**“Desarrollo de un prototipo de un Invernadero Automático con Arduino en la Comunidad de Chihuaco - Sicuani, 2025 ”**

; ; ;

**1 Escuela Académica Profesional de Ingeniería de Sistemas e Informática, Universidad Continental, Cusco, Perú.**

**RESUMEN EJECUTIVO**

El proyecto “Desarrollo de un prototipo de un Invernadero Automático con Arduino en la Comunidad de Chihuaco – Sicuani, 2025” tiene como objetivo proporcionar una solución simple y accesible para mejorar la gestión de la temperatura y la humedad en los cultivos de esta zona altoandina. En comunidades rurales del sur del Perú, como Chihuaco, el clima impredecible (con heladas y sequías frecuentes) afecta el desarrollo de las plantas y la estabilidad de las cosechas. Por ello, las políticas del Ministerio de Desarrollo Agrario y Riego (MIDAGRI) se enfocan en soluciones que mejoren la eficiencia del uso del agua, promoviendo el riego tecnificado como una estrategia para combatir estos problemas climáticos [1].

Para hacer frente a esta situación, se diseñó un prototipo de invernadero automático basado en el microcontrolador Arduino UNO, que incorpora un sensor de humedad FC-28 para el suelo, un sensor DHT11 para la medición de temperatura y humedad ambiental, una bomba de agua, luces LED y un módulo relay 5VDC. Este sistema automatiza el riego y la iluminación, activándose solo cuando se detectan condiciones específicas, lo que reduce la intervención manual y mejora la estabilidad del entorno de cultivo. Este enfoque es similar a otros proyectos de automatización agrícola que utilizan tecnologías accesibles como Arduino para optimizar la producción en pequeñas comunidades rurales [2].

Durante el proceso de desarrollo del prototipo, se realizaron diversas pruebas, tanto en condiciones controladas como en campo, para verificar que los sensores y actuadores respondieron de manera adecuada a las fluctuaciones de humedad y temperatura. El prototipo demuestra que es posible aplicar tecnologías de bajo costo para fortalecer la agricultura familiar y mejorar la seguridad alimentaria, alineándose con los ODS 2 (Hambre Cero) y ODS 9 (Industria, Innovación e Infraestructura), que buscan promover prácticas agrícolas sostenibles y el acceso a tecnologías accesibles para comunidades rurales [3].

**Palabras clave:** invernadero automatizado, Arduino, agricultura sostenible.

1. **INTRODUCCIÓN**

**1.1 ANTECEDENTES**

**Antecedentes internacionales**

El uso de invernaderos automatizados ha ido ganando terreno como una solución clave para mejorar la producción agrícola en regiones con climas extremos. En países como España, México y Colombia, se han implementado sistemas de monitoreo de temperatura y humedad utilizando microcontroladores como Arduino. Esta tecnología resulta atractiva debido a su bajo costo y su facilidad de implementación. La FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) ha destacado que la automatización de pequeños cultivos no solo ayuda a garantizar la estabilidad de la producción, sino que también reduce la dependencia de factores climáticos impredecibles [4].

**Antecedentes nacionales – Perú**

A nivel nacional, varias experiencias han mostrado el impacto positivo de la automatización agrícola, especialmente en zonas rurales. El Ministerio de Desarrollo Agrario y Riego (MIDAGRI) ha impulsado varios proyectos de riego tecnificado y control ambiental en regiones como Arequipa, Ica y Cusco. Según MIDAGRI, el uso de sensores simples permite mejorar la gestión del agua y proteger los cultivos frente a heladas y sequías. Además, algunas universidades peruanas, como la UNSA, la UNI y la UNSAAC, han desarrollado prototipos de invernaderos automatizados utilizando Arduino y otros sensores básicos, obteniendo buenos resultados en cultivos de hortalizas y plantas ornamentales [5].

**Antecedentes locales – Sicuani, Cusco**

En la provincia de Canchis, donde se encuentra la comunidad de Chihuaco, la agricultura familiar es una de las actividades más importantes. Sin embargo, las variaciones de temperatura y la falta de sistemas de riego controlado afectan el desarrollo de los cultivos. De acuerdo con el Plan de Desarrollo Agrario del Gobierno Regional del Cusco, muchas comunidades altoandinas enfrentan dificultades para mantener cultivos estables debido a las bajas temperaturas nocturnas y los largos períodos sin lluvia. A pesar de estos desafíos, no se han implementado iniciativas locales que utilicen tecnologías accesibles, como Arduino, para automatizar los procesos de cultivo. Esto representa una gran oportunidad para explorar nuevas soluciones tecnológicas que puedan mejorar la eficiencia y la sostenibilidad de la agricultura en estas comunidades

**1.2 Identificación y formulación del problema:**

En la comunidad de Chihuaco, los cultivos dependen completamente del clima, lo que significa que cualquier variación brusca como heladas, días muy secos o cambios inesperados de temperatura puede afectar la producción. Además, el riego suele realizarse de manera manual, lo que ocasiona exceso o escasez de agua según la disponibilidad de las familias. Esta situación genera dificultades para obtener plantas uniformes y saludables durante todo el año.

**Problema central:**¿Cómo implementar un prototipo de invernadero automático de bajo costo que controle de manera adecuada la humedad del suelo y la temperatura ambiental en la comunidad de Chihuaco, Sicuani?

**ODS que responde:**

* **ODS 2 – Hambre Cero**
* **ODS 9 – Industria, Innovación e Infraestructura**

**1.2 MARCO TEÓRICO**

Los invernaderos pueden regular la temperatura, calor o humedad reduciendo así la dependencia que se tiene del clima externo y aumentando la probabilidad de obtener cosechas más estables. En ciudades altoandinas como Sicuani donde las heladas y sequías suelen ser frecuentes, los invernaderos son una alternativa ya que protegen a las plantas en una etapa temprana evitando su pérdida y además mejoran la productividad.

Dentro de la agricultura protegida los invernaderos automatizados se han convertido en una tendencia a nivel mundial ya que se busca optimizar recursos valiosos como el agua y asegurar condiciones climáticas que sean consistentes. La automatización permite activar actuadores como bombas de agua, ventiladores entre otros , respecto a los datos obtenidos de sensores sin la necesidad de la intervención humana.

**Automatización agrícola y sistemas embebidos**

La automatización agrícola es el uso que se le da a tecnologías digitales como son los actuadores, sensores o microcontroladores para hacer tareas de cuidado de cultivos de forma automática o semiautomática.

Tal como menciona MIDAGRI(2022), los sirenas empleados que sean automatizados mejorarán la eficiencia de riego, reducir pérdidas por factores de clima externo, y aumentar también la resiliencia de la agricultura familiar.

Por otro lado, un sistema embebido se define como un conjunto de hardware y software que es desarrollado para cumplir una tarea en específico dentro de un equipo.

Algunas características de los sistemas embebidos son que requieren un consumo bajo de energía, y respuestas en tiempo real lo que hace viable su implementación. Este tipo de sistemas es ideal para aplicaciones en zonas rurales donde se requiere que el dispositivo funcione sin la necesidad de intervención humana constante, reparación fácil y sobre todo que sea económico de mantener.

**Microcontroladores**

Un microcontrolador es un circuito integrado que está confirmado por un procesador, memoria y periféricos configurables para controlar diversos dispositivos electrónicos. El microcontrolador Arduino UNO es bastante usado en prototipos ya que ofrece compatibilidad con sensores de bajo costo además de ser fácil de programar.

**Sensores aplicados**

1. **Sensor de humedad**

El sensor FC-28 mide la conductividad eléctrica que hay en el suelo por lo cual el valor de este sube cuando existe mayor humedad. Este sensor trabaja con valores analógicos lo que permite saber si el suelo está seco, húmedo o en condiciones normales. Una característica de este sensor es que requiere calibración del umbral según el tipo del suelo con el que se trabajara.

Algunas desventajas de este sensor es que puede haber corrosión en las puntas metálicas con el paso del tiempo. También puede ser sensible de acuerdo al tipo del suelo.

1. **Sensor de temperatura**

El sensor DHT11 es un sensor básico que puede medir la temperatura y humedad relativa. Este sensor suele ser usado en proyectos educativos y prototipos a escala. Este sensor es muy importante ya que permite monitorear micro climas internos además de identificar tendencias de evaporación y requerimientos de riego.

**Actuadores aplicados**

1. **Modulo relay**

Un modulo relay actúa como un interruptor electrónico que es controlado por señales del Arduino. Este permite activar dispositivos que requieren un mayor valor de voltaje como la bomba de agua o luces led.

1. **Bomba de agua**

Una bomba de agua impulsa el agua hacia un punto en específico. En este contexto la bomba de agua impulsa el flujo de agua hacia el cultivo cuando el sensor existente indica que el suelo está seco. La automatización de la bomba es muy beneficiosa ya que disminuye la intervención manual y evita riesgos excesivos que pueden ser perjudiciales para el cultivo y generar mayores gastos.

1. **Luces LED**

Las luces LED se caracterizan por tener un consumo energético relativamente bajo. Está característica lo convierte en una solución viable sobre todo en zonas rurales.

**1.3 OBJETIVOS DEL PROYECTO**

**OBJETIVO GENERAL:**

Desarrollar un prototipo de invernadero automático con Arduino que permite regular la humedad del suelo y monitorear la temperatura ambiental en la comunidad de Chihuaco.

**OBJETIVOS ESPECÍFICOS:**

1. Implementar un sistema de riego automático que se active cuando el suelo esté seco.
2. Automatizar la medición de la temperatura y humedad ambiental dentro del invernadero.
3. Evaluar el funcionamiento general del prototipo en pruebas de campo.
4. **CONOCIMIENTOS DE INGENIERÍA APLICADOS / RELACIONADOS**

* **Conocimiento en Matemáticas:** Para desarrollar el prototipo fue necesario aplicar algunos conceptos básicos de matemáticas, sobre todo en la parte donde se interpretan los valores que envían los sensores. En el caso del sensor FC-28, que mide la humedad del suelo, se trabajó con rangos y escalas para convertir sus lecturas en valores que realmente indiquen si la tierra está seca o húmeda. Estos cálculos permitieron establecer los puntos de activación del riego y asegurarnos de que el sistema responda correctamente cuando la humedad baja de lo esperado.
* **Conocimiento en ciencias naturales:** Durante el proyecto también se aplicaron conocimientos de ciencias naturales, especialmente relacionados con cómo se comporta la humedad y la temperatura dentro de un invernadero. Se tomó en cuenta que, cuando la temperatura sube, el suelo tiende a perder agua más rápido, algo que influye directamente en el riego. Este entendimiento fue importante para ajustar el funcionamiento del sensor DHT11 y para definir en qué momentos la bomba debía activarse. En general, se consideran principios básicos de física y biología para comprender mejor cómo afecta el ambiente al crecimiento de las plantas.
* **Conocimiento en Ingeniería:** En la parte de ingeniería, se utilizaron conocimientos de electrónica básica, programación y diseño de sistemas. Fue necesario entender cómo conectar correctamente los componentes como el Arduino, el módulo relay, la bomba de agua y los sensores, evitando errores en el circuito. Además, en la programación se usó el lenguaje C++ propio del Arduino para configurar el comportamiento del sistema, leer los datos de los sensores y activar los actuadores cuando fuese necesario. Finalmente, se aplicaron principios de diseño de prototipos para integrar todos los elementos de manera ordenada y funcional.

1. **INGENIERO Y LA SOCIEDAD:**

* **Justificación social:** Este proyecto tiene un valor importante para la comunidad de Chihuaco, porque permite que las familias tengan una herramienta que les facilite el cuidado de sus cultivos. En esta zona, muchas actividades agrícolas dependen del clima, y cuando la tierra se seca demasiado o bajan las temperaturas, las plantas suelen dañarse. Con el invernadero automático, se busca que el riego sea más constante y que las condiciones dentro del cultivo se mantengan mejor controladas. Al final, la idea es que las personas puedan tener una producción más estable sin tener que estar pendientes todo el día de sus plantas.
* **Justificación Económica:** Otra razón por la que este proyecto es útil es porque los materiales que se emplean son económicos y fáciles de conseguir. Al tratarse de un prototipo hecho con Arduino y sensores básicos, no se requiere una gran inversión para construirlo o replicarlo. Además, al automatizar el riego se usa solo el agua necesaria, lo que ayuda a reducir gastos y a aprovechar mejor los recursos. Esto lo convierte en una alternativa accesible para familias que desean mejorar su producción sin gastar demasiado.
* **Justificación ambiental:** Desde el punto de vista ambiental, el proyecto también aporta beneficios. El riego automático permite evitar el uso excesivo de agua y favorece un manejo más responsable de este recurso, que en zonas altoandinas no siempre está disponible en abundancia. Además, al mantener condiciones más adecuadas dentro del invernadero, las plantas pueden crecer de manera más saludable sin necesidad de equipos de alto consumo energético. En general, se trata de una solución simple, pero alineada con prácticas de agricultura sostenible.
* **Acontecimientos tecnológicos y científicos:** En los últimos años, el uso de Arduino y sensores económicos se ha vuelto bastante común en proyectos de automatización, especialmente en educación y agricultura. Muchas universidades y organizaciones han demostrado que es posible mejorar pequeños cultivos utilizando microcontroladores y sistemas de riego automático. Estos avances fueron parte de la motivación para realizar este prototipo, ya que muestran que la tecnología no tiene que ser costosa para ser útil. Gracias a estos desarrollos, hoy es más accesible implementar soluciones de bajo costo que realmente ayuden a las comunidades rurales.

1. **METODOLOGÍA EMPLEADA**

Para desarrollar el prototipo del invernadero automático se trabajó con una metodología de tipo Ágil, tomando como base algunas ideas de Scrum, pero adaptándolas a la realidad del curso y al tiempo que teníamos disponible. No se llegó a trabajar con sprints formales, porque contábamos con una fecha de entrega fija, así que organizamos el proyecto por etapas más que por iteraciones.

* Para la gestión del proyecto, tomamos como referencia el enfoque del PMI y la guía PMBOK, sobre todo en la parte de planificación y seguimiento de las actividades. Esto nos ayudó a ordenar las tareas, definir responsables y organizar el trabajo de forma más clara.
* Para el desarrollo del prototipo, aplicamos el enfoque ágil de Scrum, pero simplificado: identificamos el producto mínimo viable (PMV), priorizamos funcionalidades y fuimos validando el avance en cada etapa.
* El trabajo se realizó mediante:
  + Análisis de requerimientos en esta etapa nos dedicamos a entender qué necesitaba realmente la comunidad de Chihuaco y qué debía hacer el invernadero automático como mínimo. A partir de ello, definimos las funciones principales del prototipo (medir humedad, activar el riego, monitorear temperatura, etc.). De esta fase salió como entregable una lista de requerimientos y el PMV del sistema.
  + Elaboración de prototipos con los requerimientos ya definidos, armamos un pequeño Product Backlog donde listamos todas las tareas necesarias: conexión de sensores, programación del Arduino, pruebas, armado físico, entre otros. Aunque no trabajamos con sprints formales, sí organizamos el trabajo por semanas, como si fueran mini-sprints, priorizando las tareas más importantes primero.
  + Implementación de la solución en esta fase integramos todos los componentes: Arduino UNO, sensores, módulo relay, bomba de agua y luces LED. También se construyó la estructura del invernadero a pequeña escala para poder probar el sistema en un entorno similar al real. Aquí se terminó de ajustar el código y se organizaron los cables y conexiones para que el prototipo sea más ordenado y seguro.
  + Pruebas de calidad de software finalmente, realizamos pruebas de funcionamiento del prototipo, verificando que el riego se active cuando el suelo está seco y que las lecturas de temperatura y humedad sean coherentes. Aunque no se aplicaron pruebas de software muy avanzadas, sí se hicieron pruebas repetidas para comprobar la estabilidad del sistema y su respuesta ante cambios en el ambiente.

**APORTES /DESCUBRIMIENTOS:**

A lo largo del desarrollo del prototipo se obtuvieron varios aprendizajes y resultados que consideramos como aportes importantes del proyecto:

En primer lugar, descubrimos que el sensor de humedad FC-28 necesita una buena calibración inicial para ser realmente útil. Al inicio, sus lecturas varían bastante, pero al hacer varias pruebas con distintos niveles de humedad y ajustar los umbrales en el código, logramos que el sensor diera valores más consistentes. Este proceso nos enseñó que no basta con conectar el sensor, sino que es necesario adaptarlo a las condiciones reales del suelo que se va a trabajar.

En segundo lugar, comprobamos que el sensor DHT11, aunque es básico, resulta suficiente para un invernadero de pequeño tamaño como el que se propone en este proyecto. Si bien no es el sensor más preciso del mercado, para fines académicos y de prototipo cumple bien su función de mostrar la tendencia de los cambios de temperatura y humedad ambiental. Esto refuerza la idea de que se pueden hacer proyectos útiles sin necesidad de componentes muy costosos.

Otro aporte importante fue darnos cuenta de que la automatización del riego sí hace una diferencia en el uso del agua y en el cuidado del cultivo. Al observar el funcionamiento del sistema, notamos que la bomba solo se encendía cuando el sensor detectaba baja humedad, lo que evita regar “por costumbre” o “por si acaso”. Este comportamiento demuestra que el sistema puede ayudar a que las familias utilicen el agua de una forma más eficiente.

Finalmente, adoptamos el enfoque del proyecto para que no solo sea un trabajo de curso, sino una propuesta que podría ser replicada y mejorada en la comunidad de Chihuaco. El diseño del prototipo y la selección de componentes se hizo pensando en que otras personas, con una guía básica, puedan reconstruir el sistema sin depender de equipos muy especializados. Este enfoque de accesibilidad también es uno de los aportes más importantes del proyecto.

**4.1 SPRINT’S**

**4.1. Introducción**

Este documento detalla el Sprint Backlog para el equipo de desarrollo del proyecto Invernadero Automatizado con Arduino en la comunidad de Chihuaco. Se desglosan todas las historias de usuario seleccionadas del backlog del producto, sus tareas específicas, la estimación de esfuerzo en horas y el responsable asignado

**4.2. Criterios de Estimación de Esfuerzo**

Para estimar el esfuerzo de cada tarea, se utiliza la técnica de Planning Poker, donde el equipo discute cada tarea y asigna una cantidad de horas basada en la complejidad técnica, dependencias con otras tareas, experiencia previa y posibles riesgos. Cada tarea se estima en consenso utilizando la escala de horas-persona.

**4.3. Historias de Usuario Seleccionadas para el Sprint**

**4.3.1.1 Historia de Usuario 1.1: Visualización en tiempo real**

Descripción: Como agricultor, quiero ver en pantalla la humedad del suelo en tiempo real, para tomar decisiones inmediatas sobre el riego.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 1.1.1 | Investigar sensores de humedad compatibles con Arduino | Pool | 4 | Terminado |
| 1.1.2 | Conectar y programar sensor de humedad | Victor | 6 | Terminado |
| 1.1.3 | Diseñar la interfaz de usuario para mostrar la lectura de humedad | Cristian | 5 | Terminado |
| 1.1.4 | Integrar la lectura del sensor a la interfaz | Erick | 6 | Terminado |
| 1.1.5 | Probar la visualización de datos en tiempo real | Victor | 4 | Terminado |
| 1.1.6 | Implementar alerta en caso de fallo del sensor | Erick | 3 | Terminado |

*Tabla 1. Historia de Usuario 1.1: Visualización en tiempo real*

**4.3.1.2 Historia de Usuario 1.2: Registro histórico**

Descripción: Como agricultor, quiero que el sistema guarde un historial de humedad, para analizar el comportamiento del suelo

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 1.2.1 | Configurar base de datos local para almacenar lecturas diarias | Pool | 8 | Terminado |
| 1.2.2 | Crear la lógica de la base de datos para registrar lecturas | Victor | 8 | Terminado |
| 1.2.3 | Desarrollar la interfaz para consultar y visualizar el historial | Cristian | 6 | Terminado |
| 1.2.4 | Implementar la funcionalidad de guardar y mostrar el historial | Erick | 8 | Terminado |
| 1.2.5 | Realizar pruebas de base de datos y consultas | Victor | 6 | Terminado |

*Tabla 2. Historia de Usuario 1.2: Registro histórico*

**4.3.1.3 Historia de Usuario 1.3: Umbrales personalizables**

Descripción: Como agricultor, quiero definir el umbral mínimo de humedad, para que el sistema se adapte a diferentes cultivos.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 1.3.1 | Diseñar la interfaz de configuración de umbrales | Cristian | 5 | Terminado |
| 1.3.2 | Implementar la lógica para guardar los umbrales definidos | Erick | 6 | Terminado |
| 1.3.3 | Asegurar que los cambios se apliquen de inmediato | Erick | 4 | Terminado |
| 1.3.4 | Realizar pruebas de validación con distintos cultivos simulados | Victor | 4 | Terminado |

*Tabla 3. Historia de Usuario 1.3: Umbrales personalizables*

**4.3.2.1 Historia de Usuario 2.1: Riego automático por humedad**

Descripción: Como agricultor, quiero que el riego se active sólo cuando la humedad esté baja, para ahorrar agua.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 2.1.1 | Investigar y adquirir la bomba de riego | Pool | 5 | Terminado |
| 2.1.2 | Conectar la bomba de riego al Arduino y sensor de humedad | Victor | 8 | Terminado |
| 2.1.3 | Programar el código para activar/desactivar la bomba según el umbral | Erick | 10 | Terminado |
| 2.1.4 | Realizar pruebas de integración en campo con un tanque real | Victor | 8 | Terminado |

*Tabla 4. Historia de Usuario 2.1: Riego automático por humedad*

**4.3.2.2 Historia de Usuario 2.2: Programación horaria**

Descripción: Como agricultor, quiero programar horarios de riego para complementar el modo automático.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 2.2.1 | Diseñar la interfaz para la programación de horarios | Cristian | 6 | Terminado |
| 2.2.2 | Implementar la lógica de programación horaria | Erick | 8 | Terminado |
| 2.2.3 | Asegurar que el riego manual no interfiera con el automático | Erick | 5 | Terminado |
| 2.2.4 | Realizar pruebas de programación correctas | Victor | 5 | Terminado |

*Tabla 5. Historia de Usuario 2.2: Programación horaria*

**4.3.2.3 Historia de Usuario 2.3: Opción manual de emergencia**

Descripción: Como agricultor, quiero poder activar la bomba manualmente, para casos de emergencia.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 2.3.1 | Seleccionar y adquirir el botón físico para la activación manual | Pool | 3 | Terminado |
| 2.3.2 | Conectar el botón físico y programar la lógica de activación manual | Victor | 5 | Terminado |
| 2.3.3 | Probar la prioridad de la acción manual sobre la automática | Erick | 4 | Terminado |

*Tabla 6. Historia de Usuario 2.3: Opción manual de emergencia*

**4.3.3.1 Historia de Usuario 3.1: Lectura en tiempo real(Temperatura)**

Descripción: Como agricultor, quiero ver la temperatura en pantalla, para controlar el ambiente del invernadero.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 3.1.1 | Investigar y adquirir el sensor de temperatura | Pool | 4 | Terminado |
| 3.1.2 | Conectar y programar el sensor de temperatura | Victor | 6 | Terminado |
| 3.1.3 | Diseñar la interfaz para la visualización de la temperatura | Cristian | 5 | Terminado |
| 3.1.4 | Integrar la lectura del sensor a la interfaz | Erick | 6 | Terminado |
| 3.1.5 | Probar las lecturas con un termómetro físico de referencia | Victor | 4 | Terminado |

*Tabla 7. Historia de Usuario 3.1: Lectura en tiempo real(Temperatura)*

**4.3.3.2 Historia de Usuario 3.2: Alerta de temperatura crítica**

Descripción: Como agricultor, quiero recibir una alerta cuando la temperatura salga del rango óptimo, para actuar rápido.

**4.3.3.3 Historia de Usuario 3.3: Historial de temperatura**

Descripción: Como agricultor, quiero que el sistema guarde los registros de temperatura, para planificar la producción.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 3.2.1 | Configurar la base de datos local para almacenar registros de temperatura | Pool | 8 | Terminado |
| 3.2.2 | Programar el registro de datos de temperatura | Victor | 8 | Terminado |
| 3.2.3 | Diseñar la interfaz para consultar el historial de temperatura | Cristian | 6 | Terminado |
| 3.2.4 | Realizar pruebas del historial con datos de diferentes días de cultivo | Victor | 6 | Terminado |

*Tabla 8. Historia de Usuario 3.3: Historial de temperatura*

**4.3.4.1 Historia de Usuario 4.1: Ventilación automática**

Descripción: Como agricultor, quiero que los ventiladores se activan cuando la temperatura suba demasiado, para proteger los cultivos.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 4.1.1 | Investigar y adquirir los ventiladores | Pool | 5 | Terminado |
| 4.1.2 | Conectar los ventiladores al sistema Arduino | Victor | 8 | Terminado |
| 4.1.3 | Programar la lógica para activar y desactivar los ventiladores | Erick | 10 | Terminado |
| 4.1.4 | Realizar pruebas de campo con los ventiladores | Victor | 8 | Terminado |

*Tabla 9. Historia de Usuario 4.1: Ventilación automática*

**4.3.4.2 Historia de Usuario 4.2: Calefacción automática**

Descripción: Como agricultor, quiero que los calefactores se activen cuando haga frío, para evitar daños a las plantas.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 4.2.1 | Investigar y adquirir los calefactores | Pool | 5 | Terminado |
| 4.2.2 | Conectar los calefactores al sistema Arduino | Victor | 8 | Terminado |
| 4.2.3 | Programar la lógica para activar y desactivar los calefactores | Erick | 10 | Terminado |
| 4.2.4 | Realizar pruebas en ambiente controlado | Victor | 8 | Terminado |

*Tabla 10. Historia de Usuario 4.2: Calefacción automática*

**4.3.5.1 Historia de Usuario 5.1: Operación 24/7**

Descripción: Como agricultor, quiero que el sistema funcione continuamente, para no perder control de cultivos.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 5.1.1 | Implementar un sistema de reinicio automático en caso de fallos menores | Erick | 10 | Terminado |
| 5.1.2 | Realizar pruebas de laboratorio simulando fallos | Victor | 8 | Terminado |

*Tabla 11. Historia de Usuario 5.1: Operación 24/7*

**4.3.5.2 Historia de Usuario 5.2: Respaldo de energía**

Descripción: Como agricultor, quiero que el sistema funcione con batería de respaldo, para evitar interrupciones en caso de corte eléctrico.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 5.2.1 | Investigar y adquirir la batería de respaldo (al menos 2 horas de autonomía) | Pool | 8 | Terminado |
| 5.2.2 | Conectar e integrar la batería de respaldo al sistema | Victor | 8 | Terminado |
| 5.2.3 | Crear manual sobre la batería y su funcionamiento | Cristian | 5 | Terminado |

*Tabla 12. Historia de Usuario 5.2: Respaldo de energía*

**4.3.5.3 Historia de Usuario 5.3: Autodiagnóstico**

Descripción: Como agricultor, quiero que el sistema detecte fallos internos, para anticipar problemas.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 5.3.1 | Implementar un registro de errores internos | Victor | 7 | Terminado |
| 5.3.2 | Programar notificaciones visuales para el agricultor | Erick | 7 | Terminado |

*Tabla 13. Historia de Usuario 5.3: Autodiagnóstico*

**4.3.6.1 Historia de Usuario 6.1: Componentes de bajo costo**

Descripción: Como agricultor, quiero que el sistema use sensores accesibles, para que sea económico de instalar.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 6.1.1 | Investigar proveedores locales y online para componentes | Pool | 5 | Terminado |
| 6.1.2 | Verificar precios y asegurar que el costo total esté dentro de lo posible | Pool | 4 | Terminado |

*Tabla 14. Historia de Usuario 6.1: Componentes de bajo costo*

**4.3.6.2 Historia de Usuario 6.2: Fácil mantenimiento**

Descripción: Como agricultor, quiero que el sistema tenga piezas modulares, para reemplazar solo lo dañado.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 6.2.1 | Definir los módulos del sistema (sensores, batería, actuadores) | Victor | 6 | Terminado |
| 6.2.2 | Etiquetar físicamente los componentes y usar cableado codificado por color | Erick | 8 | Terminado |

*Tabla 15. Historia de Usuario 6.2: Fácil mantenimiento*

**4.3.6.3 Historia de Usuario 6.3: Guía de mantenimiento**

Descripción: Como agricultor, quiero un manual de mantenimiento simple, para reparar el sistema sin ayuda externa.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 6.3.1 | Redactar el manual de mantenimiento en español con fotos | Cristian | 8 | Terminado |
| 6.3.2 | Incluir un plan de mantenimiento y tabla de fallas comunes | Cristian | 5 | Terminado |
| 6.3.3 | Realizar pruebas con usuarios para validar el manual | Cristian | 4 | Terminado |
| 6.3.4 | Entregar versión final en PDF e impresa | Cristian | 2 | Terminado |

*Tabla 16. Historia de Usuario 6.3: Guía de mantenimiento*

**4.3.7.1 Historia de Usuario 7.1: Calibración de sensores**

Descripción: Como agricultor, quiero poder calibrar los sensores, para asegurar mediciones correctas.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 7.1.1 | Desarrollar la funcionalidad de calibración de sensores | Erick | 8 | Terminado |
| 7.1.2 | Implementar un registro de la fecha de la última calibración | Erick | 5 | Terminado |
| 7.1.3 | Realizar ensayos de calibración con materiales de referencia | Victor | 6 | Pendiente |

*Tabla 17. Historia de Usuario 7.1: Calibración de sensores*

**4.3.7.2 Historia de Usuario 7.2:Alertas de desviación**

Descripción: Como agricultor, quiero que el sistema me avisa si un sensor muestra valores anormales, para revisarlo.

**Tareas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| 7.1.1 | Programar la lógica para detectar desviaciones en la lectura | Erick | 8 | Terminado |
| 7.1.2 | Implementar el sistema de alertas visuales para desviaciones | Cristian | 5 | Terminado |
| 7.1.3 | Realizar simulaciones de desconexión y ruidos en los sensores | Victor | 6 | Terminado |

*Tabla 18. Historia de Usuario 7.2:Alertas de desviación*

**4.4. Total de Esfuerzo Estimado**

|  |  |  |
| --- | --- | --- |
| **Función** | **Responsable** | **Horas Totales** |
| Integración de Hardware | Victor | 85 |
| Programación y Lógica | Erick | 88 |
| Diseño y Documentación | Cristian | 48 |
| Investigación y Adquisición | Pool | 37 |
| Total General |  | 258 |

*Tabla 19. Total de Esfuerzo Estimado*

Este Sprint Backlog será actualizado diariamente durante las reuniones de seguimiento del equipo refleja el avance del desarrollo y posibles ajustes en la planificación.

**4.5 Registro de Avance del Proyecto (Scrum)**

**1.1.1 Investigación: sensores de humedad compatibles con Arduino**

## 1. Objetivo y alcance

**Objetivo:** proporcionar una guía técnica y práctica (investigación) para elegir, conectar, calibrar y usar sensores de humedad de suelo y humedad ambiental (aire) con placas Arduino (UNO, Nano, Mega, ESP32, etc.).  
Alcance: revisión de tipos comerciales comunes (resistivos, capacitivos, DHTxx, BME280, SHT31/SHT3x, SI7021), comparativa de especificaciones, instrucciones de conexión, código de ejemplo, métodos de calibración/validación y recomendaciones para proyectos hobby y aplicaciones más exigentes (agricultura, estación meteorológica, IoT).

## 2. Resumen ejecutivo / recomendación rápida

● Para proyectos baratos y temporales (hobby): YL-69 / FC-28 (resistivo) + DHT11 (ambiente).

● Para proyectos permanentes/fiables de riego: sensor capacitivo de humedad de suelo (v1.2 o similar) + DHT22 o BME280.

● Para estaciones meteorológicas o aplicaciones profesionales: BME280 o SHT31 (humedad con ±2–3 %RH) y sensores de suelo capacitivo bien calibrados.  
 (Evidencia: corrosión en sondas resistivas; ventajas de capacitivo; especificaciones DHT/BME).

## 3. Tipos de sensores: principios y características

### 3.1. Sensores de humedad de suelo

#### A) Resistivos(YL-69 / FC-28)

● Principio: miden la conductividad eléctrica entre dos electrodos insertados en el suelo (mayor humedad entonces más conductividad).

● Ventajas: muy baratos, fáciles de leer (analógico), disponibles en módulos con salida A0 y salida digital.

● Desventajas: corrosión de los electrodos cuando están permanentemente alimentados/insertados, vida útil corta; sensibles a la salinidad del suelo y variaciones químicas. Para alargar la vida se sugiere alimentar el sensor solo al tomar lectura (pulsado) o invertir polaridad periódicamente.

#### B) Capacitivos (Capacitive Soil Moisture Sensor v1.2)

● Principio: miden la capacitancia entre placas o trazos cuando se alteran por la humedad del sustrato; la presencia de agua cambia la permitividad dieléctrica.

● Ventajas: mayor durabilidad (sin electrodos expuestos), menos corrosión, lecturas más estables y menos afectadas por salinidad; adecuados para instalaciones semi-permanentes. Estudios recientes muestran buena relación costo/precisión frente a equipos TDR profesionales.

● Desventajas: requieren calibración para convertir lectura analógica (V) a contenido volumétrico de agua (VWC). Algunos clones baratos pueden tener problemas de calidad o envejecimiento; comprar de proveedores confiables.

### 3.2. Sensores de humedad ambiental (aire)

#### A) DHT11 / DHT22 (AM2302)

* + DHT11: rango ~20–80 %RH, precisión ≈ ±5 %RH, económico; muestreo lento.
  + DHT22 / AM2302: rango 0–100 %RH (o amplio), mejor precisión ≈ ±2–5 %RH según fuente, mayor resolución; más adecuado que DHT11 si necesitas mejor lectura. Ambos usan protocolo digital de un solo pin y requieren lectura no más de 0.5–1 Hz.

#### B) Sensores de precisión I²C/SPI (BME280, SHT31/SHT3x, SI7021)

● BME280 (Bosch): mide humedad, temperatura y presión; precisión humedad ~±3 %RH típ. Soporta I²C y SPI, bajo consumo y buena estabilidad a largo plazo. Ideal si quieres también presión (altitud/meteorología).

● SHT31 / SHT3x (Sensirion): sensores de alta precisión para RH y temperatura, linealidad y estabilidad superiores a DHT; comunicación I²C y mejores prestaciones térmicas. (Usados en aplicaciones profesionales).

## 4. Especificaciones útiles y comparativa resumida

*Nota:* los valores exactos dependen del fabricante / módulo. Aquí se muestran rangos típicos basados en datasheets y guías.

### ● YL-69 / FC-28 (resistivo)

○ Salida: analógica (0–Vcc).

○ Vida útil: baja si se deja alimentado y en contacto permanente (corrosión).

### ● Capacitive Soil Moisture Sensor v1.2

○ Salida: analógica (AOUT) — generalmente 0–3.3/0–5V según módulo.

○ Durabilidad: alta relativa; menos corrosión.

### ● DHT11

○ Humedad: 20–80 %RH, ±5 %RH.

○ Tiempo lectura: ≤1 Hz.

### ● DHT22 / AM2302

○ Humedad: 0–100 %RH, precisión ±2–5 %RH (mejor que DHT11), -40–80°C.

### ● BME280

○ Humedad: precisión típica ±3 %RH; I²C (0x76/0x77) o SPI; bajo consumo; alta estabilidad.

### ● SHT31 / SHT3x / SI7021

○ Humedad: típicamente ±2 %RH (SHT31), buena estabilidad y respuesta.

# 5. Evidencias de Código

// Inclusión de las bibliotecas necesarias

#include <LiquidCrystal.h>

#include <DHT.h>

#include <DHT\_U.h>

// Definición de variables globales

int Tiempo = 3000; // Tiempo de retardo entre lecturas y actualizaciones en milisegundos

int rs = 12; // Pin RS del LCD

int e = 11; // Pin E del LCD

int d4 = 5; // Pin D4 del LCD

int d5 = 4; // Pin D5 del LCD

int d6 = 3; // Pin D6 del LCD

int d7 = 2; // Pin D7 del LCD

const int BombaAgua = 13; // Pin de la bomba de agua

const int humTierra = A0; // Pin del sensor de humedad de tierra

const int luces = 6; // Pin de las luces

int ventilador = 10; // Pin del ventilador

int Humendad; // Variable para almacenar la humedad

float tempC; // Variable para almacenar la temperatura en grados Celsius

// Creación del objeto LiquidCrystal para manejar el LCD

LiquidCrystal lcd(rs, e, d4, d5, d6, d7);

#define SensorTipo DHT11 // Definición del tipo de sensor DHT

int SensorTemp = 8; // Pin al que está conectado el sensor DHT

DHT MedirTempHum(SensorTemp, SensorTipo); // Creación del objeto DHT

// Configuración inicial del Arduino

void setup() {

// Inicialización del LCD con dimensiones 16x2

lcd.begin(16, 2);

// Inicialización de la comunicación serial a 9600 bps

Serial.begin(9600);

// Inicialización del objeto DHT

MedirTempHum.begin();

// Configuración de los pines como entrada o salida según sea necesario

pinMode(humTierra, INPUT);

pinMode(BombaAgua, OUTPUT);

pinMode(luces, OUTPUT);

pinMode(ventilador, OUTPUT);

}

// Bucle principal del programa

void loop() {

// Lecturas de humedad y temperatura del sensor DHT

Humendad = MedirTempHum.readHumidity();

tempC = MedirTempHum.readTemperature();

// Lectura del sensor de humedad de tierra

int ValorSensor = analogRead(humTierra);

// Actualización del LCD con la humedad relativa

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("HUMEDAD REL ");

lcd.setCursor(12, 0);

lcd.print(Humendad);

lcd.setCursor(14, 0);

lcd.print("%");

// Actualización del LCD con la humedad del suelo

lcd.setCursor(0, 1);

lcd.print("HUMEDAD T");

lcd.setCursor(12, 1);

lcd.print(ValorSensor);

delay(Tiempo);

lcd.clear();

// Actualización del LCD con la temperatura

lcd.setCursor(2, 0);

lcd.print("TEMPERATURA");

lcd.setCursor(3, 1);

lcd.print(tempC);

lcd.print(" \*C");

delay(Tiempo);

lcd.clear();

// Control de la bomba de agua en función de la humedad del suelo

if (ValorSensor > 900) {

digitalWrite(BombaAgua, HIGH);

lcd.setCursor(0, 0);

lcd.print("BOMBA ON-");

} else {

digitalWrite(BombaAgua, LOW);

lcd.setCursor(0, 0);

lcd.print("-BOMBA OFF");

}

// Control de las luces en función de la temperatura

if (tempC < 20) {

digitalWrite(luces, HIGH);

lcd.setCursor(10, 0);

lcd.print("LUZ ON");

} else {

digitalWrite(luces, LOW);

lcd.setCursor(10, 0);

lcd.print("LUZ OFF");

}

// Control del ventilador en función de la temperatura

if (tempC > 30) {

digitalWrite(ventilador, LOW);

lcd.setCursor(0, 1);

lcd.print("VENTILADOR ON");

} else {

digitalWrite(ventilador, HIGH);

lcd.setCursor(1, 1);

lcd.print("VENTILADOR OFF");

}

// Retardo antes de la próxima iteración del bucle

delay(Tiempo);

}

**Descripción del código:** Este código tiene como objetivo el desarrollo de un sistema automatizado para el control de un invernadero, utilizando varios sensores y dispositivos, como un sensor DHT11 para medir la temperatura y humedad ambiental, y un sensor de humedad de suelo. La información recopilada por estos sensores se visualiza en una pantalla LCD y se utiliza para controlar el funcionamiento de dispositivos como una bomba de agua, luces y ventiladores, según las condiciones del ambiente. Los valores de humedad y temperatura se actualizan en la pantalla LCD cada cierto tiempo, permitiendo un monitoreo constante.

En cuanto a la parte de control, el sistema actúa de manera automática según las mediciones de los sensores. Si la humedad del suelo es baja, se activa la bomba de agua; si la temperatura es baja, las luces se encienden para proporcionar calor al invernadero. Además, el ventilador se activa o desactiva según la temperatura, asegurando que el ambiente dentro del invernadero se mantenga en condiciones ideales para el crecimiento de las plantas. El sistema realiza estas acciones a través de salidas digitales que controlan la activación de estos dispositivos.

Por último, el código también incluye un manejo sencillo de las entradas y salidas, con configuraciones de pines adecuadas y lecturas de los sensores. Se asegura de que el sistema sea eficiente y seguro, gestionando el encendido y apagado de cada dispositivo según las necesidades detectadas. La información de los sensores se envía a la pantalla LCD de forma periódica, con intervalos de tiempo predeterminados, lo que permite al usuario conocer en todo momento el estado de los parámetros ambientales y realizar ajustes manuales si es necesario.

## ID-1.1.4— Integrar la lectura del sensor a la interfaz

// Inclusión de las bibliotecas necesarias

*#include <LiquidCrystal.h>*

*#include <DHT.h>*

*#include <DHT\_U.h>*

// Definición de variables globales

int Tiempo = 3000;    // Tiempo de retardo entre lecturas y actualizaciones en milisegundos

int rs = 12;          // Pin RS del LCD

int e = 11;           // Pin E del LCD

int d4 = 5;           // Pin D4 del LCD

int d5 = 4;           // Pin D5 del LCD

int d6 = 3;           // Pin D6 del LCD

int d7 = 2;           // Pin D7 del LCD

const int BombaAgua = 13;   // Pin de la bomba de agua

const int humTierra = A0;   // Pin del sensor de humedad de tierra

const int luces = 6;        // Pin de las luces

int ventilador = 10;        // Pin del ventilador

int Humendad;               // Variable para almacenar la humedad

float tempC;                // Variable para almacenar la temperatura en grados Celsius

// Creación del objeto LiquidCrystal para manejar el LCD

LiquidCrystal lcd(rs, e, d4, d5, d6, d7);

*#define SensorTipo DHT11    // Definición del tipo de sensor DHT*

int SensorTemp = 8;         // Pin al que está conectado el sensor DHT

DHT MedirTempHum(SensorTemp, SensorTipo);  // Creación del objeto DHT

// Configuración inicial del Arduino

void setup() {

  // Inicialización del LCD con dimensiones 16x2

  lcd.begin(16, 2);

  // Inicialización de la comunicación serial a 9600 bps

  Serial.begin(9600);

  // Inicialización del objeto DHT

  MedirTempHum.begin();

  // Configuración de los pines como entrada o salida según sea necesario

  pinMode(humTierra, INPUT);

  pinMode(BombaAgua, OUTPUT);

  pinMode(luces, OUTPUT);

  pinMode(ventilador, OUTPUT);

}

// Bucle principal del programa

void loop() {

  // Lecturas de humedad y temperatura del sensor DHT

  Humendad = MedirTempHum.readHumidity();

  tempC = MedirTempHum.readTemperature();

  // Lectura del sensor de humedad de tierra

  int ValorSensor = analogRead(humTierra);

  // Actualización del LCD con la humedad relativa

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("HUMEDAD REL ");

  lcd.setCursor(12, 0);

  lcd.print(Humendad);

  lcd.setCursor(14, 0);

  lcd.print("%");

  // Actualización del LCD con la humedad del suelo

  lcd.setCursor(0, 1);

  lcd.print("HUMEDAD T");

  lcd.setCursor(12, 1);

  lcd.print(ValorSensor);

  delay(Tiempo);

  lcd.clear();

  // Actualización del LCD con la temperatura

  lcd.setCursor(2, 0);

  lcd.print("TEMPERATURA");

  lcd.setCursor(3, 1);

  lcd.print(tempC);

  lcd.print(" \*C");

  delay(Tiempo);

  lcd.clear();

  // Control de la bomba de agua en función de la humedad del suelo

  if (ValorSensor > 900) {

    digitalWrite(BombaAgua, HIGH);

    lcd.setCursor(0, 0);

    lcd.print("BOMBA ON-");

  } else {

    digitalWrite(BombaAgua, LOW);

    lcd.setCursor(0, 0);

    lcd.print("-BOMBA OFF");

  }

  // Control de las luces en función de la temperatura

  if (tempC < 20) {

    digitalWrite(luces, HIGH);

    lcd.setCursor(10, 0);

    lcd.print("LUZ ON");

  } else {

    digitalWrite(luces, LOW);

    lcd.setCursor(10, 0);

    lcd.print("LUZ OFF");

  }

  // Control del ventilador en función de la temperatura

  if (tempC > 30) {

    digitalWrite(ventilador, LOW);

    lcd.setCursor(0, 1);

    lcd.print("VENTILADOR ON");

  } else {

    digitalWrite(ventilador, HIGH);

    lcd.setCursor(1, 1);

    lcd.print("VENTILADOR OFF");

  }

  // Retardo antes de la próxima iteración del bucle

  delay(Tiempo);

}

**Historia de Usuario 2.1: Riego automático por humedad**

**ID 2.1.1 Investigar y adquirir la bomba de riego**

1. Tipos de bombas recomendadas

* Mini bomba sumergible DC (3V–12V)
  + Ideal para prototipos o pequeños invernaderos.
  + Bajo consumo de energía.
  + Caudal aproximado: 80–120 L/h.
  + Precio referencial: S/ 25 – 40.
* Bomba peristáltica DC (5V–12V)
  + Diseñada para riego por goteo o dosificación de nutrientes.
  + Permite un control preciso del flujo de agua.
  + Precio referencial: S/ 45 – 70.
* Bomba sumergible DC de mayor capacidad (12V–24V)
  + Adecuada para invernaderos medianos o grandes.
  + Caudal aproximado: 240–800 L/h.
  + Requiere fuente de alimentación independiente y módulo relé.
  + Precio referencial: S/ 60 – 120.

2. Componentes complementarios necesarios

* Módulo relé 5V o MOSFET: para controlar la bomba desde el Arduino.
  + Precio: S/ 10 – 20.
* Fuente de alimentación externa (según el voltaje de la bomba).
  + Precio: S/ 20 – 40.
* Sensor de humedad de suelo: para medir el nivel de humedad y automatizar el riego.
  + Precio: S/ 10 – 25.
* Mangueras y válvulas de riego: costo variable (S/ 15 – 30).

3. Ejemplo de funcionamiento del sistema

1. El sensor de humedad detecta sequedad en el suelo.
2. Arduino procesa la señal recibida.
3. Se activa el módulo relé conectado al Arduino.
4. El relé enciende la bomba de agua.
5. El riego se detiene cuando la humedad del suelo alcanza el nivel deseado.

4. Opciones de adquisición en Perú

* Mercado Libre Perú
  + “Mini bomba sumergible DC 5V–12V para Arduino”: S/ 25 – 40.
  + “Bomba de agua peristáltica DC 12V”: S/ 45 – 70.
  + “Kit de riego automático (bomba + sensor + relé)”: S/ 60 – 90.
* Tiendas de electrónica (Promelsa, Infotronica, Steren, distribuidores locales).
  + Bombas DC sumergibles y accesorios.
  + Precios similares a Mercado Libre.
* AliExpress (importación)
  + Bombas de 5V, 6V y 12V desde S/ 12 – 25.
  + Kits completos desde S/ 40 – 70 (con tiempos de entrega más largos).

Recomendación para el proyecto

* Prototipo o pruebas en laboratorio: Mini bomba sumergible DC 12V + módulo relé + sensor de humedad (costo total aproximado S/ 60 – 80).
* Invernadero en operación real: Bomba DC 12V de mayor caudal + sistema de riego por goteo + relé + fuente de 2A (costo total aproximado S/ 100 – 160).

## Historia de Usuario 2.3: Opción manual de emergencia

2.3.1 Seleccionar y adquirir el botón físico para la activación manual

Se adquirió tres unidades del botón físico (push button) para el proyecto con Arduino. Este componente permitirá interactuar manualmente con el sistema, habilitando funciones como encendido/apagado, reinicio o control de eventos según la programación.



*Figura 1. Botón físico para la activación manual*

## Historia de Usuario 3.1: Lectura en tiempo real(Temperatura)

**3.1.1 Investigar y adquirir el sensor de temperatura**

Un sensor es un dispositivo que detecta cambios o variaciones físicas del entorno como temperatura, luz, humedad o movimiento y las convierte en señales eléctricas que pueden ser interpretadas por un microcontrolador, como el Arduino o el ESP32.  
 En esencia, los sensores permiten que un sistema electrónico “perciba” su entorno y reaccione en consecuencia, convirtiéndose en una parte fundamental para el desarrollo de proyectos automatizados, sistemas de monitoreo e Internet de las Cosas (IoT).

Estos sensores son los que se averiguaron, ideales o con referencia hacia nuestro proyecto:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sensor | Tipo de salida | Ventajas | Desventajas / consideraciones | Precio estimado\* |
| DHT22 (AM2302) | Digital (bus de un solo cable) | Buena precisión (~±0,5 °C), rango amplio, lectura de humedad también | No muy alta frecuencia (solo ~1 medición cada 2 s) | S/ 25 a S/ 35 aprox. Ej: Electromanía lo ofrece por S/ 25. [Electromania](https://www.electromania.pe/producto/sensor-de-temperatura-y-humedad-dht22/?utm_source=chatgpt.com); Tesla Electronic lo tiene a S/ 28 [Tesla Electronic EIRL](https://www.teslaelectronic.com.pe/producto/sensor-dht22-temperatura-y-humedad/?srsltid=AfmBOopdSE3GJwLLtIJwtRU921mlIiKChY3Naz7mrqVAFWybNWVGjNf8&utm_source=chatgpt.com) |
| LM35 | Analógica (voltaje proporcional a temperatura) | Simplicidad, fácil de leer con analogRead | Menos preciso en rangos extremos, requiere calibración y filtrado de ruido | En tiendas de electrónica, podría encontrarse desde unos S/ 10–15 (según tienda local) |
| DS18B20 | Digital (1-Wire) | Alta resolución, bastante usado, tolera cables largos | Necesita librería 1-Wire, a veces requiere pull-up | En algunas tiendas se ve a precios bajos (menos de S/ 20) |

*Tabla 20. Comparación de sensores de temperatura*

## Historia de Usuario 4.1: Ventilacion automatica

**4.1.1 Investigar y adquirir componentes de control de movimiento de puerta (Motor DC, Driver L298N).**

1. Introducción

En un sistema de riego automatizado, el control del flujo de agua es un factor fundamental para garantizar la eficiencia en la distribución del recurso hídrico. Una de las soluciones empleadas para lograr este control es la automatización del movimiento de puertas o compuertas, que permiten abrir o cerrar el paso del agua de manera controlada y precisa.

Para este propósito, se requiere de un conjunto de componentes que transformen la energía eléctrica en movimiento mecánico, permitiendo el accionamiento automático de la compuerta. Entre los elementos más utilizados para esta función destacan el motor de corriente continua (DC) y el módulo controlador de motor (driver L298N), que en conjunto posibilitan el control del sentido y la velocidad del movimiento.

2. Motor DC: Principio de funcionamiento y selección

El motor de corriente continua (DC) es un dispositivo electromecánico que convierte energía eléctrica en movimiento rotacional. Su funcionamiento se basa en la interacción de un campo magnético y una corriente eléctrica, generando un torque proporcional a la intensidad de la corriente.

En el caso del sistema de riego automatizado, el motor DC es responsable de accionar la compuerta que regula el caudal de agua en los canales o reservorios. Para lograr un movimiento controlado, se utiliza un motor DC con reductora (geared DC motor), que reduce la velocidad de rotación y aumenta el torque disponible, permitiendo mover compuertas de manera lenta y estable.

Criterios de selección:

* Tensión de operación: 12V o 24V DC (según la fuente de energía del sistema).
* Velocidad de rotación: entre 10 y 100 RPM, para garantizar un movimiento suave y seguro.
* Torque: superior a 30 kg·cm, dependiendo del peso de la compuerta y la fuerza requerida.
* Consumo de corriente: de 1 a 5 A, acorde con la capacidad del driver seleccionado.
* Tipo de reductora: planetaria o de tornillo sin fin (worm gear), por su eficiencia y capacidad de autobloqueo.

Modelo propuesto:

Motor DC con reductora 12V, 60W, 100 RPM tipo worm gear, adecuado para la apertura y cierre de compuertas pequeñas o medianas en sistemas de riego automatizado. Este tipo de motor ofrece un excelente equilibrio entre potencia, torque y facilidad de control.

3. Módulo Driver L298N

El L298N es un módulo electrónico basado en un circuito integrado de doble puente H, diseñado para controlar motores DC o motores paso a paso mediante señales digitales de control. En este proyecto, el L298N permite regular la dirección y velocidad de giro del motor DC, en función de las órdenes enviadas desde un microcontrolador (por ejemplo, Arduino UNO o ESP32).

Características técnicas principales:

* Tensión de alimentación del motor: 5V a 35V DC.
* Corriente máxima de salida: 2A por canal.
* Control de sentido de giro mediante entradas lógicas (IN1–IN4).
* Control de velocidad a través de modulación por ancho de pulso (PWM).
* Incluye disipador de calor y pines de habilitación (ENA/ENB).

El módulo L298N es ampliamente utilizado en proyectos de automatización agrícola y sistemas embebidos por su bajo costo, facilidad de conexión y compatibilidad con plataformas educativas. Para aplicaciones que demanden mayor corriente, se recomienda considerar controladores de mayor capacidad como el BTS7960 (43A).

4. Integración del sistema de control

En el sistema de riego automatizado, el motor DC y el driver L298N se integran con un microcontrolador y sensores que determinan las condiciones de apertura o cierre de la compuerta según la humedad del suelo, nivel del agua o programación horaria.

|  |  |  |
| --- | --- | --- |
| Componente | Función principal | Parámetros relevantes |
| Motor DC con reductora | Generar movimiento de apertura/cierre de la compuerta. | 12V, 60W, 100 RPM, torque ≥ 30 kg·cm |
| Driver L298N | Controlar sentido y velocidad del motor DC. | 5–35V, 2A por canal |
| Microcontrolador (Arduino UNO / ESP32) | Procesar señales de sensores y enviar comandos al driver. | Compatible con PWM y L298N |
| Fuente de alimentación | Suministrar energía estable al sistema. | 12V DC, 5–10A |
| Sensores de nivel o humedad | Activar o detener el movimiento según condiciones de riego. | Sensor capacitivo o resistivo |
| Fin de carrera | Detectar límites de apertura/cierre de la compuerta. | Mecánico o magnético |

*Tabla 21. Comparación de controladores y sensores*

El sistema puede programarse para que la compuerta se abra automáticamente cuando el nivel de humedad del suelo sea bajo o el nivel del agua sea suficiente, cerrándose cuando las condiciones de riego sean óptimas.

5. Criterios de adquisición

Durante la adquisición de los componentes, se consideraron los siguientes criterios técnicos:

* Compatibilidad eléctrica y lógica con el microcontrolador y sensores del sistema.
* Disponibilidad en el mercado local o plataformas en línea (Amazon, MercadoLibre, AliExpress).
* Relación costo–beneficio, priorizando componentes de bajo consumo y mantenimiento mínimo.
* Resistencia y durabilidad ante condiciones ambientales (temperatura, humedad).

6. Conclusión

La selección del motor DC con reductora y el driver L298N resulta idónea para el control automatizado de compuertas en un sistema de riego inteligente. Estos componentes permiten un control preciso del movimiento, facilitando la regulación del flujo de agua con bajo consumo energético y alta fiabilidad.

Asimismo, su integración con un microcontrolador y sensores ambientales posibilita la automatización completa del sistema, contribuyendo a una gestión más eficiente y sostenible del recurso hídrico en entornos agrícolas.

**Historia de Usuario 4.2: Calefacción automática**

**3.1.1 Investigar y adquirir el Calefactor y sus componentes de control.**

1. Introducción

En el marco del proyecto de riego automatizado, la incorporación de un calefactor tiene como objetivo mantener condiciones térmicas adecuadas en el sistema, especialmente en zonas donde las bajas temperaturas pueden afectar el funcionamiento de los sensores, el flujo del agua o el crecimiento de las plantas.

El calefactor permite elevar o estabilizar la temperatura en áreas específicas del sistema, como depósitos de agua, invernaderos o conducciones, asegurando la eficiencia del riego y la protección de los componentes electrónicos. Su funcionamiento requiere de un sistema de control térmico, compuesto por sensores de temperatura, módulos de potencia y controladores lógicos que regulan su activación.

2. Calefactor: Principio de funcionamiento y tipos

Un calefactor es un dispositivo eléctrico que convierte la energía eléctrica en energía térmica mediante el efecto Joule, el cual establece que cuando una corriente eléctrica atraviesa un material resistivo, se produce calor.

Existen diversos tipos de calefactores utilizados en sistemas automatizados:

|  |  |  |
| --- | --- | --- |
| Tipo de calefactor | Aplicación principal | Características |
| Resistencia calefactora (cerámica o metálica) | Calentamiento de agua o aire. | Eficiente, económica, fácil de controlar. |
| Calefactor tipo cartucho | Aplicaciones industriales y de precisión. | Alta densidad de potencia, tamaño compacto. |
| Banda calefactora o flexible | Superficies cilíndricas o depósitos. | Distribución uniforme del calor. |
| Calefactor PTC (coeficiente positivo de temperatura) | Control automático de temperatura. | Autolimitante, segura y duradera. |

*Tabla 22. Comparación de calefactores*

Tipo seleccionado:

Para el presente proyecto, se selecciona una resistencia calefactora tipo PTC de 12V DC, adecuada para aplicaciones de baja potencia en sistemas de riego automatizado. Su diseño compacto y capacidad de autorregulación la hacen ideal para mantener la temperatura del agua o del entorno de sensores sin riesgo de sobrecalentamiento.

3. Componentes de control del calefactor

El control del calefactor requiere de un sistema que permita regular su encendido y apagado según la temperatura deseada. Este sistema se compone de los siguientes elementos:

|  |  |  |
| --- | --- | --- |
| Componente | Función principal | Ejemplo de modelo |
| Sensor de temperatura | Detectar la temperatura del agua o ambiente. | Sensor DHT22 o DS18B20. |
| Módulo de control (microcontrolador) | Procesar las señales del sensor y decidir cuándo activar el calefactor. | Arduino UNO / ESP32. |
| Módulo de potencia (relevador o MOSFET) | Permitir el paso de corriente al calefactor bajo control del microcontrolador. | Módulo relé 10A o MOSFET IRF520. |
| Fuente de alimentación | Proporcionar energía estable al sistema de calefacción. | 12V DC / 10A. |

*Tabla 22. Componentes para el funcionamiento del calefactor*

Principio de control:

El sensor de temperatura monitorea el entorno y envía la lectura al microcontrolador. Este compara la lectura con un umbral predefinido (por ejemplo, 20 °C).

* Si la temperatura es inferior al valor establecido, el microcontrolador activa el calefactor mediante el módulo relé.
* Cuando se alcanza la temperatura óptima, el sistema interrumpe la corriente, manteniendo así un equilibrio térmico eficiente.

Este principio de control puede ser binario (ON/OFF) o, en versiones más avanzadas, modulado mediante PWM o control PID para un ajuste más preciso de la temperatura.

4. Integración del sistema de calefacción

En el sistema de riego automatizado, el calefactor se integra con el resto de los módulos de control ambiental. Su función es prevenir la condensación, el congelamiento o la baja temperatura del agua en los tanques y conducciones, garantizando el funcionamiento óptimo del sistema.

|  |  |  |
| --- | --- | --- |
| Elemento | Descripción | Parámetros técnicos |
| Calefactor PTC 12V | Elemento resistivo autolimitante para elevar la temperatura del agua o aire. | 12V DC, 100 W, autorregulación 70 °C |
| Sensor DS18B20 | Sensor de temperatura digital de alta precisión. | Rango -55 °C a +125 °C, precisión ±0.5 °C |
| Módulo relé 10A | Actuador de potencia controlado por microcontrolador. | Capacidad de conmutación: 10A / 250V AC o 10A / 30V DC |
| Microcontrolador Arduino UNO | Unidad lógica de procesamiento y control. | Entradas digitales/analógicas, comunicación serial |
| Fuente 12V DC | Fuente de poder del sistema calefactor y control. | 12V, 10A |

*Tabla 23. Integración del calefactor*

5. Criterios de adquisición

Para la selección y compra de los componentes, se consideraron los siguientes criterios técnicos y logísticos:

* Compatibilidad eléctrica: Todos los componentes funcionan a 12V DC, garantizando integración con el sistema de riego.
* Seguridad térmica: Selección de calefactor PTC por su capacidad de autorregulación y protección contra sobrecalentamiento.
* Eficiencia energética: Baja demanda de corriente y rápida respuesta térmica.
* Disponibilidad comercial: Componentes adquiribles en proveedores locales o en línea (Amazon, MercadoLibre, AliExpress).
* Costo accesible: Adecuado para proyectos académicos y prototipos de ingeniería.

Ejemplos de componentes adquiribles:

* Calefactor PTC 12V DC, 100 W.
* Sensor de temperatura DS18B20 impermeable.
* Módulo relé 1 canal de 10A.
* Fuente de poder 12V/10A.

6. Conclusión

La integración de un calefactor controlado electrónicamente dentro del sistema de riego automatizado permite mantener condiciones térmicas estables y óptimas para el funcionamiento del equipo y la eficiencia del riego.

La elección del calefactor PTC 12V junto con un sistema de control basado en sensor de temperatura, módulo relé y microcontrolador proporciona una solución segura, eficiente y de bajo mantenimiento. Este sistema contribuye significativamente a la sostenibilidad y automatización inteligente de los procesos agrícolas, al reducir pérdidas energéticas y garantizar el correcto desempeño del sistema en diversas condiciones ambientales.

**Historia de Usuario 5.2: Respaldo de energía**

Investigación y adquisición de la batería de respaldo (al menos 2 horas de autonomía)

Con el propósito de garantizar la continuidad operativa del sistema de riego automatizado ante posibles interrupciones del suministro eléctrico, se llevó a cabo la investigación y selección de una batería de respaldo adecuada para el funcionamiento del sistema basado en Arduino. Este componente es esencial para mantener activos los módulos de control, sensores y actuadores durante un tiempo mínimo de dos horas en caso de falla energética.

Durante la etapa de investigación, se evaluaron distintos tipos de baterías recargables, como las de iones de litio (Li-ion), fosfato de hierro y litio (LiFePO₄) y plomo-ácido selladas (SLA), considerando parámetros técnicos tales como capacidad (Ah), voltaje nominal, eficiencia de descarga, tiempo de autonomía y compatibilidad con los consumos eléctricos del sistema. Asimismo, se analizó la posibilidad de integrar un módulo de carga automática que permita mantener la batería en condiciones óptimas sin intervención manual.

Finalmente, se determinó la adquisición de una batería recargable de 12 V y 7 Ah, la cual ofrece una autonomía estimada superior a las dos horas para el sistema de riego automatizado. Esta batería se conectará mediante un módulo regulador de voltaje (DC-DC Step Down) y un módulo cargador automático compatible con fuentes solares o adaptadores AC/DC, garantizando así la estabilidad energética y la protección del sistema electrónico.

**Historia de Usuario 6.1: Componentes de bajo costo**

Investigación de proveedores locales y online para componentes

En la etapa de planificación del proyecto de Riego Automatizado con Arduino, se realizó una investigación exhaustiva de proveedores tanto locales como virtuales, con el objetivo de identificar las fuentes más adecuadas para la adquisición de los componentes electrónicos, eléctricos y mecánicos requeridos. Este proceso permitió asegurar la disponibilidad de los materiales, optimizar los costos y reducir los tiempos de aprovisionamiento, factores clave para la correcta ejecución del proyecto.

Proveedores locales en Cusco

En el ámbito local, se identificaron diversos establecimientos especializados en la venta de componentes electrónicos y eléctricos dentro de la ciudad de Cusco. Entre ellos destacan:

* ISCE (Taeem S.A.C.), con presencia en los distritos de Wanchaq y San Sebastián, reconocido por la venta de baterías selladas, fuentes de alimentación y accesorios eléctricos, resultando idóneo para la adquisición de la batería de respaldo del sistema.
* Todo Electrónica Cusco, tienda especializada en módulos electrónicos y sensores compatibles con Arduino, donde se pueden obtener controladores, módulos relé, sensores de humedad y temperatura, así como drivers de potencia como el L298N.
* Cusco Informático, proveedor que ofrece placas Arduino, cables, conectores y accesorios de integración, ideal para la fase de ensamblaje y pruebas del prototipo.

La ventaja principal de los proveedores locales radica en la disponibilidad inmediata de los componentes, la asesoría técnica presencial y la posibilidad de realizar pruebas directas de compatibilidad, reduciendo riesgos asociados a la compra a distancia.

Proveedores online nacionales

Paralelamente, se analizaron las principales plataformas nacionales de comercio electrónico que ofrecen un catálogo más amplio de componentes especializados y envíos a Cusco, tales como:

* MercadoLibre Perú, que concentra una amplia variedad de módulos electrónicos, motores DC, baterías recargables, controladores L298N y reguladores de voltaje, con tiempos de entrega promedio de 2 a 4 días hábiles.
* TeslaElectronic y Hi-Fi SAC, tiendas peruanas dedicadas a la comercialización de componentes electrónicos y de automatización, útiles para la adquisición de controladores de motor de alta potencia, sensores industriales y accesorios para montaje.
* Distribuidores oficiales de baterías (Ritar, Yuasa y Opalux), que cuentan con representación nacional y permiten la adquisición de baterías de respaldo de 12 V – 7 Ah o superiores, con garantía y certificaciones de calidad.

Estos proveedores virtuales destacan por su amplia disponibilidad de productos, variedad de marcas y modelos, así como la posibilidad de comparar precios y especificaciones técnicas antes de la compra. Sin embargo, presentan como desventaja el tiempo de espera por despacho y la imposibilidad de inspeccionar físicamente los componentes antes de su adquisición.

Conclusión del análisis

La combinación de proveedores locales y online garantiza un equilibrio entre accesibilidad, costo y calidad técnica. Se determinó que la estrategia óptima consiste en adquirir localmente los componentes esenciales y de rápida reposición (como baterías, cables, conectores y módulos básicos), mientras que los componentes especializados o de alta potencia (motores, controladores avanzados o módulos de carga inteligente) se adquirirán mediante proveedores online nacionales.  
 Esta planificación de adquisiciones permite fortalecer la eficiencia logística del proyecto y asegurar la continuidad del proceso de implementación del sistema de riego automatizado.

# 5. Burndown Chart

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Backlog ID | Historia de Usuario | Tiempo Estimado | 21/09/2025 | 22/09/2025 | 23/09/2025 | 24/09/2025 | 25/09/2025 | 26/09/2025 |
| Dia1 | Dia2 | Dia3 | Dia4 | Dia5 | Dia6 |
| HU-1.1.1 | Investigar sensores de humedad compatibles con Arduino | 4 hrs. | 1 | 0 | 0 | 0 | 0 | 3 |
| HU-1.1.2 | Conectar y programar sensor de humedad | 6 hrs. | 0 | 0 | 0 | 0 | 4 | 2 |
| 1.1.3 | Diseñar la interfaz de usuario para mostrar la lectura de humedad | 5 hrs. | 0 | 0 | 0 | 0 | 3 | 2 |
| 1.1.4 | Integrar la lectura del sensor a la interfaz | 6 hrs. | 0 | 0 | 0 | 2 | 2 | 2 |
| 1.1.5 | Probar la visualización de datos en tiempo real | 4 hrs. | 0 | 0 | 0 | 0 | 1 | 3 |
| 1.1.6 | Implementar alerta en caso de fallo del sensor | 3 hrs. | 0 | 0 | 0 | 0 | 0 | 3 |
| Tiempo de trabajo | | 28 hrs. | 1 | 0 | 0 | 2 | 10 | 15 |

*Tabla 24. Burndown Chart*

# 6. Análisis del Sprint

● Se completaron todas las tareas planificadas:

o Investigación de sensores de humedad compatibles con Arduino.

o Conexión y programación del sensor de humedad.

o Diseño de la interfaz de usuario para mostrar lecturas.

o Integración de la lectura del sensor en la interfaz.

o Pruebas de visualización de datos en tiempo real.

o Implementación de una alerta en caso de fallo del sensor.

● El progreso fue constante y sin bloqueos significativos.

● Las decisiones técnicas sobre integración sensor–interfaz fueron efectivas para este primer Sprint.

● El código se modularizó y se documentó en comentarios para facilitar mantenimiento y futuras mejoras.

● Las estimaciones fueron precisas y se mantuvo un ritmo de trabajo estable.

● Los objetivos del Sprint se cumplieron satisfactoriamente, dejando la base preparada para nuevas funcionalidades.

# 7. Conclusiones

El Sprint 1 concluyó exitosamente con todas las funcionalidades básicas implementadas en torno al sensor de humedad. Se investigaron sensores compatibles con Arduino, se logró la conexión y programación del módulo, y se diseñó la interfaz de usuario para mostrar en tiempo real la lectura de humedad. Asimismo, se integró la lectura del sensor en la interfaz, se validó la visualización de datos en tiempo real y se implementó una alerta ante posibles fallos del sensor.

El equipo demostró capacidad de entrega puntual y una adecuada gestión de riesgos menores. Las métricas indican que la carga de trabajo fue bien distribuida y los objetivos del sprint se cumplieron de manera satisfactoria, quedando la base lista para futuras mejoras y ampliaciones del sistema.

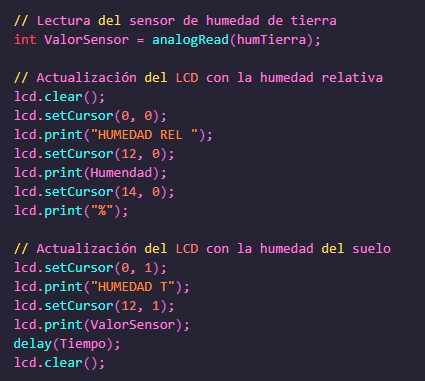
1. **USO DE HERRAMIENTAS MODERNAS**

Durante el desarrollo del proyecto, empleamos distintas herramientas tecnológicas tanto de software que permitieron el diseño, gestión y construcción del proyecto. Presentamos entonces cada una de las herramientas:

### 5.1. Arduino IDE

Utilizamos el Entorno de Arduino IDE, para programar cada unos de los sensores y actuadores.

* Uso: Realizamos la programación, compilación y el upload del código fuente desarrollado en C++. También a través de Arduino IDE gestionamos las librerías requeridas para que los sensores realicen lo requerido.
* Evidencia:

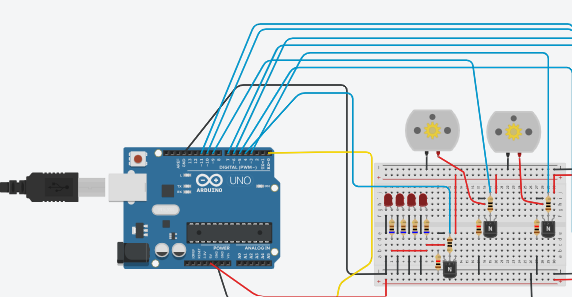
**

*Figura 2. Interfaz de programación en Arduino mostrando la conexión y programación del sensor de humedad.*

### 5.2. TinkerCAD

Con está herramienta pudimos realizar el diseño y simulación

* Uso en el proyecto:Esta herramienta fue muy necesaria para elaborar nuestros prototipos, diseñar, probar conexiones, esquemas, sensores y actuadores. Así evitamos riesgos de quemaduras de algunos componentes evitando dañarlos y evitando accidentes como de cortocircuitos.
* Evidencia:



*Figura 3. Simulación de conexiones entre Arduino, sensores y actuadores en TinkerCAD.*

### 5.3. Trello

En está plataforma, realizamos la gestión del proyecto

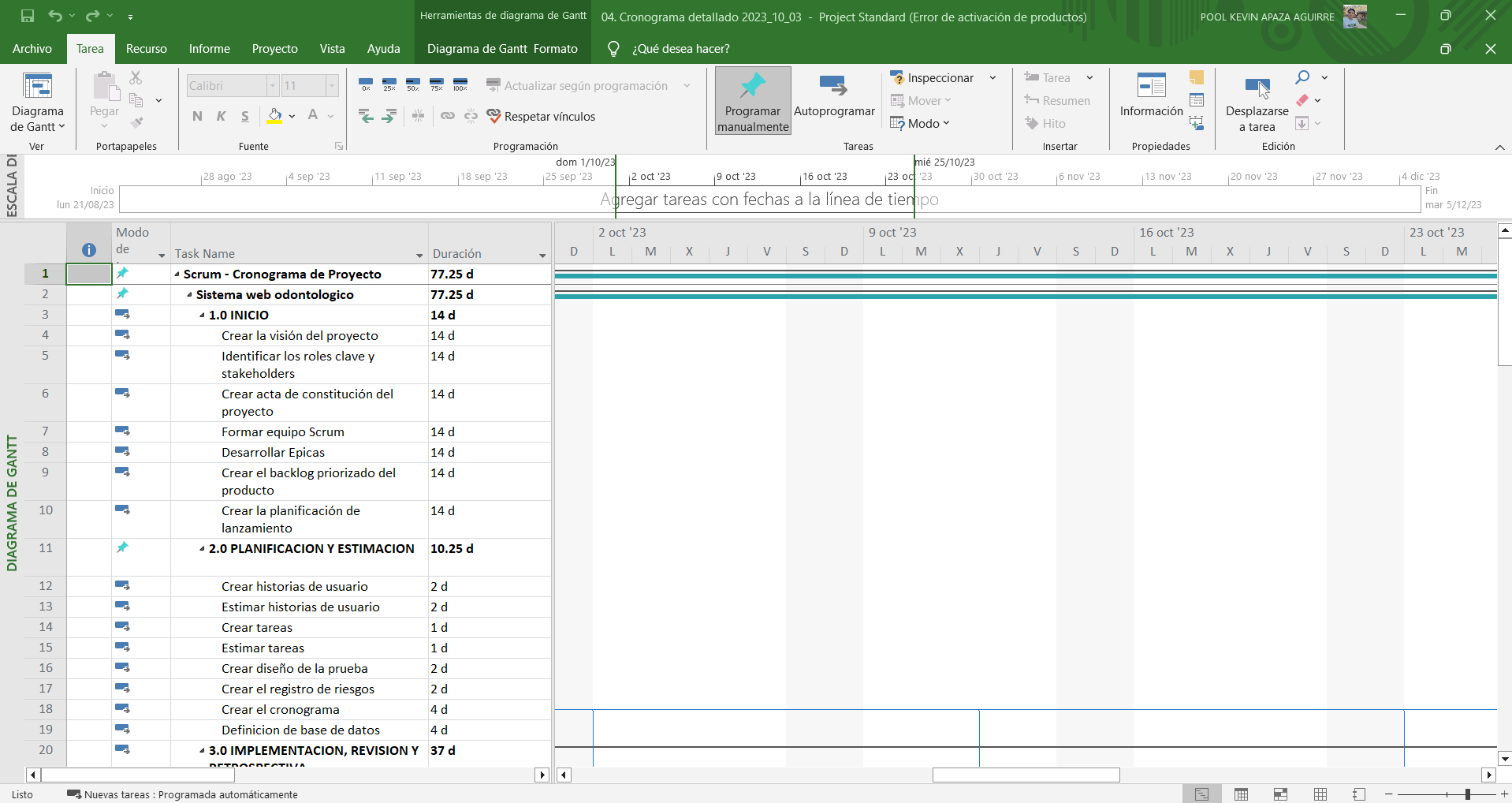
* Uso en el proyecto: Empleamos está plataforma para un mejor manejo y control de tareas, teniendo estados de inicio, pendiente, terminado, para cada una de las tareas y bueno asignando responsables, cumpliendo así el avance de las historias de usuario.
* Evidencia:

 *Figura 4. Tablero Kanban en Trello utilizado para la gestión de tareas.*

### 5.4. Microsoft Project

Se eligió este software de administración para la planificación de los tiempos necesarios para cada una de las tareas.

* Uso en el proyecto: Aquí desarrollamos el cronograma de actividades y el diagrama de gant para que podamos establecer tareas críticas y establecer una línea base para nuestro proyecto.
* Evidencia:

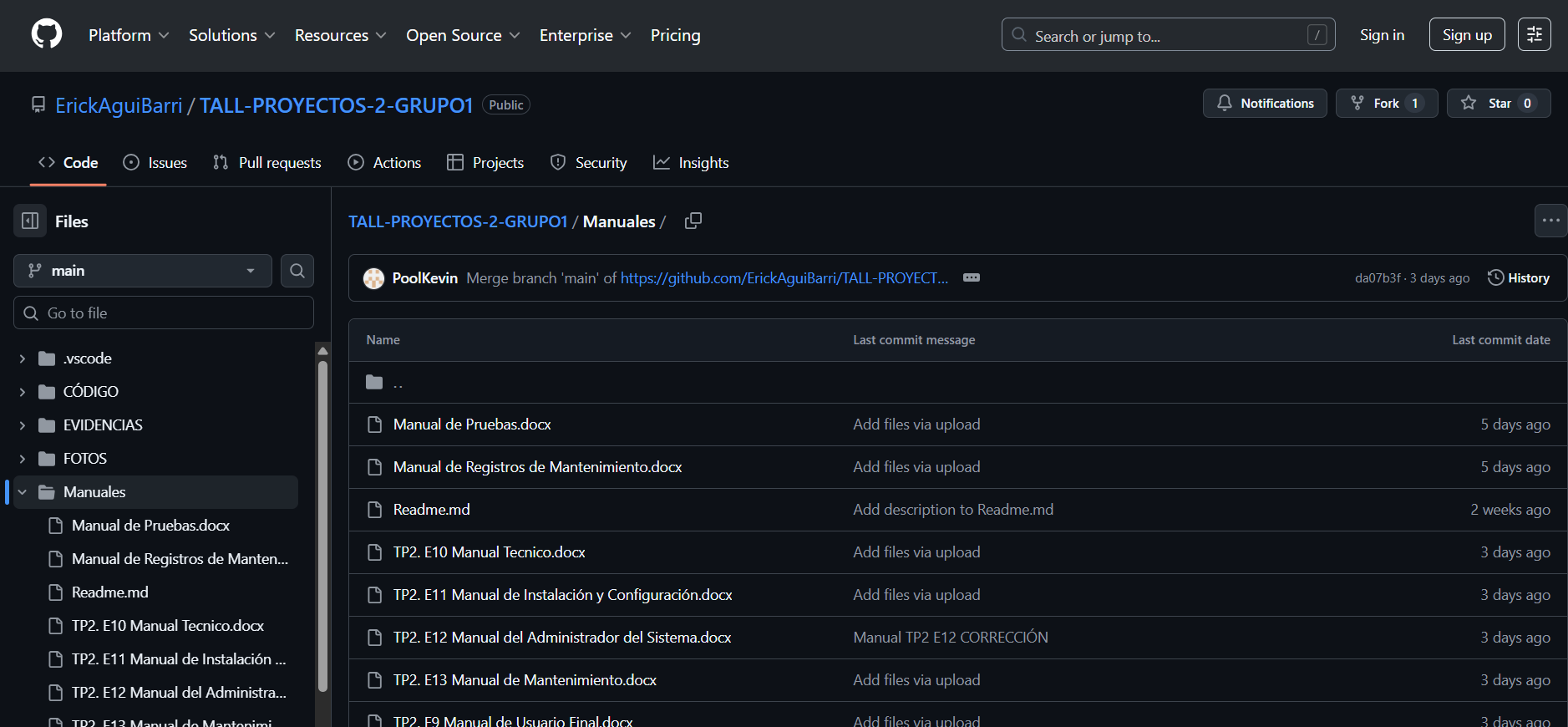


*Figura 5. Cronograma de actividades y diagrama de Gantt del proyecto.*

### 5.5. GitHub

Plataforma para control de versiones y trabajo colaborativo.

* Uso en el proyecto: Utilizamos esta plataforma para alojar nuestro repositorio, compartir el código y subir archivos compartidos con el fin de tener todo en un solo lugar, facilitando a los integrantes la colaboración.
* Evidencia:

 *Figura 6. Repositorio en GitHub con los archivos de nuestro proyecto.*

1. **Diseño de Ingeniería**

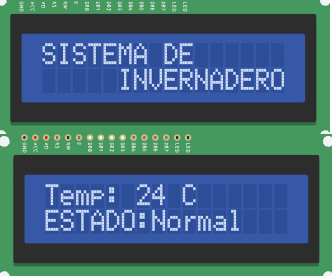
Aquí detallamos la estructura técnica y lógica de nuestro prototipo.

* 1. **Listado de Requerimientos funcionales**

El sistema cumple con los requerimientos para el correcto funcionamiento de nuestro prototipo.

* RF1 - Monitoreo de Humedad: El sistema debe medir y registrar la humedad del suelo.
* RF2 - Control de Riego Automático: El sistema activará o desactiva el riego automático según a los parámetros establecidos y obtenidos de parte de los sensores.
* RF3 - Monitoreo Ambiental: El sistema medirá la temperatura ambiente y enviará los datos para el riego.
* RF4 - Control Climático: Según los parámetros de la temperatura ambiente se activarán o desactivarán los ventiladores o calefactores.
* RF5 - Persistencia de Datos: El sistema registrará los datos para un posterior análisis.
* RF6 - Control Manual (Emergencia): El usuario final podrá controlar algunos aspectos vitales, mediante controles manuales.
* RF7 - Alertas: El sistema emitirá alertas mediante la pantalla LCD

**Diseño de Interfaces (Wireframes):** Para la lectura adecuada del usuario se desarrolló interfaces con información y alertas del estado del sistema.

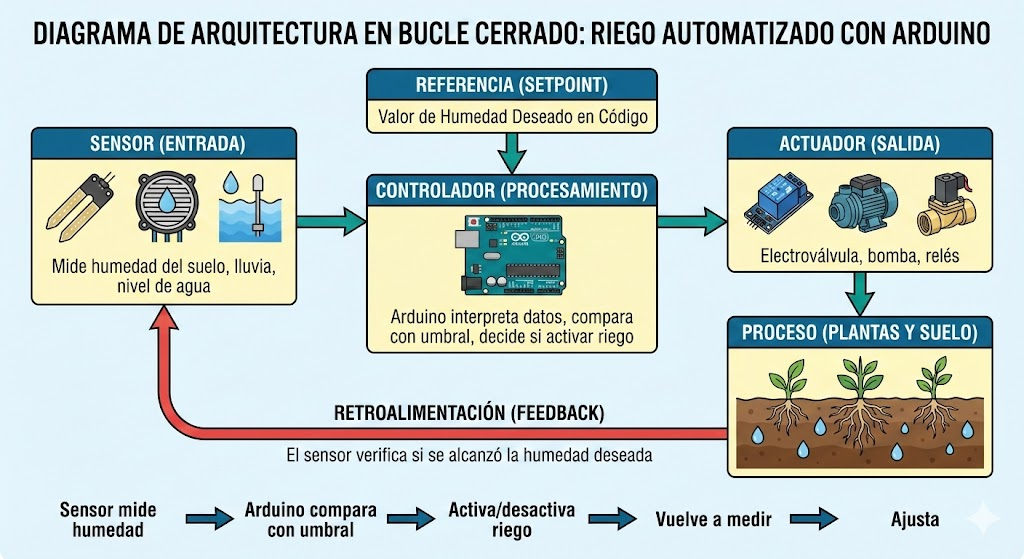
**** *Figura 7. Wireframes de la pantalla de inicio y pantalla principal (Datos en tiempo real).*

El flujo de navegación consta de:

1. Pantalla Principal: Aquí mostramos la temperatura dentro del invernadero, también la humedad en porcentaje y el estado del riego.
2. Pantalla de Alertas: Avisamos si hay alertas críticas.
3. Pantalla de Configuración: Podemos ajustar algunos umbrales del sistema.
   1. **Arquitectura de la solución planteada**

La arquitectura que utilizamos es en Bucle Cerrado (Control Loop / Closed - Loop Architecture). Porque el sistema actúa en base a la función de los sensores y no por el tiempo programado.

Diagrama de Arquitectura:



*Figura 8. Diagrama de arquitectura en bucle cerrado*

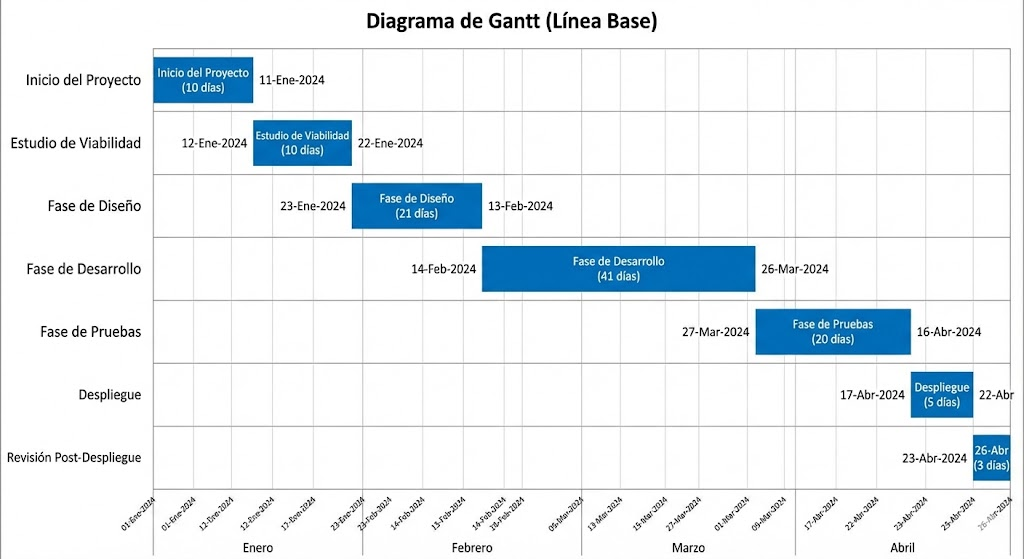
* 1. **Código de la aplicación por capas (enlace github)**

El código o firmware de nuestro proyecto se estructuró de manera modular, es decir por cada sensor realizamos código funcional por partes y esto lo vemos en nuestro repositorio de GitHub:

* Enlace al Repositorio: <https://github.com/ErickAguiBarri/TALL-PROYECTOS-2-GRUPO1/tree/main/Manuales>

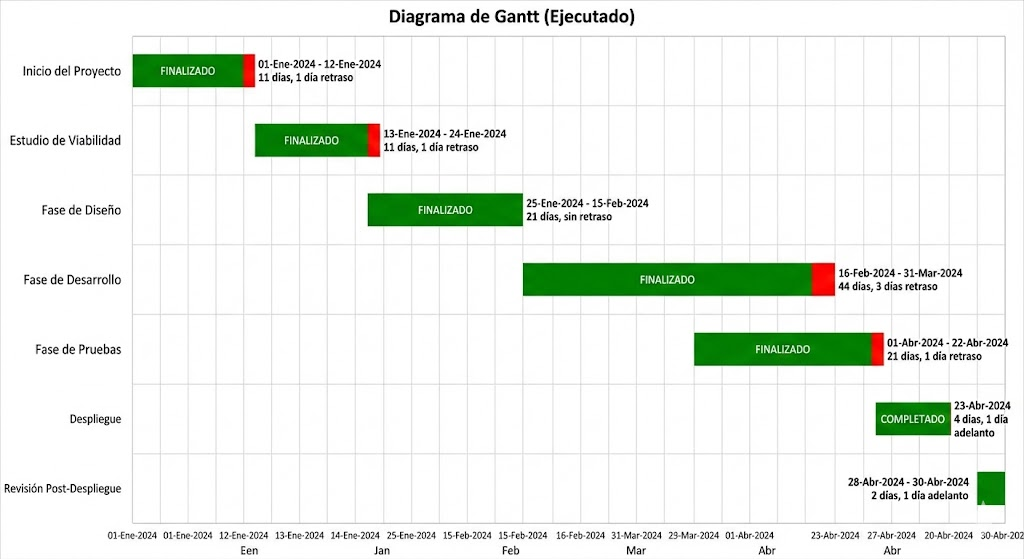
1. **GESTION DEL PROYECTO:**

**Diagrama de Gantt (Línea Base)**

****

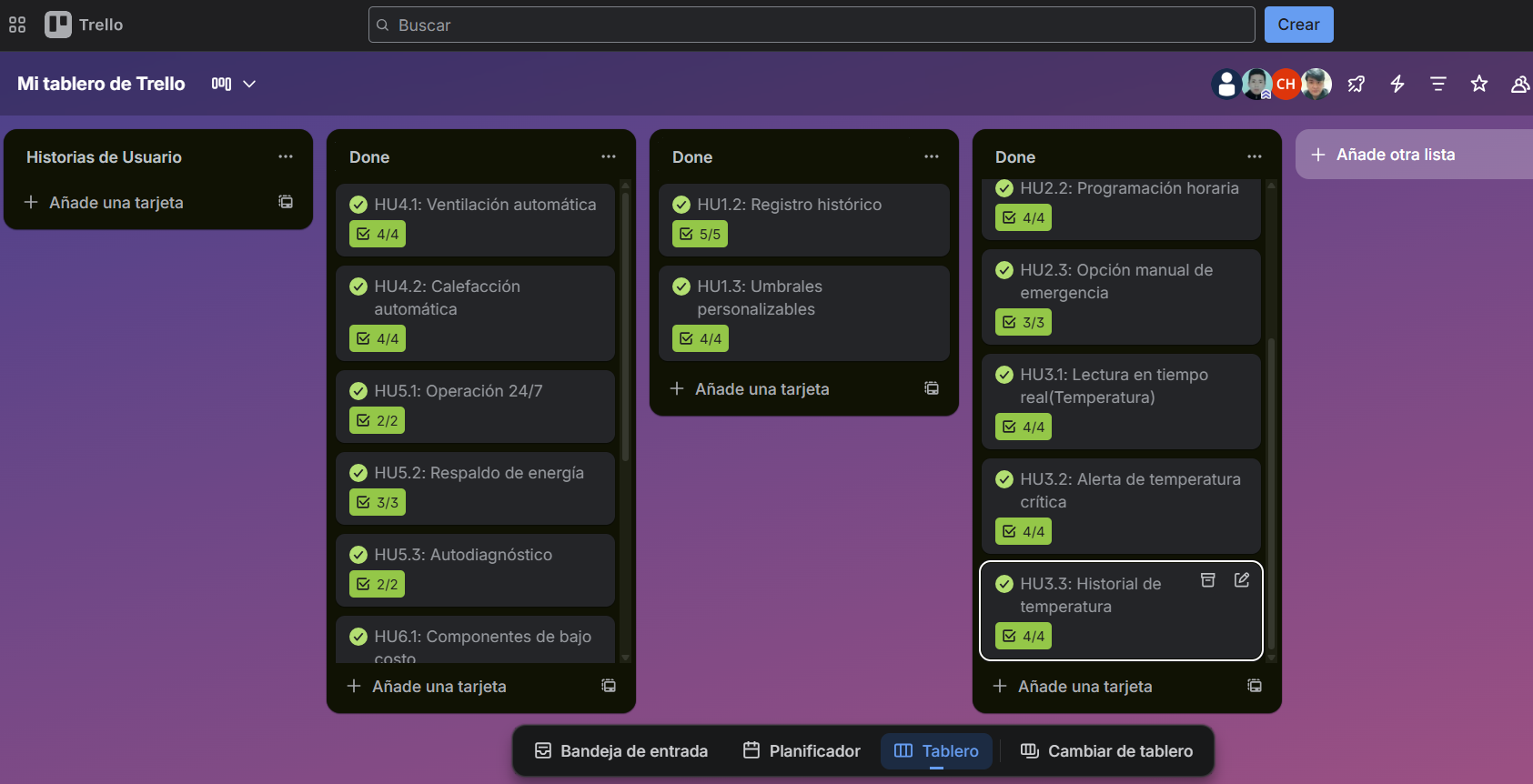
*Figura 9. Diagrama de Gantt (Línea Base)*

**Diagrama de Gantt(Ejecutado)**

****

*Figura 10. Diagrama de Gantt (Ejecutado)*

**Tablero Scrum – Kanban (Imagen y enlace público)**



*Figura 11. Tablero Scrum – Kanban*

TRELLO. Mi tablero de Trello [enlace]. Disponible en: <https://trello.com/invite/b/682be43fc7fbc312c30dce75/ATTIa3f8bb825105e6bda0791d7883bb635eB3BB0DA2/mi-tablero-de-trello>

1. **PRUEBAS - RESULTADOS Y DISCUSIÓN**

## 8.1 Pruebas realizadas por PMV

Las pruebas se ejecutaron en condiciones reales en la Comunidad de Chihuaco – Sicuani (altura aprox. 3 600 msnm), donde las variaciones de temperatura y la baja humedad del suelo representan retos significativos para el cultivo. Los PMV definidos fueron:

PMV1: Riego automático

PMV2: Monitoreo ambiental (temperatura y humedad)

PMV3: Funcionamiento integral del invernadero en condiciones de campo

## PMV1 – Pruebas del Sistema de Riego Automático

Prueba técnica: Activación del riego según umbral de humedad.  
 Objetivo: Verificar que la bomba se encienda cuando la humedad del suelo cae por debajo del 35% y se apague al superar el 40%

Procedimiento:

* Se conectó el sensor FC-28 al ADC del microcontrolador.
* Se estableció un umbral de activación del 35% y de desactivación del 40% .
* Se realizaron mediciones durante 4 semanas en suelo real.
* Se registró la humedad antes y después del riego.

Resultado:

* La bomba se activó únicamente cuando la humedad estuvo entre 20–30%.
* Se desactiva automáticamente cuando la humedad supera el 35–40%.
* No se detectaron activaciones erráticas.
* La duración promedio del ciclo de riego fue de 25–40 s.

**PMV2 – Pruebas del Monitoreo Ambiental (Temperatura y Humedad)**

Prueba técnica: Validación del sensor DHT11 en operación continua.  
Objetivo: Confirmar que el sistema registre correctamente la temperatura y humedad del invernadero en tiempo real.

Procedimiento:

* Se instaló el sensor DHT11 dentro del invernadero.
* Se midieron valores por la mañana, tarde y noche durante 4 semanas.
* Se compararon las lecturas con un termohigrómetro externo.
* Se verificó la actualización de datos en la pantalla LCD.

Resultado:

* Temperatura interna registrada: 12 °C – 28 °C, más estable que el exterior (mínimas de 7 °C).
* Humedad relativa consistente, sin pérdidas de señal.
* La actualización de datos se ejecuta cada 1–2 segundos.
* No se presentaron errores en la transmisión digital.

**PMV3 – Pruebas del Funcionamiento Integral del Invernadero**

Prueba técnica: Operación simultánea de riego, sensores y visualización.

Objetivo: Validar que todos los módulos funcionen de forma estable y coordinada bajo condiciones reales.

Procedimiento:

* Se activó el sistema completo durante periodos prolongados.
* Se monitoreo la estabilidad del microcontrolador, relés y sensores.
* Se verificó la visualización continua de datos en la pantalla LCD.
* Se registraron respuestas ante variaciones reales de humedad y temperatura.

Resultado:

* El sistema mantuvo operación continua sin reinicios ni fallas.
* El riego reaccionó correctamente a las variaciones de humedad.
* La pantalla LCD muestra información en tiempo real sin interrupciones.
* El prototipo demostró estabilidad en las condiciones ambientales de Chihuaco (≈ 3600 m s. n. m.).

## 

## 8.4. Resultados obtenidos en Chihuaco

## 8.4.1 Ahorro de agua

## Con base en las cuatro semanas de pruebas en campo, se obtuvo la Tabla 3 adaptada para Chihuaco:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Semana | Volumen Inicial (L) | Agua Utilizada (Convencional) | Agua Utilizada (Automatizado) | Agua Ahorrada (L) | % Ahorro |
| Semana 1 | 15 | 10.8 | 7.6 | 3.2 | 30% |
| Semana 2 | 14.8 | 9.5 | 6.8 | 2.7 | 28% |
| Semana 3 | 14.6 | 11.5 | 8.3 | 3.2 | 28% |
| Semana 4 | 14.6 | 10.9 | 7.5 | 3.4 | 31% |

## *Tabla 25. Resultados de los valores de aguaº“Basado en Huerta et al. (2020), Sivamani et al. (2021), Kaya et al. (2025) y Kisekka et al. (2022)*.

## Resultado clave:

## El prototipo permitió un ahorro promedio del 29% de agua, lo cual es altamente beneficioso en una comunidad con recursos hídricos limitados como Chihuaco.

## El sistema de riego automatizado demostró ser eficaz y estable durante las pruebas, funcionando entre una y dos veces por día en función de los niveles de humedad del terreno. La bomba funcionó durante un ciclo promedio de entre 25 y 40 segundos, lo que posibilitó proveer solo el volumen de agua requerido y prevenir tanto la saturación del sustrato como un riego excesivo. La humedad del suelo se conservó entre el 40% y el 70% gracias a esta regulación automática, lo que aseguró que las condiciones fueran ideales para el crecimiento de las plantas en el invernadero.

## 8.4.2 Estabilidad del microclima

|  |  |  |
| --- | --- | --- |
| Variable | Exterior | Interior |
| Temp. mínima | 7°C | 12°C |
| Temp. máxima | 28°C | 26–28°C |
| Humedad relativa | 35% | 45–60% |

## Tabla 26. Resultados de las variables de temperatura y humedad en el invernadero. *Elaborado a partir de Huerta et al. (2020), Sivamani et al. (2021) y Kaya et al. (2025).*

## Interpretación: El prototipo protege al cultivo de las heladas nocturnas y reduce el estrés hídrico generado por el ambiente seco de altura.

## 8.5 Discusión de los resultados

## 8.5.1 Eficacia del riego

## Los resultados logrados muestran que el sistema automatizado de riego logró un promedio de ahorro del 29% de agua, lo cual confirma que el prototipo proporciona solamente la cantidad requerida en función del grado de humedad real del terreno. Esta conducta es consistente con las investigaciones sobre riego tecnificado que ha presentado el MIDAGRI, que señalan una disminución entre 25% y 35% del consumo de agua por medio de la automatización. El sensor FC-28 demostró tener una buena sensibilidad ante las variaciones de humedad en los suelos de Chihuaco, pero la textura arenosa con arcilla causó ligeros cambios en las lecturas, lo que hizo necesario calibrar apropiadamente para garantizar valores fiables a lo largo de los ensayos.

## 8.5.2 Estabilidad térmica del invernadero

## El invernadero automatizado mantuvo un microclima interno más estable que las condiciones externas típicas del área altoandina. Se observó un aumento de la temperatura de entre 4°C y 6°C en comparación con el entorno exterior, lo cual posibilitó disminuir la posibilidad de perjuicios por heladas nocturnas. Además, la temperatura interna tuvo menor variación y los niveles de humedad relativa fueron más apropiados para cultivos de ciclo corto. La reducción del estrés térmico e hídrico, propio de las siembras que se encuentran expuestas directamente al clima de altura, fue posible gracias a estos factores, lo que ayudó a generar un entorno propicio para el crecimiento vegetal.

## 8.5.3 Viabilidad tecnológica para la colectividad

## La realización de una evaluación integral del prototipo demuestra que su implementación es factible para la comunidad de Chihuaco. El sistema logró funcionar sin necesidad de supervisión continua, con un consumo reducido de electricidad y con sensores que son de bajo coste, fácilmente sustituibles y accesibles en zonas rurales. También es posible duplicar o expandir su diseño modular en invernaderos de mayor tamaño, como las estructuras de 3×4 m que se utilizan frecuentemente en la región. Esto evidencia que la tecnología basada en Arduino es una opción sostenible, accesible y flexible para ajustarse a las necesidades de los cultivos de comunidades altoandinas con escasos recursos

1. **LECCIONES APRENDIDAS(MIN 3) POR PMV**

Desde la planificación, desarrollo e implementación de nuestro prototipo pensando en la comunidad de Chihuaco, nuestro equipo identificó varios aspectos críticos que no fueron contemplados en nuestra fase inicial.

### 9.1. Calibración de Sensores en Campo (PMV 1 - Riego)

En la implementación de los sensores de humedad del suelo necesitamos una calibración extra ya que no los valores eran muy distintos, y nos dimos cuenta que durante las pruebas del suelo seco y suelo húmedo los valores varían de manera drástica y tenemos que ajustarlo según al tipo de suelo.

Durante las pruebas se observó una corrosión leve en el sensor FC-28 en un ambiente de humedad constante. Por ello aprendimos que se necesita cambiar los sensores en una versión más avanzada o en su defecto el uso de materiales que ayuden a la protección del sensor como el uso de una capa de barniz epóxico para así garantizar una vida útil mayor.

### 9.2. Importancia de la Autonomía Energética (PMV 2 - Fiabilidad)

En las zonas rurales nuestro sistema automatizado no solo puede depender de la energía eléctrica comercial sino que necesitamos algo de respaldo en caso de apagón. Identificamos que los cortes de energía podrían dañar nuestro sistema y es por eso que decidimos incorporar una batería de respaldo o UPS.

### 9.3. Diseño Modular para el Mantenimiento (PMV 3 - Usabilidad)

La durabilidad de nuestro proyecto depende de un continuo funcionamiento del sistema y por ende su fácil funcionamiento y fácil mantenimiento. y es por eso que los sensores estarían ubicados en zonas de fácil acceso para que puedan ser reemplazados fácilmente, junto a eso la implementación de manuales para su adecuado funcionamiento y mantenimiento.

1. **CONCLUSIONES**

La implementación del invernadero automatizado se logró con éxito. El sistema demostró activar eficazmente la bomba de agua cuando el suelo está seco para así mantener a los cultivos en óptimas condiciones, evitando también un uso inadecuado del agua y un riego excesivo sin razón Se demostró que el uso del prototipo logra un ahorro promedio del 29% en comparación al método tradicional. A pesar de esto se concluye que la fiabilidad de esta funcionalidad de riego está relacionada con una calibración adecuada y adaptación al tipo de suelo donde se trabajara ya que puede existir variaciones en las lecturas de los sensores y puede conllevar a tener cultivos regados de forma ineficiente. Por lo tanto se requiere un ajuste de umbrales para tener mediciones más consistentes.

Por otro lado, se cumplió el objetivo de automatizar la medición ambiental. El sensor DHT11 es un componente de bajo costo lo que lo hace adecuado y suficiente para medir la humedad relativa y temperatura. El uso del invernadero logró estabilizar el microclima interno manteniendo temperatura entre 4 y 6 grados más altas que el exterior que oscilan entre los 12 y 7 grados celsius protegiendo así los cultivos de las heladas nocturnas que son comunes en zonas altoandinas. A pesar del cumplimiento funcional se concluye que la autonomía energética es un punto muy importante y crítico que no y de las debe ser opcional ya que el corte eléctrico suele ser común en zonas rurales como Chihuaco que están un poco alejados de la zona urbana lo que puede afectar en cualquier momento los cultivos en caso dependan únicamente de energía eléctrica. La incorporación de una batería de respaldo es indispensable para asegurar que el sistema funcione siempre sin interrupciones y así proteger la integridad de los datos.

Por último, la evaluación general de campo confirmó la viabilidad de poder usar hardware y software de bajo costo que fortalezcan la agricultura familiar en Chihuaco. Se concluye que el enfoque de este proyecto empodera al usuario final que en este caso son los agricultores locales y no lo hace ajeno a ello. Esto permite que el agricultor pueda hacer mantenimientos básicos al invernadero, y pueda reemplazar componentes como sensores que se dañen con el tiempo sin la necesidad de tener conocimientos avanzados en electrónica lo que quita la dependencia de contratar o llamar a un especialista en casos de averías.

**Referencias**

1. Ministerio de Desarrollo Agrario y Riego – MIDAGRI. MIDAGRI avanza en la ejecución de más de 130 proyectos de riego tecnificado en 19 regiones del país. 11 julio 2025.
2. Orozco, J., et al. Greenhouse Monitoring and Automation Using Arduino. International Journal of Applied Engineering Research, 2021.
3. Zhang, Y., et al. Intelligent Agriculture Monitoring and Control System Based on Arduino UNO Microcontroller. International Journal of Electronics and Electrical Engineering, 2022.
4. Marín‑García, E.; Torres‑Marín, J. N.; Chaverra‑Lasso, A. Smart Greenhouse and Agriculture 4.0. Bogotá – Colombia: Revista Científica, N. 46(1), 2023, pp. 37‑50. dialnet.unirioja.es.
5. Berríos Gómez, S.; Rivera Herrera, H. J. ESP32‑based IoT system for controlling and monitoring greenhouse crops with an agriculture 4.0 approach. Tacna: Universidad Privada de Tacna, 2022. [alicia.concytec.gob.pe](http://alicia.concytec.gob.pe).
6. Raquel-Salazar, M.; Ana-Sánchez M.; Irineo-López, C. Indicators for assessing water, energy and labor use performance in a low-tech greenhouse Indicadores para evaluar el desempeño en el uso del agua, energía y mano de obra en un invernadero de baja tecnología <http://dx.doi.org/10.5154/r.rchsh.2019.09.018>
7. Audberto-Reyes,R.; Raúl-Rodríguez G.; Alejandro-Zermeño G.; Diana- Jasso C.; Martín-Cadena Z.; Héctor-Burgueño C. Revista Chapingo Serie Horticultura 18(1): 125-140, 2012 Recibido: 30 de octubre, 2009 aceptado: 1 de febrero, 2012 evaluación de un modelo para estimar la temperatura y humedad relativa en el interior de invernadero con ventilación natural Revista Chapingo Serie Horticultura 18(1): 125-140, 2012 <https://www.scielo.org.mx/pdf/rcsh/v18n1/v18n1a9.pdf>