|  |
| --- |
|  |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| **Proyecto: [SINH20]** |
|  |
| **Revisión: [06]** |
| **[06/09/24]** |

|  |
| --- |
|  |

Contenido

[Ficha del documento: 4](#_Toc185175465)

[1. Introducción: 5](#_Toc185175466)

[1.1. Propósito: 5](#_Toc185175467)

[1.2. Ámbito del Sistema: 5](#_Toc185175468)

[1.3. Definiciones, Acrónimos y Abreviaturas: 6](#_Toc185175469)

[1.4. Referencias: 6](#_Toc185175470)

[1.5. Visión General del Documento: 7](#_Toc185175471)

[2. Descripción General: 7](#_Toc185175472)

[2.1. Perspectiva del Producto: 7](#_Toc185175473)

[2.2. Funciones del Producto: 7](#_Toc185175474)

[2.3. Características de los Usuarios: 8](#_Toc185175475)

[2.4. Restricciones: 8](#_Toc185175476)

[2.5. Suposiciones y Dependencias: 8](#_Toc185175477)

[2.6. Requisitos Futuros: 9](#_Toc185175478)

[3. Requisitos Específicos 9](#_Toc185175479)

[3.1 Requisitos comunes de las interfaces: 11](#_Toc185175480)

[3.1.1 Interfaces de usuario: 15](#_Toc185175481)

[3.1.2 Interfaces de hardware: 15](#_Toc185175482)

[3.1.3 Interfaces de software: 16](#_Toc185175483)

[3.1.4 Interfaces de comunicación: 17](#_Toc185175484)

[3.2 Requisitos funcionales: 19](#_Toc185175485)

[3.3 Requisitos no funcionales: 22](#_Toc185175486)

[3.3.1 Requisitos de rendimiento: 22](#_Toc185175487)

[3.3.2 Seguridad: 23](#_Toc185175488)

[3.3.3 Fiabilidad: 23](#_Toc185175489)

[3.3.4 Disponibilidad 23](#_Toc185175490)

[3.3.5 Mantenibilidad: 24](#_Toc185175491)

[3.3.6 Portabilidad: 24](#_Toc185175492)

[3.4 Otros Requisitos: 24](#_Toc185175493)

[4. Propuesta de Planificación: 25](#_Toc185175494)

[4.1 Descripción general acerca de la Planificación: 25](#_Toc185175495)

[4.1.2 Definición del Equipo de Trabajo: 26](#_Toc185175496)

[4.1.3 Definición de Actividades principales del Proyecto: 26](#_Toc185175497)

[4.1.4 Diagrama EDT: 28](#_Toc185175498)

[4.1.5 Carta Gantt: 29](#_Toc185175499)

[4.1.6 Resumen Costos del Desarrollo del Proyecto: 29](#_Toc185175500)

[4.2 Plan de Control de Cambio: 30](#_Toc185175501)

[5. Anexos 30](#_Toc185175502)

[5.1 Acta de Proyecto: 30](#_Toc185175503)

[5.2 Matriz Especificación de Requerimientos: 30](#_Toc185175504)

[5.3 Diagrama de Casos de Uso General: 31](#_Toc185175505)

[32](#_Toc185175506)

[5.4 Planilla Casos de Uso: 32](#_Toc185175507)

[5.5 Prototipado de Software: 33](#_Toc185175508)

[5.6 Resultado Análisis de Calidad Diagramas Modelamiento: 36](#_Toc185175509)

[5.7 Resultado Análisis de Calidad Prototipo No funcional del Sistema: 36](#_Toc185175510)

[5.8 Planilla entregables del Proyecto: 36](#_Toc185175511)

[5.9 Matriz de Control de Cambios: 36](#_Toc185175512)

[5.10 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo: 36](#_Toc185175513)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Ficha del documento:

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| 27/08/24 | **1.0** | Matias Mora Recabarren | Introducción, propósito, ámbito del sistema, definiciones, acrónimos y abreviaturas. |
| 29/08/24 | **2.0** | Matias Mora Recabarren | Descripción general, perspectiva del producto, funciones del producto, características del usuario, restricciones, suposiciones y dependencias, requisitos futuros. |
| 30/08/24 | **3.0** | Matias Mora, Sebastián Pino y Andrés Silva. | Requisitos específicos, requisitos comunes de las interfaces, interfaces de usuario, interfaces de hardware, Interfaces de software, Interfaces de comunicación, requisitos funcionales, no funcionales, de rendimiento y seguridad, Fiabilidad, Disponibilidad, Mantenibilidad, planificación, definición del equipo de trabajo, definición de actividades, Diagrama EDT, Carta Gantt, Resumen de costos, Acta de proyecto y Matriz EDT. |
| 04/09/24 | **4.0** | Matias Mora Recabarren | Visión General del documento, Portabilidad, Otros requisitos |
| 05/09/24 | **5.0** | Matias Mora, Sebastián Pino y Andrés Silva. | Plan de control de cambios, Referencias, matriz de especificación de requerimientos, Diagrama de casos de uso general, Planilla casos de uso, prototipo de software y planilla entregables del proyecto. |
| 06/09/24 | **6.0** | Matias Mora Recabarren | Matriz de control de cambios, |

Integrantes:

|  |  |
| --- | --- |
| **Nombre Integrante del Equipo** | **Rol Definido** |
| Matias Mora | Jefe de proyecto, Desarrollador. |
| Sebastián Pino | Desarrollador Full Stack |
| Andrés Silva | Especialista machine Learning/QA |

# 1. Introducción:

El uso eficiente de los recursos hídricos es un reto clave en la agricultura moderna, especialmente en regiones con acceso limitado al agua. Para hacer frente a este reto, el equipo Pentium D ha desarrollado el sistema de riego automatizado (SINH2O), que utiliza sensores de humedad del suelo y controladores para optimizar el riego.

Este sistema está diseñado para proporcionar la cantidad exacta de agua necesaria en cada sector de cultivo, evitando el despilfarro de recursos y contribuyendo a la sostenibilidad agrícola.

## 1.1. Propósito:

El propósito principal de este documento es proporcionar las especificaciones y requisitos del proyecto SINH2O, desarrollado por el equipo Pentium D. Su objetivo es ofrecer una solución eficaz y moderna a la gestión del riego, optimizando el uso del agua en función de la humedad del suelo para evitar su degradación.

Este documento también detalla las funcionalidades clave del sistema, así como otros aspectos técnicos relevantes para su desarrollo y funcionamiento.

## 1.2. Ámbito del Sistema:

El nombre completo del proyecto es: Sistema de irrigación por nivel de humedad de 2 orientaciones.

El sistema de irrigación automatizado SINH2O estará diseñado para:

1. Monitorear los niveles de humedad del suelo en diferentes sectores de cultivo mediante sensores de humedad conectados a un microcontrolador.
2. Automatizar el riego activando el flujo de agua solo en las áreas donde el nivel de humedad esté por debajo de un umbral predefinido.
3. Optimizar el uso del agua, proporcionando la cantidad exacta necesaria en cada sección de cultivo, evitando el desperdicio y asegurando que los recursos se utilicen de manera eficiente.
4. Registrar y almacenar datos de humedad del suelo y temperatura ambiente en una base de datos, para análisis y optimización futura del riego.
5. Generar reportes y alertas a través de una interfaz de usuario gráfica (aplicación de escritorio en Python con Tkinter), permitiendo al usuario visualizar los niveles de humedad y el estado de los estanques de agua.
6. Permitir ajustes manuales y monitoreo remoto a través de la interfaz, de modo que el usuario pueda modificar parámetros del sistema o verificar su rendimiento en tiempo real.

El sistema SINH2O no estará diseñado para:

1. Predecir cambios climáticos o ajustar el riego en función de datos meteorológicos externos. El sistema solo actuará en función de los niveles de humedad del suelo y no de factores climáticos futuros.
2. Detectar y corregir automáticamente fallas en los sensores, tuberías o componentes del sistema. La detección de fallas requerirá intervención humana.
3. Gestionar otros factores agrícolas como el control de plagas o la calidad del agua utilizada para el riego.
4. Funciona sin conexión eléctrica, ya que el sistema depende de fuentes de energía y conectividad para operar correctamente.

El sistema SINH2O se desarrollará con los siguientes objetivos principales:

1. Automatizar el proceso de riego en función de los niveles de humedad del suelo, minimizando la intervención humana y mejorando la precisión en el uso del agua.
2. Proporcionar una plataforma de monitoreo centralizada para visualizar los datos de humedad y controlar el sistema de riego desde una aplicación de escritorio.
3. Contribuir a la sostenibilidad agrícola reduciendo el uso innecesario de agua y mejorando la eficiencia de los recursos en el campo.

## 1.3. Definiciones, Acrónimos y Abreviaturas:

|  |  |
| --- | --- |
| ACRÓNIMOS O ABREVIATURAS | DEFINICIONES |
| SINH2O | Sistema de irrigación por nivel de humedad de 2 orientaciones. |
| RF | Requerimiento funcional |
| RNF | Requerimiento No funcional |
| RP | Requerimiento de rendimiento |
| RS | Requerimiento de seguridad |

## 1.4. Referencias:

* Agile Alliance. (2020). *Manual de buenas prácticas de desarrollo ágil*. Agile Alliance.
* Arduino.cc. (2023). *Documentación técnica de Arduino Uno*. Recuperado de <https://www.arduino.cc>
* Instituto de Ingenieros Eléctricos y Electrónicos. (1998). *Prácticas recomendadas para las especificaciones de requisitos de software*. IEEE Standards Association.
* ISO. (2021). *Ergonomía de la interacción persona-sistema. Parte 110: Principios de diseño de interacción* (ISO 9241-110:2020). ISO.
* Project Management Institute. (2021). *Guía del PMBOK: Guía de los fundamentos para la dirección de proyectos* (7ª ed.). Project Management Institute.

## 1.5. Visión General del Documento:

Este documento presenta los requisitos específicos para el sistema S.I.N.H.2.O a lo largo del documento se detallan las características funcionales y no funcionales del sistema, así como los casos de uso que permitirán guiar el desarrollo y garantizar que el sistema cumpla con las expectativas del cliente y los usuarios finales.

El documento está organizado de la siguiente manera:

1. Descripción general del sistema, incluyendo su contexto, propósito y objetivo.
2. Requisitos funcionales, donde se describen las funcionalidades que debe ofrecer el sistema.
3. Requisitos no funcionales, los cuales incluyen aspectos de rendimiento, seguridad y mantenibilidad.
4. Casos de uso y escenarios que muestran interacciones típicas con el sistema.

# 2. Descripción General:

En esta sección se analiza el contexto general del Proyecto SINH2O, teniendo en cuenta los factores que influyen en su desarrollo y funcionamiento.

Se ofrecerá una visión general de cómo funcionará el sistema, lo que permitirá comprender en mayor profundidad los requisitos descritos a continuación.

## 2.1. Perspectiva del Producto:

El proyecto SINH2O se enmarca en el contexto de la agricultura sostenible y eficiente, donde el uso óptimo del agua es un factor clave para mejorar la productividad agrícola y reducir el impacto ambiental.

El sistema está diseñado para automatizar y optimizar el riego de cultivos mediante el uso de sensores de humedad, controladores de válvulas de agua y un sistema de monitoreo centralizado a través de una aplicación de escritorio.

Su implementación tiene como objetivo modernizar la gestión agrícola.

## 2.2. Funciones del Producto:

1. Monitoreo en tiempo real de los niveles de humedad del suelo en diversas áreas de cultivo.
2. Activación y desactivación automática del riego cuando los niveles de humedad superen o bajen de umbrales predefinidos.
3. Registro y análisis de datos históricos de humedad del suelo y temperatura para optimización futura.
4. Notificación al usuario y ajustes manuales a través de la interfaz gráfica, permitiendo al usuario intervenir o monitorear el sistema de forma remota.
5. Optimización del uso de agua, aplicando solo la cantidad necesaria en cada área de cultivo.

## 2.3. Características de los Usuarios:

|  |  |
| --- | --- |
| Tipo de usuario | Descripción |
| Agricultores y encargados de riego | Usuarios que buscan mejorar la eficiencia en el riego de sus cultivos. Estos usuarios tendrán acceso a la aplicación de escritorio para monitorear el sistema |
| Técnicos de mantenimiento | Usuarios encargados de la instalación, configuración inicial y mantenimiento del sistema. Estos requieren acceso tanto del hardware como del software. |
| Administradores del sistema | Responsables de la gestión general del sistema, con acceso completo a los registros de datos y opciones de control. |

## 2.4. Restricciones:

* Limitación del presupuesto
* Utilización de infraestructura de riego existente
* Disponibilidad de agua estable
* Conectividad eléctrica estable
* Capacidad de procesamiento
* Condiciones climáticas extremas
* Capacitación de los usuarios
* Tiempo de desarrollo
* Regulaciones ambientales o agrícolas

## 2.5. Suposiciones y Dependencias:

1. Disponibilidad de sensores y hardware compatibles  
   Se asume que los sensores de humedad del suelo y los controladores Arduino necesarios para el sistema estarán disponibles y se integrarán sin problemas.
2. Acceso a suministro eléctrico constante  
   Se presupone que las áreas donde se instalará SINH2O tendrán acceso estable a una fuente de energía. Si la conectividad eléctrica es intermitente, el sistema no funcionará correctamente, y podrían necesitar fuentes de energía alternativas.
3. Condiciones climáticas moderadas  
   Se asume que los sensores y otros componentes del sistema funcionarán en condiciones climáticas moderadas. Si el clima en las áreas agrícolas es extremo, será necesario implementar medidas adicionales para proteger los equipos.
4. Normativas locales sobre el uso del agua  
   Se asume que el sistema cumplirá con las regulaciones locales en cuanto a la gestión del agua.
5. Presupuesto y cronograma establecidos  
   Se supone que el presupuesto y el cronograma del proyecto serán suficientes para cumplir con los requisitos establecidos.
6. Expansión futura del sistema  
   Se presupone que el sistema se diseñará de manera modular, permitiendo futuras expansiones para abarcar más terreno.

## 2.6. Requisitos Futuros:

1. Integración con servicios meteorológicos  
   En futuras versiones, se podría incorporar la capacidad de acceder a datos meteorológicos en tiempo real para ajustar automáticamente los ciclos de riego.
2. Automatización avanzada del riego  
   Se podría desarrollar un algoritmo de riego más avanzado que no solo se base en los niveles de humedad, sino que también analice el tipo de suelo, el ciclo de cultivo y la topografía del terreno.
3. Interfaz web y móvil  
   En versiones futuras, se podría desarrollar una interfaz de usuario basada en web o una aplicación móvil que permita a los usuarios acceder al sistema desde cualquier dispositivo con acceso a internet, ampliando las opciones de monitoreo y control.
4. Soporte para energías renovables  
   En futuras versiones, se podría optimizar el sistema para funcionar con fuentes de energía renovable.

# 3. Requisitos Específicos

1. El sistema SINH2O debe permitir a los usuarios visualizar los niveles de humedad de cada sección del campo en una cuadrícula, con columnas numéricas y filas alfabéticas.
2. Los usuarios deben poder seleccionar una sección específica de la cuadrícula para obtener detalles adicionales del nivel de humedad de esa área.
3. El sistema debe permitir a los usuarios fijar umbrales de humedad para cada sección y activar el riego automáticamente cuando los niveles caigan por debajo de esos umbrales.
4. El sistema debe entregar alertas en caso de que los niveles de humedad superen o caigan por debajo de los valores definidos.
5. El sistema debe incluir un apartado donde se pueda ver el historial de los niveles de humedad.
6. El sistema debe permitir la visualización y gestión del estado de los estanques de agua, incluyendo el nivel de agua restante en cada uno.
7. El sistema debe actualizar los niveles de humedad y el estado de los estanques cada 10 segundos.
8. La interfaz debe tener un tiempo de carga máximo de 5 segundos al iniciar la aplicación.
9. La interfaz gráfica debe ser sencilla, con colores que faciliten la visualización de los datos.
10. La capacidad de almacenamiento de datos históricos debe ser de al menos 1 año, registrando los niveles de humedad cada 1 hora.
11. El sistema debe procesar la entrada de datos de los sensores y actualizar los niveles de humedad en pantalla en menos de 5 segundos.
12. La interfaz gráfica debe cargar y actualizar todos los datos en menos de 5 segundos después de iniciar sesión.
13. El sistema debe soportar la comunicación con hasta 100 sensores de humedad distribuidos en el campo sin pérdida de rendimiento.
14. Las alertas críticas deben enviarse a los usuarios en menos de 1 segundo después de que el sistema detecte condiciones fuera de los parámetros establecidos.
15. El sistema debe responder a una solicitud de riego manual en menos de 3 segundos después de que el usuario lo active.
16. El sistema debe autenticar a los usuarios mediante un sistema de usuario y contraseña con cifrado.
17. Los usuarios deben ser bloqueados después de 5 intentos fallidos de inicio de sesión.
18. Los datos históricos deben estar protegidos contra modificaciones, permitiendo solo la lectura.
19. Las comunicaciones entre la interfaz de usuario y el backend del sistema deben estar protegidas contra inyecciones SQL.

## 3.1 Requisitos comunes de las interfaces:

Entradas del sistema:

1. Sensor de humedad:

* Descripción: Los sensores de humedad proporcionan datos en tiempo real sobre los niveles de humedad del suelo.
* Formato: Señal analógica convertida a un valor digital entre 0-100% de humedad.
* Frecuencia: Cada 5 segundos.
* Origen: Sensores de humedad conectados al Arduino Uno.
* Acciones: Estos datos son procesados para determinar si es necesario activar el riego o alertar al usuario.

1. Configuración de umbrales de humedad:

* Descripción: El usuario ingresa los umbrales de humedad, que activan el riego automático.
* Formato: Valores numéricos (0-100%).
* Frecuencia: Entrada manual, puede ser modificada en cualquier momento.
* Origen: Pantalla de configuración de la interfaz de usuario.
* Acciones: Estos valores determinan cuándo el sistema debe activar el riego.

1. Activación manual de riego:

* Descripción: El usuario puede iniciar manualmente el riego de una o varias secciones desde la interfaz.
* Formato: Botón de selección con el estado ON/OFF para cada sección de la cuadrícula.
* Frecuencia: Entrada manual.
* Origen: Pantalla principal de la interfaz de usuario.
* Acciones: Activa o desactiva la bomba de riego correspondiente.

1. Datos de temperatura:

* Descripción: Los sensores de temperatura reportan el estado climático en el área del cultivo, lo cual puede influir en la frecuencia de riego.
* Formato: Datos numéricos en grados Celsius.
* Frecuencia: Cada 10 minutos.
* Origen: Sensores de temperatura conectados al sistema.
* Acciones: Estos datos son utilizados para ajustar la lógica de riego en base a condiciones climáticas.

1. Nivel de agua en los estanques:

* Descripción: Sensores en los estanques miden la cantidad de agua disponible para el riego.
* Formato: Datos numéricos (litros y porcentaje).
* Frecuencia: Cada 30 segundos.
* Origen: Sensores de nivel de agua en los estanques.
* Acciones: Los datos son utilizados para gestionar la capacidad de riego y alertar al usuario en caso de niveles bajos.

1. Identificación de usuario y autenticación:

* Descripción: El sistema recibe las credenciales de los usuarios para autenticarlos antes de permitir el acceso.
* Formato: Campos de texto (usuario y contraseña).
* Frecuencia: Cada inicio de sesión.
* Origen: Pantalla de inicio de sesión de la interfaz de usuario.
* Acciones: Autentica al usuario.

1. Alarmas:

* Descripción: El sistema recibe configuraciones de las alertas que los usuarios desean recibir (sobre humedad, nivel de agua, temperatura, etc.).
* Formato: Parámetros de configuración seleccionados en la interfaz.
* Frecuencia: Entrada manual.
* Origen: Pantalla de configuración de notificaciones.
* Acciones: Configura cuándo y cómo el sistema notificará a los usuarios sobre condiciones críticas.

Salidas del sistema:

1. Visualización de niveles de humedad:

* Descripción: El sistema muestra el estado actual de humedad para cada sección en la cuadrícula.
* Formato: Valores numéricos.
* Frecuencia: Actualización cada 5 segundos.
* Destino: Pantalla principal de la interfaz de usuario.

1. Historial de humedad:

* Descripción: El sistema genera gráficos que muestran los niveles de humedad de cada sección durante los últimos 7 días.
* Formato: Gráficos.
* Frecuencia: Consultable en cualquier momento.
* Destino: Pantalla de historial de la interfaz de usuario.

1. Alertas de nivel crítico:

* Descripción: El sistema envía alertas visuales, cuando los niveles de humedad o el nivel de agua en los estanques alcanzan valores críticos.
* Formato: Mensaje emergente.
* Frecuencia: En tiempo real, cada vez que se detecta una condición crítica.
* Destino: Pantalla principal de la interfaz de usuario.

1. Estado del estanque de agua:

* Descripción: Muestra el nivel de agua disponible en los dos estanques de la granja.
* Formato: Gráficos visuales de "tubos" que representan la capacidad restante (porcentaje o litros).
* Frecuencia: Actualización cada 30 segundos.
* Destino: Pantalla principal de la interfaz de usuario.

1. Activación de las bombas de riego:

* Descripción: El sistema envía señales a las bombas de riego para activar o desactivar el flujo de agua según los niveles de humedad.
* Formato: Señal de activación/desactivación (ON/OFF).
* Frecuencia: Cada vez que el sistema detecta una necesidad de riego.
* Destino: Bombas de riego en el sistema físico.

1. Confirmación de inicio de sesión:

* Descripción: El sistema confirma si la autenticación del usuario fue exitosa y carga la interfaz.
* Formato: Mensaje de texto o redirección a la pantalla principal.
* Frecuencia: Cada inicio de sesión.
* Destino: Pantalla de inicio de sesión de la interfaz de usuario.

1. Reporte del sistema:

* Descripción: Genera un informe con el estado de las secciones, niveles de humedad, riegos realizados, y capacidad de los estanques en un período determinado.
* Formato: Archivo PDF o CSV.
* Frecuencia: Entrada manual para generación de reportes.
* Destino: Descargable en la interfaz de usuario.

1. Respuesta de error o estado invalido:

* Descripción: El sistema muestra mensajes de error si los sensores envían datos fuera del rango esperado o si hay fallos de comunicación.
* Formato: Mensaje emergente de advertencia.
* Frecuencia: Cada vez que ocurre un error.
* Destino: Pantalla principal de la interfaz de usuario

### 3.1.1 Interfaces de usuario:

Interfaz gráfica: Desarrollada en Python usando Tkinter.

Pantalla principal: Muestra una cuadrícula (de A hasta la Z en columnas y 1 a 9 en filas) donde cada celda representa una sección del terreno. Las celdas muestran los valores de humedad en tiempo real.

Visualización de estanques de agua: En la pantalla principal, dos tubos verticales simulan el nivel de agua disponible en los estanques. Estos tubos se actualizan dinámicamente a medida que se utiliza o repone el agua.

Botones de control manual: Permiten activar o desactivar el riego de forma manual en cada una de las secciones de la cuadrícula.

Configuración: Los usuarios pueden ajustar los umbrales de humedad de cada sección, configurando el nivel mínimo de humedad antes de activar el riego automático.

Notificaciones y alertas: La interfaz muestra alertas visuales en caso de niveles de humedad críticos o baja capacidad de los estanques.

Inicio de sesión: Los usuarios deben autenticarse para acceder al sistema.

### 3.1.2 Interfaces de hardware:

Sensores de humedad:

* Descripción: dispositivos colocados en las diferentes secciones del terreno que miden el nivel de humedad en el suelo.
* Comunicación: Los sensores están conectados al Arduino a través de pines analógicos, los cuales convierten la señal de humedad en datos porcentuales del 0 al 100%.

Sensores de temperatura:

* Descripción: Miden la temperatura en el área de cultivo. influyen en la decisión de activar el riego o no.
* Comunicación: Al igual que el sistema de humedad, están conectados al Arduino uno.

Actuadores de bombas de riego:

* Descripción: Las bombas de riego se activan automáticamente cuando los niveles de humedad son inferiores a los umbrales configurados.
* Comunicación: Controladas por señales digitales enviadas desde el Arduino Uno.

Arduino uno:

* Descripción: Es el controlador principal que actúa como interfaz entre los sensores, las bombas de agua y el software de la aplicación.
* Comunicación: El Arduino Uno recibe los datos de los sensores (humedad y temperatura), los procesa y los envía a la aplicación a través de una conexión serial. A su vez, recibe instrucciones del software para controlar las bombas de agua.

Sensores de nivel de agua:

### 3.1.3 Interfaces de software:

* Sistema operativo: Windows 10 u 11.

El sistema SINH2O está diseñado para ejecutarse en sistemas operativos Windows 10 o Windows 11. La aplicación de escritorio creada con Python y Tkinter será compatible con estas versiones de Windows, asegurando su correcto funcionamiento en un entorno moderno y ampliamente utilizado.

* Producto de software: Arduino IDE.

Arduino IDE es una plataforma utilizada para escribir, compilar y cargar código en el microcontrolador Arduino Uno.

Propósito: Este software es esencial para el desarrollo y carga del código que controlará el hardware del sistema SINH2O, incluyendo la lectura de sensores y el control de las electroválvulas.

Definición:

Contenido: Código de fuente en lenguaje: “Python”.

Formato: Archivos con extensión “. ino”

* Producto de software: Base de datos relacional.

Se considera utilizar MySQL o SQLite como base de datos relacional.

Propósito: La base de datos se utilizará para almacenar datos históricos de humedad y temperatura, permitiendo el análisis posteriores y optimización del riego.

Definición:

Contenido: Datos de la humedad del suelo, temperatura ambiente, estado de las electroválvulas, y configuración del sistema.

Formato: Tablas relacionales, con registros en formatos SQL.

* Producto de software: Python.

Python es el lenguaje de programación utilizado para desarrollar la lógica de negocio del sistema y la interfaz de usuario.

Propósito: Ejecutar el código que gestiona la comunicación entre el Arduino, la base de datos, y la interfaz de usuario.

Definición:

Contenido: Código de fuente en Python que incluye scripts para la lógica del sistema, el manejo de eventos en la interfaz gráfica, y la comunicación con el hardware.

Formato: Archivos “.py”.

* Producto de software: Tkinter.

Tkinter es el framework utilizado para desarrollar la interfaz gráfica de usuario del sistema SINH2O.

Propósito: Proporcionar una interfaz gráfica interactiva para que el usuario final pueda monitorear y controlar el sistema de riego.

Definición:

Contenido: Elementos GUI como botones, gráficos, y cuadros de texto, que interactúan con el usuario y con los scripts de Python.

Formato: La interfaz se define y controla a través del código Python que utiliza las clases y funciones de Tkinter.

### 3.1.4 Interfaces de comunicación:

Tipo de interfaz: Hardware.

1.- Microcontrolador: Arduino Uno

El Arduino Uno será el cerebro del sistema, encargado de recibir datos de los sensores de humedad y temperatura, y de controlar las electroválvulas o la bomba de agua. Actuará como el intermediario entre los sensores, actuadores y la aplicación de escritorio.

2.- Sensores de Humedad y temperatura:

Sensor de Humedad: FC-28

Sensor de temperatura: DHT11

Los sensores FC-28 y DHT11 enviarán señales al Arduino Uno. El sensor FC-28 proporcionará datos analógicos sobre la humedad del suelo, mientras que el DHT11 enviará datos analógicos sobre la temperatura ambiental.

3.- Actuadores: Electroválvulas o Bomba de agua

El Arduino controlará las electroválvulas o la bomba de agua mediante relés de control, para activar o desactivar el flujo de agua basado en las lecturas de humedad.

4.- Sensores Ultrasónicos:

Los sensores ultrasónicos medirán el nivel de los estanques de agua y enviarán señales al Arduino para determinar cuándo es necesario activar la bomba de agua.

5.- Relés de control:

Los relés actuarán como interruptores controlados por el Arduino para gestionar el encendido y apagado de las electroválvulas o bombas de agua, asegurando el control seguro de dispositivos de mayor potencia.

Tipo de interfaz: Software.

1.- Lenguaje de programación: Python.

Python será utilizado para desarrollar la lógica del sistema y la interfaz de usuario. Se conectará con el Arduino para recibir y enviar datos a través de un puerto de serie.

2.- Framework para la interfaz gráfica: Tkinter.

Tkinter se utilizará para crear la aplicación de escritorio con una interfaz gráfica de usuario. Esta interfaz permitirá a los usuarios monitorear los niveles de humedad y temperatura, y controlar el sistema de riego.

3.- Base de datos: (Por definir)

Los datos de humedad y temperatura serán almacenados en una base de datos relacional, permitiendo la posterior consulta y análisis para optimizar las estrategias de riego.

Tipo de interfaz: Usuario.

1.- Aplicación de escritorio:

La interfaz gráfica estará diseñada para usuarios finales que operen en sistemas Windows. Permitirá la visualización en tiempo real de los datos de sensores, el control del riego, y la configuración de parámetros del sistema. La interfaz será intuitiva y fácil de usar, dirigida tanto a usuarios técnicos como no técnicos.

## 3.2 Requisitos funcionales:

Requerimiento funcional 1: Registro de humedad de la tierra

* Actores: Usuario Administrador
* Descripción: Apartado en donde se puede ver la cantidad de humedad que ha tenido la tierra.

Requerimiento funcional 2: login

* Actores: Usuarios
* Descripción: Iniciar sesión

Requerimiento funcional 3: registro de usuario

* Actores: Usuarios
* Descripción: registro del usuario en el sistema

Requerimiento funcional 4: Recuperación de nombre de usuario y contraseña

* Actores: Sistema
* Descripción: Recuperar el nombre de usuario o contraseña mediante un correo electrónico.

Requerimiento funcional 5: Localización del dispositivo

* Actores: Usuario administrador
* Descripción: función en la que se puede ver la localización del dispositivo en tiempo real.

Requerimiento funcional 6: Apagado y encendido remoto

* Actores: Usuario administrador
* Descripción: Función que da la opción de poder apagar o encender a distancia en dispositivo

Requerimiento funcional 7: visualizador de diagrama

* Actores: All User
* Descripción: Apartado donde el usuario donde el usuario podrá visualizar a través de un diagrama el hardware que se encuentra activo/inactivo o disfuncional.

Requerimiento funcional 8: Ayuda y Guía de uso

* Actores: Usuarios
* Descripción: Soporte donde tener ayuda directamente del personal y una simple guía con la forma correcta de usar cada función.

Requerimiento funcional 9: Interfaz de opciones visuales

* Actores: All User
* Descripción: Apartado donde el usuario podrá realizar modificaciones visuales a la interfaz general.

Requerimiento funcional 10: Visualizar Temperatura

Actores: All User

Descripción: Apartado en la interfaz del inicio, que indique a tiempo real, la temperatura

Requerimiento funcional 11: Capacidad de agua

* Actores: All User
* Descripción: Apartado en la interfaz del inicio, que indique a tiempo real, la cantidad de agua restante en los estanques.

Requerimiento funcional 12: Humedad a tiempo real

* Actores: All User
* Descripción: Apartado en la interfaz del inicio, que indique a tiempo real, la humedad que posee el suelo.

Requerimiento funcional 13: Estado del clima

* Actores: Sistema
* Descripción: Apartado en dónde se muestre el clima a tiempo real.

Requerimiento funcional 14: Integración y eliminación de sensores

* Actores: Usuario
* Descripción: Gestión adecuada y cómoda para el usuario para la actualización de nuevos ascensores.

Requerimiento funcional 15: Configuración remota

* Actores: Usuarios administrador
* Descripción: Apartado en el que se puede cambiar valores de forma remota

Requerimiento funcional 16: Apagado de emergencia

* Actores: Usuarios administrador
* Descripción: Función del dispositivo que se activa de forma automática, en el momento en el cual se detecte un fallo.

Requerimiento funcional 17: Historial de recarga de agua

* Actores: Usuarios administrador y usuario
* Descripción: Apartado en donde se muestre en qué fecha y hora se recargo el tanque de agua del dispositivo.

Requerimiento funcional 18: Autentificar usuario al iniciar sesión

* Actores: Usuarios
* Descripción: Autenticar la identidad del usuario mediante su rut y nombre de usuario.

Requerimiento funcional 19: Apartado de sistema de medidas

* Actores: All User
* Descripción: Apartado donde el usuario podrá cambiar el sistema de medición de variables, ºC a K.

Requerimiento funcional 20: Apartado de versión del dispositivo

* Actores: All User
* Descripción: Apartado donde el usuario podrá visualizar la versión del software/hardware del dispositivo.

Requerimiento funcional 21: Detección de obstrucciones

* Actores: Sistema
* Descripción: Forma en la que el sistema puede dar aviso, si algún irrigador no está funcionando como debe ser.

Requerimiento funcional 22: Historial de errores

* Actores: All User
* Descripción: Serie de códigos para identificar cada error y tener mayor fluidez a la hora de contactar al servicio técnico.

Requerimiento funcional 23: Generador de reportes de errores

* Actores: Sistema
* Descripción: Cada mes el sistema mandará un reporte, en el cual informará el estado de cada componente del sistema.

## 3.3 Requisitos no funcionales:

Requerimiento no funcional 24: Navbar

* Actores: Sistema
* Descripción: Una interfaz que te permite ajustar el tamaño de las ventanas y apartados.

Requerimiento no funcional 25: Colores de interfaz

* Actores: Sistema
* Descripción: La relación de colores se distribuye en 60% base, 30% interactivo y 10% importantes.

Requerimiento no funcional 26: Selección de colores

* Actores: Sistema
* Descripción: Los colores seleccionados para el software consisten en degradados del color verde agua y color cian.

Requerimiento no funcional 27: tipo de caligrafía

* Actores: Sistema
* Descripción: El tipo de caligrafía seleccionada para los textos de la interfaz será Sans serif.

Requerimiento no funcional 28: Integración del modo oscuro

* Actores: Sistema
* Descripción: Los colores seleccionados para la interfaz en modo oscuro, consisten en color de bajo contraste y letras blancas.

### 3.3.1 Requisitos de rendimiento:

El sistema debe procesar la entrada de datos de los sensores y actualizar los niveles de humedad en pantalla en menos de 5 segundos.

La interfaz gráfica debe cargar y actualizar todos los datos en menos de 5 segundos después de iniciar sesión.

El sistema debe soportar la comunicación con hasta 100 sensores de humedad distribuidos en el campo sin pérdida de rendimiento.

Las alertas críticas deben enviarse a los usuarios en menos de 1 segundo después de que el sistema detecte condiciones fuera de los parámetros establecidos.

El sistema debe responder a una solicitud de riego manual en menos de 3 segundos después de que el usuario lo active.

### 3.3.2 Seguridad:

* El sistema debe autenticar a los usuarios mediante un sistema de usuario y contraseña con cifrado.
* Los usuarios deben ser bloqueados después de 5 intentos fallidos de inicio de sesión.
* Los datos históricos deben estar protegidos contra modificaciones, permitiendo solo la lectura.
* Las comunicaciones entre la interfaz de usuario y el Backend del sistema deben estar protegidas contra inyecciones SQL.

### 3.3.3 Fiabilidad:

La fiabilidad del sistema SINH2O, se especifica en los siguientes términos:

1. El tiempo medio entre los fallos, debe de ser de al menos 1 mes.
2. El sistema debe estar disponible más del 90% del tiempo.
3. El número máximo de fallos en los sensores será de 2 al año

### 3.3.4 Disponibilidad

El sistema SINH2O entregará los siguientes niveles de disponibilidad:

1. Disponibilidad mínima del 90%: El sistema debe estar disponible para los usuarios al menos el 90% del tiempo.
2. Horarios de operación: El sistema debe estar operativo 24/7, asegurando que los usuarios puedan acceder a las funcionalidades críticas en cualquier momento del día.
3. Tiempo de respuesta a fallos: Ante cualquier fallo crítico que afecte la disponibilidad del sistema, este debe ser restablecido en un plazo máximo de 1 hora.
4. Tiempo de mantenimiento planificado:  
   Los periodos de mantenimiento deben ser anunciados con 48 horas de antelación y siempre fuera de las horas de mayor uso.

### 3.3.5 Mantenibilidad:

|  |  |  |
| --- | --- | --- |
| Tipo de mantenimiento | Responsables del mantenimiento | Frecuencia del mantenimiento |
| Mantenimiento Preventivo: Incluye tareas como la actualización de software para corregir errores y vulnerabilidades. | Mantenimiento Preventivo y Evolutivo: Debe ser realizado por el equipo de desarrollo de software. | Actualizaciones de Software: Cada 3 meses para aplicar parches de seguridad, correcciones y optimizaciones. |
| Mantenimiento Correctivo: Consiste en la corrección de fallos o errores identificados durante el uso del sistema. | Mantenimiento Correctivo: Debe ser realizado por el equipo de soporte técnico. | Respuestas a Incidentes: Inmediato, en caso de fallos críticos.  Para problemas menores, el tiempo de respuesta debe ser dentro de 24 horas. |
| Mantenimiento Evolutivo: Implica la adición de nuevas funcionalidades o mejoras basadas en los comentarios de los usuarios. | Mantenimiento Evolutivo y Preventivo : Debe ser realizado por el equipo de desarrollo de software. | Evaluación de Requerimientos de Nuevas Funcionalidades: Cada 6 meses, basado en la retroalimentación de los usuarios. |

### 3.3.6 Portabilidad:

La portabilidad del proyecto S.I.N.H.2.O es un factor clave para garantizar que pueda ser implementado en diferentes entornos. A continuación, se especifican los atributos que aseguran la facilidad de portabilidad:

Componentes dependientes del servidor:

* Se busca que menos del 20% de los componentes del sistema dependan de un servidor específico. Esto permite que el sistema pueda funcionar en entornos tanto locales como en la nube

Uso de lenguaje por su portabilidad:

* El sistema está desarrollado en Python, dado que este lenguaje es portable y entrega mucho soporte a diversas plataformas.

## 3.4 Otros Requisitos:

Cumplimiento de normativas locales: El sistema debe cumplir con las normativas locales relacionadas con el uso de tecnología en la agricultura.

Compatibilidad con hardware adicional: El sistema debe ser compatible con sensores de humedad del suelo de otras marcas que puedan ser integrados en el futuro.

Accesibilidad: El sistema deberá ser accesible para personas con discapacidades, asegurando que la interfaz gráfica de usuario cumpla con los estándares de accesibilidad.

# 4. Propuesta de Planificación:

## 4.1 Descripción general acerca de la Planificación:

La planificación del proyecto SINH2O se enfoca en establecer un cronograma detallado y definir claramente los recursos humanos y técnicos necesarios para su ejecución exitosa. La estrategia de planificación se basa en un enfoque incremental, donde las fases del proyecto se dividen en entregables más pequeños y manejables, permitiendo una implementación gradual y un control continuo de calidad.

Duración estimada y recursos del proyecto:

Se estima una duración de 3 a 5 meses desde el inicio hasta la implementación final, este plazo considera tiempos de desarrollo, pruebas y capacitación.

Equipo de trabajo:

* Jefe de proyecto / Desarrollador: Gestiona el proyecto y participa en el desarrollo del software. Supervisa el progreso y mantiene la comunicación con el cliente.
* Desarrollador Full Stack: Desarrolla la aplicación de escritorio y la página web, abarcando tanto el Frontend como el Backend.
* Especialista machine Learning / QA: Desarrolla el modelo de predicción, realiza las pruebas funcionales e integraciones y proporciona soporte técnico.

Buenas prácticas y condiciones necesarias:

* Planificación y seguimiento: Reuniones periódicas para evaluar el avance y ajustar estrategias.
* Uso de Herramientas de Gestión: Herramientas como Jira o Trello para gestionar tareas y monitorear el progreso.
* Metodología Incremental: Desarrollo en fases para permitir ajustes y minimizar riesgos.
* Control de Calidad: Pruebas continuas para asegurar el cumplimiento de estándares.
* Capacitación y Documentación: Formación para el usuario final y manuales claros.
* Gestión de Riesgos: Identificación de riesgos y planes de contingencia.

### 4.1.2 Definición del Equipo de Trabajo:

|  |  |  |
| --- | --- | --- |
| **Rol** | **Nombre** | **Responsabilidades** |
| Jefe de proyectos / Desarrollador | Matias Mora | -Gestión general del proyecto  -Participación en el desarrollo del software.  -Supervisión del progreso y comunicación con el cliente. |
| Desarrollador Full Stack | Sebastian Pino | -Desarrollo de la aplicación de escritorio y la página web.  -Implementación del frontend y backend. |
| Especialista en Machine Learning/ QA | Andres Silva | -Desarrollo del modelo de predicción.  -Realización de pruebas funcionales e integración.  -Provisión de soporte técnico. |

### 4.1.3 Definición de Actividades principales del Proyecto:

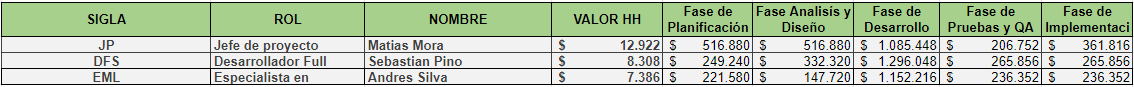
|  |
| --- |
| **Fase de Planificación** |
| Kick Off |
| Acta de Constitución de proyecto |
| Aprobación del Acta |
| Definición de requerimientos Generales del proyecto |
| Organización del equipo |
| **Fase de Análisis y diseño** |
| Captura de requerimientos específicos |
| Análisis de requerimientos |
| Diseño de la solución. Modelamientos |
| Propuesta ERS |
| Plan de proyecto |
| **Fase de Desarrollo** |
| Implementación ambiente de desarrollo |
| Desarrollo requerimiento Registro usuario |
| Desarrollo requerimiento Inicio sesión |
| Desarrollo requerimiento Autentificación inicio sesión |
| Desarrollo requerimiento Recuperación de contraseña |
| Desarrollo requerimiento Humedad tiempo real |
| Desarrollo requerimiento Registro Humedad |
| Desarrollo requerimiento Registró Temperatura |
| Desarrollo requerimiento Capacidad estanque de agua |
| Desarrollo requerimiento Detección de obstrucciones |
| Desarrollo requerimiento Configuración remota |
| Desarrollo requerimiento Apagado y encendido remoto |
| Desarrollo requerimiento Integración y eliminación de sensores |
| Desarrollo requerimiento Apagado de emergencia |
| Desarrollo requerimiento Generador de reportes |
| Desarrollo requerimiento Historial de reportes |
| Desarrollo requerimiento Historial de irrigación |
| Desarrollo requerimiento Historial de recarga de agua |
| Desarrollo requerimiento Ayuda y guia de uso |
| **Fase de Pruebas y QA** |
| Implementación ambiente de pruebas |
| Pruebas Funcionales |
| Pruebas de Integración |
| Pruebas Unitarias por Componentes |
| **Fase de implementación y cierre** |
| Migración del sistema a producción |
| Pruebas de integración final |
| Marcha blanca |
| Capacitación |
| Acta cierre de proyecto |

### 4.1.4 Diagrama EDT:



### 4.1.5 Carta Gantt:

### 4.1.6 Resumen Costos del Desarrollo del Proyecto:



## 4.2 Plan de Control de Cambio:

Tipos de cambio permitidos en el proyecto SINH2O:

1. Funcionales: Cambios que afecten directamente las características del sistema Ej:

* Ajustes en la interfaz de usuario.
* Nuevos módulos.
* Mejoras en las funcionalidades ya existentes del sistema.

1. No Funcionales: Cambios que tengan que ver con rendimiento, seguridad, mantenimiento o cualquier característica que no impacte directamente las funcionalidades visibles.

Alcance de los cambios aplicados:

1. Funcionales: Se permiten los cambios siempre y cuando no afecte al propósito que este tiene en el sistema. Los cambios deben ser aprobados.
2. No funcionales: Los cambios en rendimiento o seguridad son permitidos siempre y cuando no afecten la integridad del sistema.

Motivos por los que pueden ocurrir cambios en el sistema:

* Solicitudes del cliente o nuevas necesidades detectadas
* Mejoras identificadas por el equipo de desarrollo durante la fase de pruebas
* Correcciones de errores que afecten el funcionamiento

Motivos para no aplicar cambios en el sistema:

* Cambios que impliquen un rediseño total.
* Cambios solicitados fuera de las fases de desarrollo establecidos en el cronograma.

## 5. Anexos

### 5.1 Acta de Proyecto:

En el siguiente apartado se adjunta el [v2 Acta de constitución.docx](https://docs.google.com/document/d/1DMIg7FlZ16yKlaV3rm0qmbl7POI_5uHw/edit)

### 5.2 Matriz Especificación de Requerimientos:

En el siguiente apartado se adjunta la matriz de [Requerimientos SINH2O.xlsx](https://docs.google.com/spreadsheets/d/1sUROpyjgL8tgr3iA76RkRx8hzdLiv7um/edit?gid=75800468#gid=75800468)

### 5.3 Diagrama de Casos de Uso General:

### 

### 5.4 Planilla Casos de Uso:

En el siguiente apartado se adjuntan los [Casos de uso extendido](https://drive.google.com/drive/u/1/folders/1sxZ-QcE7vbwxhGrVqo0TaxlqWbj6QDvR)

### 5.5 Prototipado de Software:





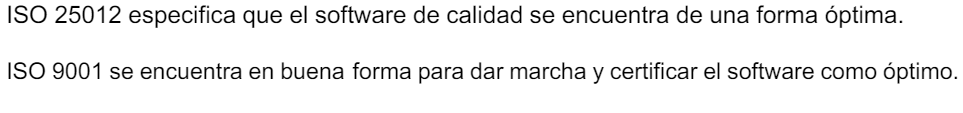








### 5.6 Resultado Análisis de Calidad Diagramas Modelamiento:



### 5.7 Resultado Análisis de Calidad Prototipo No funcional del Sistema:

En el siguiente apartado se adjunta el análisis del [Plan Gestión de Calidad SINH2O.docx](https://docs.google.com/document/d/1h5c3d-OH9HyvKM6g1MOnGnCCglU57D9U/edit)

### 5.8 Planilla entregables del Proyecto:

### 5.9 Matriz de Control de Cambios:

En el siguiente apartado se adjunta la [Matriz de control de cambios.xlsx](https://docs.google.com/spreadsheets/d/1kyEULlTIEtambuklpCl_LyetMsSU6M3e/edit?gid=1717034525#gid=1717034525)

### 5.10 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo:

En el siguiente apartado se adjunta la [MatrizEDT\_SINH2O.xlsx](https://docs.google.com/spreadsheets/d/13055r0gbAo1qK05JcTSsV7mpof3M1HxY/edit?gid=1924178595#gid=1924178595)