grove Sensor de Luz	1	\$3.400	\$3.400
Grove Buzzer	1	\$2.990	\$2.990
Grove Sensor de sonido	1	\$4.000	\$4.000
Grove Botón	1	\$1.650	\$1.650
Grove LCD RGB Backlight	1	\$17.000	\$17.000
Grove Rotary Angle Sensor	1	\$4.500	\$4.500
Grove Temperature and Humidity Sensor	2	\$6.600	\$13.200
Grove Ultrasonic Ranger	1	\$5.000	\$5.000
Grove Relay	1	\$16.000	\$16.000
Protoboard	1	\$3.000	\$3.000
Notebook	2	\$200.000	\$400.000
Chatreey ANP2	1	\$200.000	\$200.000
Monitores	2	\$90.000	\$180.000
Sensor NPK	1	\$15.000	\$15.000

Recursos Hardware

Mouse	2	\$15.000	\$30.000
Teclados	2	\$10.000	\$20.000
Tarjeta Micro SD	1	\$6500	\$6500
Caja Protectora Raspberry	1	\$3000	\$3000
Total \$			\$1.059.930

Tabla 4 - Recursos Hardware

Recursos Software

Producto	Cantidad	Precio
Unity	4	\$0
Blender	4	\$0
Canva Premium	1	\$7.990
Discord	4	\$0
Google Drive	4	\$0
Google Documentos	4	\$0
Visual Studio Code	4	\$0
Total \$		\$7.990

Tabla 5 - Recursos Software

4.1.2 Planificación de Recursos Humanos

Rol	Cantidad por Rol	Costo/Hora	Horas Totales	Costo Total CLP
Jefe de Proyecto	1	21.000	142	2.982.000
Programador	1	9.000	142	1.278.000
Diseñador	1	7.000	142	994.000
Documentador	1	7.500	142	1.065.000
				Total \$ 6.319.000

Tabla 6 - Planificación Recursos Humanos

4.1.3 Costos Totales

Tipo Costo	Costo
Costo Hardware	\$ 1.059.930
Costo Software	\$7.990
Costo Recursos Humanos	\$ 6.319.000
Total :	\$ 7.380.320

Tabla 7 - Costos Totales

4.2 Lista de Actividades

Actividad	Descripción	Responsable
Lluvia de Ideas	Lluvia de ideas para elegir el tipo de proyecto que se va a desarrollar.	- Denis Condori - Alex Muñoz - Alonso Kalise - Kary Tudela
Establecer alcance del proyecto	Establecer la solución y alcance de la misma.	- Denis Condori - Alex Muñoz - Alonso Kalise - Kary Tudela
Bosquejo Maqueta	Diseño inicial del sistema ya implementado, para ser modelado en realidad virtual.	- Alex Muñoz - Alonso Kalise
Estudio de Software a utilizar	Estudio de los softwares a utilizar para la implementación de la maqueta en realidad virtual. (Blender,Unity)	- Denis Condori - Alex Muñoz - Kary Tudela
Modelado de Maqueta	Inicio del modelado de la maqueta en el software Blender y modelado de sensores y crear la aplicación para los lentes meta quest 3.	- Alex Muñoz - Denis Condori
Grabación de Maqueta	Grabación de video en el que se expone el modelo virtual del proyecto.	- Denis Condori - Alonso Kalise
Desarrollo Informe II	Redacción del Informe de la primera fase del proyecto.	- Denis Condori - Alonso Kalise - Kary Tudela

Tabla 8 - Lista de Actividades

4.2.1 Asignación de Trabajo

Actividad	Descripción	Responsable
Bitácoras	Registro del progreso y avance de las actividades semanalmente	-Kary Tudela -Alonso Kalise
Organización	Asignación de actividades que realizará cada integrante	-Alex Muñoz
Establecer Problemática	Identificar problemática y la solución correspondiente	-Kary Tudela -Alex Muñoz -Alonso Kalise -Denis Condori
Panorámica respecto a las posibles soluciones	Análisis general respecto a los pros y contras de las soluciones propuestas	-Kary Tudela -Alex Muñoz -Alonso Kalise -Denis Condori
Creación Modelo 3D	Creación de maqueta e implementación de herramientas 3D	-Alex Muñoz -Denis Condori -Kary Tudela
Presentación	Se expone la problemática, solución, objetivos y todo lo relacionado al tema dispuesto a resolverse	-Alonso Kalise -Kary Tudela -Alex Muñoz -Denis Condori
Informe 2	Creación y diseño de la fase 1 del informe a presentar titulado "Diseño y Construcción de la maqueta"	-Kary Tudela -Alex Muñoz -Alonso Kalise -Denis Condori

Tabla 9 - Asignación de Trabajo

4.2.2 Asignación de Tiempo

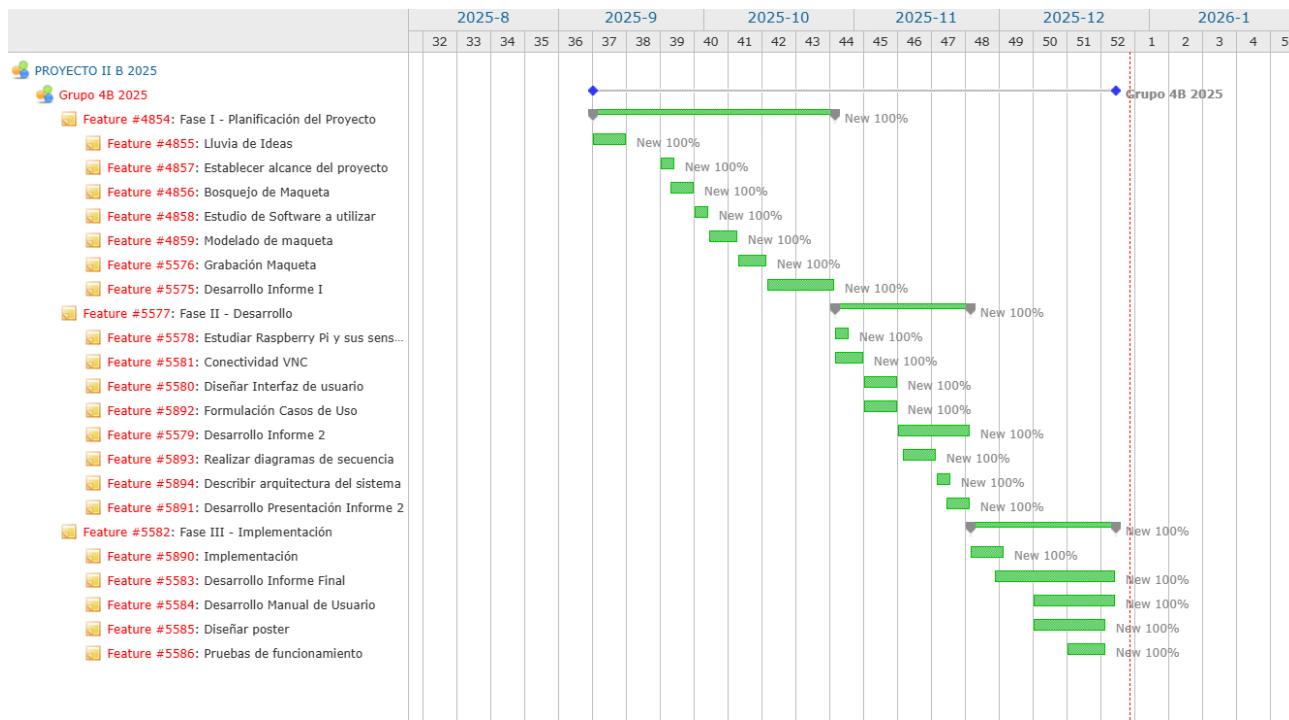


Figura 2 - Carta Gantt

4.3 Planificación Gestión de Riesgos

Para la planificación de la gestión de riesgos se consideraron los siguientes tipos de riesgos y sus factores:

Tipo de Riesgo	Descripción
Tecnológico	Retrasos en la entrega a causa de distintos problemas relacionados con el hardware y el software.
Humano	Problemas con el personal por diversas causas que retrasan el progreso del proyecto tales como: problemas de salud, problemas de agenda, problemas de comunicación entre los miembros, etc.
Requerimientos	Frecuentes cambios respecto a la solicitud del cliente.
Organizacional	Falta de liderazgo y poca claridad a la hora de organizar al equipo.
Estimación	Incumplimiento de los plazos propuestos.

Tabla 10 - Tipos de Riesgos

Para el impacto de los riesgos que podrían surgir durante el desarrollo del proyecto, se clasificaron en los siguientes cuatro niveles de impacto.

Nivel de impacto	Descripción
1. Catastrófico	Impacto crítico que puede poner en riesgo la continuidad o el éxito del proyecto.
2. Crítico	Impacto significativo que requiere recursos adicionales para ser gestionado, pero el proyecto puede continuar
Marginal	Impacto leve que puede retrasar algunos aspectos del proyecto, pero sin afectar gravemente los resultados.
Despreciable	Impacto mínimo que no requiere acción inmediata y no afectará de manera relevante el desarrollo del proyecto.

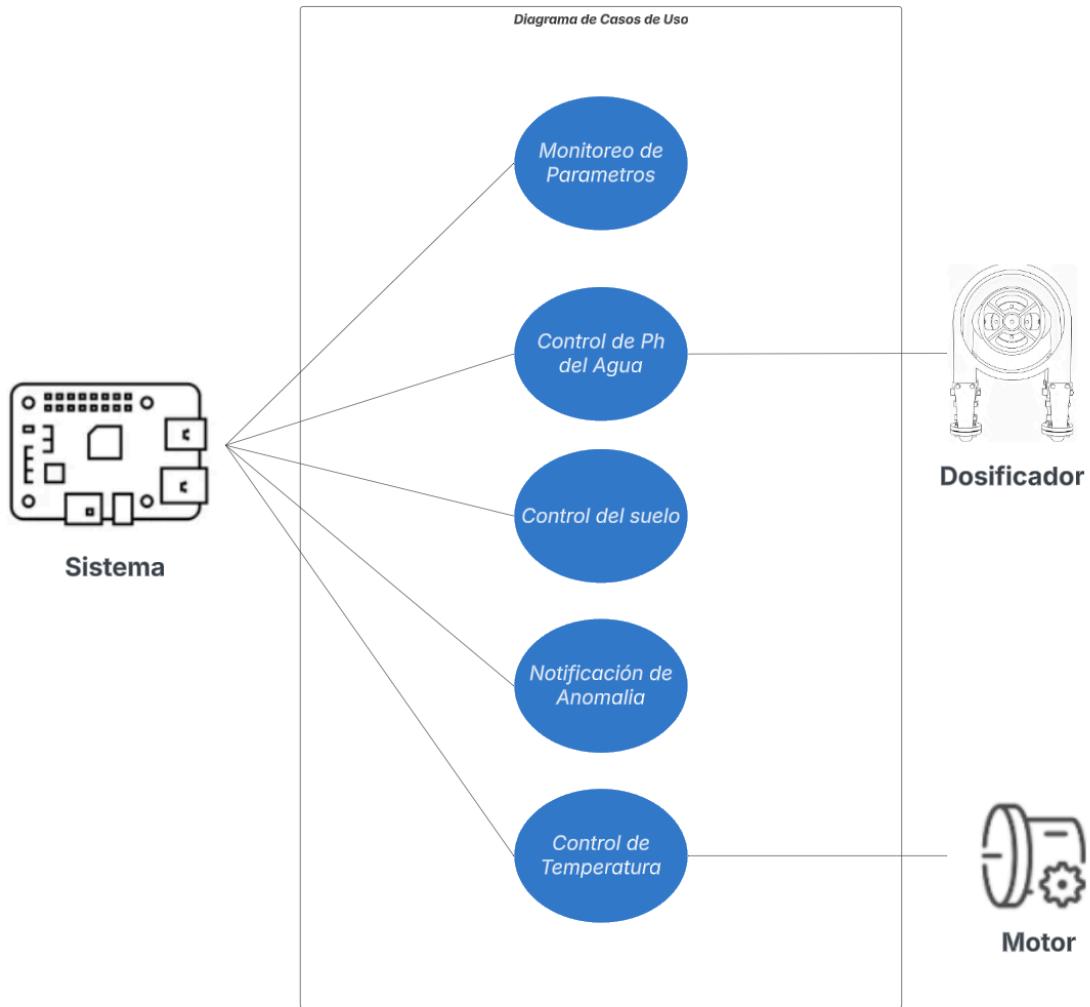
Tabla 11 - Nivel de Impacto

Riesgos	Nivel de Impacto	Probabilidad de ocurrencia	Acción Remedial
Perdida tarjeta Micro SD	2	80%	Realizar respaldos de seguridad y tener a disposición otras SD.
Pérdida de equipo notebooks	1	40%	Establecer lugares seguros para el trabajo que se realizará con los notebooks.
Pérdida de equipo Raspberry	1	20%	Utilizar el equipo Raspberry solamente en las ocasiones que se requiera y siempre en un espacio seguro.
Disponibilidad	3	40%	Establecer horarios compatibles con todos los miembros para las actividades a realizar
Mala distribución en el trabajo del equipo	2	50%	Coordinar reuniones vía WhatsApp, Discord donde se establecerá la distribución, en caso de discrepancias, por los mismos medios se llegará a una solución conjunta.
Cambio de requisitos a pedido del cliente	1	60%	Realizar una reunión con el cliente donde se discutirá la viabilidad respecto a los cambios requeridos.
Errores en el software de monitoreo	2	70%	Realizar pruebas continuas desde el inicio del desarrollo, realizando un monitoreo de cada uno de los componentes utilizados.
Ausencia prolongada de un miembro del equipo de trabajo	3	30%	Reasignar tareas de manera temporal para cumplir con el avance del proyecto.
Falta de capacitación y conocimientos técnicos	3	40%	Realizar capacitaciones, otorgar guía para el manejo de componentes específicos.
Se necesita hardware adicional al inicialmente planeado	2	20%	Realizar una petición formal para solicitar el hardware requerido.

Tabla 12 - Tabla de Riesgos

5. Planificación de los procesos técnicos

5.1 Modelo de proceso



5.1.1 Requerimientos establecidos

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 describen los requerimientos asociados a nuestro proyecto:

5.1.1.1 Requerimientos Funcionales

- Determinación de datos sensoriales: Capacidad de leer y procesar datos de los sensores conectados, incluyendo:
 - Sensor NPK.
 - Sensor pH del agua.
 - Sensor de temperatura ambiente.
 - Sensor de temperatura del suelo.
- Envío de notificaciones al usuario: Capacidad de enviar notificaciones al agricultor cuando se detectan condiciones críticas anormales (por ejemplo, temperatura del suelo baja, pH del agua de riego elevado, etc.), permitiéndole tomar acciones adicionales si es necesario.
- 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 despliegue de mallas para mantener la temperatura dentro de los rangos óptimos.
- Control del pH del agua de riego: El sistema debe monitorear y controlar el pH del agua de riego mediante el sensor de pH, activando un dosificador de agua que permita mantener el pH dentro de los rangos ideales para el correcto crecimiento del cultivo.
- Ajuste de parámetros de monitoreo: Debe permitir al agricultor ajustar los parámetros de monitoreo, tales como los rangos óptimos pH del agua, humedad y temperatura, para que el control y las notificaciones se adapten a las necesidades específicas del cultivo.

5.1.1.2 Requerimientos no Funcionales

- Escalabilidad: La solución debe permitir la integración de los distintos sensores y actuadores.
- 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.
- Confiabilidad: La recolección y procesamiento de datos de los sensores debe ser confiable, minimizando errores de lectura o fallas en el funcionamiento, 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.

5.1.2 Diagrama de Clases

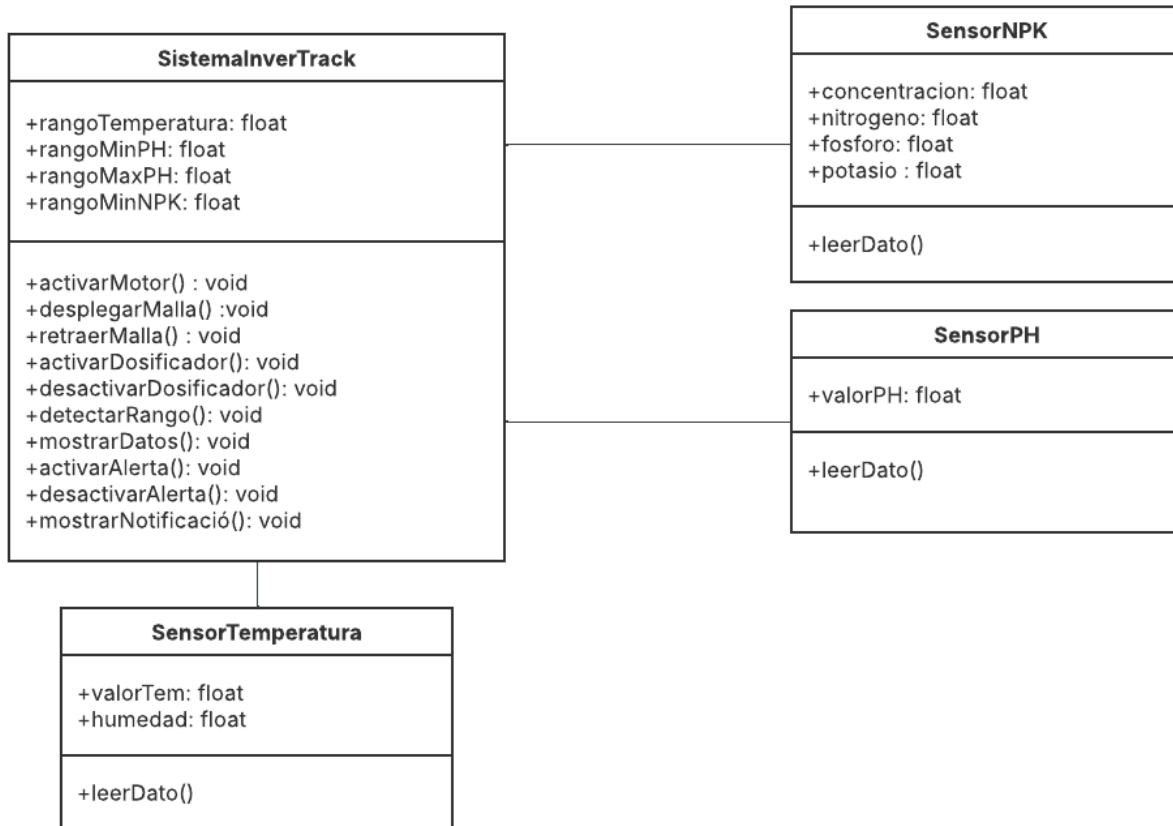


Figura 3 - Diagrama de Clases

5.1.3 Casos de Uso

ID: 01	
Nombre del CU: Monitoreo de parámetros.	
Actor(es): Sensores / Sistema	
Descripción: El sistema registra en tiempo real los datos de la temperatura ambiente, pH, temperatura del suelo y NPK. donde estos datos serán visualizados en la interfaz de usuario.	
Precondiciones: Los sensores deben estar conectados y ser reconocidos por el Raspberry Pi.	
Flujo Principal: Sensores <ol style="list-style-type: none"> 1. Se inicia con la lectura de datos de los sensores. 2. Los sensores envían los datos registrados al Raspberry Pi. 	Flujo Principal: Sistema <ol style="list-style-type: none"> 3. El sistema procesa los datos y los almacena temporalmente. 4. Los datos registrados se muestran en la interfaz de usuario.
Flujo Alternativo: Sensores	Flujo Alternativo: Sistema
Postcondiciones: Los datos de las lecturas de los sensores quedan registrados en la interfaz.	

Tabla 13 - CU Monitoreo de parámetros

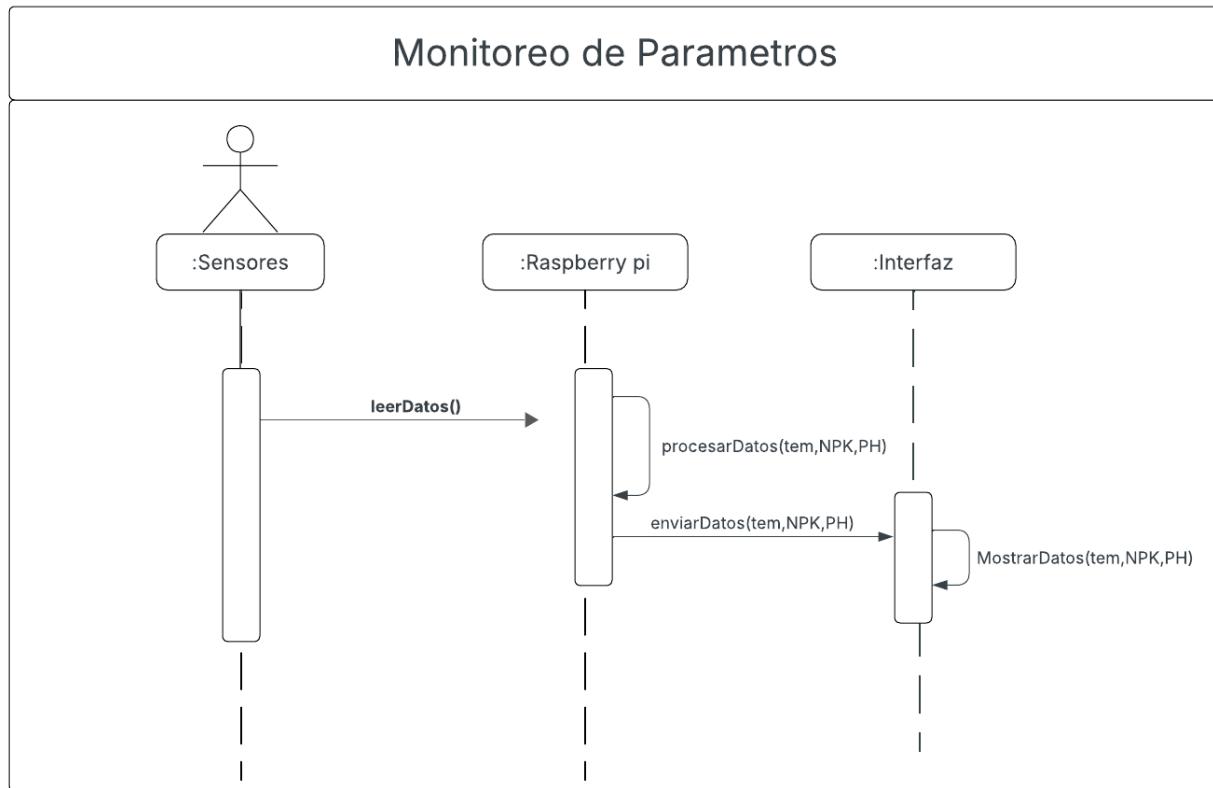


Figura 4 - DS Monitoreo de Parámetros

ID: 02	
Nombre del CU: Control de ph del agua.	
Actor(es): Sensor PH, sistema y dosificador	
Descripción: El sistema ajusta el pH del agua para mantenerlo dentro del rango definido por el usuario, activando un dosificador.	
Precondiciones: El sensor, dosificador deben estar operativos y los parámetros deben estar definidos.	
Flujo Principal: Sensor de PH 1. El sensor envía el valor de pH.	Flujo Principal: Sistema 2. El sistema verifica que el valor que entrega el sensor está fuera de rango. 3. El usuario acepta la solicitud para activar el dosificador. 4. El sistema manda una señal para activar el dosificador donde se agrega productos químicos para controlar el ph del agua. 5. Se monitorea el sensor de ph hasta que vuelva al rango óptimo.
Flujo Alternativo:	Flujo Alternativo: 2.1 El sistema continúa recibiendo datos y mostrandolos en interfaz.
Postcondiciones: El pH del agua queda dentro del rango definido por el usuario.	

Tabla 14 - CU Control del pH del agua

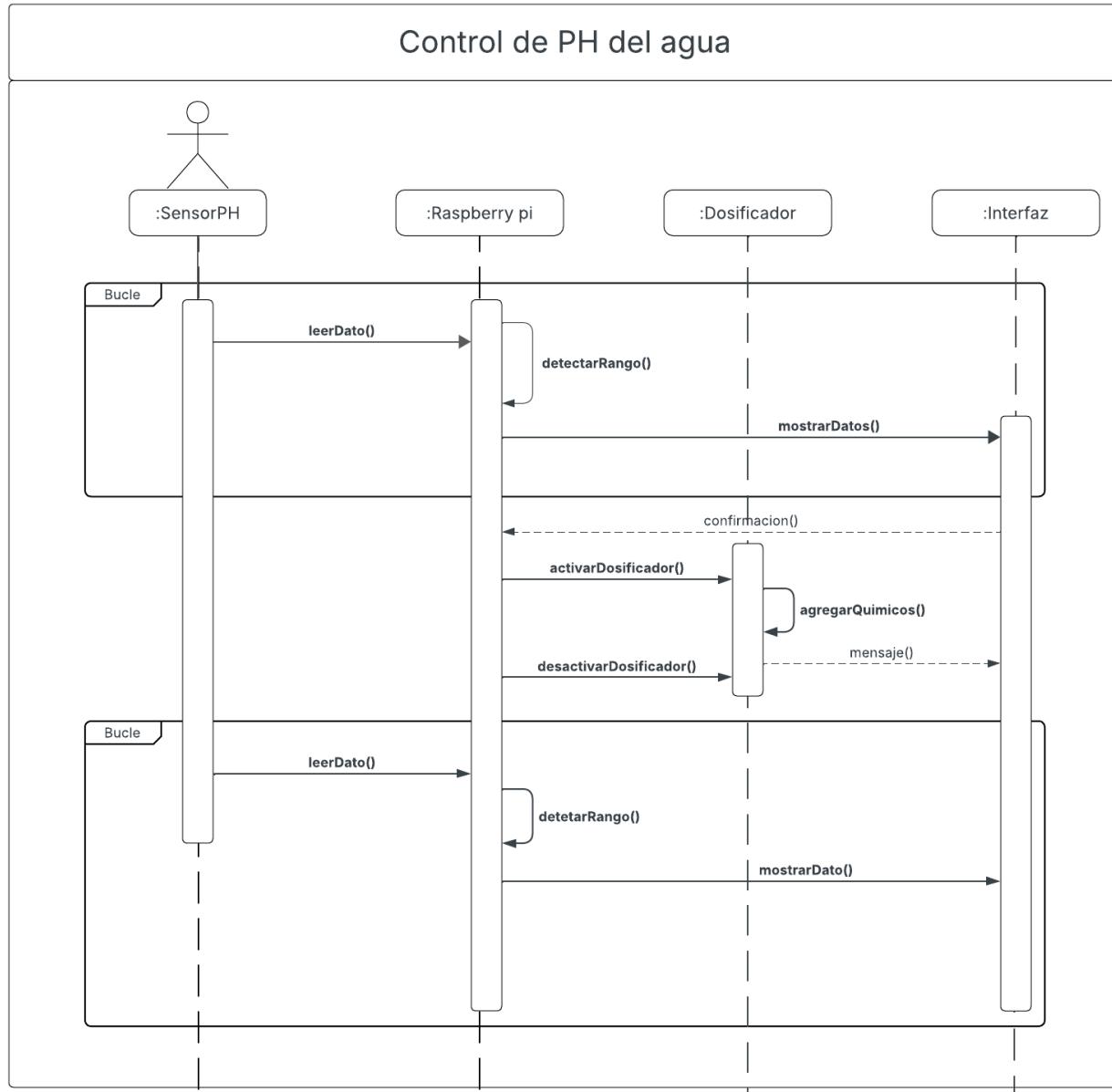


Figura 5 - DS Control de pH del agua

ID: 03	
Nombre del CU: Control de temperatura.	
Actor(es): Sensor Temperatura, Sistema y motor	
Descripción: El sistema regula la temperatura ambiental dentro del invernadero mediante el despliegue de una malla que bloquea la luz solar y reduce el calor cuando los valores superan los valores.	
Precondiciones: El sensor y las mallas deben estar instaladas y operativas , los rangos de temperatura deben estar definidos.	
Flujo Principal: Sensor Temperatura 1. Se inicia con la lectura del sensor de temperatura.	Flujo Principal: Sistema 2. El sistema detecta que datos obtenidos de la temperatura está sobre el rango establecido. 3. El sistema enciende el motor y este despliega la malla para bajar la temperatura del invernadero. 4. El sistema monitorea los valores del sensor hasta que vuelvan al rango. 5. El sistema enciende el motor para retraer la malla cuando se estabiliza la temperatura dentro del invernadero.
Flujo Alternativo:	Flujo Alternativo: 2.1 El sistema detecta que datos obtenidos de la temperatura están dentro del rango. 3.1 El sistema enciende el motor y este retrae la malla.
Postcondiciones: La temperatura del invernadero vuelve a quedar dentro de los parámetros establecidos.	

Tabla 15 - CU Control de Temperatura

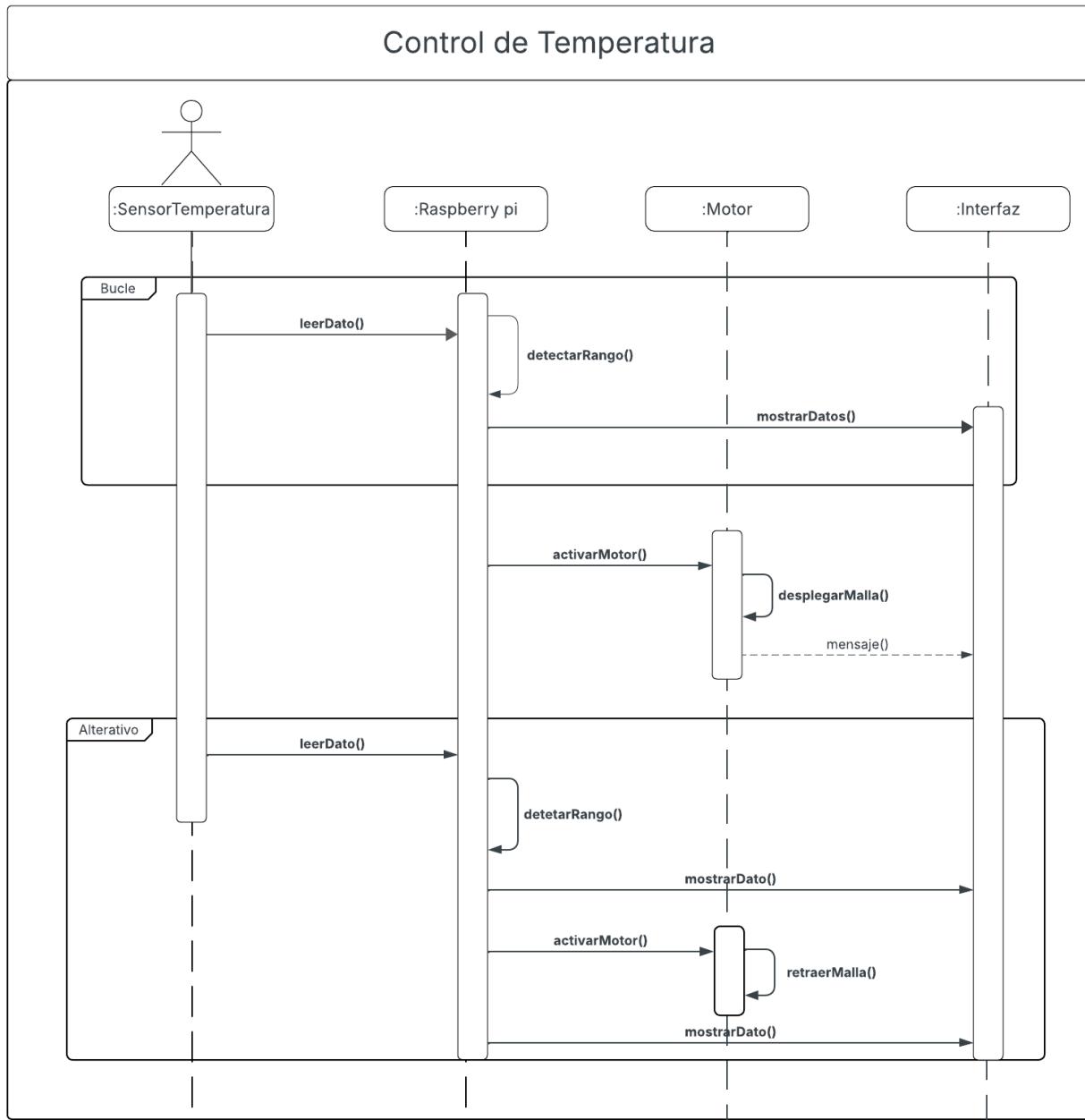


Figura 6 - DS Control de Temperatura

ID: 04	
Nombre del CU: Notificación de anomalía en el sistema	
Actor(es): Sistema, Interfaz	
Descripción: La Raspberry Pi monitorea a través de los sensores y al notar una anomalía lo notifica.	
Precondiciones: El sistema debe detectar anomalías en el sistema de monitoreo	
Flujo Principal: Sensores <ul style="list-style-type: none">1. Los sensores envían los datos obtenidos al sistema.	Flujo Principal: Sistema (Raspberry Pi) <ul style="list-style-type: none">1. La Raspberry Pi recibe un parámetro que se salió de rango.2. El sistema genera y envía una alerta del sensor que se salió de rango con el valor detectado.3. Muestra una notificación al usuario por medio de la interfaz.
Flujo Alternativo:	Flujo Alternativo:
Postcondiciones: El usuario recibe una notificación.	

Tabla 16 - CU Notificación de anomalías en el sistema

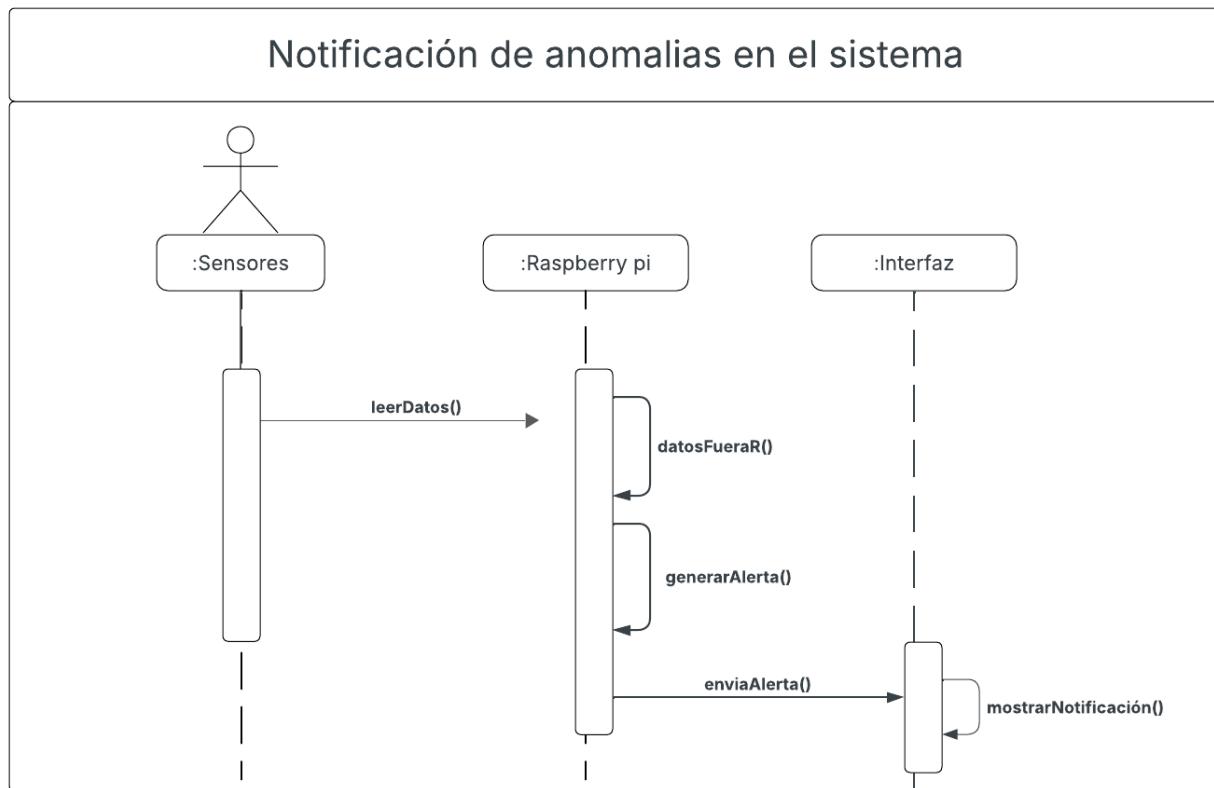


Figura 7 - DS Notificación de anomalías en el sistema

ID: 05	
Nombre del CU: Monitoreo de nutrientes del suelo	
Actor(es): Sistema y sensor NPK	
Descripción: El sistema verifica los niveles de nutrientes (NPK) del suelo, cuando los niveles de nutrientes están bajos el sistema	
Precondiciones: El sensor y el dosificador deben estar instalados y operativos, los parámetros del sensor deben estar establecidos.	
<p>Flujo Principal: Sensores</p> <ol style="list-style-type: none"> 1. El sensor NPK inicia la lectura de los parámetros. 	<p>Flujo Principal: Sistema</p> <ol style="list-style-type: none"> 2. El sistema detecta los datos obtenidos por el sensor. 3. Si el nivel detectado por el NPK está por debajo del mínimo, el sistema agrega una “alerta” para que el usuario realice acciones necesarias. 4. Una vez el sistema detecta que los valores del sensor vuelven a estar dentro del rango se desactiva la alerta.
<p>Flujo Alternativo:</p>	<p>Flujo Alternativo:</p>
Postcondiciones: Los parámetros de nutrientes del suelo vuelven a estar dentro del rango.	

Tabla 17 - CU Monitoreo de nutrientes del suelo

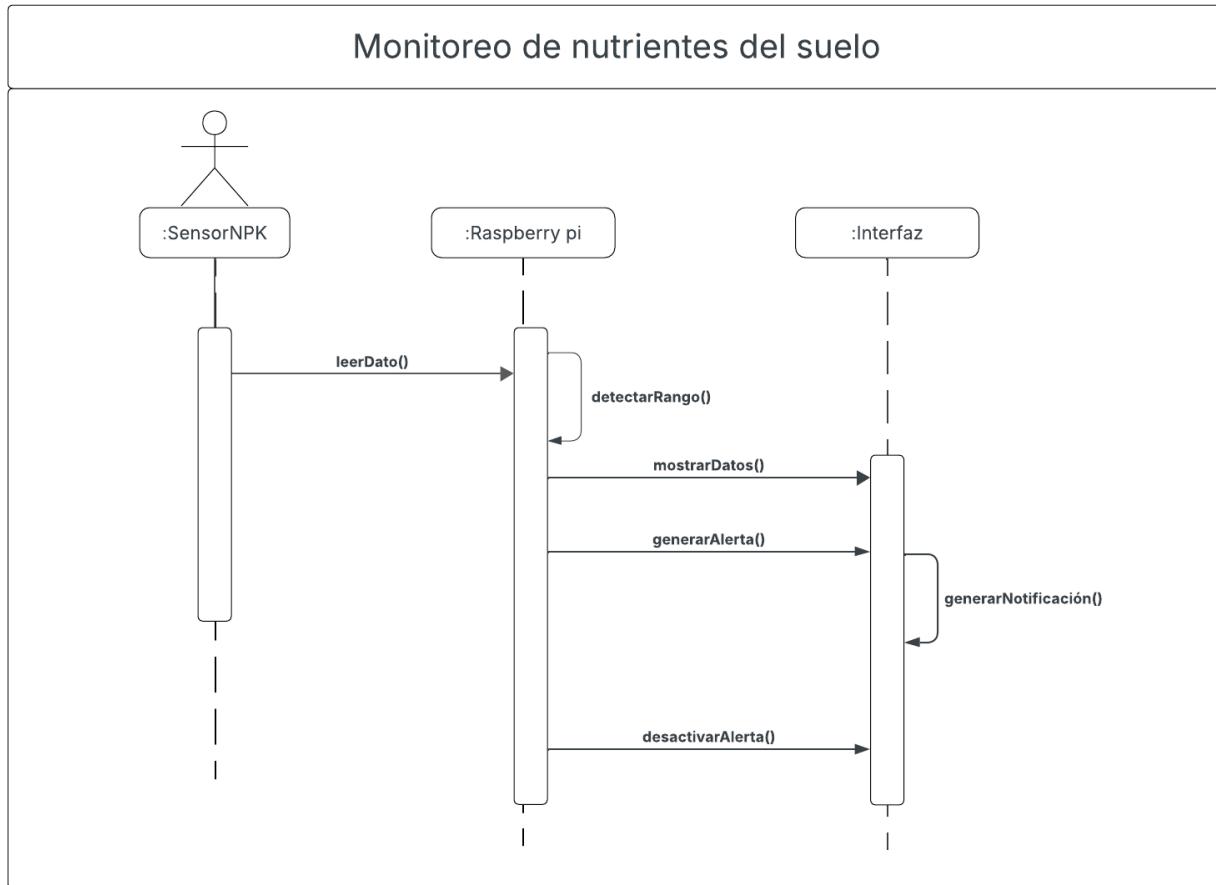


Figura 8 - DS Monitoreo de nutrientes del suelo

5.1.4 Descripción de la Arquitectura

Es necesario tener en cuenta el proceso de conexión entre el usuario final y el modelo en el cual se actuará. En este caso, se ha optado por la implementación de la Arquitectura de Cliente-Servidor, en donde:

- Cliente: Usuario, que mediante un dispositivo móvil, interactúa con el sistema, donde, además de visualizar datos y recibir notificaciones, maneja ciertas funcionalidades de manera manual.
- Servidor: Está compuesto por Raspberry Pi 4B, donde se ejecutará un software que funciona como un intermediario entre el hardware y el usuario que interactúa con este.

Para el correcto funcionamiento, tanto cliente como servidor deben estar conectados a la misma red Wi-Fi.

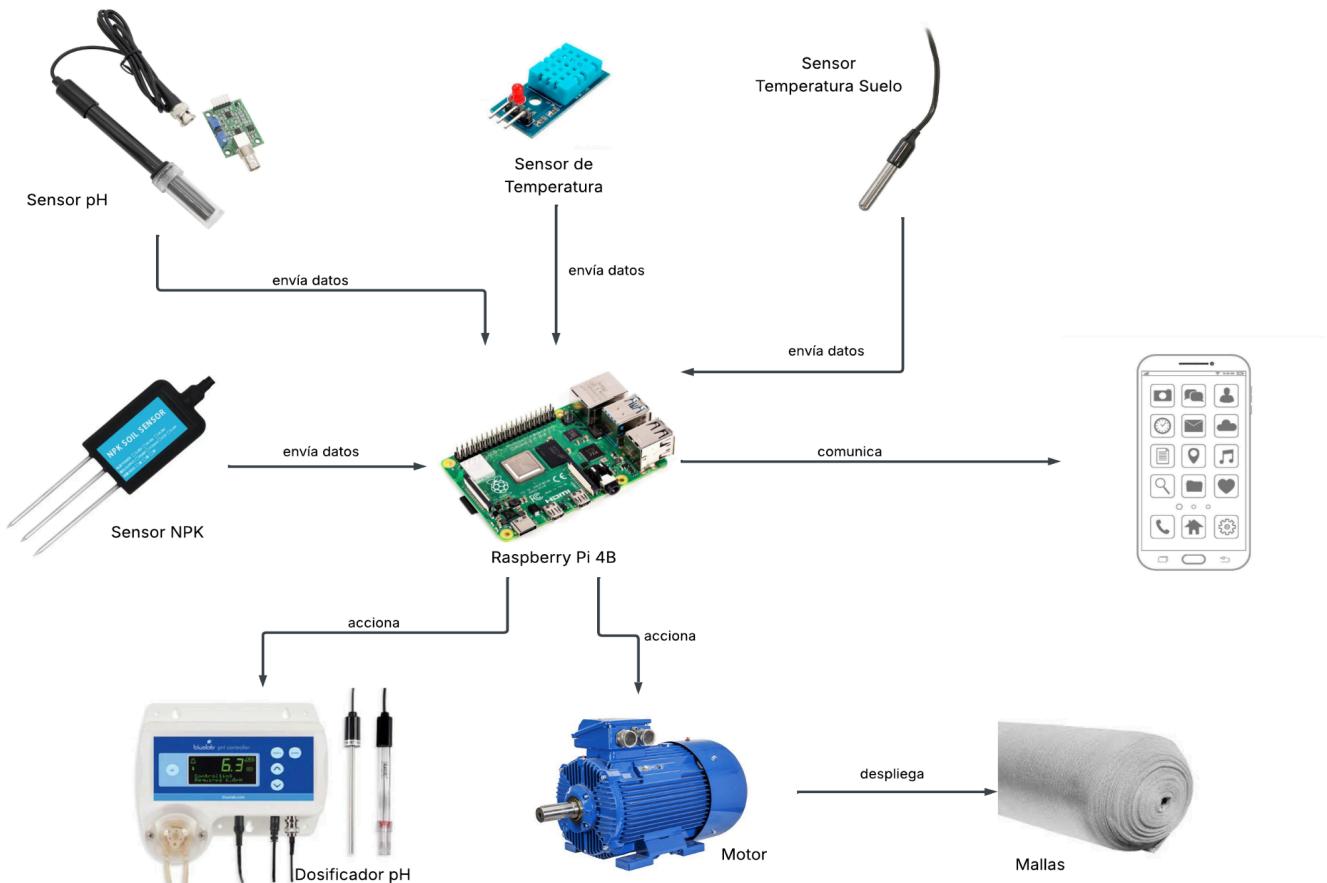


Figura 9 - Arquitectura del Sistema

- Sensor NPK: Medirá la concentración de nitrógeno (N), fósforo (P) y potasio (K) presentes en el suelo del invernadero.
- Sensor de Humedad: Medirá el nivel de humedad presente en la tierra del invernadero, permitiendo conocer si el sustrato se encuentra seco, húmedo o saturado.

- Sensor de temperatura: Medirá la temperatura ambiental dentro del invernadero, permitiendo detectar aumentos o descensos que puedan afectar el crecimiento de las plantas.
- Sensor de temperatura del suelo: Medirá la temperatura del suelo de cultivo, permitiendo detectar aumentos o descensos que puedan afectar el crecimiento de las plantas.
- Dosificador: Permite el control de pH del agua de riego, es activado cuando el usuario lo solicite en la interfaz.
- Motores: Permite el despliegue de las mallas de invernadero para controlar la temperatura ambiente del invernadero.

5.1.5 Diseño de Interface

En cuanto al diseño de la interfaz, se propuso algo sencillo, una secuencia de vistas que incluyen menús que permiten al usuario revisar los parámetros seleccionados. En caso de que se deseé realizar una acción remedial en cuanto al pH del agua o a la temperatura del ambiente, el usuario será capaz de configurar la acción y activar la función a través de los menús correspondientes.

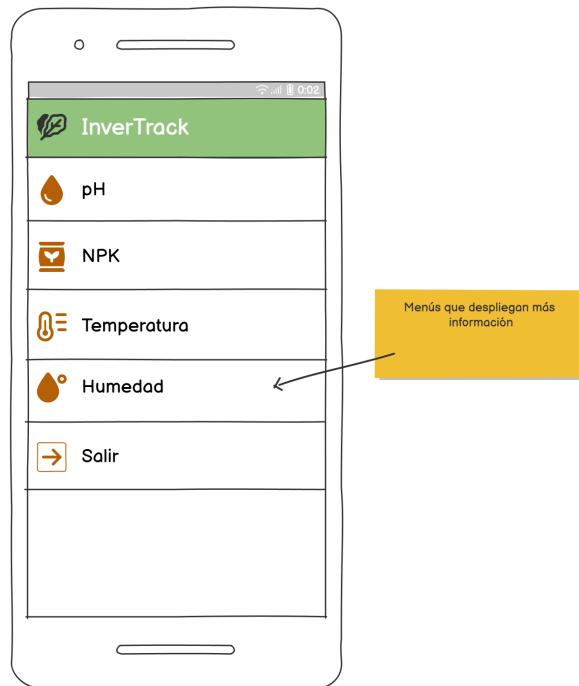


Figura 10 - Diseño menú principal

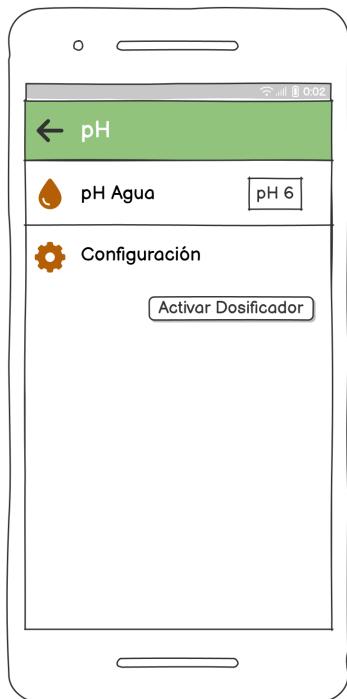


Figura 11 - Diseño menú pH

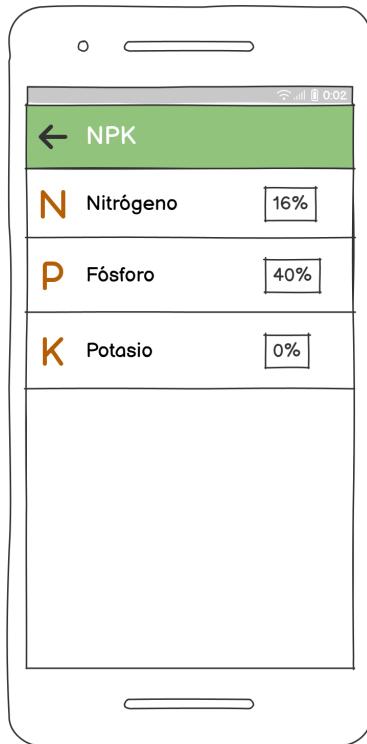


Figura 12 - Diseño menú NPK



Figura 13 - Diseño menú Temperatura

5.2 Herramientas y Técnicas

A continuación se detallan las herramientas utilizadas a lo largo del proyecto:

- Raspberry Pi 4B: Es un mini computador del tamaño de una tarjeta de crédito, de bajo costo y versátil. Funciona casi como un PC normal, pero más pequeño, de menor consumo y pensado para proyectos educativos, electrónicos y de desarrollo.
- Visual Studio Code: Editor de código fuente, diseñado para, principalmente, ser ligero, rápido y altamente personalizable, además de tener métodos y funciones detalladas con una documentación integrada, depuración integrada y tener un buen control de versiones, soporta múltiples lenguajes de programación y tiene una interfaz intuitiva y de fácil utilización.
- Python: Lenguaje de programación interpretado, orientado a objetos, y de alto nivel con una semántica dinámica. Por lo que su alto nivel construido de datos, combinado con una tacleando dinámica y unión dinámica, lo hacen muy atractivo para el desarrollo de aplicaciones rápidas, así como para conectar sus componentes existentes juntos.
- Herramientas y dispositivos:
 - Sensor NPK.
 - Sensor pH.
 - Sensor de temperatura ambiente.
 - Sensor de temperatura del suelo.
 - Dosificador de agua.
 - Motores.

6. Problemas Encontrados

A lo largo del proyecto, enfrentamos diversas dificultades, desde la instalación de SO y librerías de el raspberry pi 4 hasta la conexión con nuestra interfaz y modelo 3D en Unity.

Problema con librerías y SO

- Este problema surgió debido a la poca compatibilidad y soporte que le dan al las versiones compatibles tanto a la versión de SO y las librerías compatibles.

Solución

- Para solucionar este problema, se utilizó el SO legacy de 32 bits y para solucionar la compatibilidad con las librerías de GPIO se utilizó una versión compatible de python 3.7

Problemas de conexión

- Este problema surgió porque se desconocía cómo conectar nuestro servidor “Raspberry pi 4” con nuestra maqueta virtual para que reciba los datos de los sensores que enviaba nuestro sistema.

Solución

- Se utilizó el protocolo WebSocket en Unity para lograr la conexión con el servidor y a su vez poder conectarse con nuestra interfaz y así poder cambiar parámetros para activar los disparadores de eventos en nuestra maqueta virtual.

7.Implementación

La implementación del proyecto consiste en la integración de los distintos módulos de desarrollo cumplir los requerimientos postulados por la problemática que ha sido establecida para la realización del proyecto, realizando una especificación técnica y de códigos, hechas en el lenguaje de programación python y scripts en C en práctica en el proyecto.

Plan de Integración

El sistema implementado permite el monitoreo y control en tiempo real de parámetros ambientales dentro de un invernadero mediante sensores conectados a una Raspberry Pi y estos se comunican con la maqueta virtual en Unity y la Interfaz.

El sistema obtiene los valores y los compara con rangos previamente definidos. Cuando alguno de los parámetros se encuentra fuera de los límites establecidos, se generan notificaciones de alerta para informar la situación.

Además, el sistema integra mecanismos de control que se conecta a través de webSocket para activar eventos dentro de la maqueta , como el despliegue de una malla para reducir la temperatura ambiente y la activación de un dosificador para regular el pH del agua de riego, permitiendo mantener condiciones adecuadas para el cultivo.

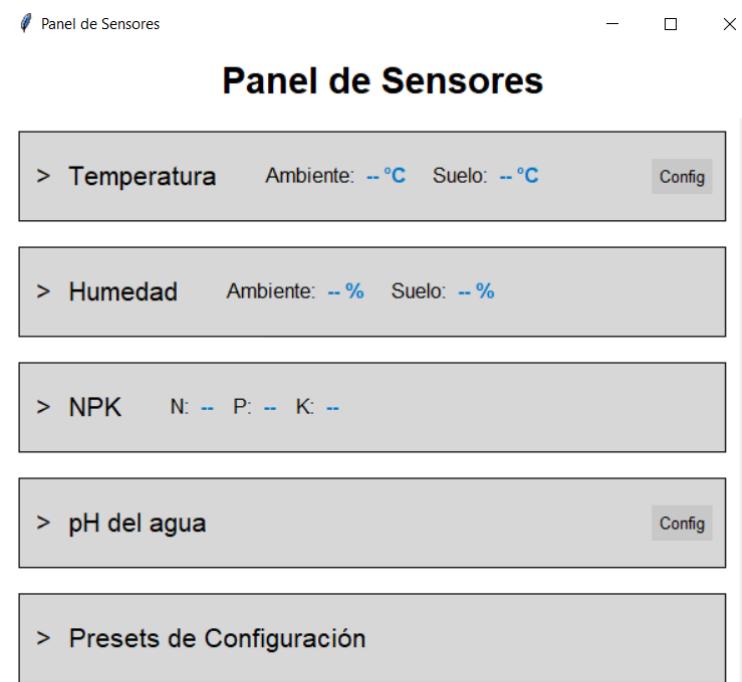
Interfaz

Este es el menú principal de la interfaz del proyecto donde se pueden observar los datos de los sensores y a su vez realizar acciones o modificar umbrales, así cumpliendo con los requerimientos de observar en tiempo real los datos obtenidos por los sensores y tomar acciones correctivas.

```

1 # =====
2 # APP PRINCIPAL      #
3 # =====
4
5 class App(tk.Tk):
6     def __init__(self):
7         super().__init__()
8         self.title("Panel de Sensores")
9         self.geometry("700x900")
10        self.configure(bg="white")
11
12        self.ws = None
13        self.ws_loop = None
14        self.ws_conectado = False
15
16
17        tk.Label(self, text="Panel de Sensores", font=("Arial", 22, "bold"), bg="white").pack(pady=10)
18
19        # ----- Scroll principal -----
20        container = tk.Frame(self, bg="white")
21        container.pack(fill="both", expand=True)
22
23        self.canvas = tk.Canvas(container, bg="white", highlightthickness=0)
24        self.v_scroll = tk.Scrollbar(container, orient="vertical", command=self.canvas.yview)
25        self.canvas.configure(yscrollcommand=self.v_scroll.set)
26
27        self.v_scroll.pack(side="right", fill="y")
28        self.canvas.pack(side="left", fill="both", expand=True)
29
30        self.content = tk.Frame(self.canvas, bg="white")
31        self.canvas_window = self.canvas.create_window((0, 0), window=self.content, anchor="nw")
32
33        def on_frame_configure(_):
34            self.canvas.configure(scrollregion=self.canvas.bbox("all"))
35
36        def on_canvas_configure(event):
37            # que el frame interno use el ancho del canvas
38            self.canvas.itemconfigure(self.canvas_window, width=event.width)
39
40            self.content.bind("<Configure>", on_frame_configure)
41            self.canvas.bind("<Configure>", on_canvas_configure)
42
43        # rueda mouse (linux/windows)
44        def _on_mousewheel(event):
45            # Windows: event.delta
46            if hasattr(event, "delta") and event.delta:
47                self.canvas.yview_scroll(int(-1 * (event.delta / 120)), "units")
48            else:
49                # Linux: Button-4/5
50                if event.num == 4:
51                    self.canvas.yview_scroll(-3, "units")
52                elif event.num == 5:
53                    self.canvas.yview_scroll(3, "units")
54
55        self.bind_all("<MouseWheel>", _on_mousewheel)
56        self.bind_all("<Button-4>", _on_mousewheel)
57        self.bind_all("<Button-5>", _on_mousewheel)
58
59        # ----- Acordeones -----
60        self.temp = AcordeonTemp(self.content, self)
61        self.temp.pack(fill="x", padx=10, pady=10)
62
63        self.hum = AcordeonHumedad(self.content)
64        self.hum.pack(fill="x", padx=10, pady=10)
65
66        self.npk = AcordeonNPK(self.content)
67        self.npk.pack(fill="x", padx=10, pady=10)
68
69        self.ph = AcordeonPH(self.content, self)
70        self.ph.pack(fill="x", padx=10, pady=10)
71
72        self.presets = AcordeonPresets(self.content, self)
73        self.presets.pack(fill="x", padx=10, pady=10)
74
75        start_websocket_client(self)
76
77    def update_from_ws(self, temp, hum, ph_raw):
78        if temp is not None:
79            self.temp.actualizar(temp, temp)
80
81        if hum is not None:
82            self.hum.actualizar(hum, hum)
83
84        if ph_raw is not None:
85            self.ph.actualizar(ph_raw)

```



Modificar Umbral

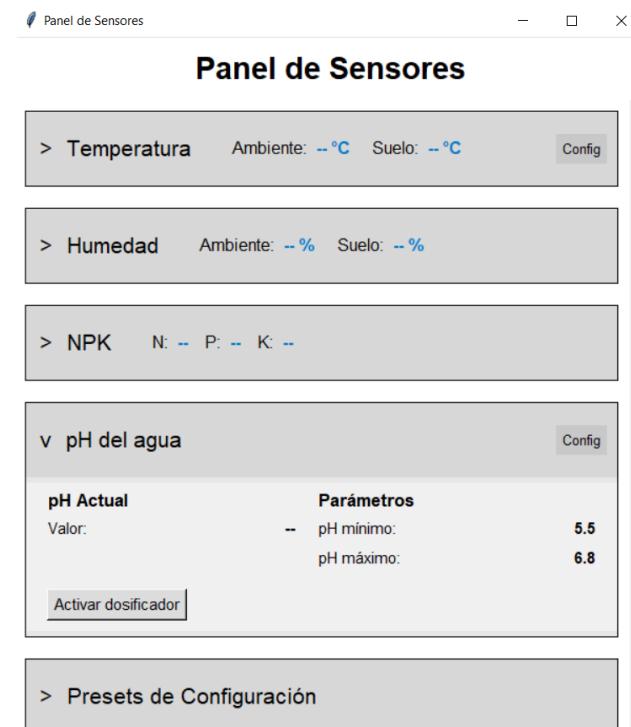
En general la interfaz se puede cambiar el umbral de un sensor dependiendo de las necesidades que requiera el usuario y esto aplica para todos los sensores y crear una configuración personalizada del umbral para una determinada plantación.

Además de poder realizar acciones como por ejemplo controlar el PH del agua como sería el caso de activar el dosificador para controlar la salinidad y alcalinidad del agua de riego.

```

1  def abrir_config(self):
2      win = tk.Toplevel(self)
3      win.title("Configurar pH")
4      win.geometry("300x200")
5
6      tk.Label(win, text="pH mínimo:", font=("Arial", 11)).pack(pady=5)
7      ent_min = tk.Entry(win)
8      ent_min.insert(0, str(self.ph_min_cfg))
9      ent_min.pack()
10
11     tk.Label(win, text="pH máximo:", font=("Arial", 11)).pack(pady=5)
12     ent_max = tk.Entry(win)
13     ent_max.insert(0, str(self.ph_max_cfg))
14     ent_max.pack()
15
16     def guardar():
17         try:
18             self.ph_min_cfg = float(ent_min.get())
19             self.ph_max_cfg = float(ent_max.get())
20             self.actualizar_parametros()
21             enviar_parametros(self.app)
22             win.destroy()
23         except ValueError:
24             messagebox.showerror("Error", "Valores inválidos")
25
26     tk.Button(win, text="Guardar", command=guardar).pack(pady=10)

```



```

1  def abrir_config(self):
2      win = tk.Toplevel(self)
3      win.title("Configurar pH")
4      win.geometry("300x200")
5
6      tk.Label(win, text="pH mínimo:", font=("Arial", 11)).pack(pady=5)
7      ent_min = tk.Entry(win)
8      ent_min.insert(0, str(self.ph_min_cfg))
9      ent_min.pack()
10
11     tk.Label(win, text="pH máximo:", font=("Arial", 11)).pack(pady=5)
12     ent_max = tk.Entry(win)
13     ent_max.insert(0, str(self.ph_max_cfg))
14     ent_max.pack()
15
16     def guardar():
17         try:
18             self.ph_min_cfg = float(ent_min.get())
19             self.ph_max_cfg = float(ent_max.get())
20             self.actualizar_parametros()
21             enviar_parametros(self.app)
22             win.destroy()
23         except ValueError:
24             messagebox.showerror("Error", "Valores inválidos")
25
26     tk.Button(win, text="Guardar", command=guardar).pack(pady=10)

```

8. Conclusión

El desarrollo del proyecto de monitoreo y control de un invernadero ha demostrado ser una iniciativa prometedora que busca optimizar el cuidado de cultivos mediante el uso de tecnología.

El uso de sensores para monitorizar condiciones ambientales resulta fundamental para garantizar un crecimiento saludable de los cultivos. La implementación de sistemas automatizados de control de temperatura y pH permite una respuesta inmediata a las variaciones en el entorno, mejorando así la eficiencia y la productividad del invernadero.

Se ha priorizado un diseño de interfaz amigable, lo que permitirá a los usuarios interactuar con el sistema de manera efectiva, incluso sin experiencia técnica previa. Esta accesibilidad es fundamental para fomentar el uso para todo tipo de usuario.

Finalmente, el proyecto no solo busca proporcionar una herramienta efectiva para el monitoreo y control de invernaderos, sino que también tiene el potencial de mejorar la calidad de vida de los agricultores al facilitarles el cuidado de sus cultivos. La combinación de tecnología y diseño centrado en el usuario sienta las bases para un futuro más sostenible en la agricultura.

9.Trabajo Futuro

Integración de Sensor NPK

- Incorporar un sensor NPK que permita medir los niveles de nitrógeno “N”, fósforo “P” y potasio “K” en el suelo.

Actualmente el sistema y la interfaz se encuentran diseñados para registrar y visualizar datos de NPK; sin embargo, debido a la limitación de hardware disponible que se nos presentó el sensor utilizado solo permitió medir temperatura y humedad del suelo debido a la versión de este sensor **NPK RS485**, impidiendo la obtención total de valores de nutrientes.

Integrado el sensor NPK adecuado, el sistema podrá habilitar el control y monitoreo completo de nutrientes, mejorando la toma de decisiones agronómicas.

Optimización Energética

- Incorporar fuentes de energía renovable, como paneles solares, para garantizar el funcionamiento continuo del sistema.

10. Referencias

1. «Raspberry Pi OS – Raspberry Pi», Raspberry Pi. <https://www.raspberrypi.com>
2. «Raspberry Pi OS Documentation» <https://www.raspberrypi.com/documentation/>
3. «Lucidchart - Diagramas» <https://www.lucidchart.com>