**Especificación de requisitos de software (Versión 3.0)**

**Sistema de control de riesgos y plagas en una platanera  
  
  
  
Proyecto SENA**

**Logotipo

El contenido generado por IA puede ser incorrecto.**

Índice

[**Especificación de Requisitos de Software** 3](#_Toc192421529)

[**Sistema de control de riegos y plagas en una platanera** 3](#_Toc192421530)

[**1. Introducción** 3](#_Toc192421531)

[**1.1. Propósito** 3](#_Toc192421532)

[**1.2. Alcance** 3](#_Toc192421533)

[**1.3. Definiciones, Acrónimos y Abreviaturas** 3](#_Toc192421534)

[**1.4. Referencias** 3](#_Toc192421535)

[**2. Descripción General** 4](#_Toc192421536)

[**2.1. Perspectiva del Producto** 4](#_Toc192421537)

[**2.2. Funcionalidades del Producto** 4](#_Toc192421538)

[**2.3. Características del Usuario** 4](#_Toc192421539)

[**3. Requisitos Específicos** 5](#_Toc192421540)

[**3.1. Requisitos Funcionales** 5](#_Toc192421541)

[**3.2. Requisitos No Funcionales** 7](#_Toc192421542)

[**3.3. Requisitos del Sistema** 7](#_Toc192421543)

[**4. Casos de uso** 8](#_Toc192421544)

[**5. Restricciones** 11](#_Toc192421545)

[**6. Aceptación de Requisitos** 11](#_Toc192421546)

[**6.1. Criterios de Aceptación** 11](#_Toc192421547)

# **Especificación de Requisitos de Software**

# **Sistema de control de riegos y plagas en una platanera**

## **1. Introducción**

## **1.1. Propósito**

El propósito de este documento es proporcionar una especificación completa de los requisitos para el desarrollo del software de control de riesgos y plagas en una plantación de plataneras. Este sistema está diseñado para ayudar a los agricultores a identificar, monitorear y gestionar plagas y riesgos, optimizando el uso de recursos y mejorando la productividad de las plantaciones.

## **1.2. Alcance**

Este sistema de software permitirá a los usuarios registrar información sobre las plataneras, detectar plagas y enfermedades, gestionar tratamientos, y generar informes. El sistema incluirá una interfaz de usuario amigable, capacidades de monitoreo y notificación, registro fotográfico y una base de datos para almacenar información relevante.

## **1.3. Definiciones, Acrónimos y Abreviaturas**

* **ERS:** Especificación de Requisitos de Software
* **Platanera:** Plantación de plátanos
* **UI:** Interfaz de Usuario
* **BD:** Base de Datos
* **PM:** Planificación de Mantenimiento
* **API:** Interfaz de Programación de Aplicaciones
* **AC:** Aseguramiento de Calidad

## **1.4. Referencias**

* IEEE 830-1998, **"IEEE Recommended Practice for Software Requirements Specifications"**
* "Agricultural Pest Management" por L. P. Pedigo
* [FAO – Organización de las Naciones Unidas para la Alimentación y la Agricultura](https://www.fao.org)

## **2. Descripción General**

## **2.1. Perspectiva del Producto**

El software de control de riesgos y plagas para una platanera se desarrollará como una aplicación web y móvil. La aplicación permitirá a los usuarios registrar información sobre las plataneras, identificar plagas y enfermedades, gestionar tratamientos y generar informes. El sistema estará basado en una base de datos relacional para almacenar y recuperar información de manera eficiente.

## **2.2. Funcionalidades del Producto**

* **Registro de Lotes:** Los usuarios pueden añadir y modificar.
* **Monitoreo de Plagas y Enfermedades:** Los usuarios pueden registrar avistamientos, síntomas y reportar incidencias.
* **Base de Datos de Plagas y Enfermedades:** Proporciona información sobre diferentes plagas, enfermedades y métodos de control.
* **Planificación de Tratamientos:** Facilita la programación y gestión de tratamientos para plagas y enfermedades.
* **Generación de Informes:** Crea informes detallados sobre el estado de las plataneras, plagas, tratamientos y estadísticas.
* **Interfaz de Usuario:** Proporciona una interfaz intuitiva para los usuarios del sistema.
* **Acceso Multinivel:** Gestión de roles y permisos para diferentes tipos de usuarios (administrador, técnico, agricultor).

## **2.3. Características del Usuario**

* **Agricultor:** Persona que trabaja en la plantación de la siembra y necesita herramientas para gestionar plagas y riesgos.
* **Técnico:** Profesional que proporciona asesoramiento sobre el control de plagas y enfermedades.
* **Administrador:** Persona que gestiona el sistema, incluyendo la configuración de usuarios y el mantenimiento del sistema.

## **3. Requisitos Específicos**

## **3.1. Requisitos Funcionales**

| **ID** | **Requisito** | **Características** | **Descripción** |
| --- | --- | --- | --- |
|  |  |  |  |
| RF-01 | **Registro de lotes** | El técnico debe tener acceso a la información del cultivo para realizar el registro de lotes. | El sistema permitirá a los técnicos registrar cada lote con la información proporcionada por el agricultor, incluyendo número de plantas, ubicación y fecha de siembra.  El registro se realizará utilizando datos específicos del cultivo y lotes, asegurando una organización eficiente de la información. |
| RF-02 | **Monitoreo de Plagas y Enfermedades** | Los agricultores deben autenticarse en el sistema para registrar y reportar el estado de sus cultivos. | El sistema permitirá a los agricultores registrar avistamientos de plagas y enfermedades, proporcionando detalles sobre síntomas y ubicación geográfica.    El agricultor iniciará sesión en la aplicación móvil con su número de cédula, seleccionará el lote correspondiente y enviará reportes sobre el estado de la planta. |
| RF-03 | **API de Plagas y Enfermedades** | El sistema debe capturar y enviar información de manera eficiente a la API para su procesamiento. | La API Crop.Health proporcionará información detallada sobre plagas, enfermedades, síntomas y métodos de control.    Las imágenes registradas por el agricultor se compararán con la base de datos de la API, identificando enfermedades y recomendando tratamientos adecuados.  La imagen proporcionada por el agricultor debe estar en formato JPG y cumplir con un tamaño máximo de 1024x1024 con un peso de 500kb - 1MB, para garantizar una correcta comparación en la API. |
| RF-04 | **Planificación de Tratamientos** | El técnico tendrá acceso al sistema para planificar tratamientos basados en la información procesada por la inteligencia artificial. | El sistema recomendará tratamientos para plagas y enfermedades, registrando los métodos y las fechas de aplicación.  La API analizará la enfermedad detectada y sugerirá tratamientos. El técnico planificará el tratamiento, y el agricultor lo implementará en el cultivo. |
| RF-05 | **Generación de Informes** | El sistema debe generar y enviar informes de manera rápida y eficiente. | Se generarán informes detallados sobre el estado de las plataneras, incidencias de plagas, tratamientos realizados y estadísticas generales.    Al enviar un reporte, el agricultor recibirá una notificación de confirmación, y el técnico será notificado sobre la nueva incidencia reportada. |
| RF-06 | **Reporte sin Anomalías** | El agricultor podrá indicar que no se han detectado anomalías durante el monitoreo. | Si al realizar la inspección el agricultor no encuentra problemas en los cultivos, tendrá la opción de marcar el reporte como “sin anomalías”.    El sistema notificará automáticamente al técnico sobre la ausencia de problemas en el lote reportado. |
| RF-07 | **Generación de Historial** | El sistema debe almacenar y organizar los datos de monitoreo para generar un historial detallado. | Se permitirá la consulta y exportación del historial de eventos y actividades realizadas en los lotes.  Los datos almacenados incluirán enfermedades detectadas, tratamientos aplicados y resultados obtenidos, facilitando el análisis histórico de los cultivos.  El historial podrá ser exportado en formato PDF para su uso en informes externos. |
|  |  |  |  |

## **3.2. Requisitos No Funcionales**

| **ID** | **Requisito** | **Descripción** |
| --- | --- | --- |
| RNF-01 | **Usabilidad** | El sistema debe ser fácil de usar para personas con conocimientos técnicos limitados. |
| RNF-02 | **Seguridad** | El sistema debe proteger la información mediante autenticación, autorización, y encriptación de datos sensibles. |
| RNF-03 | **Escalabilidad** | El sistema debe ser capaz de manejar un crecimiento en el número de plataneras y usuarios. |
| RNF-04 | **Compatibilidad** | El sistema debe ser compatible con diferentes dispositivos y navegadores web. |
| RNF-05 | **Desempeño** | El sistema debe responder de manera eficiente a las solicitudes de los usuarios y generar informes de manera oportuna. |
| RNF-06 | **Disponibilidad** | El sistema debe tener una alta disponibilidad con mínimo tiempo de inactividad para realizar mantenimientos. |

## **3.3. Requisitos del Sistema**

| **ID** | **Requisito** | **Descripción** |
| --- | --- | --- |
| RS-01 | **Plataforma de Desarrollo** | El sistema se desarrollará en un entorno basado en desarrollo web utilizando tecnologías como (Visual studio code) como editor de código, utilizaremos leguajes de programación como (HTML, CSS y JavaScript) para el desarrollo del front-end y un framework como (React) para un mejor desarrollo de las interfaces. |
| RS-02 | **Lenguaje de Programación** | El backend se desarrollará en un lenguaje robusto como Node.js, Python y JavaScript. |
| RS-03 | **Base de Datos Relacional** | El sistema utilizará una base de datos relacional como MySQL Workbench. |

**4. Casos de uso**  
  
 **Caso de Uso 1: Registro de Lotes**

**ID:** CU-01  
**Actor:** Técnico  
**Descripción:** Permite al técnico registrar lotes con información proporcionada por el agricultor.  
**Precondición:** El técnico debe estar autenticado en el sistema.  
**Flujo Principal:**

1. El técnico accede al sistema e inicia sesión.
2. Selecciona la opción "Registrar Lote".
3. Ingresa los datos del lote (número de plantas, ubicación, fecha de siembra).
4. Guarda el registro.
5. El sistema confirma el registro exitoso.  
   **Postcondición:** El lote queda registrado en la base de datos.

**Caso de Uso 2: Monitoreo de Plagas y Enfermedades**

**ID:** CU-02  
**Actor:** Agricultor  
**Descripción:** Permite a los agricultores registrar avistamientos de plagas y enfermedades.  
**Precondición:** El agricultor debe estar autenticado en el sistema.  
**Flujo Principal:**

1. El agricultor accede al sistema e inicia sesión.
2. Selecciona el lote a monitorear.
3. Registra la información de plagas/enfermedades detectadas (síntomas, ubicación, fotos).
4. Envía el reporte.
5. El sistema almacena el reporte y notifica al técnico.  
   **Postcondición:** El reporte queda registrado y el técnico es notificado.

**Caso de Uso 3: API de Plagas y Enfermedades (Crop.Health)**

**ID:** CU-03  
**Actor:** Sistema  
**Descripción:** Permite analizar imágenes de plagas y enfermedades.  
**Precondición:** El agricultor debe haber subido una imagen en formato JPG con el tamaño adecuado.  
**Flujo Principal:**

1. El agricultor carga la imagen de la planta afectada.
2. La API Crop.Health analiza la imagen.
3. Se identifican posibles enfermedades y se muestran recomendaciones.
4. El sistema almacena los resultados.  
   **Postcondición:** Se genera un diagnóstico con información relevante para el agricultor y técnico.

**Caso de Uso 4: Planificación de Tratamientos**

**ID:** CU-04  
**Actor:** Técnico  
**Descripción:** Permite al técnico planificar tratamientos según la información de la API.  
**Precondición:** El técnico debe estar autenticado en el sistema.  
**Flujo Principal:**

1. El técnico accede al sistema e inicia sesión.
2. Revisa los reportes de plagas/enfermedades.
3. Consulta la recomendación de la API Crop.Health.
4. Registra un tratamiento con detalles (método, fecha de aplicación).
5. Guarda la planificación.
6. Se notifica al agricultor.  
   **Postcondición:** El tratamiento queda registrado y el agricultor recibe la información.

**Caso de Uso 5: Generación de Informes**

**ID:** CU-05  
**Actor:** Sistema  
**Descripción:** Genera informes sobre el estado de las plataneras, plagas y tratamientos.  
**Precondición:** Debe haber datos almacenados en el sistema.  
**Flujo Principal:**

1. El técnico solicita un informe.
2. El sistema recopila datos de plagas, tratamientos y estadísticas.
3. Se genera el informe en formato PDF.
4. Se notifica al usuario que el informe está disponible.  
   **Postcondición:** Se genera un informe descargable.

**Caso de Uso 6: Reporte sin Anomalías**

**ID:** CU-06  
**Actor:** Agricultor  
**Descripción:** Permite al agricultor reportar que no hay anomalías en su lote.  
**Precondición:** El agricultor debe estar autenticado en el sistema.  
**Flujo Principal:**

1. El agricultor accede al sistema e inicia sesión.
2. Selecciona el lote correspondiente.
3. Marca la opción "Sin Anomalías".
4. Envía el reporte.
5. El sistema notifica al técnico.  
   **Postcondición:** Se almacena la información y se notifica al técnico.

**Caso de Uso 7: Generación de Historial**

**ID:** CU-07  
**Actor:** Sistema  
**Descripción:** Permite la consulta y exportación del historial de plagas y tratamientos.  
**Precondición:** Debe haber registros en la base de datos.  
**Flujo Principal:**

1. El usuario (técnico o agricultor) accede al sistema.
2. Selecciona la opción "Consultar Historial".
3. Filtra la información según fechas o lotes.
4. Descarga el historial en formato PDF.  
   **Postcondición:** Se obtiene un historial detallado del cultivo.

## **5. Restricciones**

* **Restricciones Legales y Reglamentarias:** El software debe cumplir con las regulaciones locales e internacionales sobre protección de datos y seguridad de la información.
* **Restricciones Técnicas:** El sistema debe ser compatible con las versiones actuales de navegadores web y dispositivos móviles.

## **6. Aceptación de Requisitos**

## **6.1. Criterios de Aceptación**

* **Cumplimiento de Requisitos Funcionales:** Los requisitos funcionales serán evaluados mediante pruebas de aceptación.
* **Usabilidad y Efectividad:** La interfaz debe ser evaluada por usuarios finales para asegurar su facilidad de uso.
* **Seguridad y Desempeño:** El sistema será sometido a pruebas de seguridad y rendimiento para validar que cumple con los requisitos no funcionales.Principio del formularioFinal del formulario

## **7. Clase de datos**

## **Diagrama de flujo**

