

SICLAUy – Sistema de Información de Control Logístico Agrícola del Uruguay

Amílcar Santana¹[0009-0004-9733-7618], Rebeca E. Fiss²[0000-0003-0008-880X], Juan Ignacio dos Santos³[0009-0004-6628-2749]

¹ Instituto Federal Sul-rio-grandense Campus Santana do Livramento, RS, Brasil, Universidad Tecnológica ITR-N, Rivera, Uruguay
takintosh@gmail.com

² Instituto Federal Sul-rio-grandense Campus Santana do Livramento, RS, Brasil
rebecafiss@ifsul.edu.br

³ Intendencia Departamental de Rivera, Rivera, Uruguay
juanidosantos@gmail.com

Resumen. La producción de sandías en Rivera, Uruguay, representa el 95% de la producción nacional de sandías, y es un pilar clave en la economía del departamento. Sin embargo, su ubicación fronteriza expone a los productores a competencia desleal, debido al ingreso irregular de productos sin trazabilidad, afectando su competitividad. Para abordar este problema, la Intendencia Departamental de Rivera (IDR), en conjunto con el Ministerio de Ganadería, Agricultura y Pesca (MGAP) y la Dirección Nacional de Aduanas, implementó un protocolo de control basado en remitos físicos. No obstante, la dependencia de documentación en papel dificulta la fiscalización y genera ineficiencias operativas.

Este estudio analiza el desarrollo e implementación del Sistema de Información de Control Logístico Agrícola del Uruguay, una plataforma digital diseñada para mejorar la trazabilidad y optimizar el control del transporte agrícola. La metodología utilizada combina entrevistas con actores clave de la IDR, análisis documental de normativas vigentes y la aplicación del enfoque Rapid Application Development (RAD) para un desarrollo ágil.

El sistema fue desarrollado con PHP 8.3 y Laravel 11, utilizando FilamentPHP para la generación de interfaces y MySQL 8.4.2 para la gestión de datos, garantizando integridad mediante propiedades ACID. Se incorporaron funcionalidades clave, como gestión de zafras y cultivos, registro de inspecciones, emisión automatizada de remitos electrónicos, control en ruta en tiempo real y un modelo contable automatizado. Además, el sistema cuenta con un modelo de control de acceso basado en roles (Inspector, Productor, Policía y Administrador) y un módulo de auditoría para registrar cambios en la base de datos.

Validado por la IDR, su implementación está prevista para la zafra 2025/2026, con futuras optimizaciones basadas en su uso en campo. Su arquitectura escalable permite la incorporación de nuevos cultivos sin cambios estructurales. A futuro, se proyecta la integración de un cuaderno de campo digital y herramientas de análisis predictivo.

Los resultados de este trabajo evidencian que la digitalización del proceso de fiscalización agrícola puede mejorar la eficiencia, reducir el fraude comercial y

optimizar la gestión logística en regiones fronterizas. Además, el sistema plantea un modelo replicable en otros sectores productivos que requieran mayor trazabilidad y control en la distribución agrícola.

Palabras Clave: Agroinformática, Trazabilidad, Control logístico agrícola, Digitalización, Gestión de Remitos.

SICLAUy – Agricultural Logistics Control Information System of Uruguay

Abstract. The production of watermelons in Rivera, Uruguay, accounts for 95% of the national output and is a key pillar of the department's economy. However, its border location exposes producers to unfair competition due to the irregular entry of products without traceability, affecting their competitiveness. To address this problem, the Rivera Departmental Administration (IDR), together with the Ministry of Livestock, Agriculture and Fisheries (MGAP) and the National Customs Directorate, implemented a control protocol based on physical transport documents. Nevertheless, the reliance on paper documentation hinders oversight and creates operational inefficiencies. This study analyzes the development and implementation of the Agricultural Logistics Control Information System of Uruguay, a digital platform designed to improve traceability and optimize the control of agricultural transport. The methodology used combines interviews with key IDR stakeholder, documentary analysis of current regulations, and the application of the Rapid Application Development (RAD) approach for agile development. The system was developed using PHP 8.3 and Laravel 11, with FilamentPHP employed for generating interfaces and MySQL 8.4.2 for data management, ensuring data integrity through ACID properties. Key functionalities were incorporated, including the management of harvests and crops, inspection records, automated issuance of electronic transport documents, real-time enroute control, and an automated accounting model. Additionally, the system features a role-based access control model (Inspector, Producer, Police, and Administrator) and an auditing module to record changes in the database. Validated by the IDR, its implementation is scheduled for the 2025/2026 harvest, with future optimizations planned based on field use. Its scalable architecture allows for the incorporation of new crops without structural changes. In the future, the integration of a digital field notebook and predictive analysis tools is projected. The results of this work demonstrate that digitizing the agricultural oversight process can improve efficiency, reduce commercial fraud, and optimize logistics management in border regions. Moreover, the system presents a replicable model for other productive sectors that require enhanced traceability and control in agricultural distribution.

Keywords: Agricultural Informatics, Traceability, Agricultural logistics control, Digitalization, Electronic transport management.

1 Introducción

El control y la rastreabilidad logística de la producción agrícola son fundamentales para garantizar la calidad, autenticidad y competitividad de los productos en el mercado (Zúñiga et al., 2023). En el departamento de Rivera, Uruguay, la producción de sandías representa el 95% de la producción nacional, con un volumen estimado entre 11,000 y 15,000 toneladas por zafra (MGAP, 2023). Esta actividad emplea mayormente mano de obra de baja calificación y está dominada por pequeños productores familiares con predios de menos de cinco hectáreas (UTECH, 2022), lo que la convierte en un pilar socioeconómico clave en un departamento con el nivel de desarrollo humano más bajo del país (OPP, 2018) y bajos índices educativos en su población (OPP, 2019).

A pesar de su importancia, la producción agrícola en Rivera enfrenta desafíos significativos en términos de control y fiscalización. Su ubicación fronteriza la hace vulnerable a prácticas desleales, como el ingreso de productos sin origen claro, afectando la competitividad de los productores locales. Para abordar esta problemática, la Intendencia Departamental de Rivera (IDR), en conjunto con el Ministerio de Ganadería, Agricultura y Pesca (MGAP) y la Dirección Nacional de Aduanas (DNA), implementó un protocolo para el control de cultivos y la rastreabilidad de las cargas de sandías.

El protocolo consta de tres etapas principales: (i) el registro del productor en la Dirección General Impositiva (DGI) y el Banco de Previsión Social (BPS); (ii) la inspección de cultivos, donde un funcionario de la IDR analiza las condiciones y estima la producción esperada para determinar la cantidad que puede ser transportada; y (iii) la emisión de remitos, documentos obligatorios que acompañan cada carga y que deben ser validados en los controles de ruta. Sin embargo, la dependencia de documentación en papel genera retrasos de hasta diez días en la verificación de datos, dificultando una supervisión eficiente del flujo de productos.

En este contexto, el Sistema de Información de Control Logístico Agrícola del Uruguay (SICLAUy) surge como una solución para digitalizar y optimizar estos procesos. Su objetivo es mejorar la trazabilidad y eficiencia del control logístico del transporte de sandías mediante la digitalización de inspecciones, el monitoreo de cargas y la automatización de la emisión de remitos y controles.

Para ello, se establecen los siguientes objetivos específicos: (i) Digitalizar el proceso de inspección de cultivos, permitiendo a los funcionarios registrar datos de campo y generar estimaciones de producción de manera estructurada. (ii) Implementar un módulo de gestión de saldos de producción, reflejando la cantidad de producto autorizada para transporte y reduciendo el riesgo de inconsistencias. (iii) Automatizar la emisión y control de remitos digitales, asegurando la validez documental en los puntos de verificación en ruta. (iv) Garantizar la seguridad y trazabilidad de la información, mediante un módulo de auditoría que registre todas las interacciones dentro del sistema.

De esta manera, el sistema busca proporcionar una solución eficiente y alineada con las necesidades del sector, optimizando la supervisión del transporte agrícola y fortaleciendo la competitividad de los productores locales.

2 Referencial Teórico

2.1 Trabajo Relacionado

SICTT – Sistema de Información de Cargas del Transporte Terrestre. En Uruguay, el Sistema de Información de Carga del Transporte Terrestre (SICTT) es una herramienta fundamental para la formalización y el control del transporte de carga, implementada a través de la Guía Electrónica de Transporte de Carga. Operativo desde 2014, este sistema exige que los vehículos de más de 8,500 kg de peso bruto cuenten con una guía electrónica que registre datos como la identificación del transportista, información del vehículo, origen y destino del viaje, peso bruto de la carga y horarios, entre otros (MTOP, s.f.).

La información se gestiona a través de plataformas en línea desarrolladas por el Ministerio de Transporte y Obras Públicas (MTOP) y puede ser verificada en ruta por inspectores mediante la matrícula del vehículo como identificador principal. En algunos casos, la integración de dispositivos de monitoreo electrónico complementa los datos de las guías, mejorando la precisión en el control del transporte (MTOP, s.f.).

Si bien el SICTT busca formalizar y regularizar el transporte de carga en general, el sistema desarrollado en este proyecto responde a una problemática más específica del sector agrícola en Rivera. Ambas soluciones digitalizan procesos y garantizan la trazabilidad de las cargas, pero a diferencia del SICTT, el sistema propuesto incorpora módulos especializados en gestión de inspecciones agrícolas y saldos de producción, adaptándose a las particularidades de la región fronteriza y a las demandas específicas del sector agrícola.

2.2 Herramientas y Tecnologías

Análisis y Modelado UML. El Lenguaje Unificado de Modelado (UML) fue empleado en la etapa de análisis y modelado del sistema. Este lenguaje visual de propósito general permite especificar, visualizar, construir y documentar sistemas de software de manera estándar, facilitando la comunicación entre desarrolladores y stakeholders. UML se caracteriza por su flexibilidad y capacidad para capturar decisiones y conocimientos en todas las etapas del ciclo de vida del desarrollo, siendo especialmente útil en metodologías iterativas y orientadas a objetos (Booch et al., 2007).

Stack Tecnológico: PHP y Laravel. El sistema fue desarrollado con PHP 8.3 y el framework Laravel 11, seleccionados por su flexibilidad y eficiencia en aplicaciones web. PHP es un lenguaje versátil, compatible con múltiples plataformas y servidores como Apache y Nginx, y admite bases de datos relacionales como MySQL y no relacionales como MongoDB, lo que proporciona flexibilidad en la gestión de datos (Tatroe et al., 2020).

Laravel es un framework orientado a objetos que facilita el desarrollo rápido mediante módulos para autenticación, gestión de bases de datos y otras funcionalidades esenciales. Su capacidad para soportar aplicaciones de tamaño mediano permite la escalabilidad del sistema a medida que aumenten sus demandas (Stauffer, 2019).

Laravel sigue la arquitectura Modelo-Vista-Controlador (MVC), organizando el código en tres capas: (i) Modelo: Encapsula la lógica de acceso a datos y representa las entidades del sistema. Laravel emplea Eloquent ORM, que implementa el patrón Active Record, simplificando las operaciones CRUD (Create, Read, Update, Delete) (Stauffer, 2019). (ii) Vista: Maneja la presentación de la información con el motor de plantillas Blade, que separa la lógica de presentación de la lógica de negocio, facilitando el mantenimiento del código. (iii) Controlador: Actúa como intermediario entre las solicitudes HTTP, los modelos y las vistas, procesando las peticiones y generando respuestas dinámicas (Stauffer, 2019).

Interfaz de Usuario: Stack TALL y FilamentPHP. La interfaz administrativa del sistema fue implementada utilizando FilamentPHP versión 3.x, un paquete que forma parte del stack TALL (Tailwind CSS, Alpine.js, Laravel y Livewire). Este stack combina herramientas modernas para el diseño de interfaces, interactividad en el navegador y comunicación en tiempo real con el backend (FilamentPHP, s.f.).

Filament permite la generación automática de paneles administrativos basados en los modelos Eloquent de Laravel, reduciendo el tiempo de desarrollo al proporcionar componentes reutilizables. Este enfoque se alinea con las prácticas de desarrollo ágil, maximizando la eficiencia y minimizando esfuerzos repetitivos, sin comprometer la calidad del software.

A pesar de su enfoque en la automatización, Filament ofrece una flexibilidad considerable para personalizar las interfaces administrativas. Los desarrolladores pueden ajustar formularios, tablas y acciones según los requisitos específicos del proyecto, aprovechando herramientas como el Form Builder y el Table Builder. Esto asegura que las funcionalidades administrativas se adapten a las necesidades del usuario final mientras se mantiene la rapidez en el desarrollo (Filament, s.f.).

Base de Datos: MySQL. Para el manejo de datos estructurados, se utilizó MySQL 8.4.2, una base de datos relacional que garantiza propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) esenciales para mantener la integridad de los datos en transacciones críticas. La integración con Laravel mediante migraciones facilita la evolución del esquema de la base de datos, permitiendo una administración eficiente y adaptable (Silberschatz, 2020; Stauffer, 2019).

3 Metodología

3.1 Investigación Preliminar y Análisis

La fase inicial del proyecto tuvo como objetivo comprender las necesidades específicas relacionadas con el control y la fiscalización de la producción de sandías en Rivera. Para ello, se adoptó un enfoque colaborativo con el Coordinador del Proyecto de Desarrollo Hortofrutícola de la IDR, asegurando que las soluciones propuestas se alinearan con las demandas reales del contexto local.

Durante el levantamiento de requisitos, se llevaron a cabo actividades de consulta e investigación documental. En una primera reunión, el coordinador destacó los princi-

pales desafíos de la producción agrícola en la región, enfatizando la necesidad de digitalizar el proceso de rastreabilidad para mejorar la eficiencia y la transparencia. Esta interacción inicial permitió identificar problemas clave y sentó las bases para diseñar una solución tecnológica adecuada.

Posteriormente, se realizó un análisis documental exhaustivo, que incluyó informes del Observatorio Granjero de la Dirección General de la Granja (DIGEGR, 2023), noticias sobre la producción de sandías en Rivera publicadas en medios nacionales (El Observador, 2022; Montevideo Portal, 2024), el micrositio del Proyecto Sandía de la IDR (IDR, s.f.) y normativas vigentes relacionadas con el transporte de productos agrícolas (MTO, s.f.). Este análisis permitió proporcionar un contexto histórico y técnico, además de identificar requisitos implícitos y normativos.

Con esta base, se llevó a cabo una entrevista estructurada con el coordinador del proyecto, compuesta por 25 preguntas organizadas en cuatro ejes temáticos. Esta entrevista permitió definir el alcance del proyecto y extraer los requisitos funcionales iniciales. La información recabada se documentó y analizó minuciosamente, lo que facilitó la elaboración de diagramas de Actividad y de Casos de Uso. Estos diagramas sirvieron para visualizar el flujo de trabajo actual y establecer los objetivos del sistema.

3.2 Desarrollo del Sistema

Con los requisitos y el análisis iniciales definidos, la segunda etapa del proyecto se centró en el diseño e implementación del sistema. Para ello, se adoptaron principios y herramientas de metodologías ágiles que garantizaron un desarrollo eficiente y adaptable a las necesidades del proyecto. Entre las metodologías seleccionadas se encuentran Rapid Application Development (RAD), Lean Software Development y Scrum.

El enfoque Rapid Application Development (RAD) prioriza el desarrollo ágil mediante herramientas y frameworks que permiten construir prototipos funcionales en etapas tempranas (Beynon-Davies, 1999). En este contexto, se utilizó el framework Laravel y paquetes como Filament, que facilitaron la implementación rápida de funcionalidades esenciales y aseguraron un flujo de trabajo dinámico.

Por otro lado, Lean Software Development se enfocó en la optimización de recursos y la eliminación de desperdicios, evitando la implementación de funcionalidades no esenciales y manteniendo el enfoque en las necesidades críticas del sistema (Kniberg, 2011). Esto permitió cumplir con los plazos establecidos y reducir el esfuerzo en aspectos secundarios.

Finalmente, Scrum se utilizó como marco organizacional para estructurar el desarrollo en tareas y sprints, facilitando una gestión ordenada y orientada a objetivos (Schwaber, 2020). Se completaron 10 sprints de corta duración (entre 3 y 7 días cada uno), lo que permitió iteraciones muy ágiles y validación frecuente de los incrementos. Dado que el proyecto fue ejecutado por un único desarrollador, las ceremonias tradicionales de Scrum se adaptaron a un formato ligero: las reuniones diarias se reemplazaron por autoevaluaciones de avance y las retrospectivas por registros de lecciones aprendidas al cierre de cada iteración. Algunos de estos sprints estuvieron dedicados principalmente a la organización y refinamiento de historias de usuario, mien-

tras que el resto generó incrementos de producto. Las validaciones con el profesor orientador y el coorientador (Product Owner) se realizaron al cierre de cada sprint, asegurando la alineación con los requisitos y permitiendo ajustes rápidos en el backlog. Un tablero Kanban facilitó la visualización del progreso y la resolución ágil de impedimentos.

La combinación de estas metodologías ágiles proporcionó una base sólida para el desarrollo del sistema. Este enfoque permitió entregar una solución eficaz, alineada con las expectativas de los stakeholders y adaptada a las condiciones específicas del proyecto. El desarrollo técnico del proyecto fue llevado a cabo íntegramente por el autor principal, quien se encargó del diseño, implementación y validación técnica del sistema. A lo largo del proceso, se contó con el acompañamiento del profesor orientador, quien brindó apoyo en decisiones arquitectónicas y metodológicas, y con el coorientador, quien actuó como Product Owner, aportando los requisitos funcionales, priorizando funcionalidades y validando los entregables desde la perspectiva del usuario final.

3.3 Definición de las Épicas

Para definir las actividades del desarrollo del sistema, se utilizaron las herramientas de Historias de Usuario. Estas descripciones breves y claras de funcionalidades desde la perspectiva del usuario final facilitaron la comprensión de las necesidades del usuario, la priorización de funcionalidades y mejoraron la comunicación entre los stakeholders.

En el contexto de este proyecto, las actividades se organizaron en varias épicas que permitieron descomponer el desarrollo del sistema en bloques manejables, facilitando el cumplimiento de objetivos parciales y asegurando un progreso continuo y organizado en el proyecto, cada una agrupando historias de usuario y tareas relacionadas de manera estructurada.

El Product Backlog inicial constó de 66 Historias de Usuario, organizadas en las épicas (i) Gestión de zafras y cultivos, (ii) Gestión de inspecciones agrícolas, (iii) Gestión de producción y saldos, (iv) Emisión y verificación de remitos, (v) Control y monitoreo del transporte, (vi) Gestión de usuarios y roles, (vii) Seguridad y auditoría, (viii) UI/UX.

3.4 Herramientas Utilizadas

Diversas herramientas complementaron el desarrollo del sistema. Eclipse Papyrus fue utilizado para el diseño de diagramas UML y del modelo entidad-relación (ER), proporcionando un entorno visual conforme a estándares de modelado (Eclipse Foundation, 2024). En el diseño y gestión de la base de datos, MySQL Workbench facilitó la creación de esquemas mediante una interfaz gráfica intuitiva (Oracle, 2024).

Para el entorno de desarrollo local, Docker permitió replicar la infraestructura de producción en contenedores, garantizando consistencia en el desarrollo (Docker, Inc., 2024). PHPStorm, un entorno de desarrollo integrado (IDE), optimizó la organización y escritura del código mediante autocompletado y análisis estático, y su integración con GitHub Copilot agilizó tareas repetitivas mediante sugerencias de código basadas en inteligencia artificial (JetBrains, 2024; GitHub, Inc., 2024).

El control de versiones se gestionó con GitHub, una plataforma que ofreció un historial detallado de cambios, esencial para garantizar la trazabilidad en el desarrollo (GitHub, Inc., 2024). Además, para la gestión del proyecto, se utilizó Atlassian Jira Software, una herramienta ampliamente adoptada en entornos ágiles. Atlassian Jira Software permitió organizar tareas, planificar sprints, realizar un seguimiento detallado del progreso y gestionar impedimentos (Atlassian, 2024).

4 Resultados del Sistema

4.1 Requerimientos de Software

Los requerimientos de software, como define Sommerville (2011), constituyen un elemento fundamental en el desarrollo de sistemas. Representan la expresión formal de las necesidades de los usuarios y del sistema, sirviendo como base para las fases de diseño, implementación y pruebas.

Especificación de Requisitos. Durante la fase de análisis se identificaron las principales funcionalidades requeridas para el sistema. A continuación, se presenta un resumen de estas funcionalidades.

El sistema incorpora módulos clave para la gestión de inspecciones agrícolas, que permiten registrar y programar inspecciones en cultivos, así como generar reportes con estimaciones de producción. También incluye herramientas para el control y monitoreo del transporte, lo que abarca la emisión de remitos electrónicos para acompañar las cargas y su verificación en los puntos de control logístico.

En el área de gestión de incidentes, el sistema posibilita el registro de eventos detectados en los puntos de control ante irregularidades en las cargas. Adicionalmente, proporciona herramientas para que el administrador gestione estos incidentes mediante acciones como la anulación de remitos o la retención definitiva de cargas, conforme a reglas predefinidas.

La gestión de producción y saldos está diseñada para registrar y monitorear los saldos disponibles para transporte, permitiendo la deducción automática de cantidades transportadas. Por su parte, el módulo de gestión de usuarios y roles facilita la administración de perfiles y permisos asignados a distintos roles en el sistema.

Finalmente, se definieron funcionalidades relacionadas con la auditoría y seguridad del sistema, tales como el registro histórico de las acciones realizadas y la implementación de mecanismos de autenticación y medidas de seguridad para garantizar la protección de los datos.

Casos de Uso. Los casos de uso son una técnica fundamental para modelar el comportamiento del sistema desde la perspectiva del usuario. Estos describen las interacciones entre un actor y el sistema, proporcionando una visión clara y concisa de las funcionalidades que el sistema debe ofrecer, siendo una herramienta de comunicación eficaz entre los diferentes actores involucrados en un proyecto de Software (Sommerville, 2011).

En línea con los requerimientos funcionales definidos, se identificaron casos de uso clave relacionados con la gestión de zafras, el registro de inspecciones, la emisión de remitos y los controles policiales. Estos representan las interacciones más relevantes entre los usuarios y el sistema.

Estos casos de uso reflejan las interacciones principales previstas entre los usuarios y el sistema, asegurando que las funcionalidades implementadas respondan a las necesidades específicas del proyecto.

4.2 Modelo Lógico

El modelo lógico del sistema, representado mediante un diagrama Entidad-Relación (no UML) (ver Fig. 1), proporciona una representación conceptual de los datos que serán almacenados y gestionados en la base de datos. Este modelo es una herramienta esencial para el diseño de bases de datos, ya que permite visualizar de manera estructurada las entidades y las relaciones entre ellas. Según Silberschatz, Korth y Sudarshan (2020), los modelos lógicos facilitan la comprensión del esquema de datos y garantizan su alineación con los requerimientos del sistema.

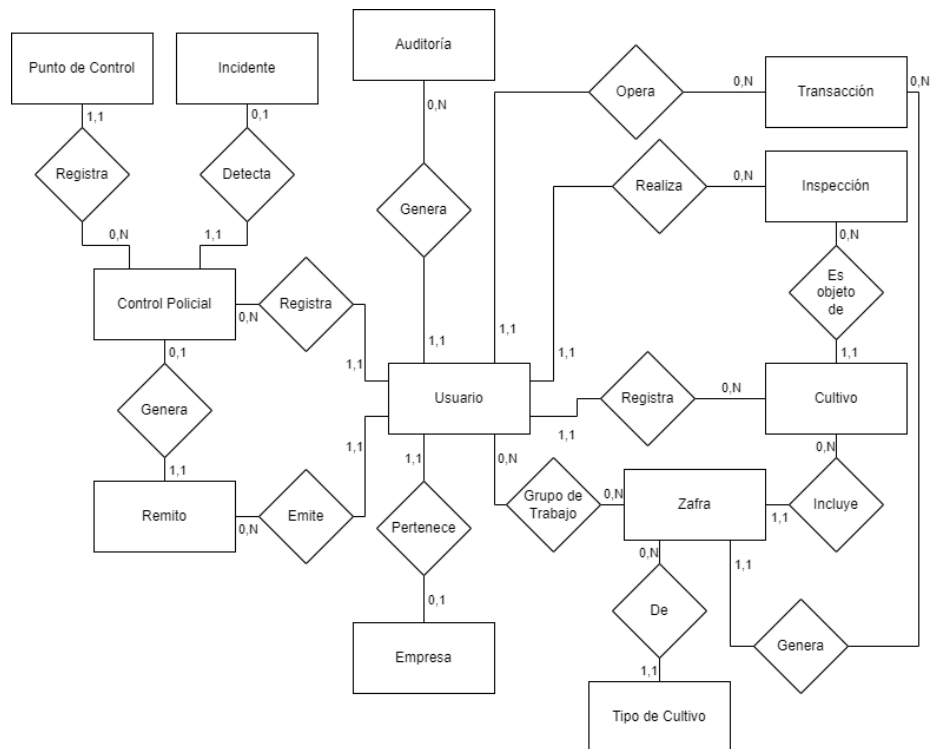


Fig.1. Diagrama Entidad-Relación (ER). Fuente: Elaboración propia.

En este proyecto, además se utilizó un modelo Entidad-Relación Extendido. Este enfoque permite capturar relaciones más complejas, como jerarquías y especializacio-

nes, reflejando con mayor precisión las especificidades de las entidades involucradas. La adopción de este modelo asegura un diseño de base de datos consistente con los requisitos funcionales y no funcionales del sistema, lo que resulta crucial para la implementación de un sistema robusto y escalable.

4.3 Arquitectura del Sistema

Como se detalló en la sección 2.2, el sistema se basa en la arquitectura Modelo-Vista-Controlador (MVC), ampliamente utilizada en el desarrollo de software por su capacidad de separar responsabilidades y mejorar la mantenibilidad, testabilidad y escalabilidad de las aplicaciones (Gamma et al., 1995). Esta estructura modular permite desarrollar cada componente de forma independiente, garantizando claridad en el diseño.

Dentro de esta arquitectura, Filament se integra de manera eficiente con Laravel, proporcionando una capa adicional de abstracción para la gestión de interfaces administrativas. Aprovechando los modelos Eloquent y las capacidades de generación automática de paneles (como se detalla en la sección 2.2.3), Filament permite alinear la lógica de negocio con la interfaz de usuario de forma fluida (Filament, s.f.).

Su implementación ha optimizado el desarrollo de la interfaz administrativa, reduciendo tiempos y facilitando la personalización. La flexibilidad de Filament para ajustar formularios y tablas según las necesidades del proyecto ha sido clave para adaptar la interfaz a los requerimientos específicos del sistema. Además, su capacidad para manejar relaciones complejas ha simplificado significativamente la gestión de datos.

4.4 Sistema en Funcionamiento

Autenticación y Autorización. Con el objetivo de garantizar la integridad y confidencialidad de los datos, se implementó un sistema de autenticación y autorización basado en roles, utilizando las herramientas de Laravel y Filament. Se definieron cuatro roles principales dentro del modelo Usuario: Inspector, Productor, Policía y Administrador, cada uno con permisos diferenciados, lo que asegura la seguridad del sistema y personaliza la experiencia del usuario al adaptar las funcionalidades según sus responsabilidades.

Además, el sistema incluye un módulo de auditoría desarrollado con Laravel Auditing, que registra acciones clave, como modificaciones de datos, creación de usuarios y eventos de inicio y cierre de sesión. Estos registros se almacenan en una tabla dedicada en la base de datos, cumpliendo con requisitos de trazabilidad y normativas aplicables.

Administración General. El panel de administración ofrece una interfaz centralizada para gestionar las operaciones diarias del sistema. Los administradores tienen privilegios extendidos, como la gestión de usuarios y roles, la configuración de tipos de cultivos y la definición de puntos de control policiales.

Originalmente concebido para la gestión de sandías, el sistema fue diseñado para ser escalable, permitiendo agregar dinámicamente nuevos tipos de cultivos a través de

un módulo dedicado. Esto reduce los esfuerzos de desarrollo futuros y adapta el sistema a diferentes escenarios agrícolas.

Gestión de Zafras y Cultivos. El módulo de gestión de zafras agrícolas permite a los administradores definir períodos específicos para cada cultivo, estableciendo fechas, asignando productores y policías, y verificando que no existan superposiciones entre campañas. Los productores, asociados a una zafra específica, pueden registrar detalles de sus cultivos como áreas sembradas y fechas clave.

Gestión de Inspecciones y Estimación de Rendimientos. Para ofrecer a los productores una estimación precisa del rendimiento esperado, el sistema incluye un módulo de inspecciones. Los inspectores programan visitas de campo para recolectar datos sobre el estado del cultivo, dimensiones, sanidad vegetal y rendimiento estimado. Los resultados se documentan en informes detallados que incluyen fotografías y observaciones, proporcionando una base sólida para la toma de decisiones comerciales.

Gestión Electrónica de Guías de Transporte (Remitos). El sistema implementa un módulo para la gestión electrónica de remitos, optimizando la trazabilidad de las cargas. Los productores generan remitos digitales que incluyen información sobre la carga, el transporte y los datos del conductor. En esta primera versión, los remitos electrónicos conviven con los físicos, permitiendo registrar ambos en paralelo para facilitar la transición.

Gestión de Saldos de Producción. El sistema emplea un modelo contable simplificado para gestionar saldos de producción, ajustando automáticamente los registros tras inspecciones y emisión de remitos. Estas herramientas aseguran que las transacciones no superen los saldos disponibles, garantizando integridad y previniendo irregularidades. También permite ajustes manuales bajo condiciones específicas, con la debida justificación.

Controles Policiales y Gestión de Incidentes. El módulo de controles vehiculares permite a los policías verificar la validez de los remitos electrónicos en los puntos de control y autorizar el remito. En caso de irregularidades, como discrepancias en la carga, el sistema registra incidentes que son notificados al administrador, quien evalúa la situación y toma las acciones correspondientes, como anulación de remitos o retención de cargas.

Adaptabilidad y Uso en dispositivos móviles. El sistema desarrollado no requirió una aplicación nativa para dispositivos móviles, ya que no se identificaron necesidades específicas que justificaran el uso de recursos exclusivos de estas plataformas. Sin embargo, para garantizar una experiencia óptima en dispositivos móviles, se aprovechó la capacidad de adaptabilidad ofrecida por Filament.

Filament, basado en el framework Tailwind CSS, permite a los paneles administrativos ajustarse de manera eficiente a diferentes tamaños de pantalla. Gracias a sus funcionalidades de diseño responsivo, el sistema ofrece una interfaz adaptativa que

asegura un acceso eficiente y cómodo tanto desde dispositivos móviles como de escritorio.

Adicionalmente, se implementó la característica de Single Page Application (SPA), lo que permite reducir significativamente los tiempos de carga al evitar la recarga completa de los assets de la página durante la navegación. Esta funcionalidad mejora la experiencia del usuario al proporcionar una interacción más fluida y rápida, especialmente en dispositivos con recursos limitados o en conexiones de red menos estables.

5 Conclusiones

El proyecto SICLAUy - Sistema de Información de Control Logístico Agrícola del Uruguay alcanzó los principales objetivos planteados inicialmente, ofreciendo una solución integral para digitalizar y optimizar los procesos de control agrícola y logístico en el departamento de Rivera. Mediante el uso de tecnologías modernas y metodologías ágiles, se desarrolló un sistema escalable que responde a necesidades específicas del sector agrícola del departamento, mejorando la trazabilidad, el monitoreo del transporte y la gestión de inspecciones.

El sistema incluye módulos esenciales como la gestión de inspecciones, la generación de remitos electrónicos y la administración de saldos de producción, lo que garantiza un control más eficiente de las cargas agrícolas. La segmentación de roles y la personalización de funcionalidades fortalecieron la seguridad y usabilidad, mientras que el diseño flexible permitió ampliar su aplicación a otros cultivos. Además, contribuyó significativamente a la digitalización de procesos, mejorando la transparencia y reforzando el control logístico mediante herramientas como el módulo de auditoría y la gestión de incidentes.

A pesar de los logros alcanzados, se identificaron limitaciones importantes, como la necesidad normativa de mantener los remitos físicos en paralelo con los electrónicos, lo que representa un desafío para la transición total a un sistema digital. Además, la implementación inicial se enfocó exclusivamente en cultivos agrícolas, limitando su aplicabilidad a otros sectores de producción.

El sistema proyecta un impacto en la competitividad y formalización de los productores agrícolas de Rivera, incluyendo una mayor transparencia y eficiencia en las operaciones agrícolas y logísticas. Esta mejora en la transparencia y eficiencia en las operaciones logísticas y agrícolas, amplía la capacidad de supervisión de las autoridades y establece una base tecnológica replicable en otros contextos productivos y departamentos fronterizos.

Con el objetivo de ampliar las funcionalidades y la adaptabilidad del sistema, se proponen las siguientes líneas de desarrollo futuro: (i) la implementación de un cuaderno de campo digital que garantice una trazabilidad integral, (ii) la incorporación de herramientas avanzadas para generar reportes personalizados y análisis predictivos, (iii) y la expansión del sistema al transporte de productos pecuarios atendiendo el interés expresado por la IDR y DIGEGRA.

Si bien la estructura actual de roles y permisos cumple con los requerimientos actuales del sistema, se considera que la implementación del paquete Laravel-permission de Spatie podría ofrecer una mayor escalabilidad y flexibilidad para futuras expansiones. Este paquete permitiría definir de manera granular y jerárquica los roles y permisos, facilitando la gestión de usuarios con diferentes niveles de acceso y adaptándose a posibles cambios en las necesidades del sistema.

Referencias

- Atlassian. (2024). Jira Software: Project and issue tracking. Recuperado el 25 de octubre de 2024 de <https://www.atlassian.com/software/jira>
- Beynon-Davies, P. et al. (1999). Rapid Application Development (RAD): an empirical review. *European Journal of Information Systems*. Recuperado el 14 de noviembre de 2024 en https://www.researchgate.net/publication/31978101_Rapid_application_development_RAD_An_empirical_review
- Booch, G. et al. (2007). El lenguaje unificado de modelado Manual de Referencia. (2º ed.) España: Pearson Educación.
- Didtrans. (2023). Software de gestión para empresas de transporte y transportistas. Recuperado el 19 de noviembre de 2024 de <https://www.didtrans.com>
- DIGEGRA – UAM (2023). Zafra de sandía 2022/23. Observatorio Granjero. Uruguay: MGAP. Recuperado el 1 de junio de 2024 de https://www.uam.com.uy/images/DESARROLLO_COMERCIAL/InformesEspeciales/Zafra_de_sandia_2022-23_-_Observatorio_Granjero.pdf
- Docker, Inc. (2024). Docker Overview. Disponible en: <https://docs.docker.com/get-started/overview/>
- Eclipse Foundation. (2024). Papyrus Documentation. Disponible en: <https://www.eclipse.org/papyrus/>
- El Observador (2022). Gobierno busca instalar al sustituto del "Gran Hermano del Transporte" de carga en 2023. Recuperado el 1 de junio de 2024 de <https://www.elobservador.com.uy/nota/gobierno-busca-instalar-al-sustituto-del-gran-hermano-del-transporte-de-carga-en-2023-20221116143257>
- FilamentPHP. (s.f.). FilamentPHP - Accelerated Laravel development framework: admin panel, form builder, table builder and more. Recuperado de <https://filamentphp.com/docs> el 14 de noviembre de 2024.
- Gamma, E., et al. (1995). Design patterns: Elements of reusable object-oriented software. Addison-Wesley Professional.
- GitHub, Inc. (2024). GitHub Documentation. Disponible en: <https://docs.github.com/>
- GitHub, Inc. (2024). GitHub Copilot Documentation. Disponible en: <https://docs.github.com/copilot>
- IDR (s.f.). Proyecto Sandía. Uruguay: Intendencia Departamental de Rivera. Recuperado el 4 de junio de 2024 de <https://www.rivera.gub.uy/medioambiente/estrategias/desarrollo-rural/proyecto-sandia/>
- JetBrains. (2024). PHPStorm Documentation. Disponible en: <https://www.jetbrains.com/phpstorm/documentation/>
- Kniberg, H. (2011). Lean from the Trenches: Managing Large-Scale Projects with Kanban. (1º ed.) USA: Pragmatic Bookshelf.
- Montevideo Portal (2024). Incautan 14.000 kilos de comida que ingresó desde Brasil; se destruirán por orden judicial. Recuperado el 21 de junio de 2024 de

<https://www.montevideo.com.uy/Noticias/Incautan-14-000-kilos-de-comida-que-ingreso-desde-Brasil-se-destruiran-por-orden-judicial-uc891788#com>
MTOP (s.f.). Guía de Carga. Recuperado el 10 de junio de 2024 de <https://www.gub.uy/ministerio-transporte-obras-publicas/comunicacion/publicaciones/preguntas-frecuentes/preguntas-frecuentes/guia-carga>
OPP, INE (2019). Perfil departamental Rivera. Observatorio Territorio Uruguay. Uruguay: OPP. Recuperado el 1 de junio de 2024 de <https://otu.opp.gub.uy/perfiles/rivera>
OPP, INE, BCU (2018). Índice de Desarrollo Humano por departamento. Observatorio Territorio Uruguay. Uruguay: OPP. Recuperado el 1 de junio de 2024 de https://otu.opp.gub.uy/?q=listados/listados_datos_formato&id=2679&cant=0&fecha=2018-01-01
Oracle. (2024). MySQL Workbench. Disponible en: <https://dev.mysql.com/doc/workbench/en/>
SAP. (s.f.). SAP Transportation Management. Recuperado el 19 de noviembre de 2024 de <https://www.sap.com/latinamerica/products/scm/transportation-logistics.html>
Schwaber, K. et al. (2020). The Scrum Guide. Recuperado el 14 de noviembre de 2024 en: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
Silberschatz, A. et al. (2020). Database System Concepts. (7º ed.) USA: McGraw-Hill Education.
Software Logística. (2024). Sistemas de gestión de transporte TMS: las 4 mejores alternativas. Recuperado el 19 de noviembre de 2024 de <https://softwarelogistica.org/mejores-sistemas-de-gestion-de-transporte-tms/>
Somerville, I. (2011). Ingeniería del Software. (9º ed.) Mexico: Pearson Educación de Mexico.
Stauffer, M. (2019). Laravel: Up & Running. (2º ed.) USA: O'Reilly.
Tatroe, K, et al. (2020). Programming PHP. (4º ed.) USA: O'Reilly.
UTEC Innova Evaluación y Estadística (2022). Informe Final de Evaluación del Proyecto “Fortalecimiento de la cadena de producción de la sandía en productores familiares del Departamento de Rivera” (2022). Uruguay: UTEC.
Zúñiga, H, et al. (2023). Trazabilidad en el sector agrícola: una revisión para el período 2017 – 2022. Costa Rica: Universidad de Costa Rica, Agronomía Mesoamericana, vol. 34, núm. 2. Recuperado el 1 de junio de 2024 de <https://www.redalyc.org/articulo.oa?id=43774024012>