**SISTEMATIZACIÓN PARA** **OPTIMIZAR LOS PROCESOS Y TOMA DECISIONES INFORMADAS BASADAS EN DATOS HISTÓRICOS Y ACTUALES.**

Mayerlin Rojas Pérez

[rojasmayer@](mailto:rojasmayer@)gmail.com

**PALABRAS CLAVE:** Software a la medida, Estimación de costos.

1. **CONTEXTUALIZACIÓN**

La Alianza de Bioversity International y el Centro Internacional de Agricultura Tropical (CIAT) trabaja para transformar los sistemas alimentarios de manera sostenible, abordando problemas globales como la malnutrición, el cambio climático, la pérdida de biodiversidad y la degradación ambiental. En este contexto, una organización de apoyo a los campesinos locales busca desarrollar un sistema de monitoreo de actividades agronómicas en fincas. Este sistema permitirá a los campesinos registrar y gestionar detalladamente sus actividades diarias, como siembra, riego, fertilización y cosecha, con el objetivo de optimizar sus procesos agrícolas y tomar decisiones informadas basadas en datos históricos y actuales.

El sistema propuesto utilizará bases de datos existentes que contienen registros de actividades agronómicas y características de parcelas, proporcionando una herramienta integral para el monitoreo y gestión de las fincas. La solución debe ser fácilmente desplegable mediante una imagen de Docker, asegurando su accesibilidad y facilidad de uso. Con esta herramienta, los campesinos podrán mejorar la eficiencia y sostenibilidad de sus prácticas agrícolas, contribuyendo a la mejora de su calidad de vida y apoyando los objetivos de la Alianza de promover sistemas alimentarios más sostenibles.

1. **OBJETIVOS**

Desarrollar un sistema de monitoreo de actividades agronómicas en fincas para que los campesinos puedan registrar y gestionar sus actividades diarias, optimizando sus procesos y tomando decisiones informadas basadas en datos históricos y actuales.

1. **ALCANCE**

El alcance de la solución propuesta incluye el desarrollo e implementación de un sistema de monitoreo y gestión de actividades agronómicas para fincas, con el objetivo de optimizar los procesos agrícolas y mejorar la toma de decisiones de los campesinos. El sistema abarcará las siguientes funcionalidades y características:

* **Registro de Actividades Agronómicas:**

Permitir a los usuarios registrar diversas actividades agrícolas como siembra, riego, fertilización, cosecha y otras labores diarias.

Incluir detalles específicos de cada actividad, tales como la fecha, tipo de actividad, insumos utilizados y duración.

* **Gestión de Parcelas de la Finca:**

Permitir a los usuarios gestionar información detallada de las parcelas, incluyendo ubicación (latitud y longitud), tamaño y tipo de cultivo actual.

Facilitar la actualización de la información de las parcelas a medida que cambian las condiciones o los cultivos.

* **Desplegabilidad y Usabilidad:**

Asegurar que el sistema sea fácilmente desplegable utilizando una imagen de Docker, lo que garantiza una instalación y configuración sencillas.

Diseñar la solución con un enfoque en la usabilidad, asegurando que los campesinos puedan usarla sin necesidad de conocimientos técnicos avanzados.

1. **LISTA DE REQUERIMIENTOS**

* **Requerimientos Funcionales**
* **Gestión de Actividades Agronómicas**
* RF1.1: El sistema debe permitir a los usuarios registrar actividades agronómicas con detalles básicos como fecha, tipo de actividad y duración.
* RF1.2: El sistema debe permitir a los usuarios visualizar un historial de actividades agronómicas registradas.
* **Gestión de Parcelas**
* RF2.1: El sistema debe permitir a los usuarios registrar parcelas con detalles básicos como ubicación (latitud y longitud) y tamaño.
* RF2.2: El sistema debe permitir a los usuarios visualizar información detallada de las parcelas registradas.
* **Requerimientos No Funcionales**
* **Usabilidad**
* RNF1.1: La interfaz de usuario debe ser intuitiva y fácil de usar.
* Rendimiento
* RNF2.1: Las operaciones CRUD deben completarse en un tiempo razonable, con tiempos de respuesta inferiores a 2 segundos en condiciones normales.

1. **MARCO DE TRABAJO Y HERRAMIENTAS DE DESARROLLO EMPLEADAS**

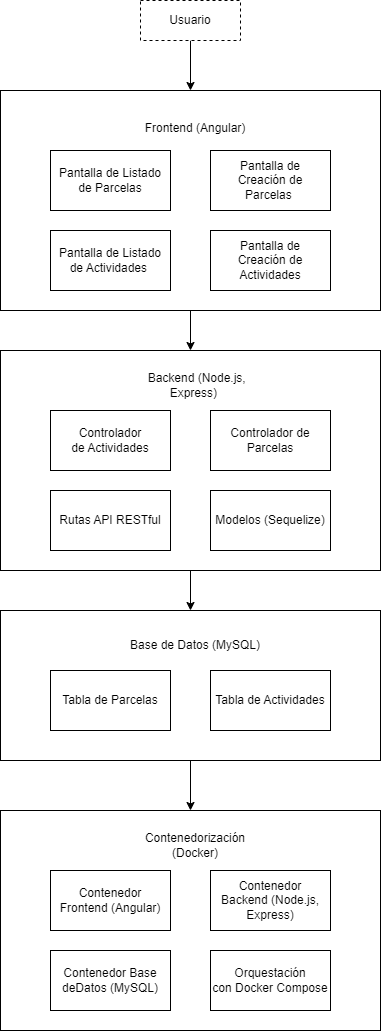
Para la realización del proyecto de monitoreo y gestión de actividades agronómicas en fincas, se utilizarán las siguientes tecnologías y herramientas de desarrollo:

* **Frontend**
* **Framework:** Angular
* **Descripción:** Angular es una plataforma de desarrollo de aplicaciones web basada en TypeScript que facilita la creación de aplicaciones web dinámicas y modernas. Proporciona un entorno robusto para el desarrollo de interfaces de usuario, con características como la vinculación bidireccional de datos, inyección de dependencias y componentes reutilizables.
* **Ventajas:**
* Estructura modular que facilita el mantenimiento y escalabilidad del código.
* Amplia comunidad y soporte, con una rica variedad de bibliotecas y herramientas disponibles.
* Excelente rendimiento en aplicaciones de una sola página (SPA).
* **Backend**
* **Framework:** Node.js con Express
* **Descripción:** Node.js es un entorno de ejecución para JavaScript construido sobre el motor V8 de Chrome. Express es un framework minimalista y flexible para Node.js que proporciona un conjunto sólido de características para aplicaciones web y móviles.
* **Ventajas:**
* Altamente eficiente y adecuado para aplicaciones en tiempo real.
* Gran ecosistema de módulos y paquetes disponibles a través de npm.
* Facilita la creación de APIs RESTful para la comunicación entre el frontend y el backend.
* **ORM:** Sequelize
* **Descripción:** Sequelize es un ORM (Object-Relational Mapping) basado en Node.js que facilita la interacción con bases de datos SQL. Permite definir modelos de datos y realizar operaciones CRUD (crear, leer, actualizar, eliminar) de manera sencilla.
* **Ventajas:**
* Abstracción de complejidades de SQL, facilitando la interacción con la base de datos.
* Soporte para múltiples sistemas de gestión de bases de datos, incluyendo MySQL.
* Facilita la definición de relaciones entre tablas y la migración de esquemas.
* **Base de Datos**
* **Sistema de Gestión de Bases de Datos:** MySQL
* **Descripción:** MySQL es un sistema de gestión de bases de datos relacional de código abierto ampliamente utilizado para aplicaciones web y empresariales. Ofrece un rendimiento sólido, seguridad y escalabilidad.
* **Ventajas:**
* Amplia adopción y soporte en la comunidad de desarrolladores.
* Alto rendimiento y capacidad de manejar grandes volúmenes de datos.
* Funcionalidades avanzadas de administración y seguridad de datos.
* **Contenedorización**
* **Herramienta:** Docker
* **Descripción:** Docker es una plataforma de contenedorización que permite a los desarrolladores empaquetar aplicaciones junto con todas sus dependencias en contenedores portátiles y consistentes.
* **Ventajas:**
* Facilita el despliegue y la gestión de aplicaciones en diversos entornos.
* Asegura la consistencia entre entornos de desarrollo, pruebas y producción.
* Simplifica la escalabilidad y la administración de las aplicaciones.
* Herramientas Adicionales
* **Control de Versiones: Git**
* **Descripción:** Git es un sistema de control de versiones distribuido que permite a los equipos de desarrollo gestionar y seguir los cambios en el código fuente a lo largo del tiempo.
* **Ventajas:**
* Facilita la colaboración entre múltiples desarrolladores.
* Permite la creación de ramas y la fusión de cambios de manera eficiente.
* Amplia integración con servicios de alojamiento de repositorios como GitHub, GitLab y Bitbucket.
* **IDE/Editor de Código: Visual Studio Code**
* **Descripción:** Visual Studio Code es un editor de código fuente desarrollado por Microsoft que es altamente extensible y configurable.
* **Ventajas:**
* Soporte para una amplia variedad de lenguajes de programación y herramientas de desarrollo.
* Extensiones y complementos que mejoran la productividad y la experiencia de desarrollo.
* Integración con sistemas de control de versiones y herramientas de desarrollo en la nube.
* **Flujo de Desarrollo**
* **Configuración del Entorno de Desarrollo:**
* Instalación de Angular CLI y creación del proyecto Angular.
* Configuración del proyecto Node.js con Express para el backend.
* Configuración de Sequelize para interactuar con la base de datos MySQL.
* Configuración de la base de datos MySQL y creación del esquema de datos necesario.
* Creación de contenedores Docker para el frontend, backend y la base de datos.
* **Desarrollo del Frontend:**
* Creación de componentes, servicios y rutas en Angular para la interfaz de usuario.
* Implementación de formularios y validaciones para el registro de actividades y la gestión de parcelas.
* Integración con el backend mediante llamadas a la API RESTful.
* **Desarrollo del Backend:**
* Creación de endpoints API RESTful en Node.js con Express para las operaciones CRUD (crear, leer, actualizar, eliminar) de actividades y parcelas.
* Implementación de la lógica de negocio y validaciones en el servidor.
* Configuración de Sequelize para interactuar con la base de datos MySQL y realizar consultas.
* **Despliegue y Pruebas:**
* Creación de imágenes Docker para el frontend, backend y la base de datos.
* Configuración de Docker Compose para la orquestación de los contenedores.
* Pruebas funcionales y de integración para asegurar el correcto funcionamiento del sistema.

1. **Arquitectura de Software de la Solución Propuesta**

**Descripción General**

La arquitectura del sistema de monitoreo de actividades agronómicas se ha diseñado para garantizar modularidad, escalabilidad y facilidad de mantenimiento. Las tecnologías utilizadas son Angular para el frontend, Node.js con Express y Sequelize para el backend, MySQL como base de datos y Docker para contenedorización.



**Componentes de la Arquitectura**

**Frontend (Angular)**

* Pantalla de Listado de Parcelas: Muestra una lista de todas las parcelas registradas.
* Pantalla de Creación de Parcelas: Permite la creación de nuevas parcelas.
* Pantalla de Listado de Actividades: Muestra una lista de todas las actividades agronómicas registradas.
* Pantalla de Creación de Actividades: Permite la creación de nuevas actividades agronómicas.

**Backend (Node.js, Express)**

* Controlador de Parcelas: Maneja las operaciones CRUD relacionadas con las parcelas.
* Controlador de Actividades: Maneja las operaciones CRUD relacionadas con las actividades agronómicas.
* Rutas API RESTful: Define los endpoints para interactuar con el frontend.
* POST /parcels: Crear una nueva parcela.
* GET /parcels: Obtener todas las parcelas.
* POST /activities: Crear una nueva actividad.
* GET /activities: Obtener todas las actividades.
* Modelos (Sequelize): Definición de los modelos de datos y su mapeo a las tablas de la base de datos.

**Base de Datos (MySQL)**

* Tabla de Parcelas (FarmPlot): Almacena información sobre las parcelas.
* Campos: id, name, latitude, longitude, size\_in\_sqm, crop\_type, created\_at, updated\_at.
* Tabla de Actividades Agronómicas (AgronomicActivity): Almacena información sobre las actividades agronómicas.
* Campos: id, date, activity\_type, duration, created\_at, updated\_at.

**Contenedorización (Docker)**

* Contenedor Frontend (Angular): Contiene la aplicación Angular.
* Contenedor Backend (Node.js, Express): Contiene la aplicación Node.js con Express.
* Contenedor Base de Datos (MySQL): Contiene la base de datos MySQL.
* Orquestación con Docker Compose: Facilita la gestión y el despliegue de múltiples contenedores.

1. **Modelo de la base de datos**

La base de datos consta de varias tablas con sus respectivas columnas.

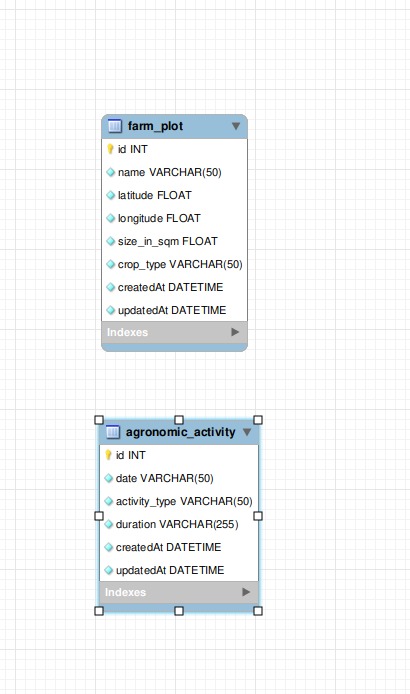
* **Tabla FarmPlot:**

Esta tabla almacena la información básica de los lotes de la finca, incluyendo su nombre, ubicación, tamaño, tipo de cultivo y las marcas de tiempo de creación y actualización del registro.

* + id: Identificador único del lote de la finca (Primary Key).
  + name: Nombre del lote de la finca.
  + latitude: Latitud de la ubicación del lote.
  + longitude: Longitud de la ubicación del lote.
  + size\_in\_sqm: Tamaño del lote en metros cuadrados.
  + crop\_type: Tipo de cultivo en el lote.
  + created\_at: Fecha y hora de creación del registro.
  + updated\_at: Fecha y hora de la última actualización del registro.
* **Tabla AgronomicActivity:**

Esta tabla almacena los registros de las actividades agronómicas realizadas, con detalles como la fecha de la actividad, tipo de actividad, duración, y las marcas de tiempo de creación y actualización del registro.

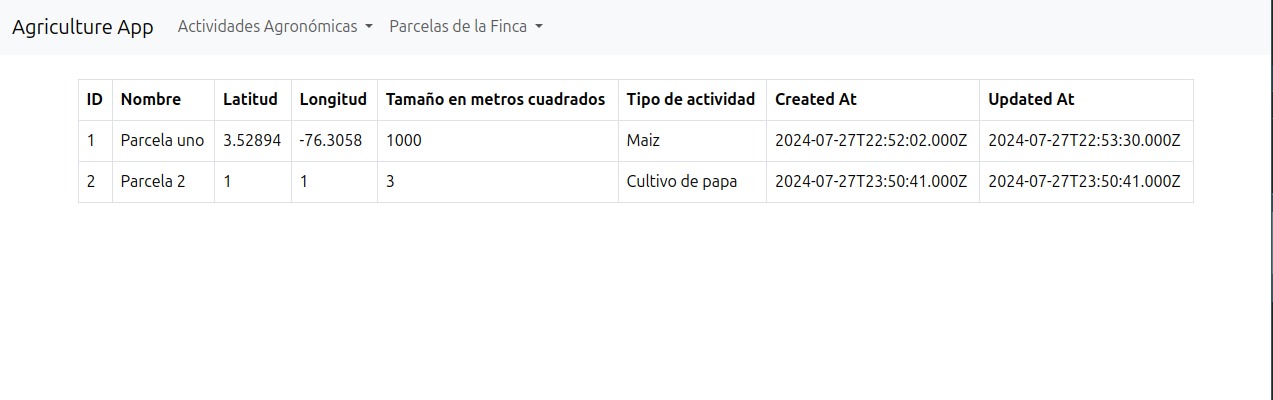
* + id: Identificador único de la actividad agronómica (Primary Key).
  + date: Fecha de la actividad agronómica.
  + activity\_type: Tipo de actividad agronómica.
  + duration: Duración de la actividad agronómica.
  + created\_at: Fecha y hora de creación del registro.
  + updated\_at: Fecha y hora de la última actualización del registro.



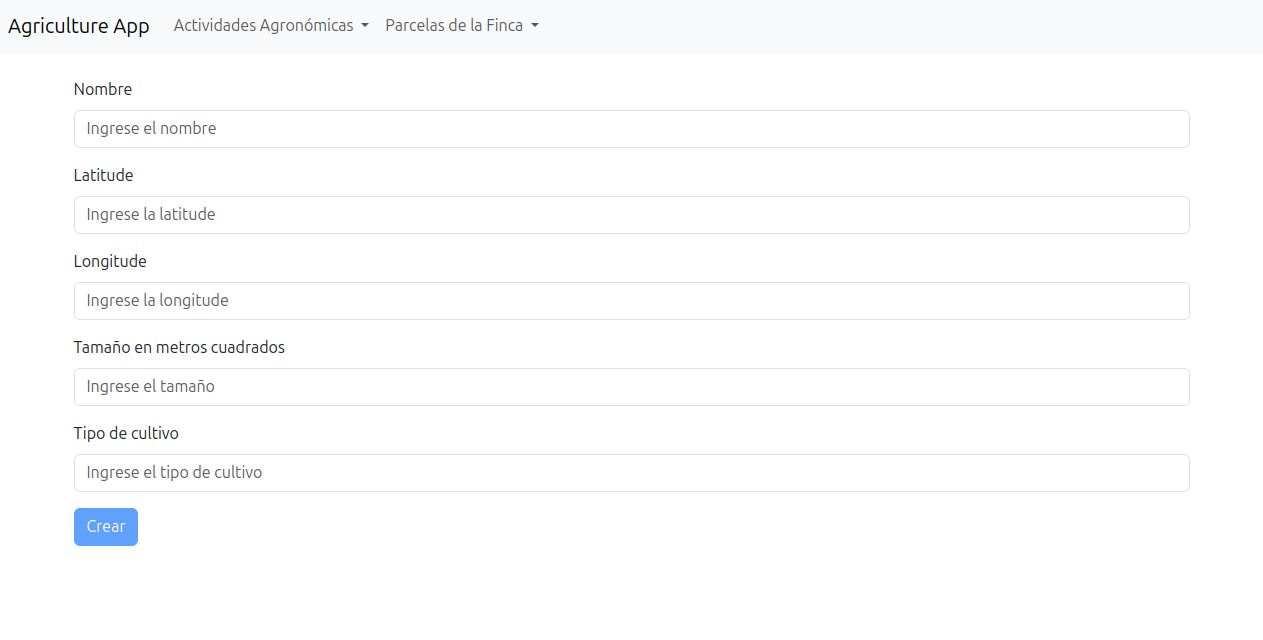
1. **DISEÑO DE INTERFAZ GRAFICA DE USUARIO**

En el manual de usuario están descritas las funciones de cada uno de los botones y segmentos.

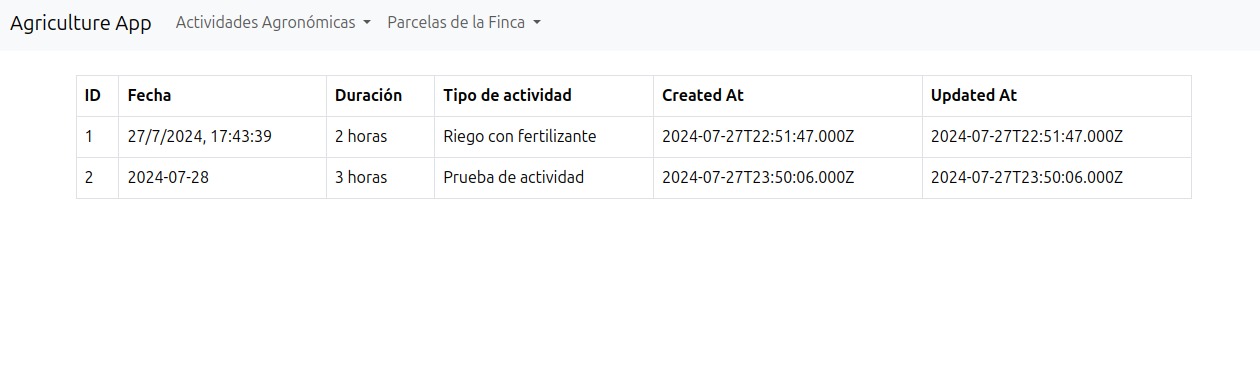
* **Farm Plot**



* **Farm Plot Agregar**



* **Agronomic Activity**

****

* **Agronomic Activity Agregar**



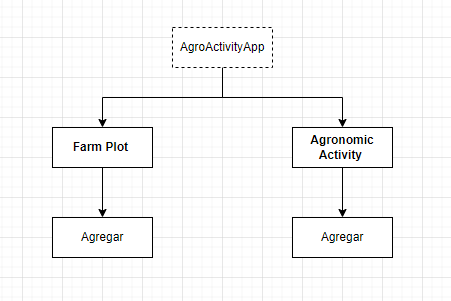
1. **ORGANIGRAMA MODULAR**

El sistema desarrollado comprende los módulos requeridos para cumplir con el objetivo del proyecto. A continuación, se describe el organigrama modular del sistema:

Explicación de los módulos:

* **Farm Plot Module:** Aquí se administran los detalles de los lotes, incluyendo su ubicación, extensión y cualquier otra información relevante.
* **Agronomic Activity Module:** Este módulo permite a los usuarios registrar y gestionar las labores agrícolas que se realizan en las fincas, incluyendo información sobre el tipo de labor, fecha y recursos utilizados.

La interacción entre estos módulos permite a los usuarios optimizar sus procesos y tomar decisiones informadas con las actividades agrícolas de manera eficiente.

****

Img. 2 organigrama modular.

1. **CONCLUSIÓN**

* El desarrollo del sistema de monitoreo de actividades agronómicas proporciona a los campesinos una herramienta eficiente para gestionar y optimizar sus actividades diarias, permitiéndoles tomar decisiones informadas basadas en datos históricos y actuales.
* La implementación del proyecto utilizando tecnologías modernas y prácticas de desarrollo ágiles ha resultado en una solución robusta y escalable, capaz de mejorar significativamente la eficiencia y sostenibilidad de la agricultura local.

**Referencias**

|  |  |
| --- | --- |
| [1] | P. Perez, Mi libro, Cali: Norma, 2018. |
| [2] | S. Coque, L. Lema y M. Arcos, Investigaciones sobre ingeniería de Software, Quito, Ecuador: Editorial Abya-Yala, 2017. |