Tecnologías de la Información y las Comunicaciones

Red de Conocimiento en Informática, Diseño y Desarrollo de Software

Tecnología en Análisis y Desarrollo de Software

SISTEMA DE GESTION DE INVENTARIO PARA LA ORGANIZACIÓN DE LOS PROCESOS DE ENTRADA Y SALIDA DE HERRAMIENTAS O EQUIPOS DE TRABAJO PARA LA FINCA DEL SENA SEDE ZULIA (NORTE DE SANTANDER)

|  |  |
| --- | --- |
| Proyecto final presentado por: | CESAR EDUARDO DIAZ QUINTERO  FREDDY ANDRES ALBARRACÍN  QUINTERO  JEAN PIER LEANDRO GOMEZ RASCH  JURGUEN ALEXIS GALLO MEZA |
| Modalidad: Presencial | Proyecto Final Etapa Lectiva |
| Instructor Líder: | NOMBRE INSTRUCTOR LÍDER |
| Fecha: | 23 de abril 2025 |

[Tipo de letra: Calibri 10, 12, 14. Alineación justificado, paginación]

**Resumen Ejecutivo:**

Nuestro proyecto “Sistema de gestión de inventario para la organización de los procesos de entrada y salida de herramientas o equipos de trabajo para la finca del SENA sede Zulia (Norte de Santander)” busca ofrecer una solución tecnológica, practica y eficaz para resolver las deficiencias actuales en el control de inventario agrícola.

Actualmente, la gestión manual del ingreso y egreso de herramientas dificulta la organización de las actividades formativas. Al no cotar con un inventario actualizado y confiable, los instructores desconocen que herramientas están disponibles, lo que complica la planeación de la formación. Además, se dificulta verificar si las herramientas prestadas fueron devueltas correctamente o no.

El objetivo principal del aplicativo es sistematizar el control de herramientas, permitiendo registrar de forma ágil y segura las entradas y salidas mediante códigos QR, generar reportes de uso, enviar notificaciones automáticas, y ofrecer una vista en tiempo real del estado, ubicación (en bodega) y disponibilidad de cada herramienta. La solución estará disponible tanto para dispositivos móviles como de escritorio, y estará orientada a facilitar el trabajo de aprendices y instructores.

**Tabla de Contenido:** (Similar a un proyecto de grado tradicional) SE HACE AL FINALIZAR EL PROYECTO

**Lista de Figuras y Tablas:** (Similar a un proyecto de grado tradicional)

**Introducción:** (Adaptada al desarrollo de software)

El presente proyecto se desarrolla en el SENA sede Zulia, ubicada en el departamento de Norte de Santander. En este entorno, los aprendices realizan actividades formativas utilizando herramientas y equipos agrícolas. Sin embargo, el control actual del inventario de dichos elementos se lleva a cabo de forma manual, lo que genera dificultades en la organización, planificación y seguimiento de la formación. A partir de un análisis a los procesos, se idéntico la necesidad urgente de implementar una solución tecnológica que permita sistematizar estos procedimientos y mejorar la eficiencia operativa, con todo esto realizamos los diagramas de uso y de flujo que manejara este proyecto.

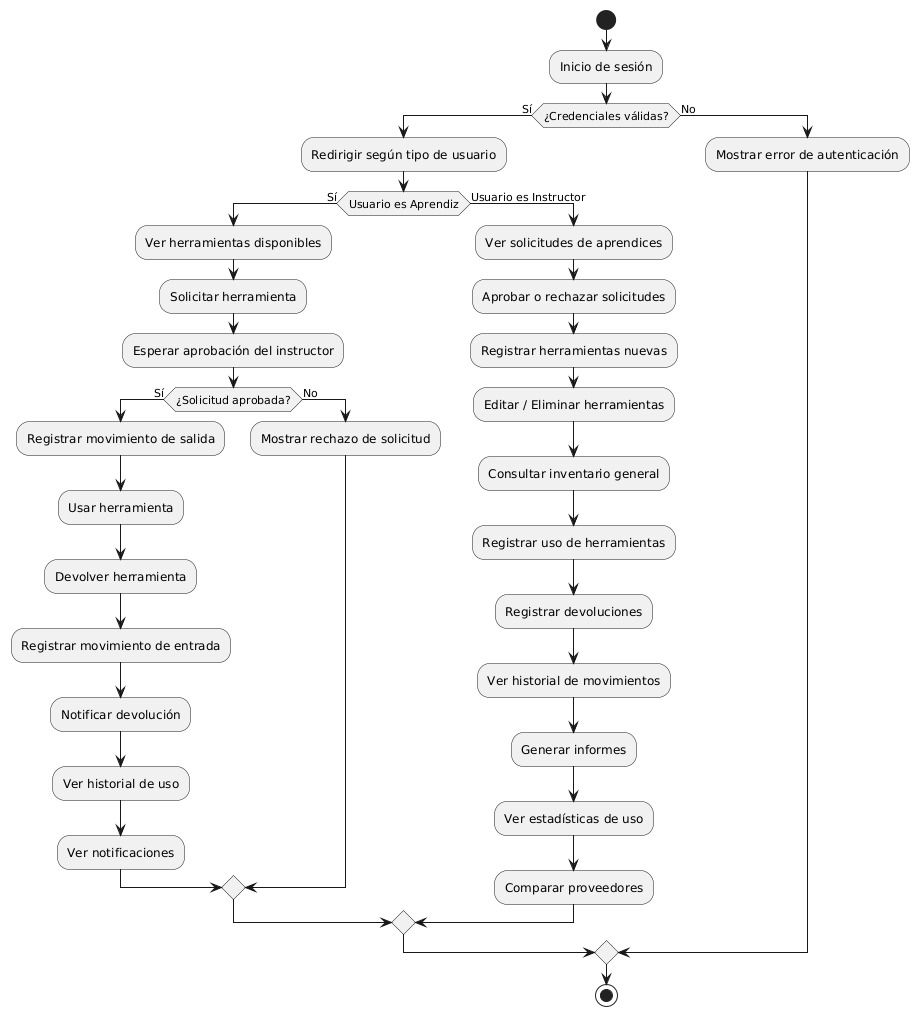


Imagen 1

Actualmente, se carece de un sistema que facilite una gestión efectiva del inventario. Esta limitación ha evidenciado dificultades en la planificación de las formaciones, en el seguimiento del uso de herramientas y en la verificación de su devolución. Los instructores no cuentan con información clara y en tiempo real sobre que herramientas están disponibles, en qué condiciones se encuentran o quien las tiene asignadas, lo que genera ineficiencias en los procesos formativos y administrativos.

Frente a esta necesidad, se propone el desarrollo de un sistema de gestión de inventario que permita automatizar los procesos de préstamo y devolución de herramientas, registrar movimientos mediante códigos QR, generar reportes detallados y ofrecer una visión general del estado del inventario. Esta solución permitirá mejorar la trazabilidad, reducir errores humanos y facilitar la toma de decisiones por parte de los diferentes actores del proceso formativo.

El objetivo general del proyecto es desarrollar un sistema de gestión de inventario que proporcione una solución práctica, eficiente y eficaz para la sistematización de los procesos de entrada y salida de herramientas en el SENA sede Zulia. Para lograrlo, se contempla el diseño de funcionalidades clave como el registro de herramientas, el escaneo mediante códigos QR, la visualización en tiempo real del inventario, el historial de uso por usuario, las notificaciones de devolución y la generación de informes periódicos.

El proyecto contempla el desarrollo de una aplicación de escritorio multiplataforma, diseñada para gestionar el inventario de herramientas utilizadas en la finca del SENA sede Zulia. Esta aplicación será operada exclusivamente por los instructores responsables del almacén y permitirá el registro de entrada y salida de herramientas mediante el uso de un lector físico de códigos QR.

Entre las funcionalidades que se desarrollarán se incluyen: el registro de herramientas y usuarios, la gestión de préstamos y devoluciones, la visualización en tiempo real del inventario, el historial de uso por usuario, la generación de alertas y notificaciones, y la creación de reportes periódicos en formatos visuales o imprimibles.

El sistema estará conectado a una base de datos MySQL, donde se almacenará toda la información relacionada con los movimientos del inventario. Asimismo, se diseñará una interfaz intuitiva y fácil de usar, desarrollada con Electron, HTML, CSS y JavaScript, que permita una interacción fluida entre el instructor y el sistema.

**Análisis de Requisitos:** Este es un apartado crucial en proyectos de software.

El sistema está dirigido a los usuarios que forman parte del entorno educativo y operativo del SENA sede Zulia, específicamente instructores y aprendices.

Los instructores requieren una herramienta que les permita planificar su formación con base en la disponibilidad de herramientas, registrar observaciones del uso, y llevar control sobre las devoluciones, por su parte, los aprendices necesitan acceder a herramientas funcionales y disponibles para sus actividades formativas, consultar su historial de préstamos, reportar fallos o daños, y recibir notificaciones de fechas de devolución, ambos perfiles requieren una interfaz intuitiva y ágil, accesible desde dispositivos móviles o de escritorio, que les permita interactuar con el sistema de forma eficiente.

Requisitos Funcionales

|  |  |
| --- | --- |
| ID del requerimiento | RF-01 |
| Nombre del requerimiento | Inicio de Sesión |
| Descripción | El sistema deberá exigir al usuario sus correspondientes credenciales para poder acceder al menú principal |
| Prioridad | Alta |
| ID del requerimiento | RF-02 |
| Nombre del requerimiento | Registro de Usuario |
| Descripción | El sistema debe permitir el ingreso de nuevos usuarios a través de una toma de datos personales para crear su usuario |
| Prioridad | Alta |
| ID del requerimiento | RF-03 |
| Nombre del requerimiento | Registro de herramientas |
| Descripción | El sistema debe permitir el ingreso de herramientas para generar su respectivo código en el sistema de inventario  RF-03.1  Editar algún dato de la herramienta (Alta) |
| Prioridad | Alta |
| ID del requerimiento | RF-04 |
| Nombre del requerimiento | Entrada y Salida de Herramientas |
| Descripción | El sistema debe permitir visualizar toda la información de entrada y salida registrada para cada herramienta  RF-04.1  Búsqueda por nombre o tipo de herramienta (Ideal)  RF-04.2  Vista de estadísticas de uso (Ideal) |
| Prioridad | Alta |
| ID del requerimiento | RF-05 |
| Nombre del requerimiento | Inventario |
| Descripción | El sistema debe de visualizar todas las herramientas disponibles junto a la información única de cada herramienta  RF-05.1  El sistema debe permitir alterar los registros (Alta)  RF-05.2  Barra de búsqueda (ideal) |
| Prioridad | Alta |
| ID del requerimiento | RF-06 |
| Nombre del requerimiento | Informes |
| Descripción | El sistema debe contar con un apartado para visualizar informes de las herramientas  RF-06.1  Capacidad para generar nuevos informes (Alta)  RF-06.2  Historial de informes (Alta)  RF—06.3  Capacidad para editar los informes (Alta)  RF-06.4  Barra de búsqueda (Ideal) |
| Prioridad | Alta |
| ID del requerimiento | RF-07 |
| Nombre del requerimiento | Notificaciones |
| Descripción | El sistema debe contar con un sistema de alertas que permita estar al tanto de cualquier modificación o alteración dentro de los otros módulos  RF-07.1  Deberá contar con alertas cuando no esté disponible una herramienta ya sea por su estado o por falta de stock (Ideal) |
| Prioridad | Alta |

Requisitos No Funcionales

|  |  |
| --- | --- |
| Nombre – Código | Descripción |
| Usabilidad | |
| RNF1 – Diseño de la aplicación | El sistema contará con una interfaz intuitiva y fácil de usar tanto en dispositivos móviles como en escritorio. |
| RNF2 – Modo Oscuro | El sistema debe permitir al usuario cambiar entre un modo claro y un modo oscuro en la interfaz gráfica, con el fin de mejorar la accesibilidad y comodidad visual, especialmente en ambientes con poca iluminación. |
| RNF3 – Mensajes de error | El sistema debe mostrar mensajes claros y específicos cuando ocurra un error (ej. QR no reconocido, herramienta no disponible, etc.). |
| Confiabilidad | |
| RNF5 – Disponibilidad de la aplicación | El sistema debe estar disponible al menos el 99% del tiempo durante horas operativas del centro. |
| RNF6 – Tolerancia a fallos leves | La aplicación debe continuar operando, aunque haya interrupciones leves en la red, almacenando datos temporalmente. |
| Seguridad | |
| RNF7 – Permisos de usuario | El sistema debe manejar roles (aprendiz, instructor, administrador) con diferentes permisos. |
| Eficiencia | |
| RNF8 - Tiempos de respuesta | El sistema debe procesar el escaneo de un código QR y registrar la acción en menos de 2 segundos. |

* + - Restricciones: Limitaciones técnicas, de presupuesto, de tiempo o de recursos que afectan el desarrollo del software.

**Diseño del Sistema:** Cómo se construirá el software para cumplir con los requisitos.

Qué debe tener:

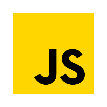
* + - Arquitectura del Software: La estructura general del sistema, los componentes principales y cómo interactúan entre sí (diagramas de arquitectura).
    - Matriz de riesgos: Estructura el cual indica la clasificación de los riesgos (Norma ISO 27001 y 27002)
    - Diseño de la Base de Datos: El modelo de datos, las tablas, las relaciones y la forma en que se almacenará la información (diagrama entidad-relación).
    - Diseño de la Interfaz de Usuario (UI): Cómo interactuará el usuario con el software (wireframes, mockups, prototipos de baja fidelidad).
    - Diseño de la Experiencia de Usuario (UX): La planificación para que la interacción del usuario sea intuitiva, stakeholders.
    - Diseño de los Módulos/Componentes: El diseño detallado de cada parte del software.
    - Tecnologías a Utilizar: Lenguajes de programación, frameworks, bases de datos, herramientas y plataformas que se emplearán.

**Electron (Framework principal)**

Electron es un framework de código abierto que permite desarrollar aplicaciones de escritorio multiplataforma utilizando tecnologías web como HTML, CSS y JavaScript.

¿Por qué usarlo?

* Permite desarrollar una sola aplicación compatible con Windows, Linux y macOS.
* Facilita la creación de una interfaz gráfica moderna e intuitiva.
* Es compatible con tecnologías backend y bases de datos, ideal para proyectos híbridos como este.
* Permite integración directa con hardware como lectores de códigos QR.



**JavaScript (Lógica de la aplicación)**

Lenguaje principal para manejar la lógica, eventos y procesamiento interno dentro del entorno Electron.

¿Por qué usarlo?

* Su integración con Electron es nativa.
* Permite desarrollar de manera ágil y con acceso a múltiples librerías útiles.
* Facilita la comunicación con la base de datos y la gestión del sistema de inventario.



**HTML y CSS (Interfaz gráfica del usuario - GUI)**

Lenguajes utilizados para estructurar (HTML) y estilizar (CSS) la interfaz del usuario.

¿Por qué usarlos?

* Permiten diseñar una interfaz visual atractiva, responsiva y personalizada.
* Ofrecen alta flexibilidad para implementar botones, formularios, tablas y componentes gráficos para el inventario.

**MySQL (Base de datos relacional)**

MySQL es un sistema de gestión de bases de datos relacional muy utilizado a nivel empresarial y educativo.

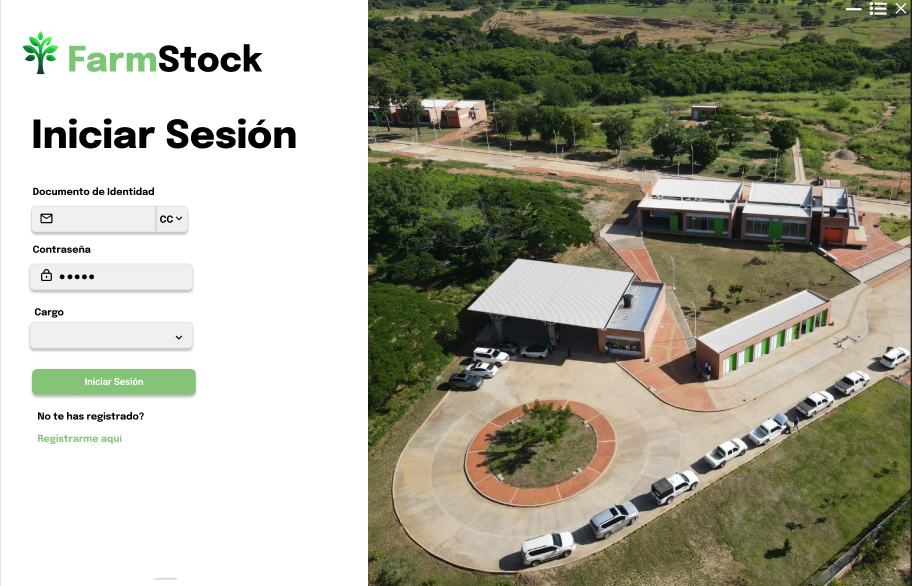
¿Por qué usarlo?

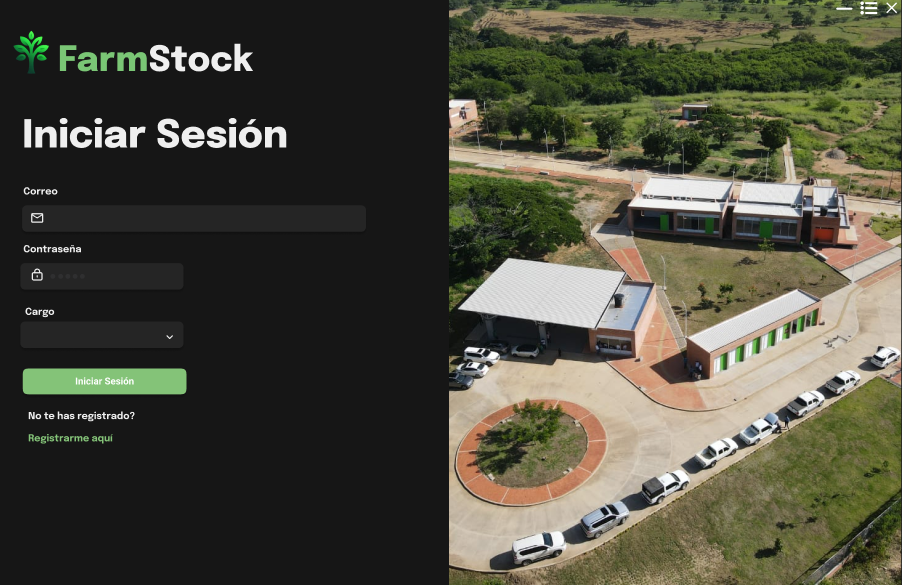
* Permite manejar grandes volúmenes de datos con integridad y eficiencia.
* Ideal para aplicaciones que requieren consultas complejas, integridad referencial y múltiples usuarios.
* Facilita la escalabilidad del proyecto, en caso de que el sistema deba integrarse en red o accederse desde varios dispositivos en el futuro.

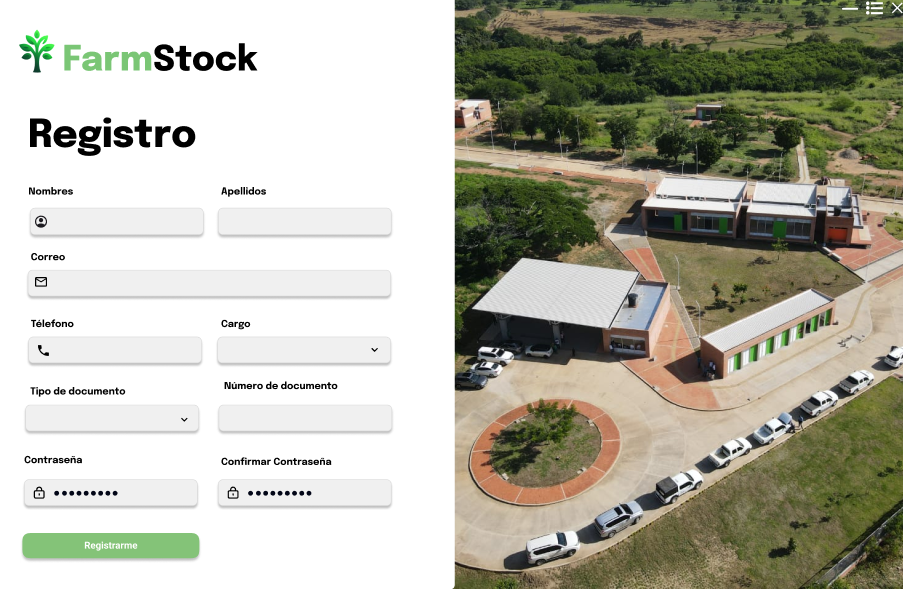
**Diseño de la interfaz de usuario(UI)**

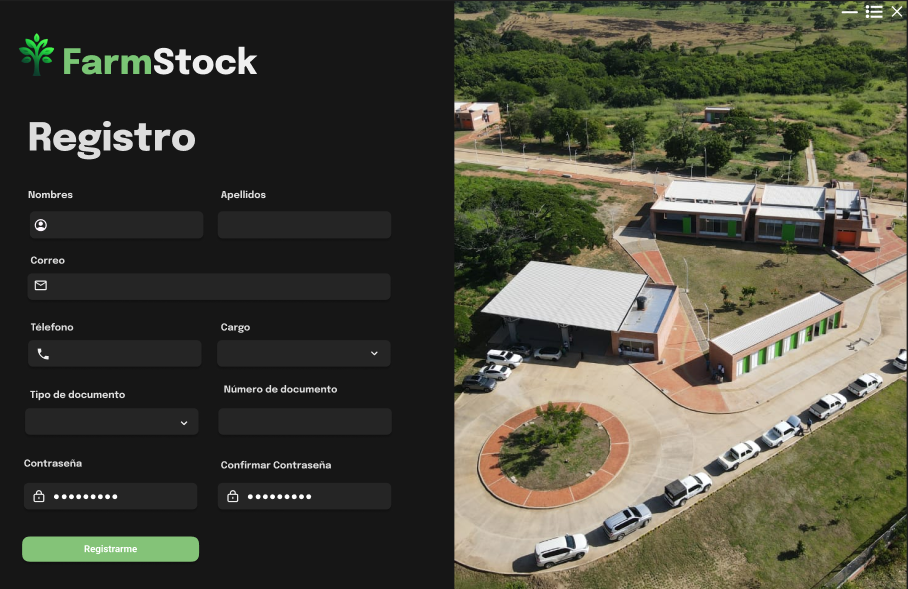
Los prototipos de interfaz permiten visualizar la estructura, flujo y diseño de la aplicación antes de su implementación. En este proyecto, los prototipos fueron diseñados utilizando Figma, siguiendo principios de usabilidad y accesibilidad. Se incluye la vista principal, formularios de registro, visualización de inventario, historial y modo oscuro.

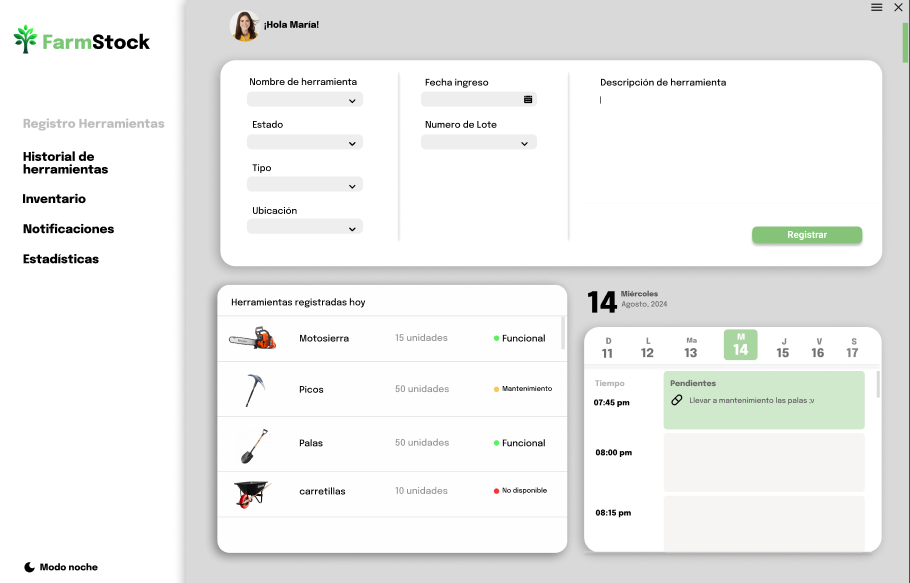
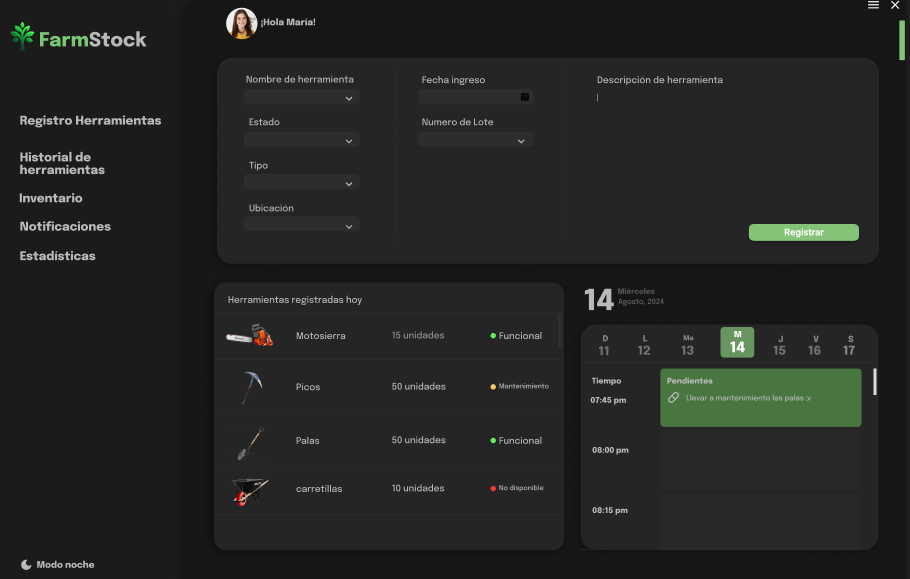
A continuación, se presentan los prototipos estáticos elaborados en Figma, junto con una breve descripción de cada vista.

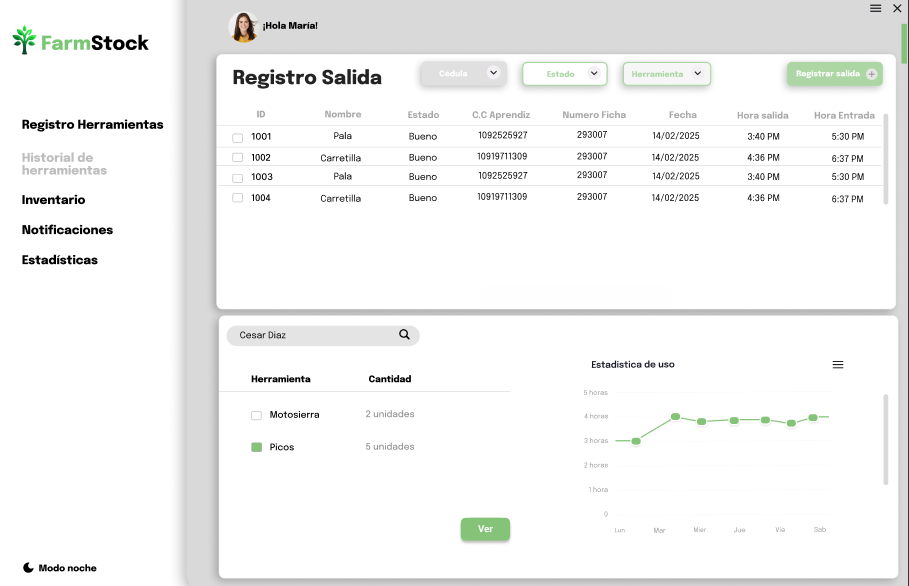
**Frame 1: Inicio de sesión**

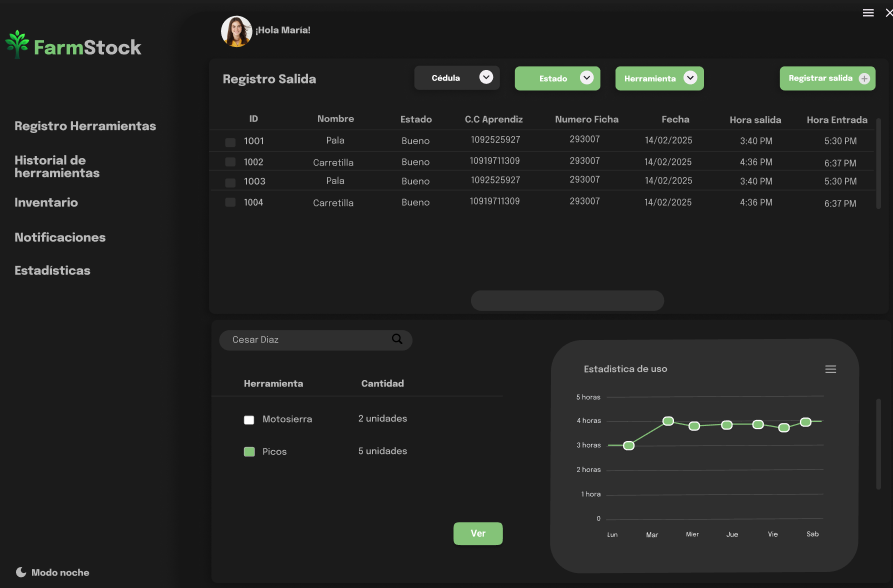
****

**Frame 2: Registro**

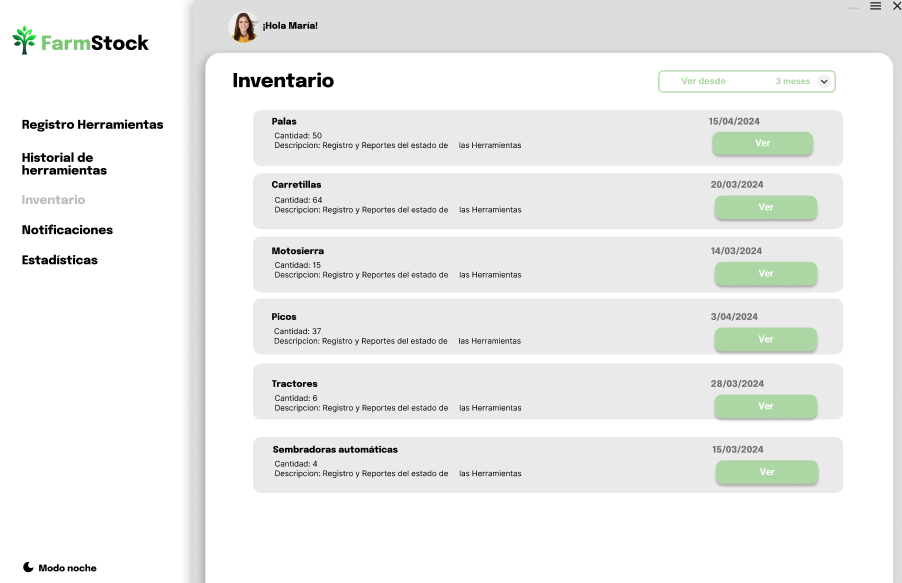
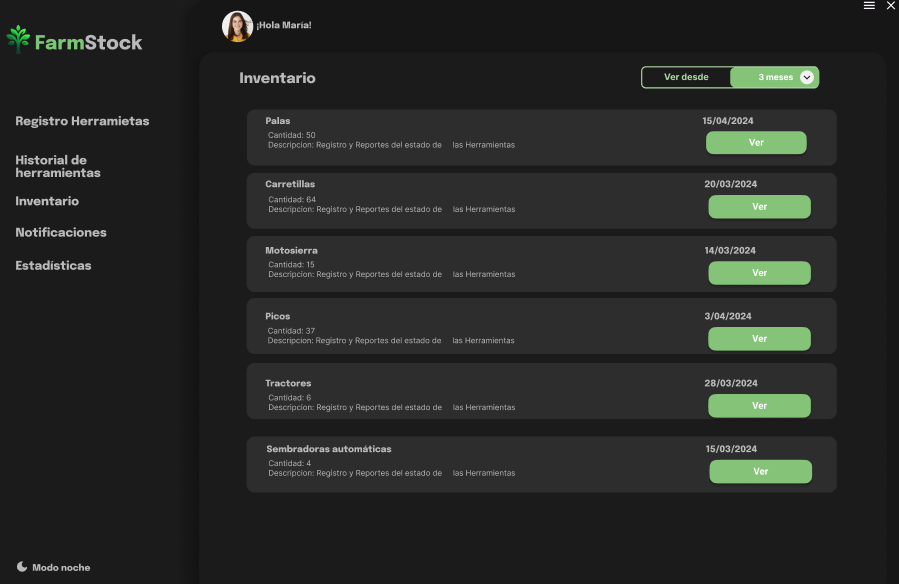
****

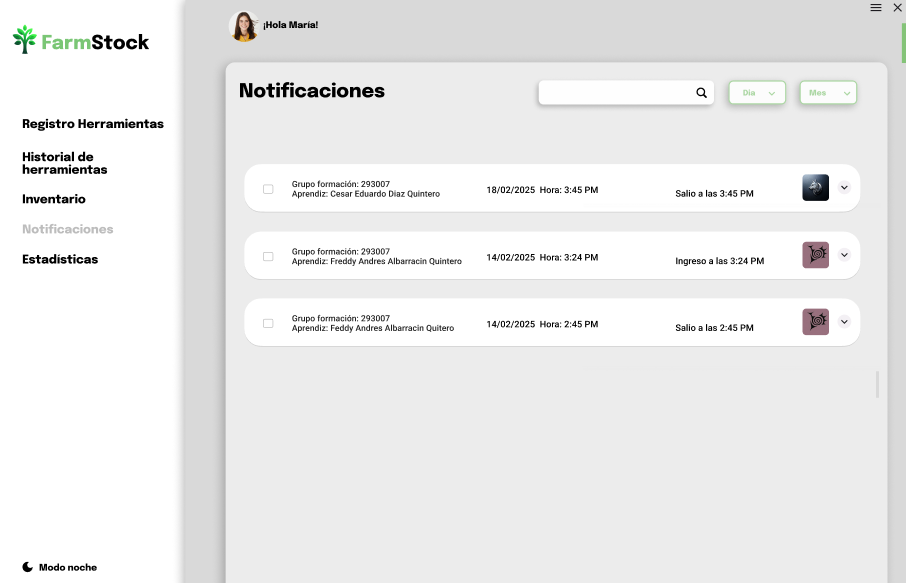
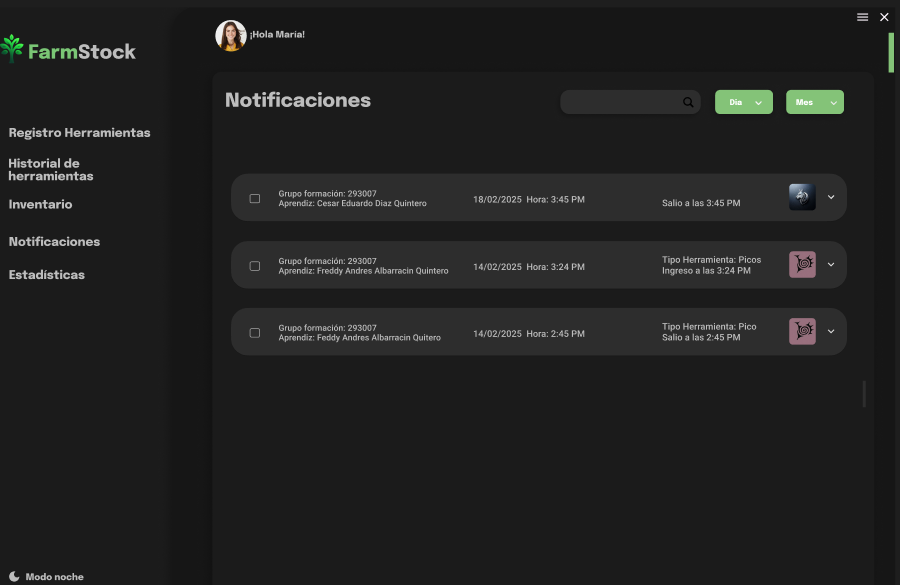
**Frame 3: Registro de herramienta**

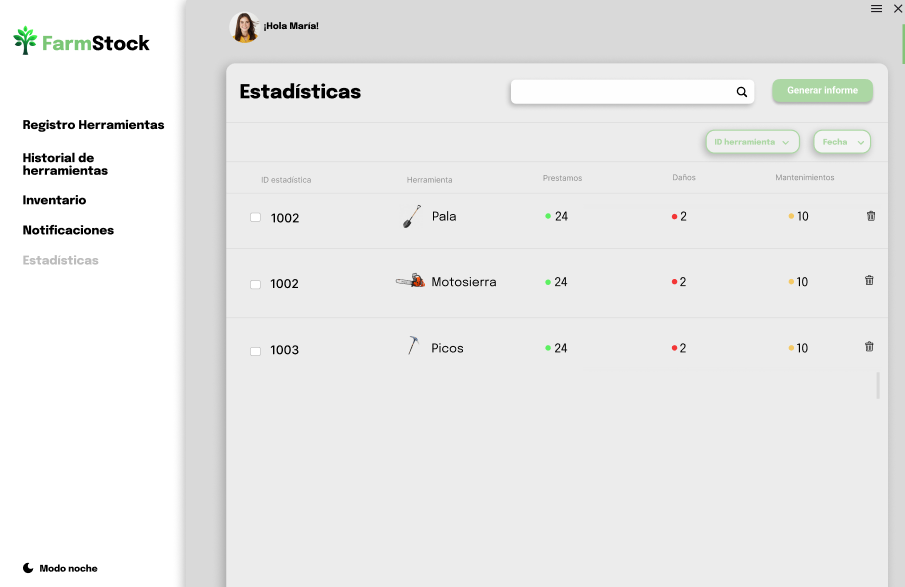
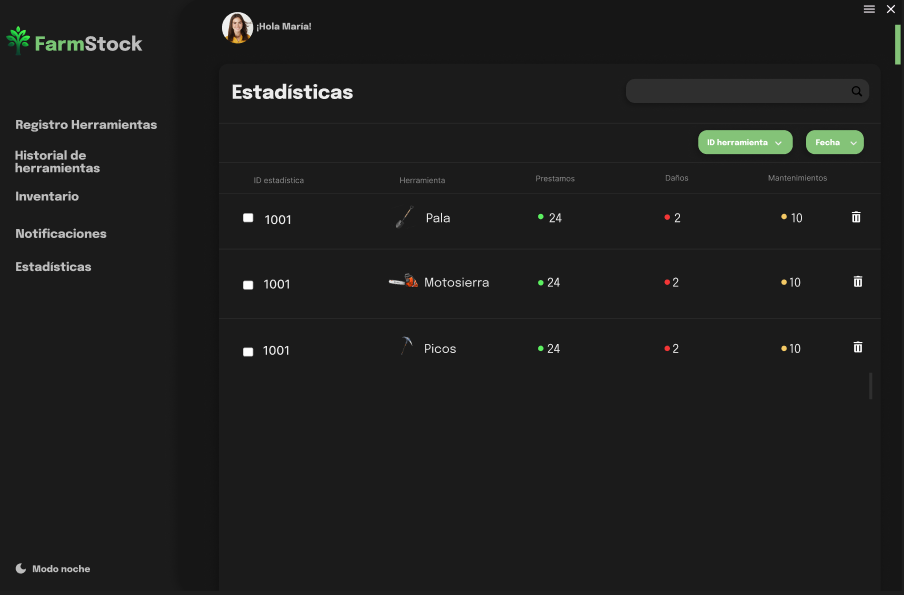
**Frame 4: Historial de herramientas**

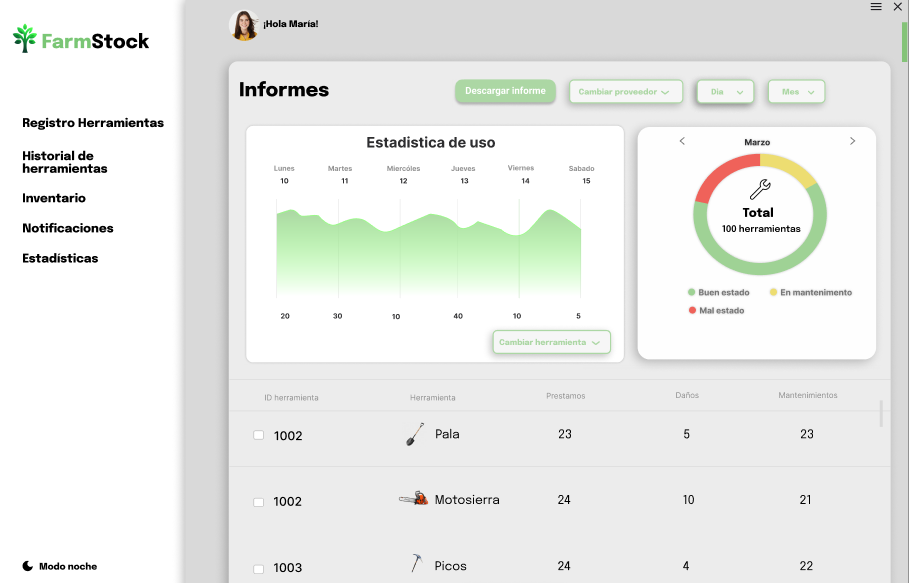
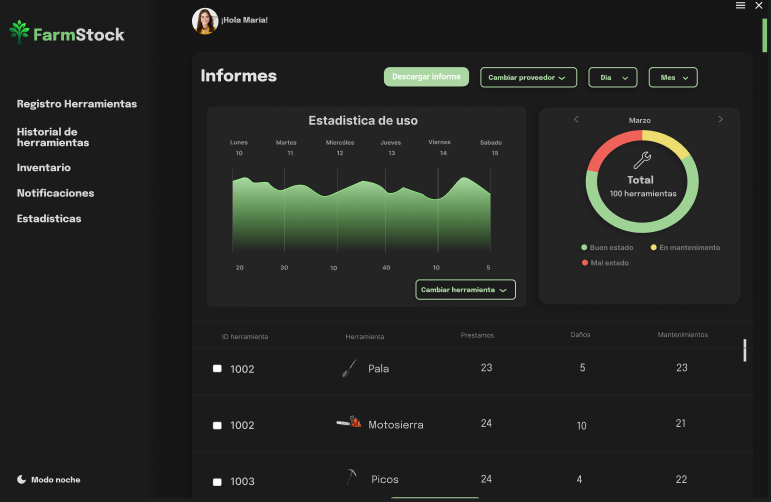


**Frame 5: Inventario**



**Frame 6: notificaciones**

**Frame 7: Estadísticas**

**Frame 7.1: Informes**

**Implementación:** La fase de codificación y construcción del software.

Qué debe tener (en el documento):

* + - Descripción del Entorno de Desarrollo: Las herramientas y configuraciones utilizadas para la programación.
    - Descripción General de la Implementación: Un resumen de cómo se llevó a cabo la construcción del software, los módulos desarrollados y las tecnologías aplicadas.
    - Ejemplos de Código (opcional y breve): Fragmentos de código relevantes para ilustrar aspectos importantes de la implementación.
    - Estrategia de Pruebas: Un plan general de cómo se probará el software.

**Pruebas:** La verificación de que el software funciona correctamente y cumple con los requisitos.

Qué debe tener:

* + - Plan de Pruebas: Los tipos de pruebas que se realizarán (unitarias, de integración, de sistema, de aceptación), los casos de prueba y los criterios de éxito.
    - Resultados de las Pruebas: Un registro de las pruebas realizadas, los errores encontrados y cómo se corrigieron.
    - Informe de Pruebas: Un resumen de la calidad del software después de las pruebas.

**Despliegue** (opcional, dependiendo del alcance del proyecto): Cómo se pondrá en funcionamiento el software.

Qué debe tener:

* + - Plan de Despliegue: Los pasos para instalar y configurar el software en el entorno de producción.
    - Requisitos del Entorno de Despliegue: Especificaciones del servidor, sistema operativo, etc.
    - Procedimientos de Instalación y Configuración: Instrucciones detalladas.

**Resultados y Discusión:**

Qué debe tener:

* + - Presentación de los Resultados del Desarrollo: Un resumen de lo que se logró construir, las funcionalidades implementadas y el estado del software.
    - Análisis del Cumplimiento de los Objetivos: Evaluar si se alcanzaron los objetivos planteados inicialmente.
    - Discusión de los Retos y Soluciones: Los problemas que surgieron durante el desarrollo y cómo se abordaron.
    - Evaluación del Software: Una valoración del software desarrollado en relación con los requisitos y los criterios de calidad.

**Conclusiones:**

Qué debe tener: Un resumen de los logros del proyecto, las lecciones aprendidas y una reflexión sobre el proceso de desarrollo.

**Trabajo Futuro:**

Qué debe tener: Sugerencias para futuras mejoras, funcionalidades adicionales o líneas de desarrollo posteriores.

III. Partes Finales:

**Referencias Bibliográficas:** (Si se utilizaron fuentes externas para el marco teórico o la investigación).

**Glosario:** Definición de términos técnicos específicos utilizados en el proyecto.

**Anexos:** Diagramas detallados, especificaciones técnicas extensas, manuales de usuario (si se desarrollaron), etc.