

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Desarrollo de una plataforma web para gestión administrativa  
de formularios de recolección de datos**

Trabajo de graduación en modalidad de Trabajo Profesional presentado  
por Kristopher Javier Alvarado López para optar al grado académico de  
Licenciado en Ingeniería en Ciencia de la Computación y Tecnologías de  
la Información

Guatemala,

2025



UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



**Desarrollo de una plataforma web para gestión administrativa  
de formularios de recolección de datos**

Trabajo de graduación en modalidad de Trabajo Profesional presentado  
por Kristopher Javier Alvarado López para optar al grado académico de  
Licenciado en Ingeniería en Ciencia de la Computación y Tecnologías de  
la Información

Guatemala,

2025



Vo.Bo.:

(f) \_\_\_\_\_

Ing. Erick Francisco Marroquín Rodríguez



Tribunal Examinador:

(f) \_\_\_\_\_

Ing. Erick Francisco Marroquín Rodríguez



(f) \_\_\_\_\_

Ing. Marlon Osiris Fuentes López



Fecha de aprobación: Guatemala, 22 de noviembre de 2025.



---

## Prefacio

---

El desarrollo de este megaproyecto representó un proceso de aprendizaje integral que combinó conocimientos técnicos, diseño de soluciones y trabajo colaborativo. La propuesta surgió a partir de la necesidad del ingenio azucarero de contar con una herramienta moderna y eficiente que facilitara la gestión de los formularios utilizados en la recopilación de datos agrícolas, optimizando el flujo de información entre las distintas áreas operativas.

A lo largo del proyecto, se enfrentaron diversos desafíos relacionados con la estructuración lógica del sistema, la organización de sus componentes visuales y la implementación de mecanismos que garantizaran una interacción fluida, intuitiva y coherente con las necesidades de los usuarios. Cada etapa implicó un proceso de análisis, diseño y validación orientado a lograr una experiencia práctica y agradable para quienes utilizarán la plataforma como parte de sus labores diarias. Si bien algunas funcionalidades —como el despliegue definitivo en los servidores institucionales y los módulos de reportería avanzada— aún se encuentran pendientes de implementación, se establece una base sólida que permitirá su integración en futuras fases del proyecto.

Este trabajo no habría sido posible sin la colaboración y el compromiso de todas las personas que aportaron sus conocimientos y tiempo durante su desarrollo. Se agradece al equipo técnico del ingenio por su constante apoyo y por brindar información valiosa durante las pruebas y validaciones del sistema. Asimismo, se expresa gratitud al asesor de proyecto por su orientación, acompañamiento y aportes técnicos, los cuales fueron esenciales para alcanzar los resultados obtenidos. Finalmente, se reconoce el esfuerzo y la dedicación del equipo de trabajo, así como el intercambio de ideas que contribuyó a consolidar este proyecto.

Este trabajo representa no solo el resultado de una etapa académica, sino también una experiencia formativa que reafirma la importancia de diseñar soluciones tecnológicas enfocadas en la funcionalidad, la claridad y la experiencia del usuario, contribuyendo al fortalecimiento del proceso de transformación digital dentro del sector agroindustrial.



---

## Índice

---

<b>Prefacio</b>	v
<b>Lista de figuras</b>	xii
<b>Lista de abreviaturas y siglas</b>	xiii
<b>Lista de cuadros</b>	xv
<b>Resumen</b>	xvii
<b>Abstract</b>	
<b>1. Introducción</b>	1
<b>2. Antecedentes</b>	3
<b>3. Justificación</b>	5
<b>3.1. Justificación</b>	5
<b>4. Objetivos</b>	7
<b>4.1. Objetivo general</b>	7
<b>4.2. Objetivos específicos</b>	7
<b>5. Alcance</b>	9
<b>6. Marco teórico</b>	11
<b>6.1. Agricultura 4.0 y la importancia estratégica de los datos</b>	11
<b>6.2. Sistemas de Información de Gestión Agrícola (FMIS) y formularios digitales</b>	12
<b>6.3. Arquitectura tecnológica de plataformas web</b>	12
<b>6.3.1. Arquitecturas tradicionales y modernas</b>	12
<b>6.4. Usabilidad y experiencia de usuario (UX)</b>	13
<b>6.4.1. Teoría del color en el diseño de interfaces para entornos administrativos agrícolas</b>	14
<b>6.5. Gestión y persistencia de datos en aplicaciones web</b>	16

6.5.1. Gestión de sincronización y consistencia en aplicaciones web . . . . .	16
6.5.2. Actualización en tiempo real en aplicaciones web . . . . .	17
6.6. Validaciones dinámicas y tipado estático en TypeScript . . . . .	18
6.6.1. Modelos conceptuales para la definición de reglas . . . . .	18
6.6.2. Aislamiento y ejecución segura en el cliente . . . . .	19
6.6.3. Principios y consideraciones de seguridad . . . . .	19
<b>7. Metodología . . . . .</b>	<b>21</b>
7.1. Selección de la Arquitectura Frontend . . . . .	21
7.2. Arquitectura Tecnológica Propuesta . . . . .	22
7.3. Diseño de Componentes y Gestión de Estado . . . . .	22
7.4. Implementación de la Interfaz de Usuario y Usabilidad . . . . .	23
7.4.1. Referente de Diseño: Aplicación Web DIGIFORMS . . . . .	23
7.4.2. Diseño Centrado en el Usuario (UX) . . . . .	23
7.4.3. Implementación de Formularios Dinámicos . . . . .	24
7.4.4. Control de Acceso en la Interfaz . . . . .	24
7.4.5. Retroalimentación Visual y Prevención de Errores . . . . .	24
7.5. Gestión de Datos y Comunicación en el Frontend . . . . .	25
7.5.1. Comunicación con la API Backend . . . . .	25
7.5.2. Gestión de Estado del Servidor con React Query . . . . .	25
7.5.3. Persistencia y Sincronización Offline . . . . .	25
7.6. Validación y Calidad de Datos en la Interfaz . . . . .	25
7.7. Implementación de Validaciones en el Frontend . . . . .	26
<b>8. Resultados . . . . .</b>	<b>27</b>
8.1. Pruebas de Usabilidad . . . . .	27
8.2. Resultados Cuantitativos y Evaluación de Desempeño . . . . .	30
8.3. Observaciones y Retroalimentación Cualitativa . . . . .	32
<b>9. Conclusiones . . . . .</b>	<b>33</b>
<b>10. Recomendaciones . . . . .</b>	<b>35</b>
<b>11. Bibliografía . . . . .</b>	<b>37</b>
<b>12. Anexos . . . . .</b>	<b>41</b>
<b>Anexo A: Manual del Módulo Frontend . . . . .</b>	<b>41</b>
12.0.1. Acceso al Sistema . . . . .	1
12.0.2. Descripción General de la Interfaz . . . . .	1
12.0.3. Barra de Navegación y Paneles de Contenido . . . . .	2
12.0.4. Panel de <i>Dashboard</i> . . . . .	4
12.0.5. Listado de categorías . . . . .	5
12.0.6. Formularios . . . . .	9
12.0.7. Fuentes de datos . . . . .	33
12.0.8. Procesos de Exportación . . . . .	35
12.0.9. Sesión: Usuarios . . . . .	37
<b>Glosario . . . . .</b>	<b>43</b>

---

## Lista de figuras

---

1.	Esquema de un sistema IoT aplicado a la Agricultura 4.0, donde los sensores de campo recopilan datos ambientales y del suelo, los cuales se envían a una plataforma en la nube para su análisis y control remoto. . . . .	11
2.	Arquitecturas web: monolítica y basada en servicios. . . . .	13
3.	Principios de usabilidad y experiencia de usuario . . . . .	16
4.	Flujo de sincronización cliente-servidor en modo offline / online . . . . .	17
5.	Diagrama conceptual: validaciones dinámicas - tipado estático en TypeScript . . . . .	19
6.	Arquitectura general del módulo <i>frontend</i> de la plataforma web. . . . .	22
7.	Sesión inicial de pruebas de usabilidad con el equipo técnico y operativo. . . . .	28
8.	Presentación de la interfaz principal de la WebApp durante la sesión de demostración. . . . .	28
9.	Participante realizando pruebas de registro y edición de formularios. . . . .	29
10.	Evaluación de los procesos de carga de datos y sincronización con la base de datos. . . . .	29
11.	Usuario probando la interfaz de la aplicación web y validando la experiencia de uso. . . . .	30
12.	Pantalla de acceso al sistema. . . . .	1
13.	Vista general del <i>dashboard</i> principal del sistema <i>Santa Ana AgroForms</i> . . . . .	2
14.	Barra de navegación y panel principal de contenido del sistema. . . . .	3
15.	Panel lateral de navegación principal del sistema. . . . .	4
16.	Vista del panel principal de <i>Dashboard</i> con indicadores y gráficos de actividad del sistema. . . . .	5
17.	Vista general de la sección <i>Listado de Formularios</i> . . . . .	6
18.	Ventana modal para la creación de una nueva categoría. . . . .	7
19.	Ventana modal para la edición de una categoría existente. . . . .	8
20.	Ventana modal de confirmación para la eliminación de una categoría. . . . .	9
21.	Botón para la creación de un nuevo formulario. . . . .	9
22.	Ventana de adición de un nuevo formulario. . . . .	10
23.	Menú de opciones disponible para cada formulario. . . . .	10
24.	Ventana para asignar un formulario a uno o varios usuarios. . . . .	11

25. Ventana de edición de un formulario existente.	12
26. Proceso de duplicación de un formulario existente.	13
27. Ventana de confirmación para suspender un formulario.	13
28. Ventana de confirmación para eliminar un formulario.	14
29. Acceso a la vista de edición de un formulario.	15
30. Campo de tipo Texto disponible en el generador de formularios.	15
31. Ventana de edición para definir los valores básicos de un campo.	16
32. Campo de tipo Número disponible en el generador de formularios.	16
33. Ventana de edición de campo numérico con límites de datos configurables.	17
34. Campo de tipo Switch disponible en el generador de formularios.	17
35. Ventana de edición del campo Switch con opciones de configuración básicas.	18
36. Campo de tipo Fecha disponible en el generador de formularios.	18
37. Ventana de edición del campo Fecha con opciones de configuración y visualización.	19
38. Campo de tipo Hora disponible en el generador de formularios.	19
39. Ventana de edición del campo Hora con opciones de configuración básicas.	20
40. Campo de tipo Combo disponible en el generador de formularios.	20
41. Ventana de edición del campo Combo con configuración de valores.	21
42. Ejemplo de valores ingresados para un campo Combo.	21
43. Ejemplo del campo Combo con sus opciones disponibles en el formulario.	22
44. Campo de tipo Dataset disponible en el generador de formularios.	22
45. Ventana de edición del campo Dataset con opciones de configuración.	23
46. Ejemplo de datos cargados desde un archivo Excel para el campo Dataset.	23
47. Selección del Dataset desde la lista desplegable de fuentes de datos.	24
48. Selección de la columna asociada al Dataset.	24
49. Configuración final del campo Dataset antes de guardar.	25
50. Campo de tipo Firma disponible en el generador de formularios.	25
51. Ventana de edición del campo Firma con opciones de configuración básicas.	26
52. Campo de tipo Calculado disponible en el generador de formularios.	26
53. Ventana de edición del campo Calculado con configuración de fórmula.	27
54. Definición de una interfaz en TypeScript utilizada dentro del campo calculado.	27
55. Declaración de la función <code>calculateTheme</code> que retorna un valor en formato String.	28
56. Ejemplo de cálculo mediante la multiplicación de dos campos numéricos.	28
57. Campo de tipo Grupo disponible en el generador de formularios.	29
58. Ventana de edición del campo Grupo con parámetros configurables.	29
59. Ejemplo de un campo de texto asociado a un grupo existente dentro del formulario.	29
60. Campo asociado a un grupo mostrado como elemento indexado dentro de la lista de campos.	30
61. Vista del formulario de configuración de páginas.	31
62. Ventana de creación de una nueva página.	32
63. Selección de página mediante el menú desplegable de navegación.	32
64. Navegación entre páginas desde la vista del teléfono móvil.	33
65. Vista general de la sección Fuentes de Datos.	34
66. Ejemplo de fuente de datos cargada con sus columnas identificadas.	34
67. Vista del proceso de exportación de formularios: listado de formularios, botón de exportar y ventana de selección de formato.	36

68. Opción para exportar todos los formularios y ventana modal de selección de formato.	37
69. Vista general de la sección de usuarios.	38
70. Ventana de edición de usuario con campos de registro y control de acceso.	39
71. Ventana de generación de código QR para autenticación de usuario.	40
72. Ventana con la opción para descargar el código QR en formato PNG.	41
73. Archivo del código QR descargado exitosamente.	41
74. Ventana de confirmación para la eliminación de usuario.	42

---

## Lista de abreviaturas y siglas

---

**API** Interfaz de Programación de Aplicaciones.

**CRUD** Create, Read, Update, Delete.

**CSS** Cascading Style Sheets.

**DOM** Document Object Model.

**FMIS** Farm Management Information System.

**GIS** Geographic Information System.

**HTML** HyperText Markup Language.

**IoT** Internet of Things.

**JSON** JavaScript Object Notation.

**JWT** JSON Web Token.

**QA** Quality Assurance.

**QR** Quick Response Code.

**RBAC** Role-Based Access Control.

**REST** Representational State Transfer.

**SPA** Single Page Application.

**SQL** Structured Query Language.

**SSE** Server-Sent Events.

**TS** TypeScript.

**UI** User Interface.

**UX** User Experience.

**WebApp** Aplicación Web.

## Lista de cuadros

## 1. Resultados de las pruebas de usabilidad. . . . . 31



---

## Resumen

---

En el ámbito agrícola, la gestión eficiente de la información recolectada en campo es un proceso esencial para garantizar la trazabilidad, el control y la toma de decisiones oportunas. Sin embargo, muchas de las tareas asociadas a la captura y administración de datos aún se realizan de forma manual o mediante herramientas dispersas, lo que dificulta la organización y el aprovechamiento de la información generada.

Con el propósito de atender esta necesidad, el presente megaproyecto se centró en el desarrollo e implementación de una plataforma web administrativa que permite gestionar, organizar y controlar los formularios utilizados en la recopilación de datos agrícolas del ingenio azucarero. La solución propuesta fue concebida con un enfoque modular, integrando funcionalidades como la creación, edición, duplicación, suspensión y eliminación de formularios, así como la asignación de permisos específicos a distintos usuarios. Además, se diseñó una interfaz intuitiva, moderna y accesible, adaptada a los procesos operativos del personal técnico y administrativo.

Durante el desarrollo se realizaron pruebas de usabilidad con usuarios reales, lo que permitió validar la facilidad de uso y el desempeño del sistema, además de identificar oportunidades de mejora orientadas a la reportería, conexión con fuentes de datos externas y registro automático de terminales.

El proyecto culmina con una plataforma robusta, escalable y adaptable, que sienta las bases para la digitalización y modernización de los procesos de gestión de información agrícola, contribuyendo directamente a la eficiencia operativa y al fortalecimiento tecnológico del ingenio azucarero.



---

## Abstract

---

In the agricultural field, efficiently managing field-collected data is essential to ensure traceability, control, and timely decision-making. However, many data collection and management tasks are still carried out manually or through isolated tools, making it difficult to organize and fully utilize the generated information.

To address this challenge, this project focused on the development and implementation of a web-based administrative platform designed to manage, organize, and control the forms used in agricultural data collection at the sugar mill. The proposed solution was conceived under a modular approach, integrating functionalities such as form creation, editing, duplication, suspension, and deletion, as well as specific permission assignment for different users. Furthermore, the system features an intuitive and accessible interface aligned with the operational needs of both technical and administrative personnel.

Throughout the process, usability tests were conducted with real users, validating the platform's usability and performance while identifying opportunities for improvement related to reporting, external data source integration, and automatic device registration.

The project concludes with a robust, scalable, and adaptable platform that establishes the foundation for the digital transformation and modernization of agricultural information management, directly contributing to operational efficiency and technological advancement within the sugar mill.



# CAPÍTULO 1

---

## Introducción

---

En las últimas décadas, la agricultura ha experimentado una transformación significativa gracias a la digitalización y la adopción de tecnologías de información. La capacidad de recolectar, organizar y analizar datos de campo —desde parámetros climáticos y del suelo hasta registros de riego y control de plagas— se ha convertido en un elemento clave para mejorar la productividad, la sostenibilidad y la resiliencia de los cultivos. Bajo este enfoque de agricultura de precisión, las soluciones digitales permiten a los ingenieros agrónomos y al personal de campo tomar decisiones basadas en información actualizada, optimizar recursos y reducir costos operativos (World Business Council for Sustainable Development (WBCSD), 2020).

En la actualidad, el sector agrícola enfrenta retos que van más allá de la variabilidad estacional. La ocurrencia de sequías prolongadas o lluvias intensas altera los ciclos de siembra y cosecha; la cobertura de Internet en zonas rurales es inestable, lo que dificulta la captura y envío de datos en tiempo real; y las parcelas, a menudo dispersas en terrenos de difícil acceso, obligan a un registro manual que consume tiempo y puede originar errores. A esto se suma la fragmentación de formatos —como papel, hojas de cálculo y aplicaciones independientes— que impide la estandarización y consolidación de la información, así como los retrasos en la validación de datos que obstaculizan una reacción ágil ante plagas, enfermedades o fluctuaciones del mercado.

Dentro de este contexto, la creación de una plataforma administrativa web que permita gestionar de manera eficiente los formularios de recolección de datos agrícolas, definir tipos de formulario según la operación (siembra, riego, monitoreo de plagas, entre otros), asignar permisos específicos a técnicos y supervisores y validar la información en campo, se presenta como una solución clave para optimizar los procesos internos en empresas agrícolas.

El presente proyecto se enfoca en desarrollar e implementar una plataforma web administrativa para la gestión, organización y control de los formularios utilizados en la recolección de datos agrícolas en un ingenio azucarero. El alcance del proyecto comprende el diseño de la arquitectura del sistema, la implementación del módulo administrativo para la creación y edición de formularios, la gestión de usuarios y roles mediante controles de acceso y la in-

tegración de validaciones para asegurar la integridad de los datos capturados en campo. No incluye el despliegue final en servidores institucionales ni el desarrollo de módulos avanzados de reportería, los cuales se plantean como fases posteriores.

Para lograr este objetivo, se empleó una metodología basada en el análisis de requisitos, el diseño modular de la solución, la implementación iterativa de los componentes y la validación continua mediante pruebas de usabilidad con usuarios representativos. Este enfoque permitió ajustar la interfaz y los flujos según las necesidades reales del personal técnico y administrativo, asegurando que la plataforma fuera intuitiva, accesible y coherente con los procesos operativos del ingenio.

La implementación de esta solución contribuirá a reducir errores manuales, mejorar la trazabilidad de la información y acelerar los procesos internos de verificación y análisis. Asimismo, fortalecerá la estandarización de los registros operativos y permitirá que los datos capturados en campo sean gestionados de manera más confiable y oportuna, facilitando la toma de decisiones estratégicas dentro del ingenio.

## CAPÍTULO 2

---

### Antecedentes

---

La digitalización de los procesos agrícolas ha cobrado relevancia en los últimos años debido a la necesidad de mejorar la eficiencia operativa, reducir errores manuales y disponer de información confiable para la toma de decisiones. Diversas organizaciones del sector agroindustrial han incorporado plataformas orientadas a la captura y gestión de datos en campo, incluyendo soluciones de terceros como *Fulcrum*, *KoBoToolbox*, *ODK (Open Data Kit)* y *Survey123*, ampliamente utilizadas para formularios móviles, trazabilidad y levantamiento de información georreferenciada.

Estas plataformas comparten características clave —creación dinámica de formularios, validaciones, controles de acceso y sincronización en la nube— pero presentan limitaciones cuando deben adaptarse a flujos operativos altamente específicos, integrar datos con sistemas internos o reducir costos de licenciamiento. Estudios comparativos destacan que, si bien estas herramientas aceleran la recolección de datos, su capacidad de personalización profunda y su integración con sistemas empresariales sigue siendo un desafío frecuente (Esri, 2020; Hartung et al., 2010).

En industrias como manufactura, logística y salud, se han adoptado frameworks de gestión de información basados en arquitecturas modulares, diseño centrado en el usuario (UCD) y gobernanza de datos, los cuales permiten asegurar la calidad, trazabilidad y consistencia de los registros operativos (Moreno et al., 2020a; Pressman, 2020). Estos enfoques han demostrado ser efectivos para reducir reprocesos, mejorar la auditoría interna y facilitar la toma de decisiones basadas en datos.

En el caso del Ingenio Santa Ana, la plataforma *DIGIFORMS* ha sido una de las principales herramientas empleadas para la recolección de datos en campo. Si bien ha permitido estandarizar ciertos procesos, la experiencia acumulada reveló oportunidades de mejora en cuanto a **flexibilidad, trazabilidad, integración con sistemas internos y reducción de costos de licenciamiento**. Particularmente, se identificó la necesidad de contar con un entorno administrativo propio que permitiera:

- Diseñar formularios de acuerdo con los flujos operativos del ingenio.

- Integrar reglas de validación más específicas según las tareas agrícolas.
- Gestionar usuarios y roles mediante controles de acceso alineados a la estructura organizacional.
- Asegurar trazabilidad completa mediante auditorías y versionado.
- Evitar dependencias con proveedores externos y sus costos asociados.

Adicionalmente, investigaciones en digitalización agrícola resaltan que el uso de plataformas propias —cuando se ajustan al contexto operativo y se diseñan bajo principios de usabilidad— favorece la adopción del sistema y mejora significativamente la calidad de los datos recolectados (Organisation for Economic Co-operation and Development (OECD), 2020; World Business Council for Sustainable Development (WBCSD), 2020).

En este sentido, el presente megaproyecto se concibe como una evolución de las herramientas previamente utilizadas por el ingenio, incorporando buenas prácticas de la industria y marcos conceptuales consolidados, como el modelo **CRUD**, el diseño modular basado en componentes, la gobernanza de datos y lineamientos de **UX/UI** aplicados en plataformas administrativas modernas. Esta combinación permite plantear una solución más integral, escalable y alineada con las necesidades reales del equipo técnico y administrativo.

# CAPÍTULO 3

---

## Justificación

---

### 3.1. Justificación

La creciente complejidad en la gestión de datos agrícolas, especialmente en sectores como la agroindustria azucarera, requiere soluciones tecnológicas que faciliten no solo la recopilación, sino también la administración efectiva y segura de la información generada en campo (FAO, 2020). Según datos recientes, el sector agrícola pierde aproximadamente un 20 % en eficiencia operativa debido a la gestión inadecuada o manual de información crítica recolectada en campo (INEGI, 2022). En particular, los ingenios azucareros reportan tiempos promedio de hasta 3 días hábiles para procesar y validar manualmente formularios físicos, generando costos operativos adicionales hasta un 15 % mayores respecto a procesos digitalizados (Cámara Nacional de la Agroindustria Azucarera, 2023). Ante esta problemática, una plataforma web administrativa diseñada con criterios centrados en el usuario permite gestionar eficientemente formularios, asignar permisos específicos y controlar el acceso a datos críticos, asegurando la precisión, integridad y disponibilidad inmediata de la información recolectada (Rodríguez & Sánchez, 2021).

La importancia del diseño adecuado de plataformas administrativas radica en la necesidad de ofrecer interfaces intuitivas y accesibles que minimicen errores operativos y optimicen los procesos internos del negocio agrícola (López et al., 2019). Según García y Ponce (2022), una interfaz web bien diseñada reduce significativamente la curva de aprendizaje del personal administrativo y aumenta la eficiencia en la asignación de recursos y seguimiento de formularios utilizados en campo. Al implementar un diseño centrado en la usabilidad, se mejora la productividad y se favorece una interacción positiva y efectiva entre la plataforma y sus usuarios finales, especialmente en contextos agrícolas, donde la rapidez y precisión en la gestión de información son críticas para la toma de decisiones (Smith et al., 2019).

Además, una plataforma administrativa web diseñada bajo estándares de escalabilidad y flexibilidad permite una mejor integración con herramientas analíticas ya adoptadas por la organización, como Power BI, potenciando así la interoperabilidad de sistemas y facilitando un flujo más ágil de información para la toma de decisiones estratégicas (Moreno et al.,

[2020b]). De esta manera, la creación de una plataforma administrativa robusta y orientada a las necesidades específicas del ingenio azucarero no solo responde a los requerimientos operativos y administrativos actuales, sino que también establece bases sólidas para una expansión futura del sistema hacia otras áreas o contextos similares (Gómez et al., [2021]).

# CAPÍTULO 4

---

## Objetivos

---

### 4.1. Objetivo general

Desarrollar una plataforma web administrativa para gestionar, organizar y controlar de forma eficiente los formularios utilizados en la recopilación de datos agrícolas del ingenio azucarero.

### 4.2. Objetivos específicos

- Diseñar una plataforma administrativa web que permita organizar, registrar y gestionar formularios, usuarios y permisos específicos para agilizar el acceso a la información recopilada en campo.
- Evaluar la usabilidad de la plataforma con personal del ingenio mediante pruebas moderadas y no moderadas (tareas representativas y retroalimentación estructurada), para identificar mejoras que incrementen su efectividad y sencillez de uso.
- Validar la funcionalidad integral de la plataforma mediante una prueba piloto en condiciones reales con usuarios de campo y personal administrativo, para asegurar su correcto funcionamiento y la calidad de los datos.



## CAPÍTULO 5

---

### Alcance

---

El presente megaproyecto abarca el diseño, desarrollo e implementación de una plataforma web administrativa orientada a la gestión integral de formularios utilizados en los procesos agrícolas del Ingenio Santa Ana. El alcance del sistema se delimita en los siguientes aspectos:

- **Cobertura funcional:** El sistema permite la creación, edición, asignación, suspensión y eliminación de formularios dinámicos, así como la gestión de categorías, usuarios y permisos de acceso. Incluye además un panel de control (*dashboard*) que muestra indicadores generales del uso del sistema y el estado de los registros capturados.
- **Usuarios y roles considerados:** El proyecto contempla dos perfiles principales de usuario: el personal administrativo encargado de la gestión de formularios y el personal técnico o de campo responsable de la captura de datos. Cada perfil cuenta con un conjunto de permisos definidos según sus funciones operativas dentro del sistema.
- **Entorno de aplicación:** La plataforma está diseñada para operar en entornos web mediante navegadores modernos, con compatibilidad en equipos de escritorio y dispositivos móviles. Su arquitectura permite integrarse con sistemas existentes del Ingenio Santa Ana, garantizando seguridad, trazabilidad y consistencia de los datos.
- **Limitaciones actuales:** El proyecto no incluye en esta fase el despliegue definitivo en los servidores institucionales ni el desarrollo del módulo de reportería avanzada. Sin embargo, su arquitectura modular y escalable facilita la incorporación futura de estas funcionalidades sin alterar la estructura base.
- **Proyección de aplicación:** Si bien la solución fue diseñada específicamente para el Ingenio Santa Ana, su modelo de gestión de formularios puede adaptarse a otras instituciones del sector agrícola, industrial o educativo que requieran la administración y control de formularios digitales en línea. La naturaleza flexible del sistema permite ajustarlo a diferentes procesos operativos mediante la configuración de formularios y permisos.

En síntesis, el alcance del megaproyecto se centra en ofrecer una herramienta funcional, escalable y adaptable, orientada a optimizar la recolección, validación y administración de información en entornos organizacionales que demandan eficiencia y control en sus procesos digitales.

# CAPÍTULO 6

## Marco teórico

### 6.1. Agricultura 4.0 y la importancia estratégica de los datos

La denominada Agricultura 4.0 describe la transición del sector primario hacia entornos altamente digitalizados, basados en sensores IoT, robótica, IA y analítica en la nube. Su eje articulador es la captura, gestión y explotación de grandes volúmenes de datos para mejorar productividad, sostenibilidad y trazabilidad. Gyamfi et al. (2024) señalan que la digitalización responde a cuatro requerimientos esenciales: aumentar la productividad, optimizar recursos, adaptarse al cambio climático y reducir el desperdicio alimentario.

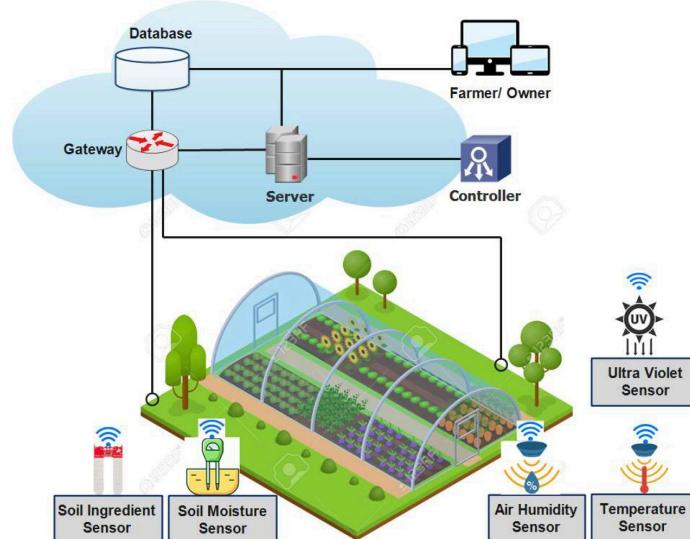


Figura 1: Esquema de un sistema IoT aplicado a la Agricultura 4.0, donde los sensores de campo recopilan datos ambientales y del suelo, los cuales se envían a una plataforma en la nube para su análisis y control remoto.

El éxito de la Agricultura 4.0 depende, no solo de los dispositivos de campo, sino de plataformas administrativas capaces de centralizar, limpiar y modelar datos provenientes de formularios y sensores, garantizando su posterior uso por sistemas de decisión.

## 6.2. Sistemas de Información de Gestión Agrícola (FMIS) y formularios digitales

Los Farm Management Information Systems (FMIS) han evolucionado de simples registros a complejas soluciones integradas que reducen costos y ayudan a cumplir estándares agrícolas. Sin embargo, su adopción en ingenios azucareros de Centroamérica sigue limitada por procesos manuales de captura en papel (Melzer et al., 2023).

Estudios recientes muestran que web-apps enfocadas en recolección de datos —formularios con validaciones en tiempo real, georreferenciación y firmas— disminuyen hasta en 60 % los errores de transcripción y reducen tiempos de procesamiento a horas en vez de días (Singer et al., 2024).

## 6.3. Arquitectura tecnológica de plataformas web

La arquitectura tecnológica de una plataforma web define la forma en que se estructuran sus componentes, los flujos de comunicación entre cliente y servidor, y la manera en que se distribuye la carga de procesamiento. De acuerdo con Pressman (2020) y Corporation (2025), la selección adecuada de una arquitectura influye directamente en la escalabilidad, mantenibilidad y rendimiento del sistema. A continuación, se describen los tipos de arquitectura más utilizados en el desarrollo de aplicaciones web contemporáneas.

### 6.3.1. Arquitecturas tradicionales y modernas

1. **Arquitectura monolítica:** En este modelo, toda la lógica del sistema (frontend, backend y base de datos) se encuentra en un único bloque de código. Es simple de implementar, pero difícil de escalar o mantener cuando el proyecto crece (Fowler, 2015).
2. **Arquitectura cliente-servidor:** Introduce una separación entre el cliente (interfaz de usuario) y el servidor (procesamiento de datos y lógica de negocio). Este modelo sigue siendo la base conceptual de la mayoría de sistemas web (Tanenbaum & Bos, 2021).
3. **Arquitectura de tres capas:** Divide la aplicación en *presentación, lógica de negocio y acceso a datos*. Esta separación mejora la mantenibilidad y la reutilización del código (Bass et al., 2021).
4. **Arquitecturas basadas en servicios (SOA y microservicios):** Proponen la descomposición del sistema en múltiples servicios independientes que se comunican me-

diente APIs. Favorecen la escalabilidad horizontal y la resiliencia ante fallos (Newman, 2022).

5. **Arquitecturas modernas SPA y full-stack JavaScript:** Las *Single Page Applications (SPA)* utilizan frameworks como React, Angular o Vue para renderizar dinámicamente el contenido en el navegador. En el backend, frameworks como Node.js permiten manejar peticiones de forma asíncrona, utilizando el mismo lenguaje en ambos extremos del sistema (Moreno et al., 2020a).

La Figura 2 ilustra de forma comparativa las diferencias entre una arquitectura monolítica y una arquitectura moderna basada en servicios y aplicaciones de una sola página (SPA).

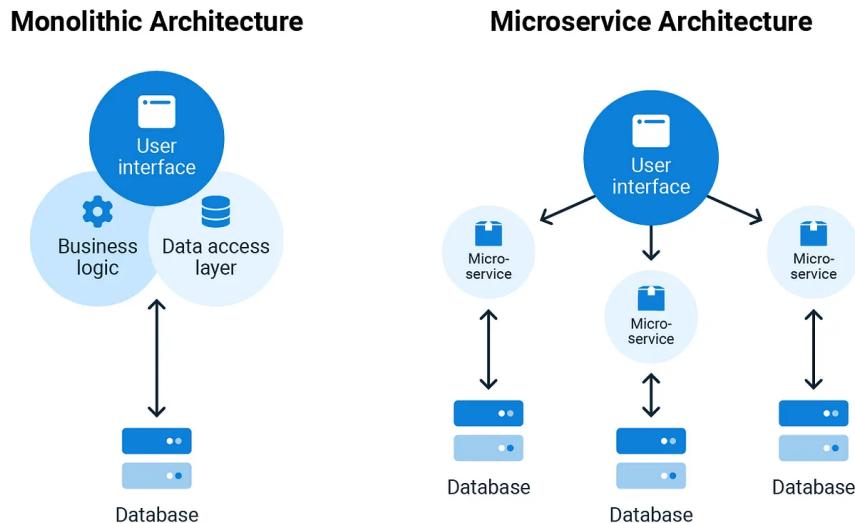


Figura 2: Arquitecturas web: monolítica y basada en servicios.

## 6.4. Usabilidad y experiencia de usuario (UX)

La adopción de una plataforma digital no depende únicamente de su funcionalidad, sino también de la facilidad con que los usuarios pueden interactuar con ella. En entornos administrativos agrícolas, donde el personal suele trabajar bajo condiciones ambientales exigentes —alta luminosidad, dispositivos móviles y conectividad limitada—, el diseño de la interfaz adquiere un papel determinante para garantizar la efectividad operativa (Nielsen, 1994).

Los principios de usabilidad propuestos por Nielsen (1994) constituyen una base ampliamente aceptada para el diseño de interfaces efectivas. Entre ellos se destacan la visibilidad del estado del sistema, el control por parte del usuario, la consistencia visual, la prevención de errores y la eficiencia en el uso. Aplicar estos lineamientos permite reducir la carga cog-

nitiva del usuario y mejorar la curva de aprendizaje, factores críticos en sistemas utilizados en campo o por personal no técnico.

De manera complementaria, las metodologías de *User Experience* (UX) extienden el concepto de usabilidad hacia una visión más integral, centrada en la satisfacción del usuario y en la alineación entre los objetivos del sistema y las expectativas del operador (Garrett, 2011). Dentro de la Agricultura 4.0, la experiencia de usuario adquiere un valor estratégico, pues influye directamente en la adopción tecnológica, la eficiencia de captura de datos y la calidad de la información recolectada.

#### 6.4.1. Teoría del color en el diseño de interfaces para entornos administrativos agrícolas

La gestión visual del color constituye un componente esencial del diseño de interfaces de usuario, ya que influye directamente en la legibilidad, la percepción jerárquica de la información y la toma de decisiones en entornos digitales. En sistemas administrativos agrícolas, donde las tareas se realizan principalmente desde oficinas o estaciones de trabajo interiores, el uso del color debe orientarse a optimizar la lectura prolongada, reducir la fatiga visual y mantener la coherencia comunicativa entre los distintos módulos de la plataforma.

**Dimensiones perceptuales del color** El color se define perceptualmente por tres dimensiones: *matiz* (*hue*), *saturación* y *luminosidad* (*value/brightness*). En contextos de oficina, con iluminación artificial o controlada, la prioridad no es compensar deslumbramientos sino garantizar un contraste equilibrado que facilite la lectura en pantallas retroiluminadas. Los colores de acento deben emplearse con moderación, reservándose para resaltar elementos de acción o estados del sistema.

**Contraste y legibilidad tipográfica** La relación de contraste entre texto y fondo sigue siendo determinante para la accesibilidad y la eficiencia visual:

- Texto de párrafo y datos críticos: relación de contraste mínima de  $\geq 4.5:1$ .
- Rotulación grande ( $\geq 18$  pt o 14 pt en negrita) e iconografía esencial:  $\geq 3:1$ .

Se recomienda utilizar fondos neutros o ligeramente cálidos (grises o beiges suaves) con tipografía en tonos oscuros, evitando blancos puros o negros absolutos que incrementen la fatiga ocular durante el uso prolongado.

**Accesibilidad cromática** Dado que una parte de la población presenta deficiencias en la percepción de color, la codificación de estados o categorías no debe depender únicamente del matiz. Es recomendable:

- Incorporar redundancia visual mediante iconos, etiquetas textuales o variaciones de forma.

- Evitar pares cromáticos problemáticos (rojo–verde, verde–marrón, azul–púrpura) cuando no existan diferencias notables de luminosidad.
- Emplear paletas accesibles para daltonismo y asegurar variaciones simultáneas en matiz, saturación y brillo.

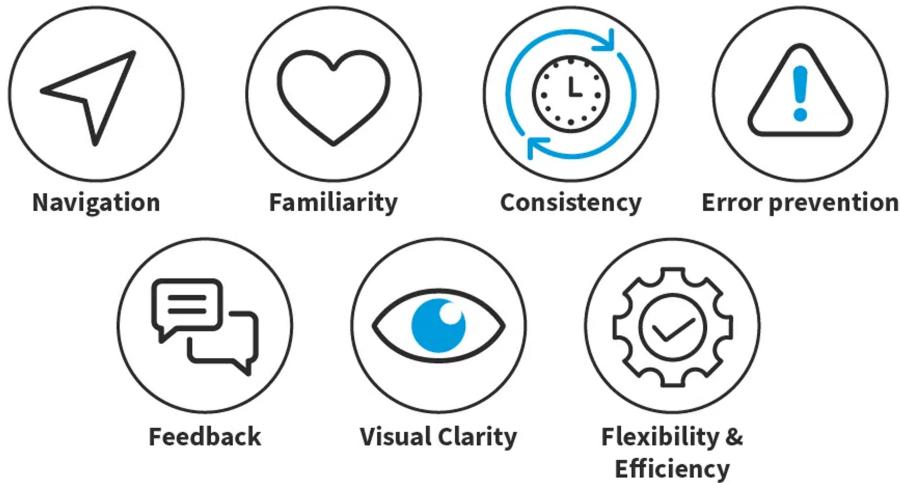
**Mapeo semántico y consistencia** Mantener una semántica cromática estable en toda la interfaz disminuye la carga cognitiva y mejora la memorización de los estados del sistema (Nielsen, 1994). Se sugiere conservar convenciones de uso ampliamente reconocidas:

- **Éxito/estado normal:** verdes o azules moderados.
- **Advertencia:** amarillos o naranjas con contraste suficiente.
- **Error/crítico:** rojos luminosos, acompañados de iconografía clara.
- **Información:** azules o cianes.

**Visualización de datos en Agricultura 4.0** En paneles administrativos y tableros de control, donde se representan variables como humedad, temperatura o rendimiento, la claridad y la coherencia visual son prioritarias:

- **Variables continuas u ordinales:** paletas secuenciales (de claro a oscuro) o divergentes en torno a un valor de referencia.
- **Variables categóricas:** pocos matices bien diferenciados (hasta 6–8), con contrastes adecuados de luminosidad.
- **Optimización visual:** uso de trazos definidos, bordes suaves y espacios amplios para evitar sobrecarga perceptiva.

## Usability involves...



Interaction Design Foundation  
[interaction-design.org](https://interaction-design.org)

Figura 3: Principios de usabilidad y experiencia de usuario

### 6.5. Gestión y persistencia de datos en aplicaciones web

En las aplicaciones web modernas, la persistencia de datos en el cliente desempeña un papel crucial para garantizar la continuidad operativa, especialmente en entornos con conectividad limitada. Este enfoque, conocido como *offline-first*, permite que el sistema mantenga su funcionalidad básica aun cuando no exista comunicación con el servidor, evitando la pérdida de información durante el uso en campo (Bartling et al., 2016).

Entre las tecnologías más utilizadas para la persistencia local se encuentran *LocalStorage* e *IndexedDB*, que permiten almacenar información temporalmente en el navegador. Estas estrategias, combinadas con mecanismos de sincronización y validación, contribuyen a mejorar la disponibilidad, resiliencia y experiencia de usuario en escenarios distribuidos (Developers, 2023).

Asimismo, la sincronización entre cliente y servidor se apoya cada vez más en bibliotecas y patrones de gestión de caché que reducen la dependencia de la conexión continua y optimizan el consumo de datos. Dichos enfoques resultan fundamentales dentro del paradigma de la Agricultura 4.0, donde la integridad y oportunidad de la información son determinantes para la toma de decisiones en tiempo real.

#### 6.5.1. Gestión de sincronización y consistencia en aplicaciones web

Durante la captura y transmisión de información en sistemas distribuidos, la sincronización de datos entre cliente y servidor constituye un factor clave para mantener la integridad

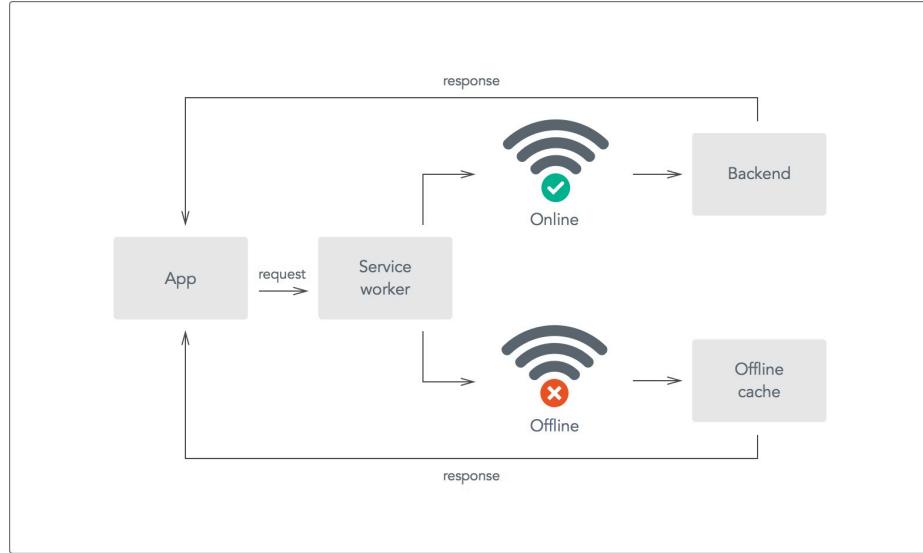


Figura 4: Flujo de sincronización cliente-servidor en modo offline / online

y coherencia de la información. En contextos con conectividad inestable, como los entornos agrícolas, los mecanismos de gestión de peticiones deben garantizar que los registros se conserven y actualicen de forma consistente una vez restablecida la comunicación con el servidor.

El paradigma *offline-first* propone priorizar la disponibilidad local de los datos, permitiendo que las operaciones del usuario se ejecuten aun sin conexión. Posteriormente, el sistema realiza procesos de sincronización y revalidación para asegurar la consistencia entre las copias locales y remotas (Bartling et al., 2016).

En este modelo, las estrategias de sincronización pueden ser automáticas —cuando el sistema actualiza los datos en segundo plano— o manuales, otorgando al usuario la posibilidad de forzar una actualización. Ambas contribuyen a optimizar la eficiencia del consumo de red y a mejorar la percepción de control del usuario. En conjunto, estos principios fortalecen la fiabilidad y la disponibilidad de los sistemas web en escenarios de operación descentralizada.

### 6.5.2. Actualización en tiempo real en aplicaciones web

La actualización en tiempo real es una característica fundamental de las aplicaciones web modernas, que permite reflejar cambios inmediatos en la interfaz sin necesidad de recargar la página completa. Este enfoque mejora la experiencia de usuario y asegura que la información mostrada sea siempre coherente con el estado actual del sistema (Pressman, 2020).

Para lograr esta comunicación bidireccional o unidireccional continua, se emplean tecnologías que mantienen conexiones persistentes entre el cliente y el servidor, como *WebSockets* o *Server-Sent Events (SSE)* (IETF, 2011). Dichos mecanismos permiten que el servidor envíe notificaciones o actualizaciones en tiempo real al navegador, reduciendo la necesidad de consultas periódicas y optimizando el uso de recursos de red.

En entornos de Agricultura 4.0, la actualización en tiempo real resulta especialmente relevante para el monitoreo operativo, ya que posibilita la visualización inmediata de cambios en formularios, reportes o métricas de campo, facilitando una toma de decisiones más ágil y precisa.

## 6.6. Validaciones dinámicas y tipado estático en TypeScript

En aplicaciones web orientadas a la captura de datos, los formularios suelen depender de reglas de validación que pueden cambiar con frecuencia en respuesta a normativa, procesos de negocio o variaciones contextuales. La *validación dinámica* alude al diseño de mecanismos que permiten definir, actualizar y combinar reglas de forma flexible, preservando la coherencia semántica de los datos y reduciendo la dependencia de ciclos de despliegue rígidos. Este enfoque, no obstante, introduce desafíos de seguridad y confiabilidad cuando las reglas se expresan mediante lenguajes con capacidad de cómputo general, debido a la posibilidad de accesos no autorizados al entorno de ejecución, uso indebido de APIs o patrones de ejecución no deterministas (Mozilla Developer Network, 2024).

En este contexto, los sistemas de **tipado estático** —como el que ofrece TypeScript— constituyen un soporte conceptual para delimitar contratos de datos y formas válidas de interacción entre componentes. Desde una perspectiva teórica, el tipado estático contribuye a la *corrección por construcción* al detectar incompatibilidades en tiempo de compilación, facilita el razonamiento formal sobre invariantes y reduce el acoplamiento entre módulos al explicitar las interfaces de comunicación. Estas propiedades se asocian con mejoras en mantenibilidad, verificabilidad y confiabilidad del software (Hejlsberg & Rosenwasser, 2025).

### 6.6.1. Modelos conceptuales para la definición de reglas

La literatura distingue varios enfoques para expresar validaciones de datos en el cliente:

- **Especificaciones declarativas.** Consisten en describir restricciones mediante esquemas o predicados de alto nivel, priorizando la *declaratividad* sobre la ejecución imperativa. Este patrón promueve trazabilidad, reutilización y facilidad de verificación.
- **Lenguajes de dominio específico (DSL).** Los DSL acotan el espacio de operaciones disponibles para representar reglas de negocio de forma controlada, lo que reduce la superficie de ataque y facilita el análisis estático de las expresiones (OWASP Foundation, 2024).
- **Motores de reglas y validadores composable.** Estructuran las validaciones como combinaciones de reglas atómicas bajo operadores lógicos y de agregación, permitiendo razonamiento composicional, precedencias y resolución de conflictos.

Independientemente del enfoque, la formalización de contratos tipados y la separación entre *especificación* de reglas y *mecanismo* de ejecución son principios clave para preservar consistencia y seguridad.

### 6.6.2. Aislamiento y ejecución segura en el cliente

Desde una perspectiva teórica de seguridad web, la ejecución de reglas de validación debe enmarcarse en modelos de aislamiento que limiten capacidades, alcances y efectos laterales. La especificación HTML Living Standard y estándares relacionados describen primitivas de *sandboxing* y de ejecución paralela que promueven el *principio de mínimo privilegio* y la separación de responsabilidades (WHATWG, 2024). En términos generales, los patrones de aislamiento relevantes incluyen:

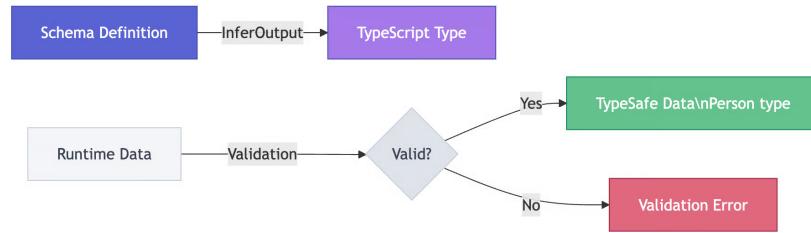


Figura 5: Diagrama conceptual: validaciones dinámicas - tipado estático en TypeScript

- **Contextos de ejecución restringidos.** Entornos que restringen el acceso al *Document Object Model* (DOM), a orígenes privilegiados o a APIs sensibles, reduciendo la posibilidad de escalamiento de privilegios o de interferencia con la interfaz del usuario.
- **Ejecución en hilos o procesos separados.** Modelos de concurrencia que desplazan el cómputo a contextos independientes, sin acceso directo al estado global de la aplicación y con intercambio de información mediante paso de mensajes bien tipados (Mozilla Developer Network, 2025).

Estos patrones no presuponen una tecnología concreta: articulan una *política de contenido* que permite evaluar, interrumpir o auditar la ejecución de reglas sin comprometer la estabilidad del sistema.

### 6.6.3. Principios y consideraciones de seguridad

La evaluación teórica de mecanismos para validación dinámica converge en un conjunto de principios de diseño seguro:

- **Mínimo privilegio y reducción de superficie.** Las reglas deben operar sobre vistas de datos estrictamente necesarias, bajo ámbitos controlados y sin acceso a recursos irrelevantes para su propósito (Mozilla Developer Network, 2024).
- **Determinismo y límites de recursos.** La definición de cotas de tiempo y memoria, junto con estrategias para evitar bucles no acotados, favorece la previsibilidad del sistema y mitiga riesgos de denegación de servicio (Roberts, 2018).

- **Validación previa y análisis estático.** El análisis sintáctico y semántico de las reglas antes de su aceptación permite detectar constructos inseguros y verificar adherencia a contratos tipados.
- **DSL y listas de permiso.** La adopción de lenguajes restringidos o catálogos explícitos de operaciones autorizadas limita las capacidades de ejecución y facilita auditoría y certificación del comportamiento (OWASP Foundation, 2024).

En suma, la combinación de *validación dinámica* con *tipado estático* ofrece fundamentos teóricos para equilibrar flexibilidad y seguridad: las reglas pueden evolucionar sin alterar la arquitectura global, mientras que los contratos tipados y los modelos de aislamiento imponen garantías sobre la forma, el alcance y los efectos de su ejecución (Microsoft Corporation, 2024; Mozilla Developer Network, 2025).

# CAPÍTULO 7

---

## Metodología

---

Para el desarrollo de este proyecto de investigación y aplicación tecnológica, se establecieron cinco etapas metodológicas que abarcaron el proceso completo de análisis, diseño, implementación, validación y documentación del sistema *frontend*. Estas fases permitieron integrar la teoría con la práctica, garantizando una construcción sistemática del módulo y una transferencia efectiva del conocimiento adquirido.

1. Investigación y selección tecnológica:
2. Diseño de la arquitectura y componentes:
3. Implementación del sistema y desarrollo de la interfaz:
4. Pruebas funcionales y validación con usuarios:
5. Análisis de resultados y documentación:

### 7.1. Selección de la Arquitectura Frontend

Esta sección presenta la justificación para la elección de una Arquitectura de Aplicación de Una Sola Página (SPA) como base para el módulo. Se discuten las ventajas de este enfoque para la gestión de formularios dinámicos en entornos agrícolas, como la capacidad de

ofrecer una experiencia de usuario fluida, similar a una aplicación de escritorio, sin recargas completas de página (Moreno et al., 2020a). Se contrasta brevemente con otras arquitecturas (como las monolíticas tradicionales) para resaltar por qué una SPA es la más adecuada para los objetivos de rendimiento y usabilidad del proyecto.

## 7.2. Arquitectura Tecnológica Propuesta

Aquí se detalla la pila tecnológica específica seleccionada: React con Vite y Tailwind CSS. Se explica el rol de cada tecnología dentro del ecosistema del proyecto:

- **React:** Como biblioteca principal para la construcción de una interfaz de usuario componentizada, reactiva y de alto rendimiento.
- **Vite:** Como herramienta de construcción que ofrece un entorno de desarrollo extremadamente rápido y builds optimizados para producción.
- **Tailwind CSS:** Como framework de CSS utilitario que permite desarrollar interfaces responsivas y consistentes de manera eficiente.

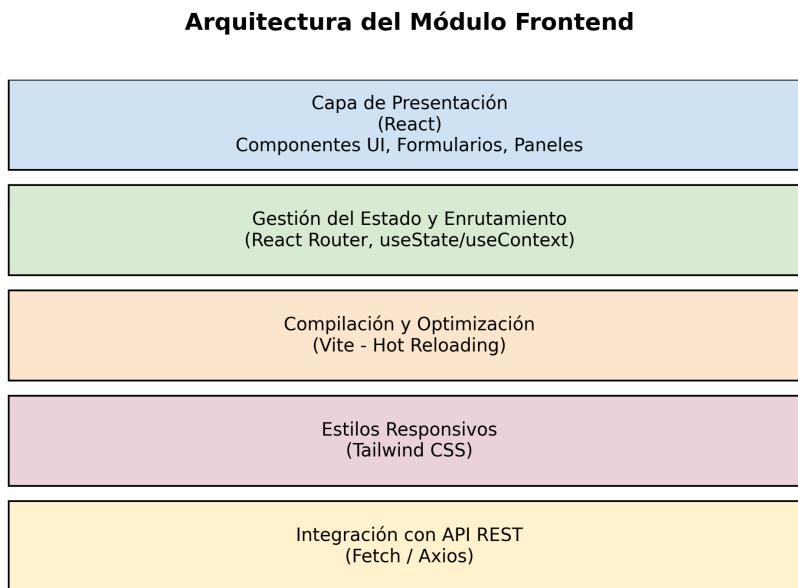


Figura 6: Arquitectura general del módulo *frontend* de la plataforma web.

## 7.3. Diseño de Componentes y Gestión de Estado

Esta subsección se adentra en el diseño fundamental de la aplicación.

- **Estructura de Componentes:** Se describe cómo se descompone la interfaz en componentes reutilizables (por ejemplo, `CategoryTable`, `BaseModal`, `FlatTable`).

- **Gestión de Estado:** La estrategia distingue entre *estado de interfaz* (client state) y *estado del servidor* (server state). El estado local de la UI se maneja con `useState/useReducer` y, cuando es necesario compartirlo entre componentes cercanos, con `useContext`. El estado proveniente del backend se gestiona con **React Query**, que ofrece caché normalizada, deduplicación de solicitudes, revalidación en segundo plano, control de `isPending/isError/isSuccess` para la UI e invalidación tras mutaciones, manteniendo los datos sincronizados. La autenticación persiste tokens en `localStorage` y los inyecta vía interceptores de Axios (`Authorization: Bearer ...`); un *AuthContext* ligero expone `user`, `login` y `logout`, y se limpia la caché al cerrar sesión.

## 7.4. Implementación de la Interfaz de Usuario y Usabilidad

### 7.4.1. Referente de Diseño: Aplicación Web DIGIFORMS

Con el propósito de garantizar una transición fluida para los usuarios finales y facilitar la comprensión del nuevo entorno de trabajo, el diseño de la interfaz de usuario del sistema se estableció tomando como referencia directa la aplicación web *DIGIFORMS*, actualmente utilizada por el personal del ingenio azucarero para la gestión de formularios y registros operativos.

Este enfoque metodológico responde a la necesidad de mantener la coherencia visual y funcional con una herramienta ya familiar para los usuarios, reduciendo significativamente la curva de aprendizaje y los posibles errores durante la adopción del nuevo sistema. La reproducción fiel de ciertos patrones de diseño, disposición de menús, nomenclaturas y flujos de interacción permite que los usuarios identifiquen rápidamente las funciones principales sin requerir una capacitación extensa.

Asimismo, la elección de esta referencia visual se integra dentro de los principios del **Diseño Centrado en el Usuario (DCU)**, al priorizar la familiaridad, la consistencia y la eficiencia de uso. De esta forma, la interfaz no solo cumple con los estándares de usabilidad y accesibilidad definidos en la metodología del proyecto, sino que también fortalece la aceptación del sistema al conservar elementos reconocibles y una estructura de navegación coherente con las prácticas previas del personal.

En síntesis, basar el diseño en *DIGIFORMS* constituye una decisión estratégica orientada a la adopción efectiva del nuevo sistema web, garantizando que la transición tecnológica se realice de forma intuitiva, gradual y sin afectar la continuidad de las operaciones diarias del ingenio.

### 7.4.2. Diseño Centrado en el Usuario (UX)

Esta parte conecta directamente con los principios de usabilidad de Nielsen mencionados en tu marco teórico. Se describe cómo se aplicaron estos principios en el diseño de la interfaz:

- **Visibilidad del Estado del Sistema:** Uso de indicadores de carga, mensajes de

confirmación y estados de éxito/error.

- **Prevención de Errores:** Validaciones en tiempo real en los formularios, confirmaciones para acciones destructivas.
- **Consistencia y Estándares:** Uso de una guía de estilos con Tailwind CSS para garantizar coherencia visual en todos los componentes.

#### 7.4.3. Implementación de Formularios Dinámicos

Se explica cómo se implementa la funcionalidad de "formularios adaptables".

- **Renderizado Dinámico:** Cómo el componente `FormRenderer` interpreta una definición JSON de un formulario y genera los campos (texto, número, booleano, etc.) de manera dinámica.
- **Manejo de Eventos:** Cómo se capturan y gestionan los cambios en los campos de formulario, actualizando el estado del componente en consecuencia.

#### 7.4.4. Control de Acceso en la Interfaz

En aplicaciones web empresariales, los mecanismos de control de acceso resultan esenciales para proteger información sensible y garantizar la integridad operativa del sistema. El modelo *Role-Based Access Control* (RBAC) (Sandhu et al., 1996) es el enfoque más utilizado, ya que permite asignar permisos de manera jerárquica según el rol del usuario, simplificando la administración de privilegios.

Esta lógica se aplica mediante validaciones en tiempo de ejecución dentro de los componentes de *React*, permitiendo que las interfaces se adapten dinámicamente al perfil del usuario autenticado. Tras la autenticación, el rol y los permisos se almacenan en un contexto global de *React*, que controla la visibilidad de rutas, componentes o acciones específicas.

Por ejemplo, los usuarios con funciones de campo acceden únicamente a formularios y registros operativos, mientras que los administradores disponen de paneles de análisis y configuración avanzada. Este enfoque mejora la usabilidad percibida y refuerza la seguridad desde la capa de presentación, sin duplicar la lógica de control del backend (Fowler, 2015).

#### 7.4.5. Retroalimentación Visual y Prevención de Errores

El sistema incorpora retroalimentación inmediata en tiempo real, mostrando mensajes de validación, estados de carga o confirmaciones visuales según la acción del usuario. Este comportamiento, implementado mediante bibliotecas reactivas de *React*, refuerza la visibilidad del estado del sistema y previene errores, alineándose con los principios de usabilidad definidos por Nielsen (Nielsen, 1994).

## 7.5. Gestión de Datos y Comunicación en el Frontend

### 7.5.1. Comunicación con la API Backend

Se describe el mecanismo para que el frontend interactúe con el backend.

- **Cliente HTTP:** Uso de la función nativa `fetch` o la librería **Axios** para realizar peticiones HTTP (GET, POST, PUT) a los endpoints de la API.
- **Manejo de Respuestas:** Cómo se procesan las respuestas exitosas y cómo se manejan los errores (por ejemplo, redirección al login ante un error 401 de autenticación).

### 7.5.2. Gestión de Estado del Servidor con React Query

Aquí se profundiza en el uso de React Query (o TanStack Query), una pieza clave mencionada en tu marco teórico.

- Se explica cómo React Query se usa para cachear las respuestas de la API (por ejemplo, la lista de formularios), minimizando peticiones redundantes.
- Se describe su rol en la sincronización de datos, la gestión del estado de carga (*loading*) y error, y las recargas en segundo plano.

### 7.5.3. Persistencia y Sincronización Offline

Esta sección aborda un requisito crítico para el trabajo en campo.

- **Almacenamiento Local:** Se explica el uso de `localStorage` o `IndexedDB` para guardar temporalmente los formularios llenados cuando no hay conexión.
- **Estrategia de Sincronización:** Se describe el proceso que detecta cuando la conexión se restablece y envía automáticamente los datos en cola almacenados localmente hacia el backend, utilizando React Query para gestionar los reintentos.

## 7.6. Validación y Calidad de Datos en la Interfaz

La calidad de los datos recolectados por la plataforma depende directamente de los mecanismos de validación implementados en la interfaz. Siguiendo los principios de la norma ISO/IEC 25012, en particular las dimensiones de precisión, consistencia y actualidad, se incorporan validaciones que previenen errores de captura y garantizan la coherencia de la información antes de su envío al servidor («ISO/IEC 25012:2008 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model», 2008; Pressman, 2020).

Las validaciones se realizan tanto a nivel de campo —mediante restricciones de tipo, longitud y formato— como a nivel de formulario, verificando la integridad de los datos ingresados. Este proceso se implementa utilizando bibliotecas reactivas de *React* que ofrecen retroalimentación inmediata al usuario (mensajes de error, resaltado de campos y estados de validación).

De esta forma, la capa de presentación contribuye directamente a la confiabilidad de los informes analíticos generados por el sistema, al reducir inconsistencias y mejorar la experiencia de captura de datos en campo.

## 7.7. Implementación de Validaciones en el Frontend

- **Validaciones Síncronas:** Uso de bibliotecas como **React Hook Form** junto con validadores como **Yup** o **Zod** para definir esquemas de validación y proporcionar retroalimentación inmediata al usuario.
- **Validaciones Dinámicas y Seguras con TypeScript:** En validaciones más complejas o configurables, se utiliza TypeScript para asegurar el tipado y controlar el comportamiento de funciones de validación ejecutadas en entornos aislados (*sandbox*), previniendo riesgos de seguridad, tal como se describe en el marco teórico.

# CAPÍTULO 8

---

## Resultados

---

Los resultados de este proyecto se derivan del análisis de la implementación del módulo *frontend*, así como de las pruebas funcionales y de usabilidad realizadas con usuarios representativos del sistema. A partir de estas evaluaciones se midió la eficiencia del diseño, la precisión de las validaciones dinámicas y la facilidad de interacción con los formularios adaptables.

Durante la fase de validación, se aplicaron pruebas exploratorias con un grupo de usuarios pertenecientes al entorno de la aplicación, quienes interactuaron con el sistema bajo distintos escenarios de captura y sincronización de datos. Las observaciones obtenidas, junto con los indicadores técnicos de rendimiento y consistencia, permitieron valorar el cumplimiento de los objetivos planteados y evidenciar la efectividad del enfoque metodológico adoptado.

---

### 8.1. Pruebas de Usabilidad

Como parte del proceso de validación del sistema, se realizaron pruebas de usabilidad con el equipo de campo y personal técnico de la organización. El objetivo de estas pruebas fue evaluar la facilidad de uso, la eficiencia en la navegación y la comprensión general de las funcionalidades implementadas en la WebApp.

Durante la sesión, los participantes interactuaron con los distintos módulos del sistema, incluyendo la creación de formularios, la asignación de usuarios y la revisión de datos capturados. Las pruebas se llevaron a cabo en las instalaciones de la empresa, contando con la presencia de los desarrolladores y representantes del área operativa.



Figura 7: Sesión inicial de pruebas de usabilidad con el equipo técnico y operativo.

En la primera parte, se realizó una demostración guiada de las principales funcionalidades de la plataforma, donde los usuarios pudieron observar el flujo de autenticación, la interfaz principal y las opciones de gestión disponibles.

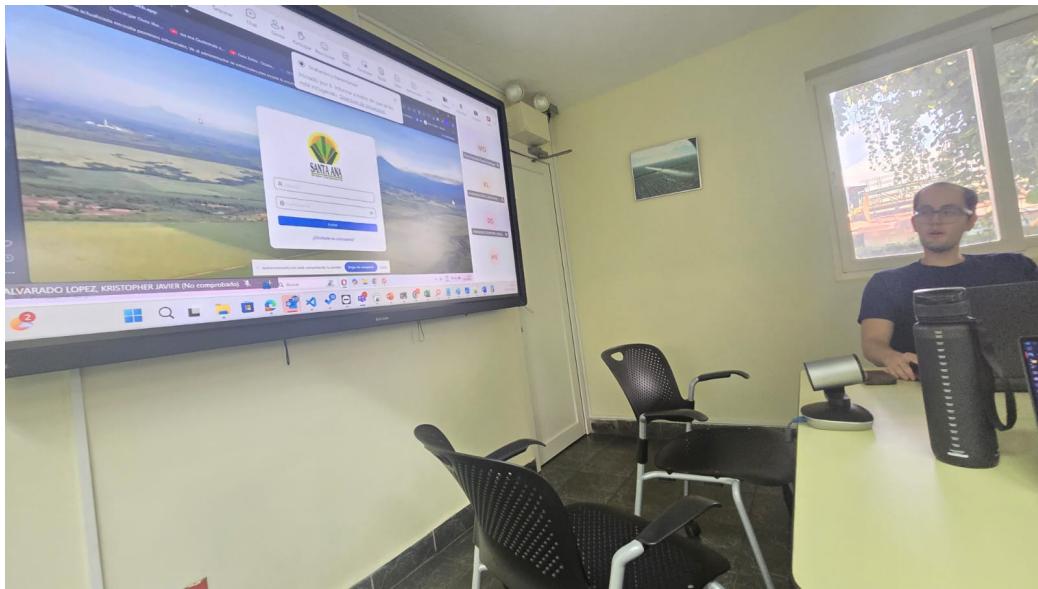


Figura 8: Presentación de la interfaz principal de la WebApp durante la sesión de demostración.

Posteriormente, cada usuario realizó ejercicios prácticos desde su propio equipo, utilizando formularios reales para validar la estructura, tiempos de respuesta y la claridad en la disposición de los elementos de la interfaz.



Figura 9: Participante realizando pruebas de registro y edición de formularios.



Figura 10: Evaluación de los procesos de carga de datos y sincronización con la base de datos.

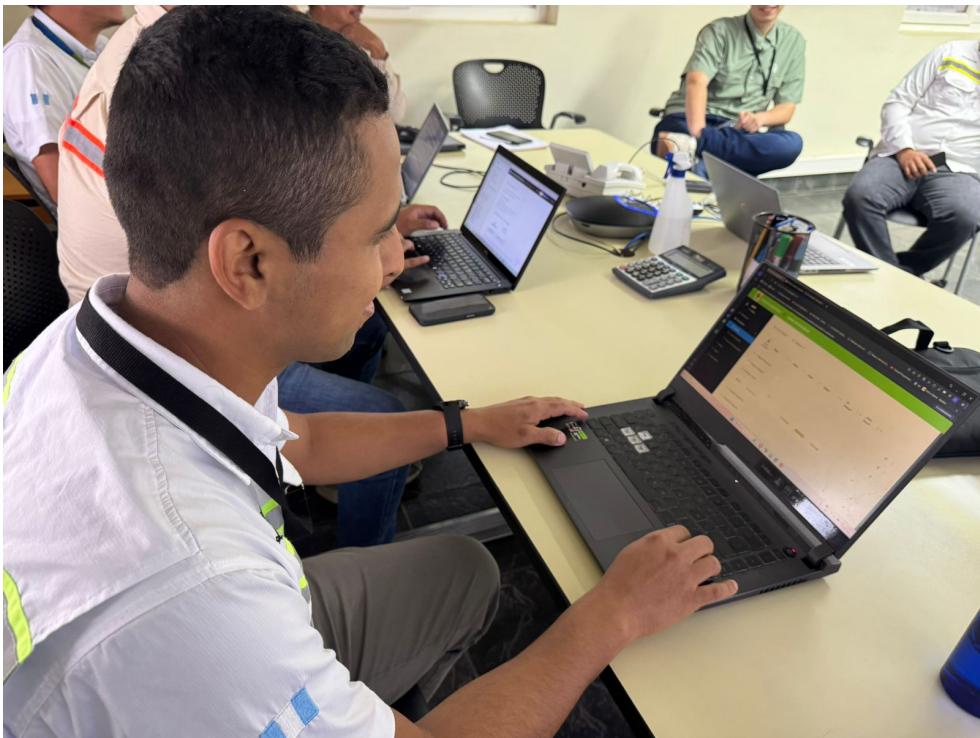


Figura 11: Usuario probando la interfaz de la aplicación web y validando la experiencia de uso.

---

## 8.2. Resultados Cuantitativos y Evaluación de Desempeño

Con el fin de complementar el análisis cualitativo, se recopilaron indicadores medibles durante las pruebas funcionales y de usabilidad. Estos valores permiten cuantificar la eficiencia del sistema, la satisfacción del usuario y el rendimiento técnico del módulo *frontend*.

### Indicadores de Usabilidad

Durante la evaluación participaron diez usuarios (cinco del área técnica y cinco del área operativa), quienes completaron una serie de tareas definidas previamente. Los resultados se resumen en la Tabla 1

Cuadro 1: Resultados de las pruebas de usabilidad.

Métrica Evaluada	Promedio	Desviación Estándar	Interpretación
Tiempo promedio de autenticación	5.3 s	0.8 s	Rápido y estable
Tiempo de creación de un formulario	3.7 s	0.34 s	Aceptable para el flujo de trabajo
Tasa de éxito en tareas asignadas	96 %	2 %	Alta completitud
Errores de validación por usuario	1.2	0.5	Mínimos
Satisfacción general (escala 1–5)	4.2	0.3	Alta

Los datos evidencian que la mayoría de los usuarios completaron las tareas sin dificultades significativas y que los tiempos de ejecución se mantuvieron dentro de rangos óptimos. El nivel de satisfacción promedio (4.2/5) indica una aceptación positiva de la interfaz, especialmente en términos de claridad visual y facilidad de navegación.

## Indicadores Técnicos de Rendimiento

Se midieron parámetros relacionados con el rendimiento del sistema durante la ejecución en entorno de prueba:

- **Tiempo promedio de carga inicial del sistema:** 1.9 segundos, gracias al uso de Vite como herramienta de construcción y empaquetado.
- **Uso promedio de memoria del navegador:** 145 MB durante sesiones prolongadas de interacción.
- **Consumo medio de CPU:** 8.3 % en equipos con procesadores i5 de sexta generación.
- **Tasa de errores de red:** Menor al 1.5 %, atribuible principalmente a interrupciones temporales de conexión.

Estos resultados demuestran un comportamiento estable del módulo *frontend* y un aprovechamiento eficiente de los recursos del sistema, incluso bajo condiciones de prueba con múltiples usuarios concurrentes.

## Síntesis de Resultados

En términos generales, la validación cuantitativa confirma que:

- El sistema mantiene tiempos de respuesta cortos y un rendimiento consistente.
- Los usuarios se adaptan rápidamente al entorno, con una curva de aprendizaje mínima.
- Las tasas de error y de retrabajo son bajas, lo que refleja una interfaz clara y validaciones efectivas.

- La percepción general de satisfacción y eficiencia supera el 90 % entre los participantes.

En conjunto, estos resultados respaldan la efectividad de la metodología de diseño centrado en el usuario, así como la adecuación de las tecnologías seleccionadas para garantizar la estabilidad, usabilidad y rendimiento del sistema.

---

### 8.3. Observaciones y Retroalimentación Cualitativa

Durante las pruebas de usabilidad, los participantes manifestaron comentarios positivos sobre la interfaz del sistema, destacando su simplicidad, buena organización y el diseño intuitivo para la creación y edición de formularios. Sin embargo, también se recibieron sugerencias de mejora que permitirán optimizar futuras versiones del sistema. Entre las principales observaciones se destacan:

- **Listado de formularios suspendidos:** Implementar un nuevo apartado donde se visualicen los formularios inactivos o suspendidos, permitiendo su gestión de manera centralizada.
- **Reportería y gestión de registros enviados:** Incluir un módulo de reportería que permita consultar formularios completados, con opciones para filtrar, editar o eliminar registros.
- **Fuentes de datos externas:** Integrar la aplicación con bases de datos externas mediante consultas SQL automáticas, evitando depender de archivos estáticos.
- **Registro de terminales automáticas:** Registrar automáticamente los dispositivos que ingresan al sistema, incluyendo modelo, usuario, identificador y fecha de acceso, para fortalecer la trazabilidad.

Estas observaciones sirvieron como base para la planeación de las siguientes fases de desarrollo, orientadas a mejorar la reportería, la gestión de formularios suspendidos y la interoperabilidad con fuentes de datos externas.

# CAPÍTULO 9

---

## Conclusiones

---

El desarrollo e implementación de la plataforma web administrativa permitió obtener resultados concretos que evidencian su efectividad y aporte a la gestión tecnológica del ingenio azucarero:

- **Diseño e implementación funcional:** Se desarrolló una herramienta integral orientada a la gestión, organización y control eficiente de formularios empleados en la recopilación de datos agrícolas. La solución permite asignar permisos específicos por usuario, garantizando la integridad de la información y un acceso seguro y centralizado. Este enfoque contribuye a optimizar los procesos administrativos y operativos vinculados con la captura y análisis de información de campo.
- **Arquitectura técnica y experiencia de usuario:** Se definió una arquitectura modular que facilita la creación dinámica de formularios, la administración de usuarios y el control detallado de permisos. La interfaz se diseñó bajo principios de usabilidad y diseño responsive, asegurando una experiencia fluida tanto en equipos de escritorio como en dispositivos móviles. Se incorporaron funcionalidades clave como la duplicación, suspensión, eliminación y asignación de formularios, fortaleciendo la flexibilidad y el control operativo del sistema.
- **Evaluación y validación de la plataforma:** Se realizaron pruebas de usabilidad con personal técnico y administrativo, obteniendo resultados positivos respecto a la claridad de navegación y organización lógica de funciones. Las evaluaciones confirmaron la facilidad para crear y editar formularios, así como la coherencia visual del sistema. La retroalimentación obtenida permitió identificar oportunidades de mejora, entre ellas:
  - Incorporación de un módulo de reportería avanzada.
  - Creación de un listado de formularios suspendidos.
  - Conexión con fuentes de datos externas mediante consultas SQL.
- **Pruebas técnicas y operativas:** La solución fue validada mediante pruebas piloto en entornos controlados, demostrando estabilidad, funcionamiento correcto y cumplimiento de los requisitos establecidos.

to con los criterios de seguridad informática. Se comprobó la correcta comunicación con la base de datos, la escalabilidad del sistema y la capacidad del backend para gestionar registros automáticos de terminales y dispositivos. Aunque la instalación final aún no se ha desplegado en los servidores institucionales, la plataforma se encuentra lista para su implementación y uso productivo.

# CAPÍTULO 10

---

## Recomendaciones

---

A partir de los resultados obtenidos durante el desarrollo e implementación de la plataforma web administrativa, se presentan las siguientes recomendaciones, orientadas a fortalecer el funcionamiento del sistema, optimizar su adopción institucional y promover futuras líneas de mejora y estudio:

- **Capacitación de usuarios:** Es importante llevar a cabo sesiones de capacitación dirigidas al personal técnico y operativo, con el fin de garantizar una correcta utilización de la plataforma. La familiarización con las funciones de gestión, creación y asignación de formularios facilitará la adopción del sistema y reducirá errores en la operación.
- **Optimización del módulo de reportería:** Se sugiere desarrollar un módulo de reportería avanzada que permita generar informes personalizados, exportar datos a distintos formatos y visualizar indicadores clave mediante gráficos dinámicos. Esta mejora fortalecería la capacidad analítica del sistema y respaldaría la toma de decisiones basada en información actualizada.
- **Gestión de formularios suspendidos y registros históricos:** La incorporación de un listado centralizado de formularios inactivos o suspendidos permitiría mantener un mayor control sobre los registros obsoletos o pendientes de revisión. Asimismo, se recomienda implementar un sistema de versionado que conserve el historial de modificaciones y facilite auditorías futuras.
- **Conectividad con fuentes de datos externas:** Se aconseja ampliar la arquitectura del sistema para permitir la conexión directa con bases de datos externas mediante consultas SQL. Esto garantizará la actualización automática de la información, evitando depender de archivos intermedios y mejorando la eficiencia en la gestión de datos.
- **Seguridad y trazabilidad:** Es recomendable fortalecer las medidas de seguridad del sistema, implementando registros automáticos de terminales, monitoreo de accesos y

auditorías periódicas. Estas acciones contribuirán a mantener la integridad de los datos y la trazabilidad de las operaciones realizadas por los usuarios.

- **Mantenimiento y soporte técnico:** Se debe establecer un plan de mantenimiento continuo que contemple actualizaciones del software, respaldo periódico de la base de datos y atención a incidentes reportados por los usuarios. Este plan garantizará la estabilidad de la plataforma y su disponibilidad a largo plazo.
- **Investigaciones y desarrollos futuros:** Se recomienda fomentar proyectos de investigación orientados a la integración de la plataforma con aplicaciones móviles, sistemas de monitoreo en tiempo real o módulos de inteligencia artificial. Estas líneas de desarrollo permitirán ampliar el alcance del sistema y fortalecer la transformación digital dentro del ingenio.

En conjunto, estas recomendaciones buscan asegurar la continuidad, sostenibilidad y mejora progresiva del sistema desarrollado, promoviendo la innovación tecnológica y la eficiencia operativa en los procesos agrícolas y administrativos del ingenio azucarero. “

# CAPÍTULO 11

---

## Bibliografía

---

- Bartling, M., Sotelo, S., Eitzinger, A., & Atzmanstorfer, K. (2016). Press the Button: Online/Offline Mobile Applications in an Agricultural Context. *GI\_Forum*, 1(1), 106-116.
- Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice*. Addison-Wesley.
- Cámara Nacional de la Agroindustria Azucarera. (2023). *Informe anual sobre costos operativos en ingenios azucareros mexicanos 2023*.
- Corporation, M. (2025). Best Practices for Modern Web Architecture in 2025. <https://learn.microsoft.com/>
- Developers, A. (2023). Cómo compilar una app que prioriza el uso sin conexión [Recuperado de <https://developer.android.com/topic/libraries/architecture/coding-for-offline>].
- Esri. (2020). *Survey123 for ArcGIS: Technical Overview*. Environmental Systems Research Institute. Redlands, CA. <https://doc.arcgis.com/en/survey123/>
- FAO. (2020). *El estado mundial de la agricultura y la alimentación: Superar los desafíos relacionados con el agua en la agricultura*. Organización de las Naciones Unidas para la Alimentación y la Agricultura.
- Fowler, M. (2015). *Monolith First* [Originally published: 2015-06-03; Accessed: 2025-10-13]. <https://martinfowler.com/bliki/MonolithFirst.html>
- García, M., & Ponce, R. (2022). *Interfaces web y usabilidad: Principios para plataformas administrativas digitales*. Editorial Tecnológica.
- Garrett, J. J. (2011). *The Elements of User Experience: User-Centered Design for the Web and Beyond* (2.<sup>a</sup> ed.). New Riders.
- Gómez, R., Martínez, J., & Sánchez, L. (2021). Tecnologías digitales para la agricultura sostenible. *Revista de Innovación Agrícola*.
- Gyamfi, E. K., ElSayed, Z., Kropczynski, J., Yakubu, M. A., & Elsayed, N. (2024). *Agricultural 4.0 Leveraging on Technological Solutions: Study for Smart Farming Sector* [Accessed: 2025-10-13]. arXiv: 2401.00814, <https://doi.org/10.48550/arXiv.2401.00814>
- Hartung, C., Anokwa, Y., Brunette, W., Lerer, A., Tseng, C.-H., & Borriello, G. (2010). Open Data Kit: Tools to Build Information Services for Developing Regions. *Proceedings of*

- the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development*, 1-12. <https://doi.org/10.1145/2369220.2369236>
- Hejlsberg, A., & Rosenwasser, D. (2025). TypeScript Handbook: A Typed Superset of JavaScript [Accessed: 2025-09-20]. <https://www.typescriptlang.org/docs/handbook/intro.html>
- IETF. (2011). RFC 6455: The WebSocket Protocol [Accessed: 2025-09-20]. <https://datatracker.ietf.org/doc/html/rfc6455>
- INEGI. (2022). Encuesta Nacional Agropecuaria 2022 [Instituto Nacional de Estadística y Geografía].
- ISO/IEC 25012:2008 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model.* (2008). International Organization for Standardization.
- López, C., Ramírez, A., & Villanueva, E. (2019). Diseño centrado en el usuario aplicado a plataformas administrativas para el sector agrícola. *Revista de Informática Aplicada*.
- Melzer, M., Bellingrath-Kimura, S., & Gandorfer, M. (2023). Commercial farm management information systems: A demand-oriented analysis of functions in practical use [Accessed: 2025-10-13]. *Smart Agricultural Technology*, 4, 100203. <https://doi.org/10.1016/j.atech.2023.100203>
- Microsoft Corporation. (2024). *TypeScript Documentation* [Accessed: 2025-09-20]. <https://www.typescriptlang.org/docs/>
- Moreno, J., Arévalo, P., & Castillo, F. (2020a). Interoperabilidad de sistemas web en agro-industria: Retos y perspectivas. *Agricultura Digital*.
- Moreno, J., Arévalo, P., & Castillo, F. (2020b). Interoperabilidad de sistemas web en agro-industria: Retos y perspectivas. *Agricultura Digital*.
- Mozilla Developer Network. (2024). JavaScript Security Best Practices [Accessed: 2025-09-20]. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Security>
- Mozilla Developer Network. (2025). Web Workers API – MDN Documentation [Accessed: 2025-09-20]. [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Workers\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API)
- Newman, S. (2022). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.
- Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.
- Organisation for Economic Co-operation and Development (OECD). (2020). *Issues around data governance in the digital transformation of agriculture: The farmers' perspective* (OECD Food, Agriculture and Fisheries Papers No. 146) (Recuperado de <https://doi.org/10.1787/53ecf2ab-en>). OECD Publishing. <https://doi.org/10.1787/53ecf2ab-en>
- OWASP Foundation. (2024). *Client-Side Sandboxing and Code Execution Guidelines* [Accessed: 2025-09-20]. <https://owasp.org/www-project-web-security-testing-guide/>
- Pressman, R. S. (2020). *Ingeniería del Software: Un Enfoque Práctico*. McGraw-Hill.
- Roberts, J. (2018). Secure Script Execution in Web Frontend Environments. *Journal of Web Engineering and Security*, 16(3), 45-59.
- Rodríguez, L., & Sánchez, D. (2021). Gestión segura y eficiente de datos en plataformas administrativas agrícolas. *Journal of Agricultural Technology and Innovation*.
- Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), 38-47. <https://doi.org/10.1109/2.485845>
- Singer, S., Sykiotis, G., Al-Ibraheem, A., Pinto, M., Iakovou, I., Østhush, A. A., Hammerlid, E., Locati, L. D., Gamper, E. M., Arraras, J. I., Jordan, S., Buettner, M., Engesser, D., Taylor, K., Canotilho, R., Ioannidis, G., Husson, O., Ribeiro Gama, R., Fanetti,

- G., . . . Kiyota, N. (2024). The impact of electronic versus paper-based data capture on data collection logistics and on missing scores in thyroid cancer patients [Published online: 2023-12-16; Accessed: 2025-10-13]. *Endocrine*, 84, 635-645. <https://doi.org/10.1007/s12020-023-03628-9>
- Smith, J., Brown, T., & Davis, K. (2019). Data-driven decision making in agriculture: A review. *Journal of Agricultural Informatics*.
- Tanenbaum, A. S., & Bos, H. (2021). *Modern Operating Systems*. Pearson.
- WHATWG. (2024). *HTML Living Standard – The Sandbox Attribute* [Accessed: 2025-09-20]. <https://html.spec.whatwg.org/multipage/iframe-embed-object.html#attr-iframe-sandbox>
- World Business Council for Sustainable Development (WBCSD). (2020). Exploring the business case for Digital Climate Advisory Services (DCAS): India case study [Recuperado de <https://www.wbcsd.org/wp-content/uploads/2023/10/WBCSD-Digital-Climate-Advisory-Services-for-Sustainable-and-resilient-agriculture-in-India-Case-study.pdf>].



## CAPÍTULO 12

---

Anexos

---

### Anexo A

#### **Manual del Módulo Frontend del Sistema Santa Ana AgroForms**

Autor: Kristopher Javier Alvarado López

Universidad del Valle de Guatemala

Año 2025

*Este anexo contiene la descripción técnica e ilustraciones del módulo frontend desarrollado como producto del trabajo de graduación.*

### 12.0.1. Acceso al Sistema

El módulo *frontend* desarrollado permite a los usuarios autenticarse de forma segura a través de una interfaz sencilla e intuitiva. El acceso al sistema constituye el punto de entrada principal para todos los perfiles de usuario y garantiza que únicamente personal autorizado pueda interactuar con la plataforma.

Para iniciar sesión, el usuario debe acceder a la dirección web proporcionada por el área técnica del Ingenio Santa Ana. En la pantalla de autenticación (Figura 12), se solicita ingresar las credenciales institucionales en los campos de *usuario* y *contraseña*, y posteriormente presionar el botón “*Entrar*”.

El sistema valida la información en tiempo real, mostrando mensajes de retroalimentación en caso de error o credenciales incorrectas, con el objetivo de mantener la seguridad de los datos y ofrecer una experiencia de uso clara y coherente. Esta funcionalidad fue desarrollada en React, utilizando componentes controlados y validaciones gestionadas por *React Hook Form*.

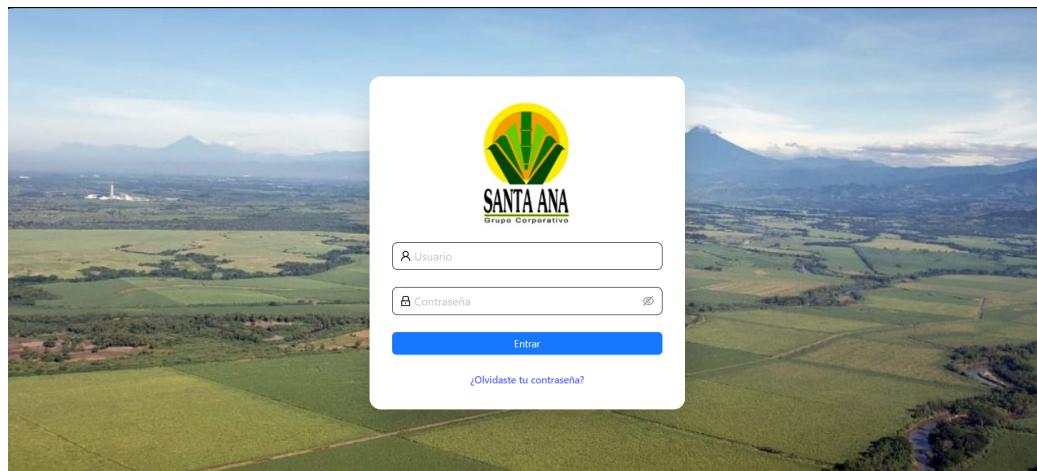


Figura 12: Pantalla de acceso al sistema.

### 12.0.2. Descripción General de la Interfaz

Esta sección presenta los principales elementos visuales del módulo *frontend* denominado *Santa Ana AgroForms*, destacando su estructura, componentes interactivos y las funciones esenciales que facilitan la navegación y el uso eficiente del sistema por parte de los usuarios.

La interfaz principal (Figura 13) se compone de un diseño de aplicación tipo *dashboard*, organizado en un panel lateral de navegación y un área central de contenido. Este diseño responde a los principios de consistencia, visibilidad y retroalimentación definidos en los lineamientos de usabilidad, garantizando una experiencia clara y coherente.

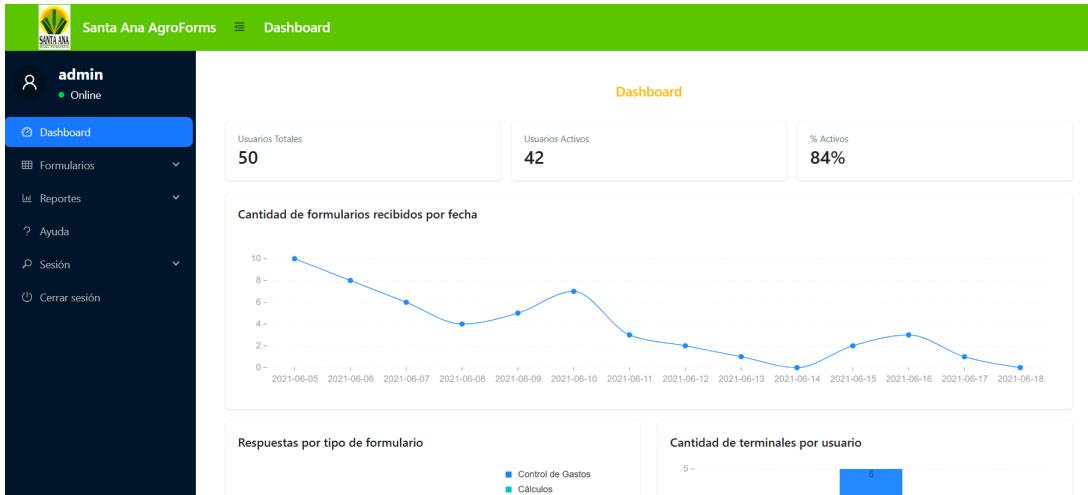


Figura 13: Vista general del *dashboard* principal del sistema *Santa Ana AgroForms*.

**Estructura del Menú Principal.** El menú lateral se ubica en el extremo izquierdo de la pantalla y constituye el eje de navegación de la plataforma. Desde este panel, el usuario puede acceder a las secciones funcionales más relevantes del sistema:

- **Dashboard:** Presenta un resumen general de la actividad del sistema mediante gráficos y estadísticas generadas dinámicamente.
- **Formularios:** Permite crear, editar y consultar los formularios registrados en la base de datos.
- **Reportes:** Genera informes basados en los datos recolectados por los usuarios de campo.
- **Ayuda:** Ofrece acceso a documentación técnica y guías de soporte.
- **Sesión / Cierre de sesión:** Facilita la gestión de la sesión de usuario actual, permitiendo cerrar sesión de forma segura.

Este enfoque estructural permite que la información esté organizada jerárquicamente y minimizando la carga cognitiva del usuario.

### 12.0.3. Barra de Navegación y Paneles de Contenido

La barra de navegación y los paneles de contenido conforman los elementos principales de interacción del módulo *frontend*, facilitando al usuario un acceso rápido, claro y estructurado a las distintas secciones del sistema *Santa Ana AgroForms*.

En la parte superior se encuentra la barra de navegación (Figura 14), la cual muestra el nombre del sistema y la sección activa. Esta barra se mantiene visible en todo momento, garantizando al usuario la identificación del módulo en uso y una orientación constante dentro de la plataforma.

El panel principal de contenido, ubicado en la zona central de la interfaz, despliega la información correspondiente al módulo seleccionado. En el ejemplo mostrado se visualiza la sección *Dashboard*, donde se presentan indicadores y resúmenes de actividad del sistema. Asimismo, el ícono de menú ( $\equiv$ ) permite ocultar o mostrar el panel lateral, adaptando la visualización del entorno según las preferencias del usuario o el tamaño del dispositivo.



Figura 14: Barra de navegación y panel principal de contenido del sistema.

El menú principal se ubica en la parte izquierda de la pantalla (Figura 15) y actúa como eje de navegación de la aplicación. Desde este panel, el usuario puede acceder a las diferentes secciones funcionales del sistema, organizadas de forma jerárquica y con identificadores visuales que favorecen la usabilidad. Cada opción del menú se representa mediante un ícono y un nombre descriptivo, lo que facilita la localización y comprensión de las funciones disponibles.

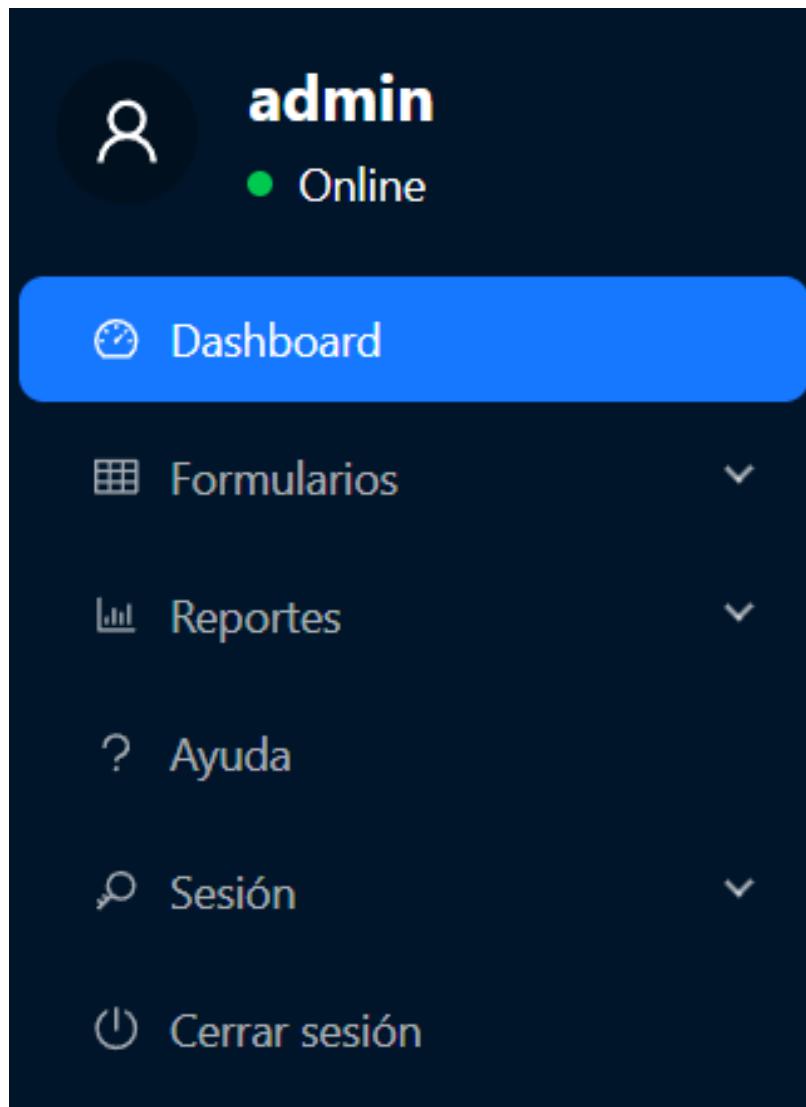


Figura 15: Panel lateral de navegación principal del sistema.

Este diseño visual y funcional responde a los principios de usabilidad establecidos durante la fase metodológica, priorizando la consistencia, visibilidad y retroalimentación en la interacción con el usuario.

#### 12.0.4. Panel de *Dashboard*

El *Dashboard* o panel principal del sistema proporciona una vista general del estado y la actividad de la plataforma (Figura 16). Desde esta sección, el usuario puede visualizar información resumida y en tiempo real sobre los indicadores más relevantes del sistema, facilitando la toma de decisiones y el seguimiento del rendimiento operativo.

El diseño del panel integra componentes gráficos dinámicos que presentan métricas clave como el número de usuarios totales y activos, el porcentaje de sesiones registradas, así

como estadísticas relacionadas con el flujo de formularios enviados y procesados. Estos datos se representan mediante gráficas de líneas, barras y sectores, desarrolladas con librerías especializadas de visualización que permiten una interpretación clara e intuitiva.

El panel de indicadores superiores muestra el resumen cuantitativo de la actividad del sistema, mientras que las gráficas inferiores permiten analizar la evolución temporal de los formularios, la distribución por tipo de registro y la cantidad de terminales utilizadas por usuario. Este enfoque visual refuerza la comprensión de la información y contribuye a una gestión más eficiente de los procesos operativos dentro del Ingenio Santa Ana.

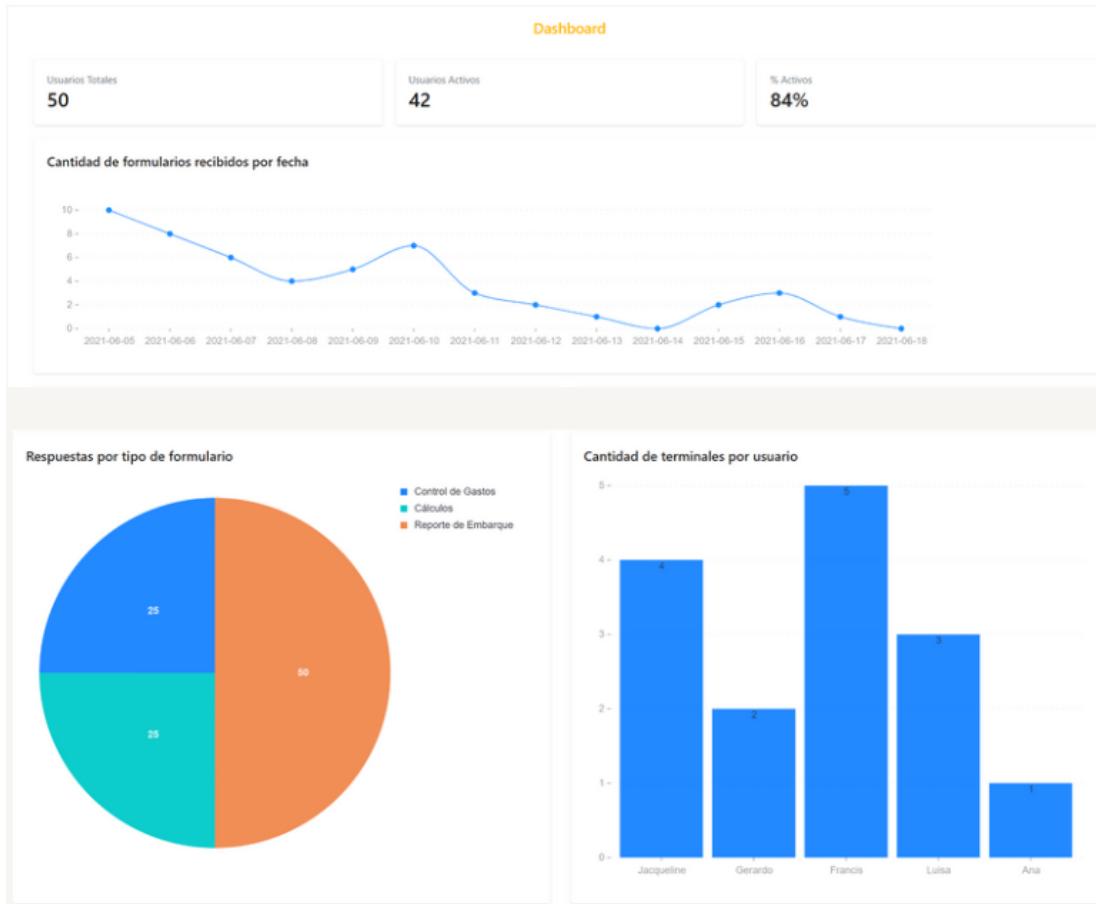


Figura 16: Vista del panel principal de *Dashboard* con indicadores y gráficos de actividad del sistema.

### 12.0.5. Listado de categorías

La sección *Listado de categorías* permite al usuario visualizar, gestionar y organizar todos los formularios disponibles dentro del sistema agrupados vía categorías. Desde esta interfaz, el usuario puede acceder de manera centralizada a los formularios creados, clasificados por categorías, así como crear nuevos formularios o modificar los existentes.

Este módulo se diseñó con el objetivo de optimizar la administración de los formularios, facilitando la búsqueda, clasificación y edición a través de una interfaz intuitiva y coherente con el resto del sistema. Además, la disposición de los elementos sigue principios de usabilidad

orientados a reducir la carga cognitiva y mejorar la experiencia de interacción del usuario.

The screenshot shows a user interface for managing forms. At the top, there are buttons for 'Crear categoría' (Create category) and 'Categoría ↑' (Category up). To the right is a search bar with the placeholder 'Introduzca el texto a buscar...' (Enter text to search...). Below these are filter options: 'Nuevo formulario' (New form), 'Título' (Title), 'Desde' (From), 'Hasta' (To), 'Estado' (State), '¿Es Público?' (Is it public?), and '¿Auto Envío?' (Automatic sending?). The main area displays a table with five rows of data:

▶ Categoría: Básicos		
▶ Categoría: CategoríaPruebaXD123		
▶ Categoría: ChavalesxdHolaaaaaa		
▶ Categoría: Seguridad 107		
▶ Categoría: Siembra Mecanizada 9655		

Figura 17: Vista general de la sección *Listado de Formularios*.

Como se observa en la Figura 17, los formularios se agrupan en categorías, lo que permite una organización jerárquica y clara. Desde esta vista, el usuario puede identificar rápidamente cuáles son las categorías existentes y qué formularios están asociados a cada una de ellas.

En la parte superior de la interfaz se ubican los botones *Crear categoría* y *Nuevo formulario*, que facilitan la gestión de nuevos elementos sin necesidad de salir del módulo principal. Asimismo, se incluye un campo de búsqueda que permite filtrar formularios por título o categoría, incrementando la eficiencia en la localización de registros.

Cada fila de la tabla muestra información relevante sobre los formularios registrados, incluyendo:

- **Título:** Nombre del formulario.
- **Desde / Hasta:** Periodo de vigencia o disponibilidad del formulario.
- **Estado:** Indica si el formulario se encuentra activo o inactivo.
- **¿Es público?:** Determina si el formulario está disponible para todos los usuarios o restringido a ciertos perfiles.
- **¿Auto envío?:** Indica si los datos capturados se envían automáticamente al sistema.

Esta vista constituye una herramienta central dentro del sistema *Santa Ana AgroForms*, ya que permite un control integral sobre los formularios utilizados en los distintos procesos operativos del Ingenio Santa Ana, manteniendo una estructura ordenada y accesible para los usuarios administrativos.

## Crear Categoría

Las acciones que se pueden realizar con las categorías dentro del módulo *Listado de Formularios* incluyen la creación, modificación y eliminación de agrupaciones. A continuación, se describe la acción **Crear categoría**, la cual permite generar nuevas agrupaciones para organizar los formularios dentro del sistema.

Al hacer clic en el botón *Crear categoría*, se despliega una ventana emergente que permite ingresar los datos necesarios para registrar una nueva categoría dentro del sistema. En esta ventana (Figura 18), el usuario debe completar los campos obligatorios de *Nombre* y *Descripción*, asegurando que cada categoría cuente con una identificación clara y un propósito definido.

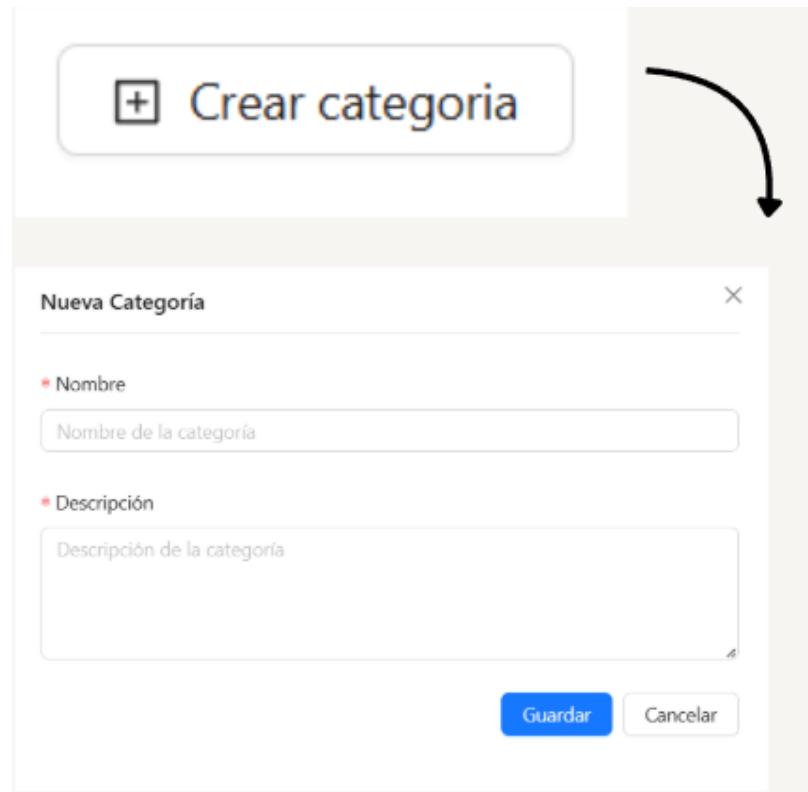


Figura 18: Ventana modal para la creación de una nueva categoría.

Una vez completados los campos, el usuario puede presionar el botón *Guardar* para registrar la nueva categoría, o bien seleccionar *Cancelar* para descartar la acción. Esta funcionalidad facilita la administración jerárquica de los formularios y mantiene la coherencia estructural del sistema, contribuyendo a una organización más eficiente de la información y a una experiencia de uso más fluida.

## Editar Categoría

La acción *Editar categoría* permite modificar el nombre de una categoría existente dentro del módulo *Listado de Formularios*.

Para editar una categoría, el usuario debe hacer clic en el ícono de lápiz ubicado junto al nombre de la categoría que desea actualizar (Figura ??). Al hacerlo, se despliega una ventana emergente (Figura 19) que permite modificar los campos correspondientes.

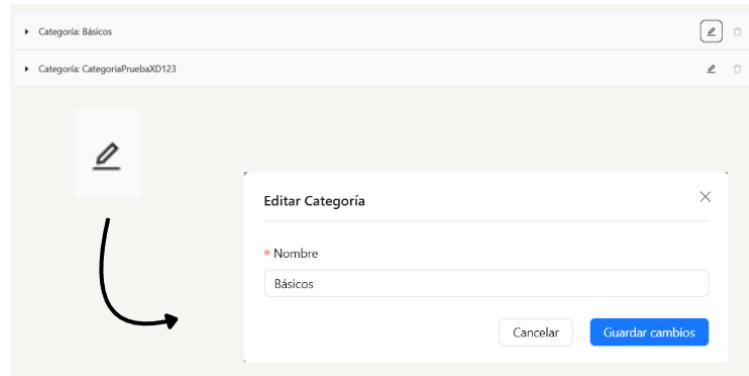


Figura 19: Ventana modal para la edición de una categoría existente.

En la ventana de edición, el sistema muestra el nombre actual de la categoría, permitiendo al usuario modificarlo según sea necesario. Una vez realizados los cambios, se debe presionar el botón *Guardar cambios* para confirmar la modificación o *Cancelar* para descartarla. Esta funcionalidad proporciona flexibilidad y control sobre la organización del sistema, permitiendo ajustes dinámicos sin comprometer la integridad de los datos ni la estructura de los formularios asociados.

## Eliminar Categoría

La acción *Eliminar Categoría* permite borrar de forma definitiva una categoría existente dentro del módulo *Listado de Formularios*.

Para realizar esta operación, el usuario debe hacer clic en el ícono de papelera ubicado junto al nombre de la categoría que desea eliminar (Figura ??). Al seleccionar esta opción, el sistema despliega una ventana emergente de confirmación (Figura 20) que advierte al usuario sobre la irreversibilidad de la acción.

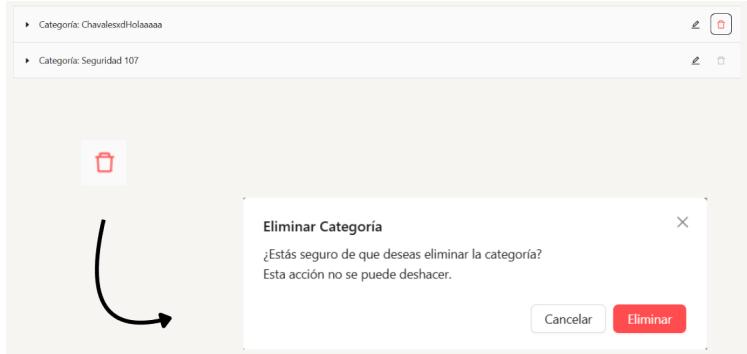


Figura 20: Ventana modal de confirmación para la eliminación de una categoría.

El sistema solicita la confirmación del usuario antes de proceder con la eliminación, mostrando el mensaje: “*¿Estás seguro de que deseas eliminar la categoría? Esta acción no se puede deshacer.*”

Si el usuario confirma la acción mediante el botón *Eliminar*, la categoría se elimina de manera permanente del sistema. No obstante, para que esta funcionalidad esté disponible, la categoría no debe tener formularios asociados; en caso contrario, el sistema impedirá su eliminación.

### 12.0.6. Formularios

#### Creación de Formularios

La funcionalidad *Crear formulario* permite generar nuevos formularios dentro del sistema, facilitando la recolección y organización de información según las necesidades del usuario.

Para iniciar la creación de un nuevo formulario, el usuario debe hacer clic en el botón *Nuevo formulario* (Figura 21), ubicado en la interfaz principal del módulo *Listado de Formularios*.

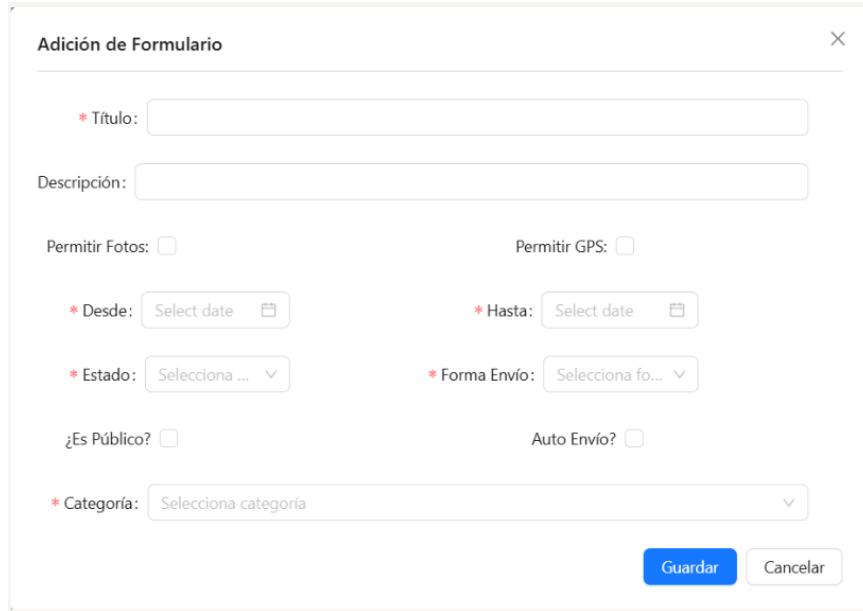


Figura 21: Botón para la creación de un nuevo formulario.

#### Adición de Formularios

Al seleccionar la opción *Nuevo formulario*, se despliega una ventana donde el usuario debe completar los campos requeridos, como título, descripción, fechas de inicio y finalización, estado, forma de envío y categoría (Figura 22). También es posible configurar opciones

adicionales, como permitir el uso de fotos o GPS, marcar el formulario como público o habilitar el autoenvío de información.



La imagen muestra una ventana de diálogo titulada "Adición de Formulario". La interfaz es la siguiente:

- Título:** Campo de texto obligatorio.
- Descripción:** Campo de texto.
- Permitir Fotos:** Botón de opción.
- Permitir GPS:** Botón de opción.
- Desde:** Campo para seleccionar una fecha con botones "Select date" y "Calendar".
- Hasta:** Campo para seleccionar una fecha con botones "Select date" y "Calendar".
- Estado:** Botón desplegable "Selecciona ...".
- Forma Envío:** Botón desplegable "Selecciona fo...".
- ¿Es Público?** Botón de opción.
- Auto Envío?** Botón de opción.
- Categoría:** Botón desplegable "Selecciona categoría".
- Guardar:** Botón azul.
- Cancelar:** Botón blanco.

Figura 22: Ventana de adición de un nuevo formulario.

Una vez ingresados los datos, se debe presionar el botón *Guardar* para registrar el nuevo formulario o *Cancelar* para descartar la operación. Esta funcionalidad facilita la creación y personalización de formularios, adaptándose a las necesidades operativas del usuario.

## Opciones dentro los formularios

Cada formulario cuenta con un menú de opciones que permite realizar distintas acciones de gestión (Figura 23). Para acceder a este menú, el usuario debe hacer clic en el ícono de los seis puntos ubicado junto al nombre del formulario.

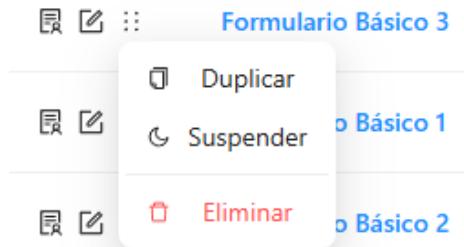


Figura 23: Menú de opciones disponible para cada formulario.

Al desplegar el menú, se muestran distintas acciones disponibles, como *Duplicar*, *Suspender* o *Eliminar* el formulario. Estas opciones brindan al usuario un control más completo

sobre la administración de los formularios, facilitando su mantenimiento y actualización dentro del sistema.

## Asignar Formulario

La opción *Asignar formulario* permite vincular uno o varios usuarios a un formulario específico (Figura 24). Al realizar esta acción, el formulario se habilita automáticamente en la aplicación móvil de los usuarios seleccionados, permitiéndoles acceder y completar la información correspondiente desde sus dispositivos.



Figura 24: Ventana para asignar un formulario a uno o varios usuarios.

Esta funcionalidad facilita la distribución de tareas y garantiza que cada usuario cuenta con los formularios necesarios para sus actividades en campo, optimizando la gestión operativa dentro del sistema.

## Editar Formulario

La opción *Editar formulario* permite modificar la información de un formulario existente (Figura 25). Desde esta ventana, el usuario puede actualizar campos como título, descripción, fechas de vigencia, estado, forma de envío, categoría y opciones adicionales como visibilidad pública, autoenvío, GPS o permiso para fotos.

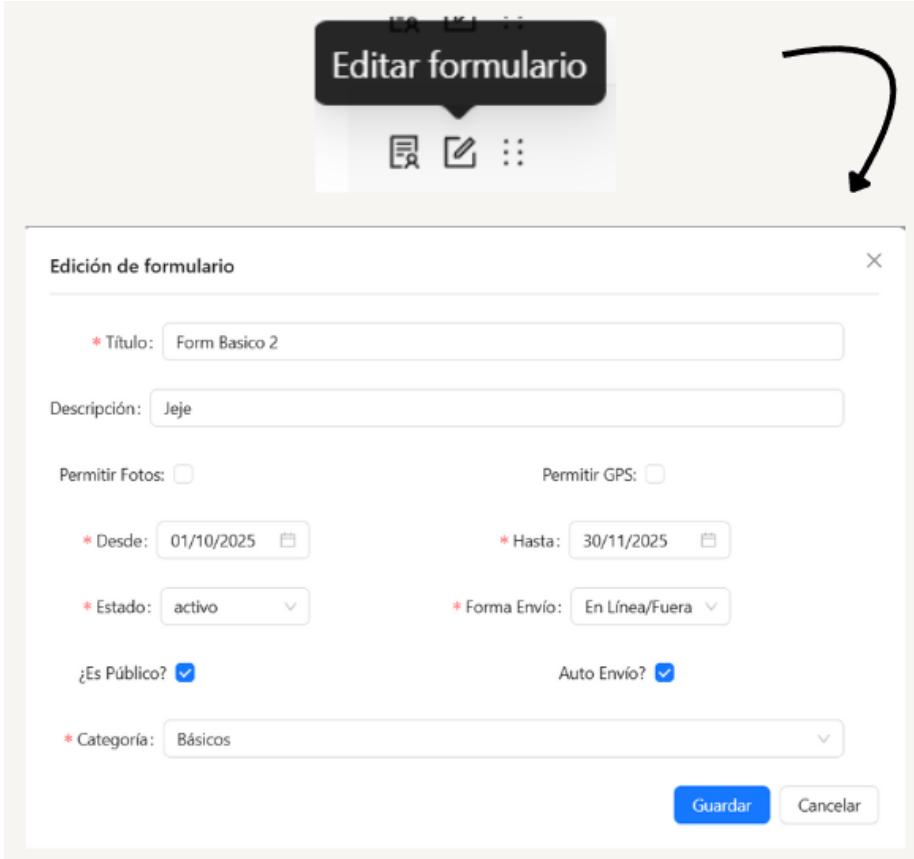


Figura 25: Ventana de edición de un formulario existente.

Una vez realizados los cambios, se debe presionar el botón *Guardar* para aplicar las modificaciones o *Cancelar* para descartarlas. Esta funcionalidad permite mantener actualizados los formularios sin necesidad de crear nuevos, optimizando la gestión y continuidad del trabajo en la plataforma.

## Duplicar Formulario

La opción *Duplicar formulario* permite crear una copia exacta de un formulario existente, conservando toda su estructura y contenido original (Figura 26). Al seleccionar esta acción, el sistema muestra una ventana de confirmación donde el usuario debe validar si desea proceder con la duplicación. Una vez confirmada, se genera un nuevo formulario con las mismas configuraciones, el cual puede ser editado y personalizado de manera independiente.

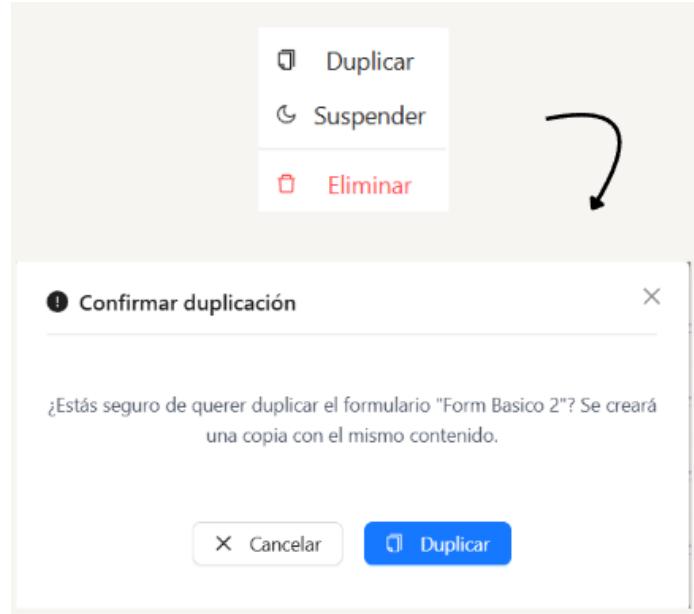


Figura 26: Proceso de duplicación de un formulario existente.

Esta funcionalidad resulta especialmente útil para agilizar la creación de formularios similares sin necesidad de configurarlos desde cero. Además, contribuye a mantener la coherencia entre formularios que comparten una estructura común, reduciendo errores y optimizando el tiempo de gestión.

### Suspender Formulario

La opción *Suspender formulario* permite desactivar temporalmente un formulario para que no esté disponible en la aplicación móvil (Figura 27). Al seleccionar esta acción, el sistema solicita una confirmación antes de proceder. Una vez suspendido, el formulario deja de estar visible para los usuarios, aunque puede reactivarse en cualquier momento si es necesario.

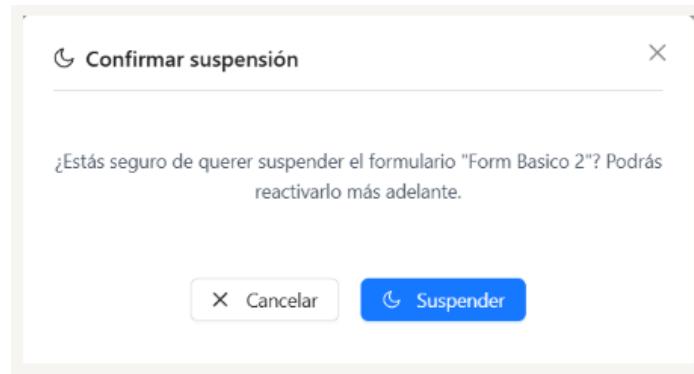


Figura 27: Ventana de confirmación para suspender un formulario.

Esta funcionalidad es útil cuando se requiere detener temporalmente la recolección de información sin eliminar el formulario, permitiendo mantener su configuración y contenido para futuras reactivaciones.

## Eliminar Formulario

La opción *Eliminar formulario* permite borrar de manera permanente un formulario del sistema (Figura 28). Al seleccionar esta acción, el sistema muestra una ventana de confirmación que advierte que el proceso no se puede deshacer. Si el usuario confirma la eliminación, el formulario y toda su configuración asociada son removidos definitivamente.

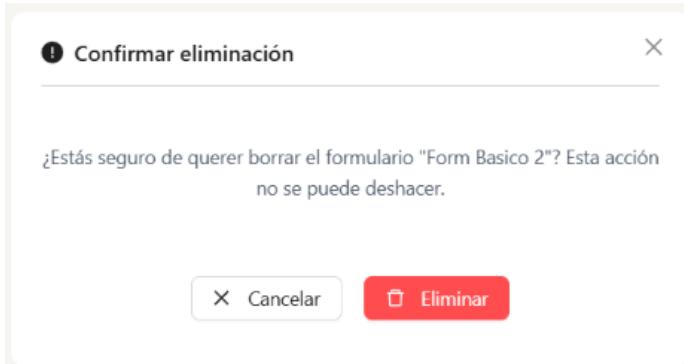


Figura 28: Ventana de confirmación para eliminar un formulario.

Esta funcionalidad debe utilizarse con precaución, ya que elimina de forma irreversible el formulario junto con todos sus datos y configuraciones, garantizando que no quede rastro en el sistema.

## Acceso a un Formulario

Para acceder al contenido de un formulario, el usuario debe hacer clic sobre su nombre, identificado en color celeste dentro del listado (Figura 29). Esta acción abre la vista de edición, donde se pueden visualizar y gestionar los elementos que lo componen, como campos de texto, fechas y opciones de selección.

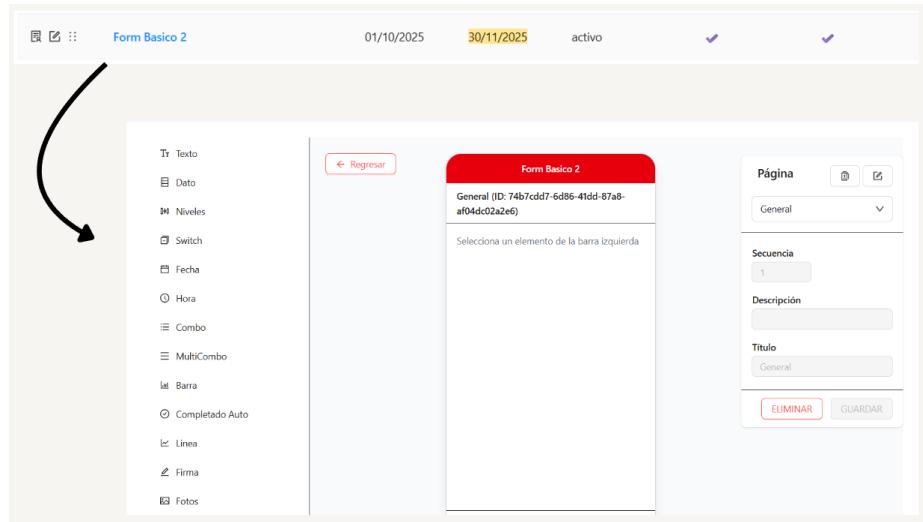


Figura 29: Acceso a la vista de edición de un formulario.

Desde esta interfaz también es posible modificar la estructura del formulario o actualizar su configuración según las necesidades del usuario.

### Campo de Texto

A continuación, se presenta una lista de campos que pueden crearse desde esta interfaz.

- **Texto:** El campo *Texto* permite ingresar información en formato de texto libre, siendo ideal para capturar nombres, descripciones, observaciones y cualquier tipo de contenido alfabético o alfanumérico que no requiera un formato específico (Figura 30).

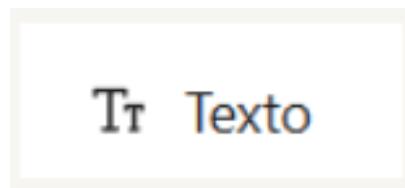


Figura 30: Campo de tipo Texto disponible en el generador de formularios.

### Valores básicos para crear un campo

En todos los formularios de creación de campo, se deben definir ciertos valores fundamentales que determinan el comportamiento y la visualización del campo dentro del formulario (Figura 31).

The dialog box has a title bar 'Edición de campo: texto' and a close button 'X'. Inside, there are four input fields with red asterisks indicating they are required: 'Nombre' (Name), 'Etiqueta' (Label), 'Ayuda' (Help), and 'Requerido' (Required). Below these is a dropdown menu labeled 'Grupo' (Group) with the option 'Selecciona grupo' (Select group). At the bottom right is a blue 'Guardar' (Save) button.

Figura 31: Ventana de edición para definir los valores básicos de un campo.

En todos los formularios de creación de campo, el **Nombre** se refiere siempre al identificador interno del campo, mientras que la **Etiqueta** corresponde al nombre que los usuarios de la aplicación móvil visualizarán. Ambos valores son obligatorios para todos los campos.

El campo **Ayuda** permite incluir información adicional o instrucciones para guiar al usuario sobre cómo completar el campo. El atributo **Requerido** indica si el campo es obligatorio antes de enviar el formulario, garantizando que el usuario no omita información esencial. Finalmente, la opción **Grupo** permite asociar el campo a un grupo específico dentro del formulario, en caso de que este esté organizado por secciones (lo cual se detalla más adelante).

### Campo de Número

- **Número:** El campo *Número* está especializado para el ingreso de valores numéricos, aceptando exclusivamente datos de tipo número. Resulta ideal para registrar cantidades, precios, edades, mediciones y cualquier información matemática que requiera cálculos posteriores (Figura 32).



Figura 32: Campo de tipo Número disponible en el generador de formularios.

Además, al crear o editar un campo numérico, se pueden configurar límites mínimos y máximos para los valores permitidos (Figura 33). Esto permite controlar el rango de datos que los usuarios pueden ingresar, asegurando la validez y consistencia de la información capturada.

The screenshot shows a configuration dialog for a numerical field named 'numero'. It includes fields for 'Nombre' (Name) and 'Etiqueta' (Label), both marked with a red asterisk indicating they are required. There is an 'Ayuda' (Help) field and a 'Requerido' (Required) checkbox. A 'Grupo' (Group) dropdown menu is set to 'Selecciona grupo'. Under 'Límites de datos' (Data limits), there are fields for 'Cantidad mínima' (Minimum value) and 'Cantidad máxima' (Maximum value). A blue 'Guardar' (Save) button is at the bottom right.

Figura 33: Ventana de edición de campo numérico con límites de datos configurables.

En este caso, es posible definir valores mínimos y máximos que el campo numérico puede aceptar, dependiendo del tipo de información que se desea registrar.

### Campo de Switch

- **Switch:** El campo *Switch* funciona como un interruptor booleano que alterna entre dos estados opuestos, tales como verdadero/falso, sí/no o activo/inactivo. Es especialmente útil para opciones binarias donde se necesita una respuesta clara entre dos posibilidades (Figura 34).

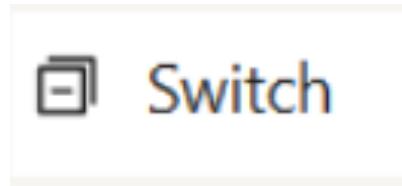


Figura 34: Campo de tipo Switch disponible en el generador de formularios.

Al crear o editar un campo de tipo Switch, se presentan las mismas opciones básicas de configuración, como el nombre, la etiqueta, la ayuda, la opción de requerido y la asignación a un grupo (Figura 35). Esto permite adaptar su comportamiento y presentación según las necesidades del formulario.

The screenshot shows a modal window titled "Edición de campo: switch". It contains the following fields:

- \* Nombre: A text input field.
- \* Etiqueta: A text input field next to an "Ayuda:" text input field.
- Requerido: A checkbox labeled "Requerido".
- Grupo: A dropdown menu labeled "Selecciona grupo".
- Guardar: A blue button at the bottom right.

Figura 35: Ventana de edición del campo Switch con opciones de configuración básicas.

## Campo de Fecha

- **Fecha:** El campo *Fecha* está diseñado para seleccionar o ingresar fechas específicas, incorporando generalmente un selector de calendario que facilita la entrada precisa de datos cronológicos y evita errores de formato (Figura 36).



Figura 36: Campo de tipo Fecha disponible en el generador de formularios.

Durante la creación o edición de un campo de tipo Fecha, el sistema permite definir parámetros adicionales como el valor inicial y reglas de visualización condicional (Figura 37). Estas opciones otorgan mayor flexibilidad al comportamiento del campo, adaptándolo a distintos escenarios de uso.

Edición de campo: fecha

\* Nombre:

\* Etiqueta:  Ayuda:

Requerido  Valor inicial:

Grupo:

Regla de Visualización:

Figura 37: Ventana de edición del campo Fecha con opciones de configuración y visualización.

### Campo de Hora

- **Hora:** El campo *Hora* se especializa en el ingreso de horarios, permitiendo capturar horas y minutos según sea necesario. Es fundamental para aplicaciones que requieren gestión de tiempo, programación de actividades o control de horarios (Figura 38).

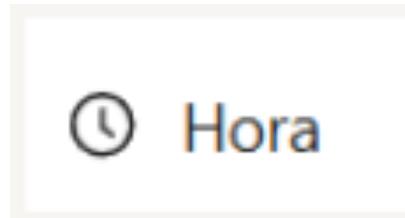


Figura 38: Campo de tipo Hora disponible en el generador de formularios.

Durante la creación o edición del campo Hora, se mantienen las opciones básicas de configuración como nombre, etiqueta, ayuda y grupo (Figura 39). Este tipo de campo permite capturar información temporal precisa, facilitando la organización y el control de registros dependientes del tiempo.

The screenshot shows a modal window titled 'Edición de campo: hora'. Inside, there are four input fields: 'Nombre' (with placeholder 'Nombre:'), 'Etiqueta' (with placeholder 'Etiqueta:' and a 'Ayuda:' field next to it), and 'Grupo' (a dropdown menu with 'Selecciona grupo'). At the bottom right is a blue 'Guardar' button.

Figura 39: Ventana de edición del campo Hora con opciones de configuración básicas.

## Campo de Combo

- **Combo:** El campo *Combo* funciona como una lista desplegable de opciones predefinidas que permite seleccionar una única opción de un conjunto de valores establecidos. Es ideal para categorías, tipos, estados u otras situaciones donde las opciones disponibles son limitadas y conocidas (Figura 40).



Figura 40: Campo de tipo Combo disponible en el generador de formularios.

Durante la creación o edición del campo *Combo*, se despliega una ventana con las opciones básicas de configuración, tales como: **Nombre**, **Etiqueta**, **Ayuda**, **Grupo** y **Valores** (Figura 41).

En el apartado **Valores**, se deben escribir los elementos que conformarán la lista desplegable, ingresándolos *uno por línea*. Cada línea representa una opción distinta que el usuario podrá seleccionar al momento de llenar el formulario.

The screenshot shows the configuration interface for a 'Combo' field. At the top, it says 'Edición de campo: combo'. Below that are fields for 'Nombre' (Name) and 'Etiqueta' (Label). There is also a 'Ayuda' (Help) field and a 'Requerido' (Required) checkbox. A dropdown menu for 'Grupo' (Group) is set to 'Selecciona grupo'. The main section is titled 'Datos' (Data) and contains a text area for 'Valores' (Values) with the placeholder 'Escribe los valores separados por filas' (Write values separated by rows). A blue 'Guardar' (Save) button is located at the bottom right.

Figura 41: Ventana de edición del campo Combo con configuración de valores.

Por ejemplo, si se desea crear una lista desplegable con diferentes tipos de caña, bastará con colocar en el campo **Valores** los datos separados por filas, como se muestra a continuación:

The screenshot shows the 'Valores' (Values) input field. It contains three lines of text: 'Caña Criolla', 'Caña de Alta Rendimiento', and 'Caña Transgénica', each on a separate line. The entire input field is highlighted with a blue border.

Figura 42: Ejemplo de valores ingresados para un campo Combo.

De esta forma, el formulario mostrará un menú desplegable con las opciones *Caña Criolla*, *Caña de Alta Rendimiento* y *Caña Transgénica*, facilitando la selección rápida y evitando errores de escritura.

Una vez completadas las opciones de respuesta para este campo en el formulario, bastará con hacer clic en el botón **Guardar** para crear el campo y que quede disponible en el generador de formularios.

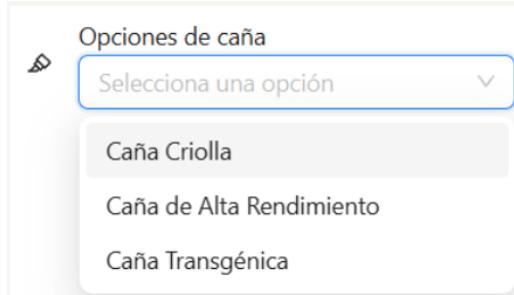


Figura 43: Ejemplo del campo *Combo* con sus opciones disponibles en el formulario.

De esta forma, se obtiene una lista desplegable funcional dentro del formulario, desde la cual el usuario puede seleccionar una única opción entre las disponibles, garantizando uniformidad en los datos registrados y evitando errores de escritura.

### Campo de Dataset

- **Dataset:** El campo *Dataset* es similar al campo *Combo*, con la diferencia principal de que las opciones no se ingresan manualmente, sino que se cargan desde una fuente de datos externa. Esto permite vincular un conjunto de registros almacenados en un archivo de Excel previamente cargado al sistema, facilitando la reutilización de datos y garantizando coherencia entre distintos formularios (Figura 44).

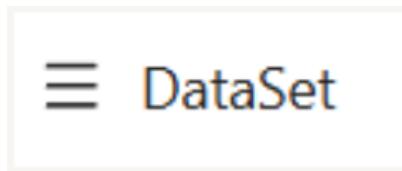


Figura 44: Campo de tipo Dataset disponible en el generador de formularios.

Para utilizar este tipo de campo, es necesario haber cargado previamente un archivo de Excel como fuente de datos dentro del sistema. Durante la configuración del campo, el usuario puede seleccionar la fuente correspondiente y asociarla al formulario. De esta manera, las opciones del desplegable se generan automáticamente a partir de la información contenida en dicho archivo (Figura 45).

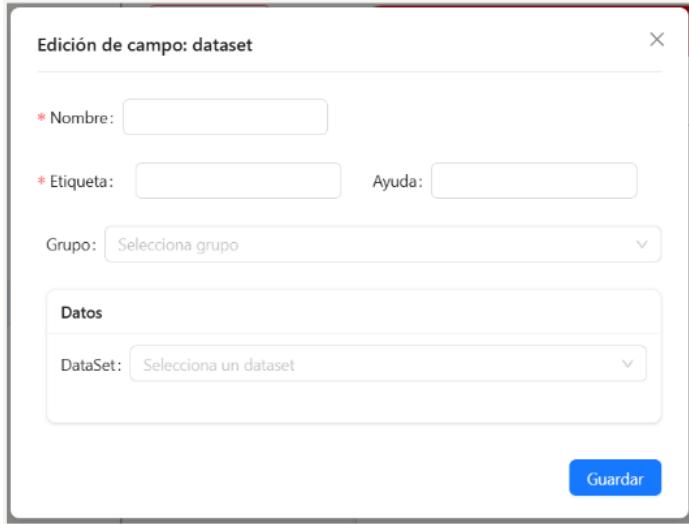


Figura 45: Ventana de edición del campo Dataset con opciones de configuración.

**Nota:** La correcta utilización del campo *Dataset* requiere que las fuentes de datos estén definidas y cargadas con anterioridad en el sistema. La sección de **Fuentes de datos** del presente documento explica este proceso en detalle y constituye un requisito previo para la creación de este tipo de campo.

Al abrir el Modal, tendremos los datos habituales a llenar de cualquier campo; lo único diferente será una lista desplegable de *Datasets* que ya están cargados en el sistema.

<b>Id</b>	<b>Fecha</b>	<b>Área</b>	<b>Producción (toneladas)</b>	<b>Consumo de bagazo (toneladas)</b>	<b>Horas de operación</b>	<b>Rendimiento (%)</b>	<b>Observaciones</b>
1	1/10/2025	Energía	249.67	69.94	17.15	89.62	Parada programada
2	2/10/2025	Campo	476.73	283.19	18.92	80.91	Alta eficiencia
3	3/10/2025	Mantenimiento	444.89	63.36	22.86	85.18	Normal
4	4/10/2025	Laboratorio	186.97	196.31	22.39	84.88	Normal
5	5/10/2025	Energía	255.07	221.32	21.42	71.47	Revisión de calidad
6	6/10/2025	Campo	373.2	195.36	19.51	71.8	Normal
7	7/10/2025	Campo	401.71	263.45	19.1	90.55	Mantenimiento menor
8	8/10/2025	Energía	160.93	259.83	23.56	84.37	Alta eficiencia
9	9/10/2025	Fábrica	367.94	160.08	20.42	91.42	Normal
10	10/10/2025	Campo	163.41	228.92	22.07	88.9	Parada programada
11	11/10/2025	Energía	408.45	152.49	21.47	83.78	Revisión de calidad
12	12/10/2025	Laboratorio	108.39	164.52	23.18	72.64	Normal
13	13/10/2025	Energía	358.75	244.13	17.99	80.22	Alta eficiencia
14	14/10/2025	Campo	471.2	151.93	19.28	90.87	Normal

Figura 46: Ejemplo de datos cargados desde un archivo Excel para el campo Dataset.

Para este ejemplo, se utiliza un archivo Excel de prueba con datos ficticios, el cual ya se encuentra cargado en la sección de **Fuentes de datos**. Este archivo contiene columnas como *Fecha*, *Área*, *Producción (toneladas)*, *Consumo de bagazo (toneladas)*, *Horas de operación*, *Rendimiento (%)*, y *Observaciones*, entre otras.

**IMPORTANTE:** Debe existir una columna llamada **id** que numere las filas. Si esta columna no está presente, es posible que el campo *Dataset* no funcione correctamente al momento de vincular los datos con el formulario.

Este mecanismo permite que los formularios se alimenten de fuentes de datos externas previamente definidas, garantizando coherencia y reutilización de la información a lo largo

del sistema.

Durante la configuración del campo *Dataset*, el usuario podrá desplegar una lista con todas las fuentes de datos previamente cargadas en el sistema. Desde este listado, simplemente se selecciona el nombre del Dataset que se desea utilizar, tal como se muestra en la Figura 47.

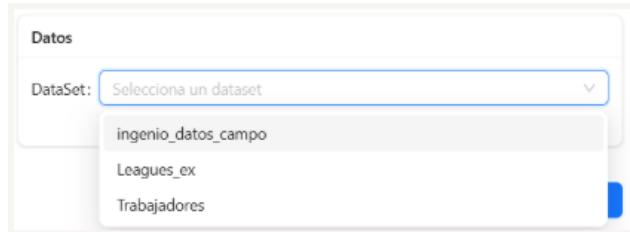


Figura 47: Selección del Dataset desde la lista desplegable de fuentes de datos.

Al abrir la lista de opciones, seleccionamos el nombre del Dataset correspondiente; en este ejemplo, sería `ingenio_datos_campo`. Una vez seleccionado, el sistema habilitará automáticamente un nuevo campo llamado **Columna**, el cual permitirá definir qué conjunto de datos dentro del archivo será utilizado como variable principal en el formulario (Figura 48).

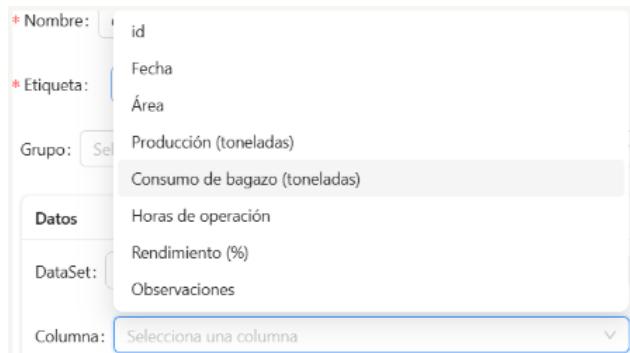


Figura 48: Selección de la columna asociada al Dataset.

En esta lista desplegable de *Columnas*, aparecerán todos los encabezados del archivo Excel vinculado al Dataset, tales como *id*, *Fecha*, *Área*, *Producción (toneladas)*, *Consumo de bagazo (toneladas)*, *Horas de operación*, *Rendimiento (%)*, y *Observaciones*.

De esta manera, el usuario podrá asignar dinámicamente la columna que se mostrará o utilizará dentro del formulario, manteniendo la consistencia y reutilización de los datos cargados en el sistema.

Una vez finalizada la configuración, simplemente se debe hacer clic en el botón **Guardar** para almacenar los cambios realizados en el campo (Figura 49).

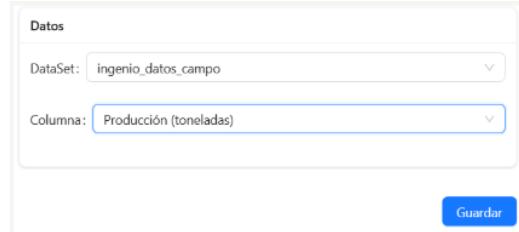


Figura 49: Configuración final del campo Dataset antes de guardar.

De esta forma, el formulario quedará vinculado con la fuente de datos externa seleccionada. Cuando un usuario complete este formulario, podrá visualizar y seleccionar los valores correspondientes a la columna de datos elegida en el archivo de Excel. Esto permite que los formularios mantengan coherencia con la información existente en el sistema y facilita la reutilización de datos sin necesidad de ingresar la información manualmente.

### Campo de Firma

- **Firma:** El campo *Firma* funciona de tal manera que permite al usuario, desde la aplicación móvil, dibujar su firma en un área tipo *Canvas*. Este campo es especialmente útil en formularios que requieren validación o confirmación por parte del usuario (Figura 50).

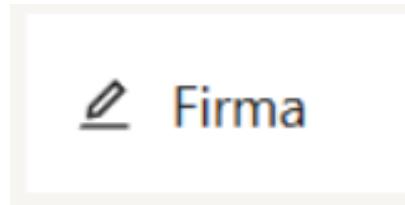


Figura 50: Campo de tipo Firma disponible en el generador de formularios.

Durante su configuración, el campo de Firma mantiene las propiedades básicas como nombre, etiqueta, ayuda, requerido y grupo (Figura 51). Una vez implementado, el usuario puede trazar su firma directamente sobre la pantalla del dispositivo, garantizando un registro visual y auténtico dentro del formulario.

Edición de campo: firma

\* Nombre:

\* Etiqueta:  Ayuda:

Requerido

Grupo:

Datos:

Figura 51: Ventana de edición del campo Firma con opciones de configuración básicas.

### Campo Calculado

- **Calculado:** El campo *Calculado* permite realizar operaciones matemáticas utilizando los valores de otros campos numéricos, ya sean de la misma página o de páginas anteriores del formulario. Este tipo de campo es especialmente útil para automatizar cálculos y reducir errores manuales en la captura de datos (Figura 52).

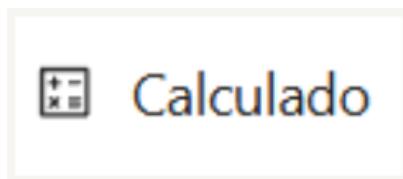


Figura 52: Campo de tipo Calculado disponible en el generador de formularios.

Por ejemplo, supongamos que tenemos dos campos numéricos: “Cantidad de caña” (`num1`) y “Porcentaje de extracción de azúcar” (`num2`). Podemos crear un campo calculado que multiplique ambos valores para obtener un resultado automático basado en los datos ingresados.

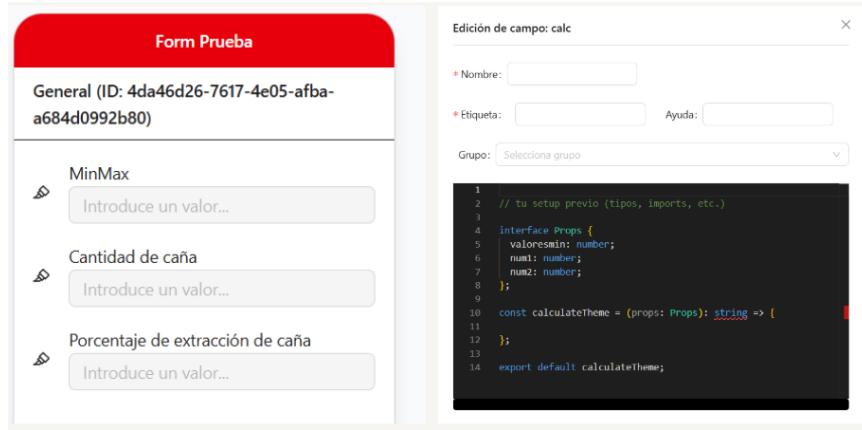


Figura 53: Ventana de edición del campo Calculado con configuración de fórmula.

Como se observa en el modal de creación del campo calculado, el sistema permite definir la lógica matemática o expresión que se ejecutará automáticamente al ingresar los valores base. Esto proporciona gran flexibilidad para crear formularios inteligentes y dinámicos que realicen cálculos sin intervención adicional del usuario.

Como vemos en el modal de creación del campo calculado, se despliega un editor de código en lenguaje de programación *TypeScript*, el cual es un lenguaje similar a *JavaScript*, pero con la característica principal de ofrecer tipado estático de datos. En este editor, se define la estructura que describe las propiedades numéricas que se utilizarán dentro del cálculo (Figura 54).

```
interface Props {
    valoresmin: number;
    num1: number;
    num2: number;
};
```

Figura 54: Definición de una interfaz en *TypeScript* utilizada dentro del campo calculado.

En *TypeScript*, una **interfaz** es una forma de definir la estructura que deben seguir los objetos, especificando sus propiedades y los tipos de datos asociados a cada una. Por ejemplo, al declarar `interface Props { ... }`, se crea una plantilla que describe cómo debe ser un objeto, indicando qué propiedades debe contener y qué tipo de valor puede asignarse a cada una.

En este caso, la interfaz `Props` incluye tres propiedades numéricas: `valoresmin`, `num1` y `num2`, las cuales serán empleadas para realizar las operaciones dentro del campo calculado, garantizando coherencia en el tipo de datos y evitando errores de ejecución.

Esto significa que esta estructura (`Props`) se utilizará para los campos numéricos previa-

mente creados en el formulario, permitiendo que los valores sean accedidos dentro del código del cálculo.

```
const calculateTheme = (props: Props): string => [
];
export default calculateTheme;
```

Figura 55: Declaración de la función `calculateTheme` que retorna un valor en formato String.

En esta parte del código se define qué valor será retornado por el campo calculado. Aunque el resultado final siempre debe ser de tipo `String`, es posible realizar operaciones matemáticas con los campos numéricos antes de convertirlos. De esta forma, el campo calculado puede contener cualquier fórmula que involucre valores numéricos del formulario.

Por ejemplo, si se desea realizar la multiplicación de dos variables numéricas (`num1` y `num2`), se debe utilizar el objeto `props`, como se muestra en el siguiente ejemplo:

```
const calculateTheme = (props: Props): string => [
    return String(props.num1 * props.num2);
];
export default calculateTheme;
```

Figura 56: Ejemplo de cálculo mediante la multiplicación de dos campos numéricos.

En este caso, la función `calculateTheme` retorna el resultado de la multiplicación de los valores `num1` y `num2`, representando, por ejemplo, la operación *Cantidad de caña × Porcentaje de extracción de azúcar*.

De esta manera, es posible realizar cualquier tipo de operación matemática dentro del campo calculado, y posteriormente devolver el resultado como un valor de tipo cadena (`String`), asegurando compatibilidad con el sistema y flexibilidad en el diseño de los formularios.

## Campo de Grupo

- **Grupo:** El campo *Grupo* funciona como un mecanismo contenedor que permite agrupar varios campos relacionados entre sí, facilitando la organización de los elementos dentro de secciones lógicas y coherentes. Esta funcionalidad resulta ideal para estructurar formularios complejos, mejorar la experiencia del usuario y mantener la consistencia visual en interfaces con múltiples campos (Figura 57).



Figura 57: Campo de tipo Grupo disponible en el generador de formularios.

Durante la creación de un campo de tipo Grupo, se pueden definir parámetros básicos como nombre, etiqueta, ayuda y asociación con otros grupos (Figura 58). Estos parámetros permiten estructurar jerárquicamente los campos dentro del formulario, facilitando su manejo.

Una ventana de diálogo titulada 'Edición de campo: grupo'. Contiene los siguientes campos: 'Nombre' (con placeholder 'Nombre:'), 'Etiqueta' (con placeholder 'Etiqueta:' y 'Ayuda:' a su lado), y 'Grupo' (una lista desplegable titulada 'Selecciona grupo').

Figura 58: Ventana de edición del campo Grupo con parámetros configurables.

Una vez creado un grupo, este aparecerá como opción disponible al momento de generar otros tipos de campos. Por ejemplo, al crear un nuevo campo de texto, se puede seleccionar el grupo previamente configurado para incluirlo dentro de esa sección específica (Figura 59).

Una ventana de diálogo titulada 'Edición de campo: texto'. Los campos son similares a la Figura 58, pero incluye un checkbox 'Requerido' y un campo 'Grupo' que ya tiene 'Selecciona grupo' resaltado. Abajo de 'Grupo' hay un input text 'Grupo\_nombre' y un botón 'Guardar'.

Figura 59: Ejemplo de un campo de texto asociado a un grupo existente dentro del formulario.

Una vez creado el campo con el grupo asociado, es posible visualizarlo dentro de la lista

general del formulario, donde aparecerá como *indexado*, indicando que forma parte de dicho grupo (Figura 60).



Figura 60: Campo asociado a un grupo mostrado como elemento indexado dentro de la lista de campos.

De esta manera, el sistema organiza visualmente los campos pertenecientes a un grupo, facilitando la identificación jerárquica y la administración de estructuras complejas dentro del formulario.

## Páginas

La vista de **Páginas** permite gestionar las diferentes secciones o módulos que conforman el proyecto. Desde esta interfaz, el usuario puede crear, editar o eliminar páginas según sea necesario, facilitando la organización jerárquica del contenido dentro del sistema.

En la parte superior se encuentran los iconos principales que controlan las acciones disponibles: *Creación* y *Edición de páginas*. Estas herramientas permiten mantener un flujo de trabajo ordenado al estructurar las distintas áreas del proyecto.

Debajo de estos controles, se muestra un formulario con la información correspondiente a la página seleccionada, incluyendo su tipo, secuencia, descripción y título (Figura 61).

The screenshot shows a user interface for configuring a page. At the top, there's a title bar with the word 'Página' and two icons: a floppy disk for saving and a pencil for editing. Below this is a dropdown menu set to 'General'. The main area contains three fields: 'Secuencia' with the value '1', 'Descripción' (an empty text input field), and 'Título' with the value 'General'. At the bottom right of the main area is a red-bordered button labeled 'ELIMINAR' (Delete).

Figura 61: Vista del formulario de configuración de páginas.

A través de esta vista, el usuario puede modificar las propiedades de cada página, asignar una secuencia dentro del flujo de navegación y definir una descripción que facilite su identificación. Además, la interfaz incluye un botón de **Eliminar**, el cual permite remover la página seleccionada de manera controlada y segura, asegurando una correcta gestión del proyecto.

La opción de **Crear página** permite generar una nueva sección dentro del sistema. Esta funcionalidad es útil para estructurar las distintas partes del proyecto, ya que cada página puede representar un módulo, formulario o componente independiente.

- **Crear página:** Permite generar una nueva página dentro del sistema.

Al seleccionar esta opción, se mostrará una ventana emergente (modal) que solicita la información básica de la nueva página, incluyendo el **Título** y la **Descripción** (Figura 62).

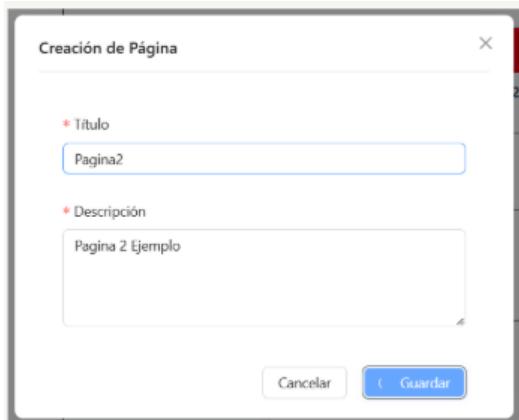


Figura 62: Ventana de creación de una nueva página.

Una vez completados los campos requeridos, el usuario puede presionar el botón **Guardar** para registrar la nueva página dentro del sistema.

De esta manera, la página creada quedará disponible en la lista de páginas configuradas, lista para ser editada o asociada a otras secciones del sistema según la estructura del proyecto.

Una vez creadas las distintas páginas dentro del sistema, es posible navegar entre ellas de manera sencilla. Para ello, se puede utilizar el *dropdown* ubicado debajo de los controles principales de la sección de **Página**, como se muestra en la Figura 63.

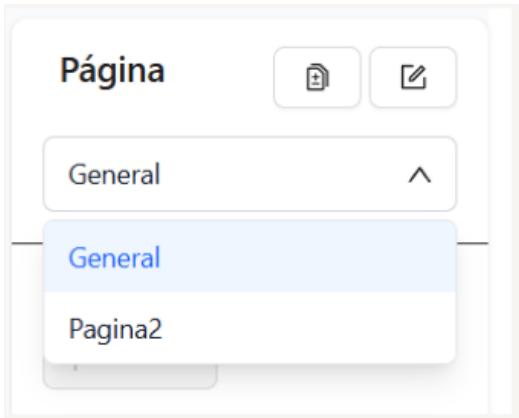


Figura 63: Selección de página mediante el menú desplegable de navegación.

Este menú desplegable permite al usuario seleccionar rápidamente la página deseada, mostrando una lista con los nombres de todas las páginas disponibles, por ejemplo: **General** y **Página2**. Al seleccionar una opción, el sistema actualizará automáticamente la vista del formulario, cargando la información correspondiente a la página elegida.

De esta forma, la navegación entre páginas se vuelve intuitiva y eficiente, facilitando el acceso y la edición de los distintos módulos que conforman el proyecto.

También es posible navegar entre las páginas utilizando las flechas de desplazamiento

incluidas en el diseño del teléfono móvil dentro del entorno de previsualización. Estas flechas permiten avanzar o retroceder entre las diferentes páginas creadas en el proyecto, ofreciendo una experiencia más intuitiva para el usuario durante la edición o prueba del formulario (Figura 64).



Figura 64: Navegación entre páginas desde la vista del teléfono móvil.

Esta funcionalidad resulta especialmente útil cuando se trabaja con formularios extensos que contienen múltiples secciones, ya que permite desplazarse rápidamente sin necesidad de regresar a la vista principal de páginas.

#### 12.0.7. Fuentes de datos

La sección **Fuentes de Datos** permite administrar los conjuntos de información que se utilizan dentro del sistema. Desde esta vista, el usuario puede cargar archivos de datos, principalmente en formato *Excel*, para integrarlos con la aplicación web y utilizarlos como base de datos local o fuente de información dinámica.

Esta herramienta resulta útil para importar información existente, como listados de empleados, productos, clientes u otros tipos de datos estructurados, evitando así la carga manual campo por campo.

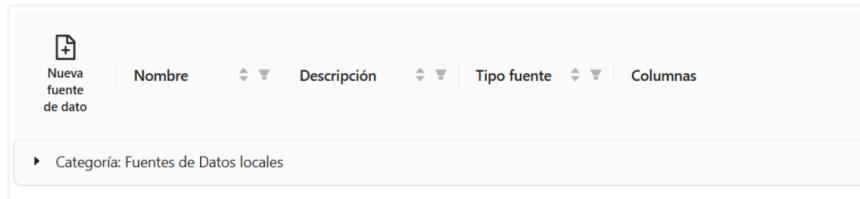


Figura 65: Vista general de la sección Fuentes de Datos.

Para visualizar las fuentes de datos, basta con desplegar la categoría **Fuentes de Datos locales**, la cual mostrará la lista de archivos cargados en el sistema. Cada registro representa una fuente de datos disponible, con la información de su nombre, tipo y columnas incluidas.

Actualmente, las columnas que se muestran en esta sección son las siguientes:

- **Nombre:** Nombre del archivo de Excel cargado.
- **Descripción:** Texto que describe el contenido del archivo al momento de su creación.
- **Tipo fuente:** Indica el formato o tipo de archivo, generalmente **Excel**.
- **Columnas:** Lista de columnas disponibles dentro del archivo cargado.

Nueva fuente de dato	Nombre	Descripción	Tipo fuente	Columnas
▼ Categoría: Fuentes de Datos locales				
	Trabajadores	Probando	excel	<a href="#">id, nombre, puesto</a>

Figura 66: Ejemplo de fuente de datos cargada con sus columnas identificadas.

Esta funcionalidad facilita la integración de información proveniente de archivos externos, permitiendo su reutilización en distintos formularios mediante los campos de tipo *Dataset*. De esta manera, se optimiza la gestión de datos y se mejora la coherencia entre las distintas partes del sistema.

### Crear una nueva fuente de datos

En la parte superior izquierda de la vista se encuentra el botón “**Nueva fuente de dato**”, identificado con un ícono de documento y un símbolo de “+”. Al hacer clic sobre este botón, se abrirá una ventana de adición de fuente de datos donde el usuario podrá registrar un nuevo archivo en el sistema (Figura ??).

Esta interfaz está diseñada para facilitar la carga de archivos externos, como hojas de cálculo en formato **Excel** (.xlsx), los cuales el sistema procesará para generar un conjunto de datos interno reutilizable en diferentes módulos.

En esta ventana, el usuario debe completar algunos campos básicos:

- **Nombre:** Campo obligatorio que permite identificar la fuente dentro del sistema (por ejemplo, *Trabajadores* o *Clientes*).
- **Descripción:** Campo opcional donde se puede agregar una breve explicación del contenido o propósito de los datos.
- **Tipo fuente:** Campo obligatorio que define el tipo de archivo a cargar; actualmente el formato soportado es **Excel**.

Una vez completados los campos requeridos, se puede arrastrar el archivo directamente al área de carga o hacer clic para seleccionarlo manualmente. El sistema validará el formato del archivo y, al presionar el botón **Guardar**, procesará la información para incorporarla al catálogo de fuentes de datos disponibles.

### 12.0.8. Procesos de Exportación

En esta sección es posible visualizar todos los formularios que han sido completados, junto con la cantidad de respuestas registradas en cada uno.

#### Exportar un formularios

Desde aquí se pueden exportar formularios individuales o descargar todos los formularios en el formato que se prefiera (XLSX, CSV o JSON). La interfaz incluye una barra de búsqueda que permite filtrar formularios por nombre o identificador, así como botones de exportación para generar los archivos descargables.

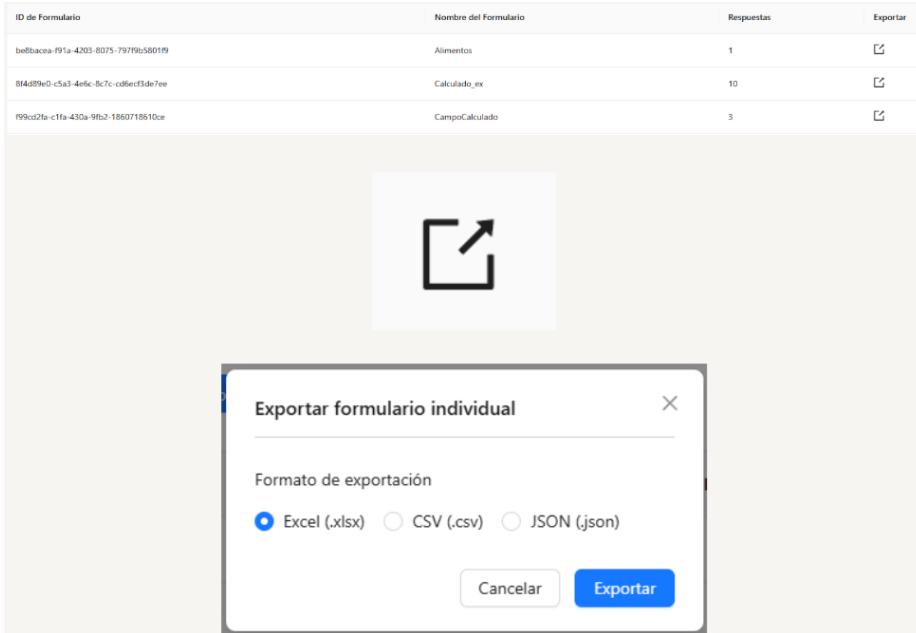


Figura 67: Vista del proceso de exportación de formularios: listado de formularios, botón de exportar y ventana de selección de formato.

**Exportar un formulario individualmente:** Cada registro dentro de la tabla de formularios completados cuenta con un botón de exportación ubicado en la columna final. Al hacer clic en este botón, se abrirá una ventana emergente que permite seleccionar el formato de salida para el formulario (Excel (.xlsx), CSV o JSON). Una vez elegido el formato, basta con presionar el botón **Exportar** para generar y descargar el archivo correspondiente.

Esta funcionalidad facilita el respaldo y análisis de la información recopilada, permitiendo exportar los datos en formatos compatibles con diferentes herramientas de procesamiento y análisis.

### Exportar todos los formularios

También es posible realizar una exportación masiva de los formularios completados mediante el botón “**Exportar todo**”, disponible en la parte superior de la sección de exportación. Esta opción permite descargar simultáneamente todos los formularios registrados en el sistema, facilitando la generación de respaldos o reportes globales.



Figura 68: Opción para exportar todos los formularios y ventana modal de selección de formato.

Al hacer clic sobre el botón, se abrirá una ventana modal donde el usuario podrá seleccionar el formato en el que desea exportar los datos. Las opciones disponibles son:

- Excel (.xlsx)
- CSV
- JSON

Una vez elegido el formato, bastará con presionar el botón **Exportar Todo** para que el sistema genere el archivo correspondiente con toda la información consolidada de los formularios completados. Esta función es especialmente útil para obtener una copia completa de los datos del sistema o integrarlos en otros procesos de análisis externo.

#### 12.0.9. Sesión: Usuarios

La sección **Usuarios** permite administrar las cuentas registradas dentro del sistema. Desde esta vista, los administradores pueden crear, editar, activar o eliminar usuarios, así como visualizar la información básica de cada uno.

The screenshot shows a user management interface with a search bar at the top containing placeholder text "Introduzca el texto a buscar...". Below the search bar is a table with the following columns: "Nuevo usuario" (User icon), "Nombre" (Name), "Contraseña" (Password), "Activo" (Active), and "Email". The table lists seven users:

Nuevo usuario	Nombre	Contraseña	Activo	Email
<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	danielv	*****	<input checked="" type="checkbox"/>	danielV@gmail.com
<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	dahernandez	*****	<input checked="" type="checkbox"/>	dahernandez@gmail.com
<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	SolanoOrozco	*****		sol21212@uvg.edu.gt
<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	HectorV10	*****	<input checked="" type="checkbox"/>	hvasquez@gmail.com
<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	lindain1	*****	<input checked="" type="checkbox"/>	user1@example.com
<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	lindain2	*****	<input checked="" type="checkbox"/>	user2@example.com
<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	lindain3	*****	<input checked="" type="checkbox"/>	user3@example.com

Figura 69: Vista general de la sección de usuarios.

La interfaz presenta una tabla organizada que muestra los principales datos de los usuarios, incluyendo su nombre de acceso, contraseña, estado activo y correo electrónico asociado. En la parte superior se encuentra una barra de búsqueda que facilita localizar rápidamente a un usuario específico mediante texto libre.

En la parte izquierda de la tabla se ubica el botón “**Nuevo usuario**”, identificado con un ícono de persona, el cual permite registrar nuevas cuentas dentro del sistema. Junto a cada usuario listado se muestran íconos de acción que permiten **editar** la información existente o **eliminar** el registrado.

## Edición de Usuario

La ventana de **Edición de Usuario** permite registrar o actualizar la información de los usuarios del sistema. En este formulario se deben completar los campos obligatorios, como *nombre completo*, *nombre de usuario*, *contraseña* y *correo electrónico*, los cuales identifican y permiten el acceso del usuario a la aplicación.

The screenshot shows a modal dialog titled 'Edición de Usuario'. It contains the following fields:

- \* Nombre completo: Text input field
- \* Nombre de usuario: Text input field
- \* Contraseña: Password input field with a visibility icon
- Activo: A checkbox labeled 'Activo'
- Acceso a la web: A checkbox labeled 'Acceso a la web'
- \* Email: Text input field

At the bottom right are two buttons: 'Guardar' (in blue) and 'Cancelar'.

La ventana de Edición de Usuario permite registrar o actualizar la información de los usuarios del sistema. En este formulario se deben completar los campos obligatorios, como nombre completo, nombre de usuario, contraseña y correo electrónico, los cuales identifican y permiten el acceso del usuario a la aplicación. También incluye las opciones Activo, para habilitar o deshabilitar la cuenta, y Acceso a la web, que define si el usuario puede ingresar al entorno web. Una vez ingresada la información, se puede guardar para aplicar los cambios o cancelar para salir sin modificar los datos.

Figura 70: Ventana de edición de usuario con campos de registro y control de acceso.

Además, el formulario incluye las opciones **Activo**, que permite habilitar o deshabilitar la cuenta, y **Acceso a la web**, que define si el usuario tiene permisos para ingresar al entorno web.

Una vez completada la información, se puede presionar el botón **Guardar** para aplicar los cambios, o bien seleccionar **Cancelar** para salir sin modificar los datos. Esta ventana es utilizada tanto para la creación de nuevos usuarios como para la actualización de registros existentes dentro del sistema.

## Generación de Código QR

La sección de **Generación de Código QR** permite a los administradores gestionar los accesos de los usuarios mediante un segundo método de autenticación. A través de esta opción, el sistema genera un código QR único asociado al usuario, el cual puede ser escaneado para ingresar a la aplicación. Es importante destacar que estos códigos QR son de *uso único*, por lo que, una vez escaneado, no será posible reutilizarlo.

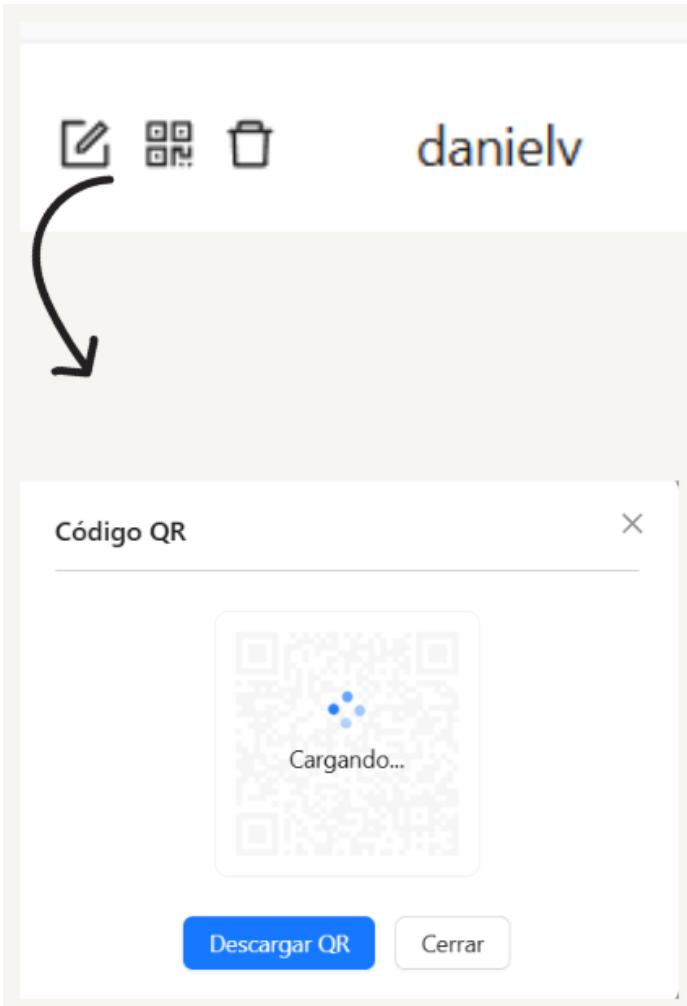


Figura 71: Ventana de generación de código QR para autenticación de usuario.

Durante el proceso, el sistema mostrará un cuadro de carga hasta completar la creación del código. Posteriormente, el administrador podrá **descargar el QR** o cerrar la ventana. Basta con esperar unos segundos para que el código QR esté disponible y pueda visualizarse correctamente.

### Descarga del Código QR

Una vez generado el código QR, el sistema mostrará una ventana emergente que permitirá al administrador **descargar la imagen del código QR** en formato PNG o cerrar la ventana, como se observa en la Figura 72. El botón *Descargar QR* facilita la obtención del archivo, el cual puede ser compartido posteriormente con los usuarios a través de medios electrónicos como correo electrónico o aplicaciones de mensajería instantánea (por ejemplo, Whatsapp).

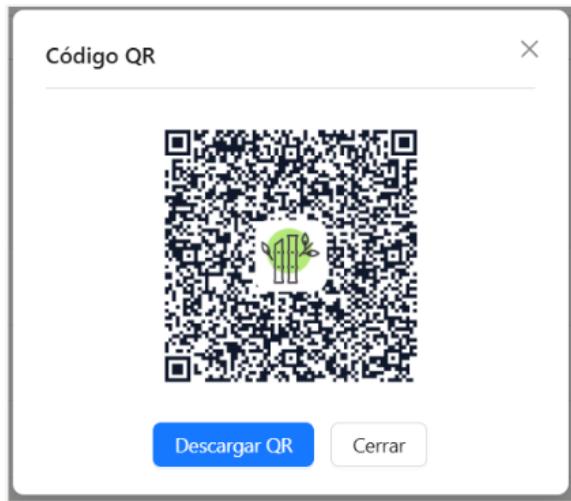


Figura 72: Ventana con la opción para descargar el código QR en formato PNG.

Al completar la descarga, el archivo se guarda localmente con un nombre identificador (por ejemplo, `qr-code-danielv.png`), tal como se muestra en la Figura 73.



Figura 73: Archivo del código QR descargado exitosamente.

## Eliminación de Usuarios

La opción de **Eliminación de Usuarios** permite al administrador dar de baja a los usuarios que ya no deban permanecer activos dentro del sistema. Esta funcionalidad resulta útil para mantener actualizada la base de datos y garantizar que únicamente los usuarios autorizados conserven acceso a la aplicación.

Dado que esta operación es irreversible, el sistema advierte que no se podrá deshacer una vez confirmada.



Figura 74: Ventana de confirmación para la eliminación de usuario.

Finalmente, al presionar el botón **Eliminar**, el registro del usuario será eliminado del sistema, impidiendo su acceso tanto a la aplicación web como a la aplicación móvil.

---

## Glosario

---

**Agricultura 4.0** Integración de tecnologías digitales (sensores, datos, automatización) en procesos agrícolas para optimizar decisiones.

**Agricultura de precisión** Uso de datos y herramientas tecnológicas para optimizar insumos y mejorar el manejo del cultivo.

**Backend** Capa encargada de la lógica de negocio, validaciones y acceso a datos.

**Base de datos local** Conjunto de datos cargados en el sistema para uso interno como Datasets.

**Dashboard** Vista de indicadores y datos agregados para toma de decisiones.

**Dataset** Archivo de datos (generalmente Excel) usado para llenar campos dinámicos.

**Despliegue (*deploy*)** Proceso de publicar la aplicación en un entorno de prueba o producción.

**Diseño responsive** Adaptación de la interfaz a diferentes pantallas y dispositivos.

**Formulario digital** Estructura electrónica con campos configurables y validaciones.

**Frontend** Capa visual del sistema con la que interactúa el usuario.

**Metadatos** Datos que describen otros datos (autor, fecha, estado, ubicación, etc.).

**Mockup** Prototipo visual de media o alta fidelidad.

**Sandbox** Entorno aislado para ejecutar validaciones o scripts de forma segura.

**Sincronización offline** Proceso de almacenar temporalmente datos sin conexión y enviarlos al restablecerla.

**Trazabilidad** Registro del ciclo de vida de los datos (creación, edición, responsable).

**Validación dinámica** Reglas evaluadas en tiempo real según condiciones o configuraciones variables.

**Versionado** Control del historial de cambios de formularios o código.

**WebSockets** Tecnología para comunicación bidireccional en tiempo real.

**Wireframe** Representación de baja fidelidad de una interfaz.