



PRODUCTO TÉCNICO PEDAGÓGICO

SOFTWARE DIDÁCTICO PARA LA GESTIÓN DE LA INFORMACIÓN DE LA PRODUCCIÓN GANADERA
EN LA FINCA VILLA LUZ DEL CENTRO DE GESTIÓN AGROEMPRESARIAL DEL ORIENTE (CGAO)

PRESENTADO POR:

MIGUEL ÁNGEL TEJEDOR MENDOZA INSTRUCTOR PROGRAMACIÓN DE SOFTWARE G 11

EDUAR JAVIER CAMACHO VENEGAS INSTRUCTOR ESPECIES MAYORES G 15

SERVICIO NACIONAL DE APRENDIZAJE SENA

REGIONAL SANTANDER

CENTRO DE GESTIÓN AGROEMPRESARIAL DEL ORIENTE

VÉLEZ, SANTANDER DICIEMBRE DE 2025



1. Resumen ejecutivo:

El presente documento expone el diseño, desarrollo, implementación y uso pedagógico del Software Didáctico para la Gestión de la Información de la Producción Ganadera de la Finca Villa Luz, un producto técnico–pedagógico creado de manera interdisciplinaria entre los programas de Producción Pecuaria y Análisis y Desarrollo de Software (ADSO) del Centro de Gestión Agroempresarial del Oriente (CGAO) del SENA.

A pesar de la importancia de la digitalización en los sistemas productivos pecuarios, la finca Villa Luz —principal ambiente de formación agropecuario del CGAO— presenta una limitada disponibilidad de software especializado, lo que afecta dos áreas críticas:

La formación de aprendices pecuarios, quienes no adquieren habilidades en herramientas de gestión digital y la gestión técnica de la finca, que depende de registros manuales y herramientas no integradas, dificultando decisiones confiables en tiempo real. La ausencia de una herramienta propia representa una oportunidad para articular formación y tecnología, fortaleciendo competencias técnicas y respondiendo a necesidades reales del sector ganadero regional y nacional.

El producto técnico pedagógico propone desarrollar un software didáctico, escalable y seguro que fortalezca competencias de los aprendices de ADSO mediante la construcción de un sistema desde una problemática real, además mejore las capacidades de registro, análisis y trazabilidad en los programas pecuarios optimizando la toma de decisiones técnicas en la finca Villa Luz mediante datos confiables y centralizados.

En cuanto a la arquitectura del Software el sistema se implementa bajo un enfoque cliente–servidor con tecnologías modernas y de uso extendido en la industria:



Frontend: React + Vite + TypeScript (SPA responsiva, menús contextuales, dashboard, búsqueda offline).

Backend: Python + Flask + SQLAlchemy (API REST modular, seguridad con JWT, validaciones estrictas).

Base de datos: MySQL, normalizada hasta 3FN.

Despliegue: Contenedores Docker mediante Coolify, certificados SSL automáticos y CI/CD desde GitHub.

A través de sus módulos, el sistema permite gestionar animales, razas, sanidad, vacunación, tratamientos, genética, potreros, alimentación y usuarios con control de roles y permisos.

Las funcionalidades Principales ofrecen un panorama completo de todas las acciones productivas y administrativas, registro técnico detallado (salud, alimentación, pastoreo, genética), dashboard analítico con más de 30 indicadores productivos con un respaldo de seguridad robusta: autenticación JWT + cookies HttpOnly , además permite la búsqueda global offline con IndexedDB y validaciones que garantizan integridad referencial y calidad de datos.

En cuanto a la estrategia pedagógica el proyecto se fundamenta en el Modelo Pedagógico Institucional del SENA (MPI) permite aplicar el aprendizaje Basado en Proyectos (ABP), Trabajo colaborativo interdisciplinario (ADSO + Pecuaria) y procura el aprendizaje significativo mediante solución de problemas reales usando estrategias activas como la simulación, estudios de caso, mapas conceptuales, seminarios investigativos. De esta forma el software se convierte en un instrumento formativo, permitiendo ejercitar competencias técnicas, tecnológicas y de análisis de datos en contextos reales de producción.

Para la Validación y Ajustes se realizaron pruebas piloto con instructores y aprendices del área pecuaria, verificando usabilidad y pertinencia técnica, también se trabajó con personal de la finca Villa Luz, garantizando coherencia con procesos productivos reales y los resultados permitieron mejorar formularios, flujo de navegación, indicadores y reportes.



Se espera que el Impacto en la formación sea a Más de 340 aprendices tecnólogos y técnicos beneficiados en una primera fase y que los Aprendices de ADSO fortalecerán competencias en desarrollo real de software, arquitectura, APIs y despliegues profesionales, en cuanto a los aprendices del área pecuaria mejorarán habilidades en gestión digital, trazabilidad y análisis productivo.

En el sector productivo La finca Villa Luz contará con un sistema propio que permitirá decisiones basadas en datos, mejorando eficiencia y calidad del manejo ganadero y se espera impactar a productores rurales vinculados a CAMPESENA y a su vez los aprendices de articulación con la media podrán acceder a una herramienta gratuita, simple y con enfoque educativo.

En lo referente a propiedad Intelectual El software es una obra protegida conforme a la guía GIC-G-0008 del SENA y se distribuye bajo licencia MIT, permitiendo su uso, adaptación y redistribución con reconocimiento de autoría.

Este documento incorpora Manual de usuario, Manual técnico, Diagramas de arquitectura, Historial de usuario, Código fuente (backend y frontend), Guías de mantenimiento y actualización y enlaces a repositorios GitHub y tablero Kanban.

2. Justificación técnica y pedagógica: Pertinencia del software frente a los resultados de aprendizaje y necesidades del programa.

El manejo de datos en la producción pecuaria es una necesidad sentida en los sistemas de grandes, pequeños y medianos ganaderos ya que permite la toma de decisiones acertadas. A su vez la evolución en el uso de herramientas digitales ha llevado a disponer de una serie de recursos tanto en la producción como en la educación que dinamizan y optimizan los procesos, generando un impacto positivo en los sectores mencionados.

A pesar de la oferta variada en el mercado y de la contratación de softwares especializados por parte del Servicio Nacional de Aprendizaje para el desarrollo de la formación en los programas pertenecientes a la red pecuaria, en el Centro de Gestión Agroempresarial del Oriente especialmente en la granja Villa Luz la disponibilidad, uso e implementación de los mismo es escaso.



Lo anterior nos lleva a dos escenarios desfavorables, primero los aprendices quedan limitados en el acceso a herramientas de gestión de la producción ganadera propias (software especializado) afectando el alcance de las competencias relacionadas con el manejo de datos de forma digital y la calidad de la formación. Segundo la producción ganadera desarrollada en la granja Villa Luz se ve relegada al manejo de datos de forma manual o al uso de softwares externo o gratuitos que limitan la toma de decisiones en tiempo real y que no garantiza la seguridad de la información.

2.1 Delimitación del Tema y Justificación:

El Centro de Gestión Agroempresarial del Oriente cuenta con una sede principal ubicada en el municipio de Vélez, una subsele ubicada en el municipio de Cimitarra y una sede agropecuaria ubicada en la vereda Pozo Negro del municipio de Barbosa denominada Finca Vila Luz. En estas sedes se desarrollan programas relacionados con la producción pecuaria como son: Tecnólogo en producción ganadera, tecnólogo en gestión de empresas pecuarias, tecnólogo en gestión Agroempresarial, técnico en producción agropecuaria, técnico en mayordomía de empresas ganaderas, técnico en producción pecuaria y profundización técnica en producción ganadera bovina intensiva. Estos programas tienen la gran ventaja que sus aprendices pueden afianzar las competencias en la sede Villa Luz donde se cuenta con un escenario real de aprendizaje ya que allí existen sistemas de producción avícola, piscícola, producción de ovinos y de ganadería bovina.

Es de resaltar que todos los programas mencionados anteriormente se incluyen en las diferentes ofertas del centro durante el transcurso del año (Ver tabla 1).

En ese mismo sentido El Centro de formación también ofrece el programa de tecnólogo en análisis y desarrollo de software (ADSO) el cual busca formar aprendices que se desempeñen desarrollando soluciones de software a través de contratos de prestación de servicios en empresas públicas o privadas en el sector TIC. De igual forma puede participar como analista, diseñador y programador de software en proyectos específicos de acuerdo con la necesidad del servicio.

De acuerdo con lo anterior para la ejecución de los programas de formación tanto del área pecuaria como en el área de desarrollo de software, el centro realiza el desarrollo curricular basado en la formación por competencias y utilizando la estrategia metodológica del aprendizaje basado en proyectos, lo que permite que los aprendices se formen resolviendo problemas de la vida real generando un aprendizaje significativo y de esta forma dar respuesta a las necesidades del sector productivo.



En ese orden de ideas se plantea el desarrollo de este producto técnico pedagógico de manera interdisciplinaria (ADSO-Pecuaria) donde los instructores y aprendices jugaran un papel importante en la creación del software. Por otra parte se reconoce que la producción ganadera es una de las actividades productivas más importantes en el país tanto por su participación en el producto interno bruto –PIB- como por la población económicamente activa que ocupa (DANE, 2019). La Provincia de Vélez, en el Departamento de Santander, no es la excepción a esta constante dentro del conjunto del territorio nacional a la vez que conviene subrayar la ascendencia de los oficios pecuarios como parte esencial de la identidad de la región. (Camacho et al. 2024)

A pesar de la importancia económica del sector, el uso de herramientas tecnológicas de gestión en la ganadería es mínimo y en muchas ocasiones puede resultar de alto costo para los productores y Aunque se contara con el recurso para la inversión, se carece de personal capacitado para poner en marcha el monitoreo sistematizado de la ganadería.

Esta situación representa una oportunidad para que el gremio que asocia a estos productores haga la inversión en equipo y software, y le permita a un profesional calcular por medio de la herramienta, los indicadores técnicos y económicos para mejorar la toma de decisiones y la incorporación de un verdadero proceso administrativo. (Obregón et al. 2022)

La granja Villa Luz del CGAO no está exenta de la problemática anterior, por ejemplo el manejo de la información de los bovinos se realiza de forma manual o en el mejor de los casos en tablas en Excel que no contienen toda la información o que no permite la toma de decisiones acertadas en tiempo real, lo anterior toma mayor importancia si le sumamos que es un ambiente de formación especializado para los programas de la producción pecuaria donde se supone los aprendices se forman con las tecnologías de vanguardia que le permitan implementar soluciones en los sectores productivos

De acuerdo a lo expuesto existe la necesidad de desarrollar una herramienta para la gestión de la información de la producción bovina en la granja Villa Luz que permita a los aprendices del programa de análisis y desarrollo de software crear una solución a un problema real, a los aprendices de programas pecuarias aprender el uso e implementación del mismo adquiriendo competencias que les mejore su desempeño en el sector productivo y a la granja Villa Luz llevar un control más técnico de los datos de la ganadería presente allí para tomar decisiones acertadas en tiempo real.

**Oferta 2025 Centro de Gestión Agroempresarial
del Oriente**



Oferta	Nivel	Nombre del programa
1	TECNÓLOGO	PRODUCCIÓN GANADERA
2	TECNÓLOGO	GESTIÓN DE EMPRESAS PECUARIAS
2	TÉCNICO	PRODUCCIÓN AGROPECUARIA
2	TÉCNICO	MAYORDOMÍA DE EMPRESAS GANADERAS.
3	TECNÓLOGO	ANÁLISIS Y DESARROLLO DE SOFTWARE.
3	TECNÓLOGO	GESTIÓN AGROEMPRESARIAL
3	TECNÓLOGO	GESTIÓN DE EMPRESAS PECUARIAS
3	PROFUNDIZACIÓN TÉCNICA	PRODUCCIÓN GANADERA BOVINA INTENSIVA

Tabla 1. Programas ofrecidos relacionados con el producto técnico pedagógico. Fuente Administración educativa CGAO

3. Objetivos (general y específicos)

3.1 Objetivo general:

Desarrollar una herramienta de gestión didáctica para la producción ganadera de la granja Villa Luz del centro de gestión Agroempresarial del Oriente.

3.2 Objetivos específicos:



Crear un software que permita el desarrollo de competencias técnicas a los aprendices del programa de análisis y desarrollo de software del Centro de Gestión Agroempresarial del oriente (CGAO).

Fortalecer tres competencias técnicas a través del uso y la implantación de software especializado en los aprendices de programas relacionados con el sector pecuario del centro de Gestión Agroempresarial del Oriente (CGAO).

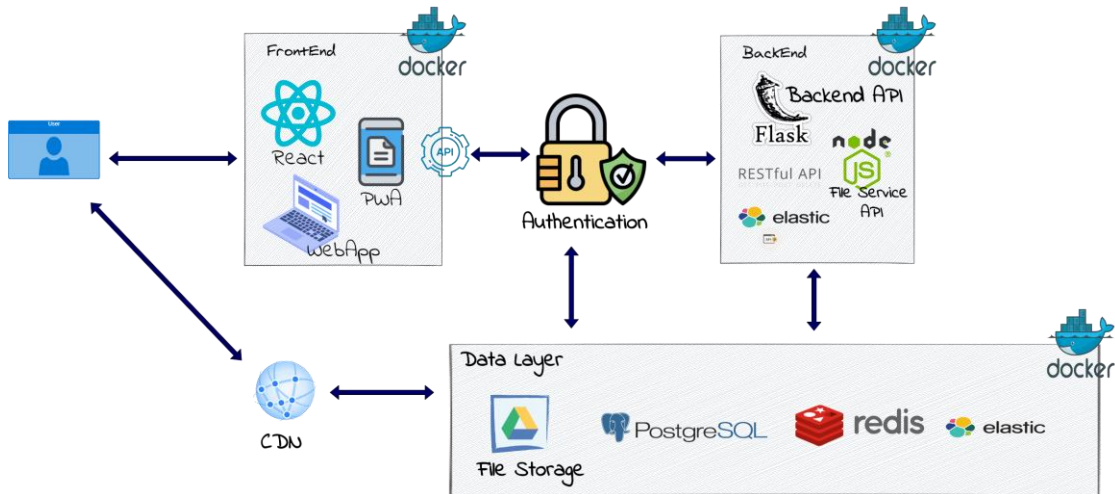
Mejorar el manejo de la información de un sistema de producción pecuario a través del uso de un software propio en la granja Villa Luz del CGAO.

4. Descripción técnica del software: Arquitectura, lenguaje de programación, requisitos, módulos y funcionamiento.

El software didáctico para la gestión de la producción ganadera de la Finca Villa Luz es una plataforma web desarrollada bajo una arquitectura moderna cliente-servidor, diseñada para fortalecer procesos formativos en ADSO y las áreas pecuarias del SENA. Se compone de un frontend interactivo, un backend robusto, una base de datos relacional y un entorno de despliegue contenerizado mediante Docker y Coolify.

El sistema facilita el registro, control, trazabilidad y análisis de información de animales, enfermedades, tratamientos, pastoreo, alimentos, mejoras genéticas y usuarios, apoyándose en principios de ingeniería de software y buenas prácticas de desarrollo seguro.

La arquitectura está basada en el modelo SPA (Single Page Application) + API REST, en donde:



4.1 Frontend (Cliente)

Construido con React + Vite + TypeScript.

Arquitectura SPA (Single Page Application): rápida, modular y responsiva.

Implementa autenticación basada en cookies HttpOnly.

Implementa:

Navegación por roles.

Menús de acciones contextuales (:).

Formularios y tablas dinámicas.

Dashboard analítico.

Búsqueda global offline con IndexedDB.

Se comunica con el backend a través de HTTP/HTTPS usando Axios.

4.2 Backend (Servidor)



Basado en Flask (Python) con una arquitectura modular por responsabilidades:

controllers/ → Controladores (capa de endpoints y lógica de flujo)

models/ → Modelos SQLAlchemy y relaciones

namespaces/ → Organización de módulos complejos (analytics, animales, usuario, etc.)

utils/ → Funciones auxiliares, API centralizada, respuestas estándar

templates/ → Plantillas Jinja2 para vistas mínimas o correos

api.py → Registro de rutas, blueprints y configuración central de la API

app/__init__.py → Inicialización del proyecto: DB, CORS, JWT, logs

migrations/ → Migraciones Alembic

instance/ → Configuraciones sensibles (DB URL, JWT secret)

certs/, static/, tools/, docs/ → Recursos del sistema

Provee una API RESTful organizada por módulos.

Contiene validaciones, reglas de negocio y acceso seguro a datos.

Utiliza SQLAlchemy como ORM para manejo de base de datos.

El backend es responsable de:

CRUD de todas las entidades.

Validaciones estrictas.

Integridad referencial preventiva.

Sistema de Analytics con más de 30 KPIs.

Seguridad y autenticación mediante JWT + Cookies HttpOnly.

Manejo de imágenes y archivos.



4.3 Base de Datos

Motor sugerido: MySQL (producción) o SQLite (desarrollo).

Normalizada hasta 3FN.

Maneja relaciones uno-a-muchos, muchos-a-muchos y referencias genealógicas.

4.4 Despliegue y Contenerización

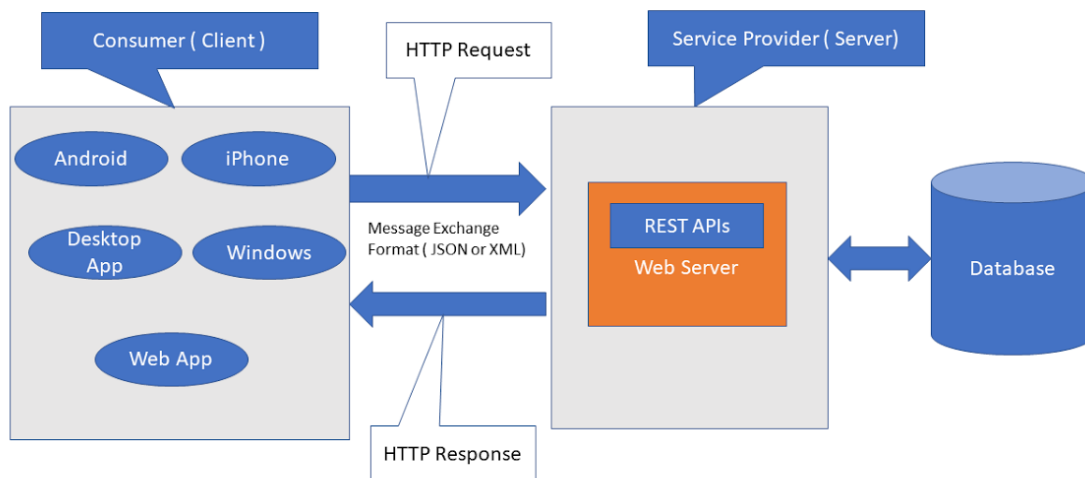
Despliegue en Coolify dentro de un VPS Debian.

Servicios ejecutados en contenedores Docker (backend y frontend).

Certificados SSL generados automáticamente por Caddy.

Escalabilidad horizontal y control de versiones vía GitHub.

REST – Architecture





Capa	Tecnología	Justificación
Frontend	React + Vite + TypeScript	Interfaces rápidas, mantenibles, tipado seguro, experiencia moderna.
Backend	Python + Flask	Ligero, modular, ideal para APIs educativas y de producción.
Base de Datos	MySQL	Transaccional, confiable, relacional y escalable.
Autenticación	JWT + Cookies HttpOnly	Seguridad contra XSS y manejo de sesiones robusto.
Despliegue	Docker + Coolify	Portabilidad, CI/CD, administración visual del servidor.

4.5 Requisitos del Software

4.5.1 Requisitos Funcionales (resumen técnico)

Basado en los documentos de requisitos cargados (RF01–RF17)

CRUD de especies, razas, animales, campos, alimentos.

Registro de:

Mejoras genéticas

Controles de salud

Enfermedades

Medicamentos

Tratamientos

Vacunas

Vacunaciones

Relaciones tratamiento–medicamento y tratamiento–vacuna



Gestión de usuarios y roles.

Manejo de genealogías y referencia a padres.

Cálculo automático de duración de pastoreo.

Control de capacidad de campos.

Validaciones estrictas de formatos, unicidad y coherencia.

4.5.2 Requisitos No Funcionales

Disponibilidad del 99% en servidor Coolify.

Seguridad con JWT, cookies HttpOnly y validación CORS.

Usabilidad para aprendices con interfaz intuitiva.

Escalabilidad mediante contenedores Docker.

Rendimiento óptimo gracias a la recarga instantánea de Vite.

4.6 Módulos del Sistema

4.6.1. Módulo de Usuarios y Roles

Funcionalidades:

Registro y autenticación.

Gestión de roles: Administrador, Instructor, Aprendiz.

Control de acceso basado en permisos.



4.6.2. Módulo de Animales

Permite registrar:

Datos generales (peso, sexo, nacimiento).

Raza y especie.

Referencias genealógicas (padre y madre).

Incluye validación automática y control de duplicados.

4.6.3. Módulo de Salud Animal

Incluye:

Controles de salud.

Historial médico.

Enfermedades.

Tratamientos.

Medicamentos.

Relación tratamiento–medicamento.

4.6.4. Módulo de Vacunación

Registro de vacunas.

Registro de vacunaciones realizadas.

Asignación a animales y aprendices.



Verificación del plan nacional.

4.6.5. Módulo de Mejoras Genéticas

Documentación del tipo de intervención genética.

Resultados y técnica usada.

Fecha y responsable.

4.6.6 Módulo de Pastoreo y Campos

Registra:

Campos de la finca.

Asignación de animales.

Cálculo automático de duración.

Control de capacidad según área y manejo.

4.6.7 Módulo de Alimentos

Tipos de alimento disponibles.

Fechas de siembra y cosecha.

Mediciones ambientales.

Áreas asignadas.



4.7 Funcionamiento del Sistema

Paso 1: Autenticación

El usuario ingresa su identificación y contraseña.

El backend valida credenciales.

Se genera un JWT seguro en cookie HttpOnly.

Paso 2: Acceso a Módulos según Rol

Administrador: gestión total.

Instructor: animales, salud, campo, vacunación.

Aprendiz: registros operativos supervisados.

Paso 3: Operaciones CRUD

Ejemplo: Registro de un animal

React renderiza el formulario.

El usuario ingresa datos.

Axios envía JSON al endpoint /animals.

Flask valida datos mediante SQLAlchemy.

Se guarda en PostgreSQL.

Respuesta: "Animal registrado exitosamente".

Paso 4: Visualización

Listados con paginación.



Búsquedas y filtros.

Tablas responsivas y gráficas futuras.

Paso 5: Generación de Reportes (Futuro)

Reportes por tratamiento, campos, capacidad, etc.

KPIs de genética, salud, pastoreo, ocupación, razas.

4.8 Diagramas Didácticos de Arquitectura Interna

4.8.1 Arquitectura Lógica

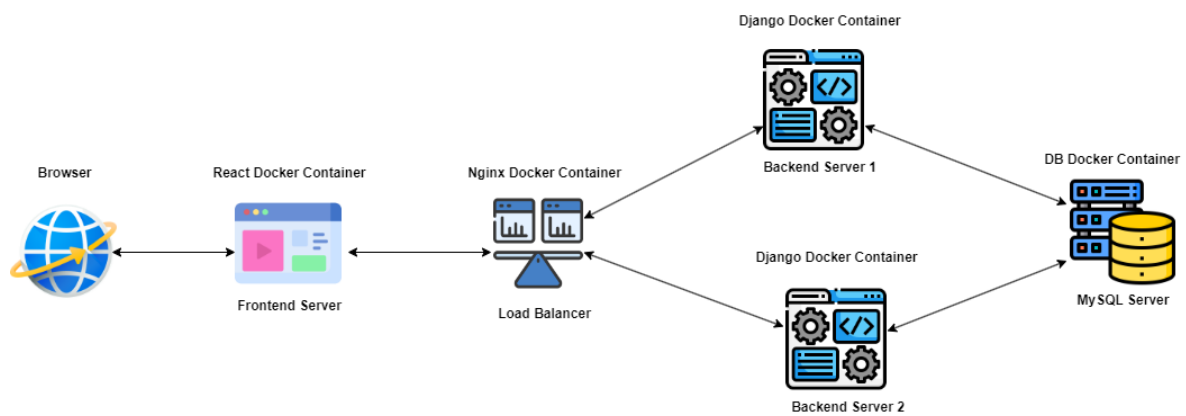


```
Frontend (React + Vite)
├ Componentes
├ Pages
├ Services (Axios)
└ Auth (Cookies)

Backend (Flask)
├ Models (SQLAlchemy)
├ Routes (Blueprints)
├ Controllers
└ Middlewares (JWT, CORS)

Base de Datos (PostgreSQL)
├ tablas animales
├ enfermedades
├ campos
├ vacunas
└ usuarios
```

4.8.2 Arquitectura Física



4.9 Ventajas Técnicas y Formativas

Software real para escenarios reales de la Finca Villa Luz.



Aprendizaje interdisciplinario entre ADSO y programas pecuarios.

Mejora la toma de decisiones basada en datos.

Fomenta competencias TIC aplicadas al sector agropecuario.

5. Manual de usuario: Instalación, navegación, actividades y orientaciones para aprendices e instructores.

5.1 Instalación del Software

El software puede ser utilizado en dos modalidades:

5.1.1 Instalación para uso local (aprendices ADSO / pruebas internas)

Requisitos Previos

Componente	Versión sugerida	Función
Python	3.10+	Ejecutar el backend Flask
Node.js + NPM	Node 18+	Ejecutar el frontend React/Vite
Git	Última versión	Clonar repositorios
Docker Desktop (opcional)	Última versión	Usar contenedores
MySQL	8	Base de datos local

Instalación del Backend (Flask)

Abrir una terminal.

Ejecutar:



```
git clone https://github.com/miuguel/BackFinca.git
cd BackFinca
python -m venv venv
source venv/bin/activate # En Windows: venv\Scripts\activate
pip install -r requirements.txt
```

Configurar variables de entorno (por ejemplo):

```
export DATABASE_URL="postgresql://usuario:password@localhost:5432/finca"
export SECRET_KEY="clave_segura"
```

Iniciar servidor: `python app.py`

Instalación del Frontend (React + Vite)

```
git clone https://github.com/miuguel/VillaLuzFront.git
cd VillaLuzFront
npm install
npm run dev
```

El sistema quedará disponible en: <http://localhost:5173>

5.1.2 Instalación en servidor (Producción – Coolify)

Flujo de instalación:

El administrador dispone de un VPS Debian configurado con Coolify.

Se crean dos aplicaciones:



Backend Flask en contenedor Docker

Frontend React como servicio estático

Coolify obtiene el código desde GitHub y genera despliegue automático.

Caddy o Traefick asigna dominio y genera certificado SSL.

Este procedimiento solo lo realiza el equipo técnico o el instructor ADSO.

5.2 Acceso al Sistema

Ingresar al dominio proporcionado por el instructor.

Se mostrará la pantalla de Inicio de Sesión.

Ingresar:

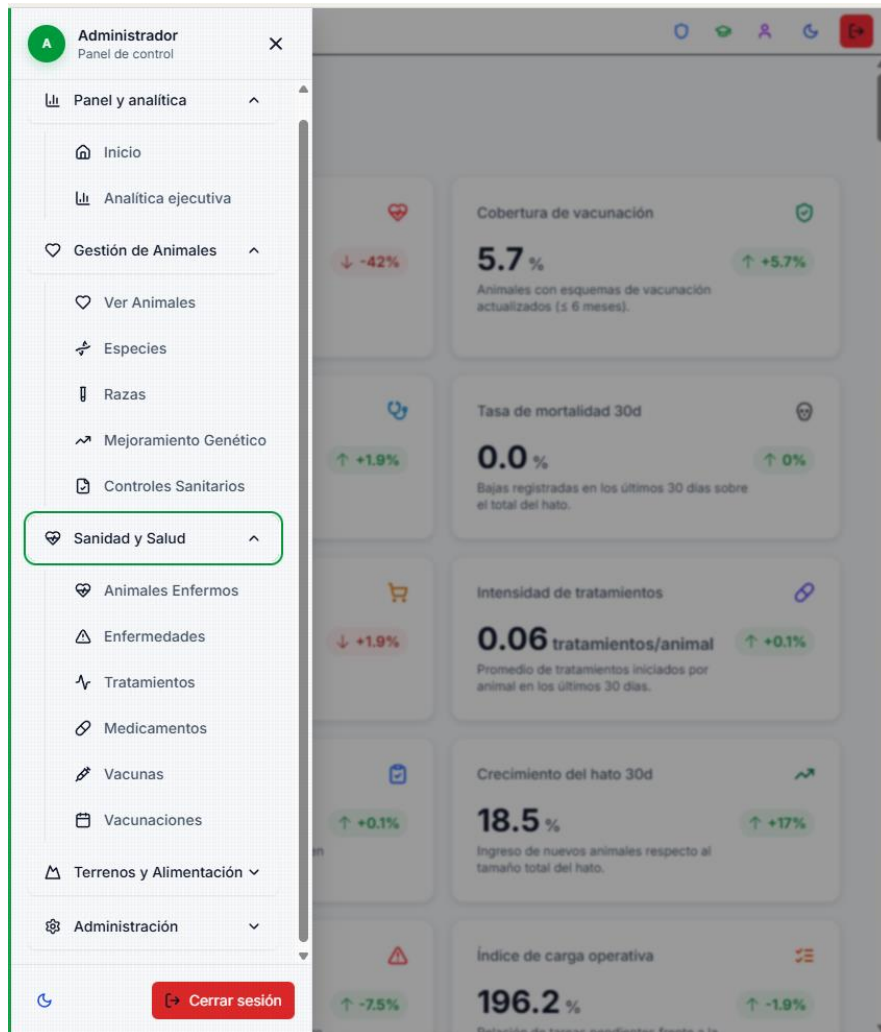
Identificación

Contraseña

Roles existentes:

Rol	Permisos principales
Administrador	Acceso total a todos los módulos
Instructor	Manejo operacional de animales, salud, campos, vacunas
Aprendiz	Registro de actividades supervisadas (pastoreo, tratamientos, etc.)

5.3 Navegación General del Sistema



Al autenticarte, verás el Panel Principal (Dashboard), que incluye:

Estructura de la interfaz

Menú lateral izquierdo

Animales

Especies y Razas

Salud



Vacunas

Mejoras Genéticas

Alimentación

Campos y Pastoreo

Usuarios (solo admin)

Zona de trabajo central

Formularios

Tablas

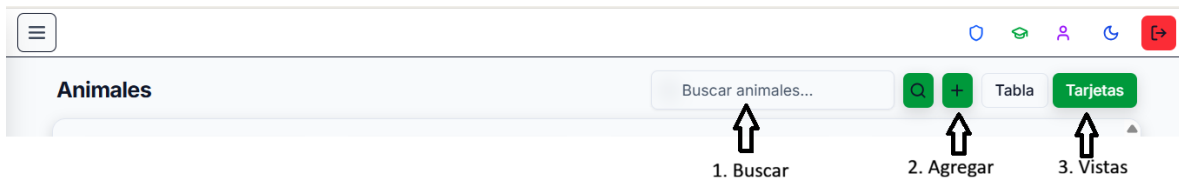
Filtros

Visualizaciones

Encabezado superior

Rol del usuario

Todas las vistas tienen la misma estructura:



1. Buscador general por vista, busca en todos los atributos del modelo
2. Despliega un modal con un formulario para insertar un nuevo registro
3. Algunas vistas tienen dos modos, tarjeta o tablas.

5.4 Actividades y Uso de Cada Módulo

A continuación, se describe paso a paso cómo interactuar con cada sección del sistema. Esta parte del manual está diseñada de manera didáctica para aprendices e instructores.



5.4.1 Módulo de Usuarios

Para Administradores

Seleccionar Usuarios en el menú.

Dar clic en Registrar Usuario.

Diligenciar:

Rol

Identificación

Nombre completo

Correo

Contraseña

Guardar.

The screenshot displays the 'Usuarios' management page. The left sidebar contains navigation options: 'Administrador Panel de control', 'Panel y analítica', 'Gestión de Animales', 'Sanidad y Salud', 'Terrenos y Alimentación', and 'Administración' with a sub-option 'Gestión de Usuarios'. The main content area shows a grid of user cards. Each card includes a status indicator (Activo), a role (Administrador or Aprendiz), and fields for 'Identificación', 'Nombre', 'Email', 'Teléfono', 'Dirección', 'Creado', and 'Actualizado'. A search bar 'Buscar usuarios...' and buttons for 'Tabla' and 'Tarjetas' are located at the top right of the grid. A red 'Cerrar sesión' button is visible in the bottom left corner of the page.

Identificación	Rol	Nombre	Email	Teléfono	Dirección	Creado	Actualizado
1098170955	Administrador	Miguel	centavendi...	313361378	Velez Santa...	10/12/2025	10/12/2025
14789	Aprendiz	Miguel Teje...	mario@mar...	123456789	-	9/12/2025	10/12/2025
19991	Administrador	Miguel Teje...	centavendi...	3133619	-	10/12/2025	10/12/2025
10999	Aprendiz	Miguel Teje...	leugimtejed...	3133612568	-	10/12/2025	10/12/2025
97686554	Aprendiz	asdfasdfas...	leugimtejed...	3133619145	-	9/12/2025	9/12/2025
123432123	Aprendiz	Miguel trer...	centavendi...	3133611035	-	9/12/2025	9/12/2025
1231245	Aprendiz	Miguel asdf...	centavendi...	3133619039	-	9/12/2025	9/12/2025
2354768	Aprendiz	asdfasdfas...	centavendi...	31336190587	-	9/12/2025	9/12/2025



Crear Usuario

• Información Básica

Identificación * • Campo obligatorio Ej: 123456789 <small>Este campo es obligatorio.</small>	Nombre completo * • Campo obligatorio Ej: Juan Pérez <small>Este campo es obligatorio.</small>
Email * • Campo obligatorio usuario@dominio.com <small>Este campo es obligatorio.</small>	Rol * • Campo obligatorio Aprendiz
Contraseña Requerida al crear	Activo <input checked="" type="checkbox"/> Activo

• Información de Contacto

Teléfono *	Dirección
-------------------	------------------

Cancelar Crear

Actividad práctica para aprendices ADSO

Registrar tres usuarios:

1 Administrador

1 Instructor

1 Aprendiz

5.4.2 Módulo de Especies y Razas

Registrar una especie



Abrir Especies.

Clic en Nueva especie.

Ingresar nombre.

Guardar.

Administrador
Panel de control

Panel y analítica

Gestión de Animales

Ver Animales

Especies

Razas

Mejoramiento Genético

Controles Sanitarios

Sanidad y Salud

Terrenos y Alimentación

Administración

Gestión de Usuarios

Cerrar sesión

Especies

Buscar especies...

ID	NOMBRE	CREADO	ACTUALIZADO	ACCIONES
#16	porcinos	10/12/2025	-	
#15	caninos	10/12/2025	-	
#10	Exotico	6/9/2025	-	
#9	PecuaríaMixta	6/9/2025	-	
#8	Felino	6/9/2025	-	
#7	Canino	6/9/2025	-	
#6	Aviar	6/9/2025	-	
#5	Equino	6/9/2025	-	
#4	Porcino	6/9/2025	-	
#3	Caprino	6/9/2025	-	

Página 1 de 2

Crear Especie

• Información Básica

Nombre *

• Campo obligatorio

Ej: Bovino

Este campo es obligatorio.

Cancelar

Crear



Registrar una raza

Abrir Razas.

Seleccionar la especie correspondiente.

Registrar nombre de raza.

Administrador
Panel de control

Panel y analítica

Gestión de Animales

- Ver Animales
- Especies
- Razas**
- Mejoramiento Genético
- Controles Sanitarios

Sanidad y Salud

Terrenos y Alimentación

Administración

Cerrar sesión

Razas

Buscar razas...

NOMBRE	ESPECIE	CREADO	ACCIONES
porcino_landrace	porcinos	10/12/2025	[Ver] [Editar] [Eliminar] [Más]
perro salchicha	Canino	10/12/2025	[Ver] [Editar] [Eliminar] [Más]
gyr	Bovino	10/12/2025	[Ver] [Editar] [Eliminar] [Más]
Nelore	Bovino	10/12/2025	[Ver] [Editar] [Eliminar] [Más]
Nelore	Bovino	10/12/2025	[Ver] [Editar] [Eliminar] [Más]
holstein	Bovino	10/12/2025	[Ver] [Editar] [Eliminar] [Más]
otra	Bovino	30/9/2025	[Ver] [Editar] [Eliminar] [Más]

Página 1 de 6

Crear Raza

Información Básica

Nombre *
• Campo obligatorio
Ej: MixedFeline
Este campo es obligatorio.

Especie *
• Campo obligatorio
Bovino

Cancelar

Crear



Actividad para aprendices pecuarios

Registrar:

Especie: Bovino

Raza: Brahman, Gyr, Holstein

5.4.3 Módulo de Animales

Pasos para registrar un animal

Ir a Animales > Registrar nuevo.

Ingresar:

Fecha de nacimiento

Sexo

Peso inicial

Raza

ID padre / madre (opcional)

Guardar.

El sistema valida formato y unicidad del ID.



Administrador
Panel de control

Panel y analítica

Gestión de Animales

- Ver Animales
- Especies
- Razas
- Mejoramiento Genético
- Controles Sanitarios

Sanidad y Salud

Terrenos y Alimentación

Administración

Cerrar sesión

Animales

Buscar animales... Tabla Tarjetas

RAZA	PESO	NACIMIENTO	EDAD	PADRE	MADRE
Brahma	638 kg	9/12/2023	24 meses	-	-
Gyr	500 kg	13/8/2025	4 meses	REC0008	REC0017
Brahma	693 kg	9/12/2020	60 meses	-	-
Holstein	75 kg	14/11/2025	1 meses	Animal1	Animal3

Página 1 de 6

Crear Animal

Información Básica

Registro *
Campo obligatorio
Ej: REC0001
Este campo es obligatorio.

Fecha de Nacimiento *
Campo obligatorio
11/12/2025

Raza *
Campo obligatorio
Criollo

Sexo *
Campo obligatorio
Macho

Estado
Sano

Peso (kg) *
Campo obligatorio
Ej: 250
Este campo es obligatorio.

Genealogía y Adquisición

Padre
Seleccionar padre

Madre
Seleccionar madre

Cancelar Crear



5.4.4 Módulo de Control de Salud

Registrar control de salud

Seleccionar Gestión de Animales > Controles Sanitarios.

Clic en Nuevo control.

Completar:

Animal

Fecha de revisión

Estado de salud

Descripción

Actividad para aprendices:

Registrar un control de salud real tomado en campo.



Administrador
Panel de control

Panel y analítica

- Inicio
- Analítica ejecutiva

Gestión de Animales

- Ver Animales
- Especies
- Razas
- Mejoramiento Genético
- Controles Sanitarios

Sanidad y Salud

Terrenos y Alimentación

Administración

Cerrar sesión

Controles

Buscar controles... Tabla Tarjetas

Estado de Salud	Estado de Salud	Estado de Salud	Estado de Salud
Excelente	Excelente	Bueno	Excelente
Información del Control	Información del Control	Información del Control	Información del Control
Animal: 00000040	Animal: 0004	Animal: fortaleza	Animal: 0004
Fecha: 9/12/2025	Fecha: 9/12/2025	Fecha: 9/12/2025	Fecha: 9/12/2025
Métricas	Métricas	Métricas	Métricas
Peso: 548 kg, Altura: 150 m	Peso: 450 kg, Altura: 140 m	Peso: 78 kg, Altura: 1 m	Peso: 23 kg, Altura: 12 m
Descripción: aumento_de_peso			
Estado de Salud: Excelente	Estado de Salud: Regular	Estado de Salud: Sano	Estado de Salud: Bueno

Página 1 de 4

Crear Control

Información Básica

Animal *
Campo obligatorio

Fecha de Chequeo *
Campo obligatorio

Selección de animal: [Seleccionar animal] (Debe seleccionar animal)

Fecha de Chequeo: [11/12/2025]

Métricas Básicas

Peso (kg): [Peso en kg]

Altura (m): [Altura en metros]

Estado de Salud *
Campo obligatorio

Selección de estado: [Seleccionar estado] (Debe seleccionar estado de salud)

Cancelar [Crear]



5.4.5 Módulo de Enfermedades y Tratamientos

Registrar enfermedad

Ir a Sanidad y Salud > Enfermedades.

Registrar nombre, síntomas y detalles.

Administrador
Panel de control

Panel y analítica
Gestión de Animales
Sanidad y Salud

Animales Enfermos
Enfermedades
Tratamientos
Medicamentos
Vacunas
Vacunaciones

Terrenos y Alimentación
Administración

Cerrar sesión

Enfermedades

Buscar enfermedades...

NOMBRE	SÍNTOMAS	DETALLES	CREADO	ACCIONES
estomatitis vesicular	fiebre	decaimiento	10/12/2025	
estomatitis vesicular	fiebre, diarrea, falta de apetito,	Apartar al animal del resto de los b...	10/12/2025	
Bronquitis	Tos persistente	Inflamación de vías respiratorias	6/9/2025	
ParásitosExternos	Prurito, pérdida de pelo	Pulgas, garrapatas, ácaros	6/9/2025	
TosDeLasGranjas	Tos, fiebre moderada	Síndrome respiratorio de origen mix...	6/9/2025	
Hipocalcemia	Temblor, debilidad	Trastorno mineral en lactantes	6/9/2025	
Dermatofitosis	Alopecia, lesiones circulares	Infección fúngica en piel	6/9/2025	
Tuberculosis	Tos crónica, pérdida de peso	Enfermedad zootópica	6/9/2025	
Artritis	Cojear, inflamación articular	Causas infecciosas o mecánicas	6/9/2025	
Gastroenteritis	Diarrea, inapetencia	Trastorno digestivo general	6/9/2025	

Página 1 de 3



Crear Enfermedad

• Información Básica

Nombre de la Enfermedad *
• Campo obligatorio
Ej: Brucelosis
Este campo es obligatorio.

Sintomas
Describe síntomas...

Detalles
Detalles y descripción general...

Cancelar **Crear**

Registrar tratamiento

Seleccionar Tratamientos.

Ingresar:

Enfermedad detectada

Fecha inicio y fin

Frecuencia

Dosis

Observaciones



Administrador
Panel de control

Panel y analítica

Gestión de Animales

Sanidad y Salud

- Animales Enfermos
- Enfermedades
- Tratamientos**
- Medicamentos
- Vacunas
- Vacunaciones

Terrenos y Alimentación

Administración

Cerrar sesión

Tratamientos

Buscar tratamientos...

ANIMAL	FECHA DE TRATAMIENTO	DESCRIPCIÓN	DOSIS	FRECUENCIA	OBSERVACIONES	CREADO	ACCIONES
0004	9/12/2025	fiebre	2_ml	2_veces_al_año	-	10/12/2025, 10:39:27	
0004	15/8/2027	sdf	5	10	asdf	17/9/2025, 23:03:14	
00000040	9/12/2025	Mejora_total	2ml	Una_vez_al_dia	-	10/12/2025, 9:09:17	
REC0040	9/10/2024	Control potrillo	5ml	Una vez	Evaluación	10/10/2024, 4:30:00	
REC0039	7/9/2024	Cuidados potro	5ml	Diaria	Observación	8/9/2024, 4:30:00	
REC0038	5/8/2024	Tratamiento joven	5ml	Diaria	Observación	6/8/2024, 4:30:00	
REC0037	3/7/2024	Control potra	5ml	Una vez	Evaluación	4/7/2024, 4:30:00	
REC0036	1/6/2024	Control potro	5ml	Una vez	Evaluación	2/6/2024, 4:30:00	
REC0035	23/3/2024	Desparasitación	10ml	Una dosis	Prevención	24/3/2024, 4:30:00	
REC0034	21/2/2024	Antibiótico	20ml	5 días	Reposición	22/2/2024, 4:30:00	

Página 1 de 5

Crear Tratamiento

Información básica

Animal *
• Campo obligatorio
Seleccionar animal
⚠ Debe seleccionar animal

Fecha de tratamiento *
• Campo obligatorio
11/12/2025

Diagnóstico *
• Campo obligatorio
Ej: Fiebre, infección...
Este campo es obligatorio.

Descripción
Descripción del tratamiento

Frecuencia
Ej: Una vez

Cancelar Crear



5.4.6 Módulo de Vacunas y Vacunación

Registrar vacuna

Datos requeridos:

Dosis

Vía de administración

Enfermedad objetivo

Intervalo

NOMBRE	TIPO	DOSIS	RUTA ADMIN.	INTERVALO (DÍAS)	ENFERMEDAD OBJETIVO	PLAN NACIONAL	CREADO
VacunaEspecial	Recombinante	1ml	Intravenosa	Anual	Mastitis	Sí	6/9/2025
VacunaCutanea	Toxoide	1ml	Tópica	Anual	Dermatitis	Sí	6/9/2025
VacunaAntiparasitaria	Inactivada	2ml	Oral	Trimestral	Parásitosintestinales	Sí	6/9/2025
VacunaMix2	Subunidad	1.5ml	Intramuscular	Semestral	Neumonía	Sí	6/9/2025
VacunaExotica	Adn	1ml	Oral	Anual	Bronquitis	Sí	6/9/2025
VacunaLeukosis	Inactivada	2ml	Intramuscular	Anual	Tuberculosis	Sí	6/9/2025
VacunaReproductiva	Recombinante	2ml	Subcutánea	Anual	Brucelosis	Sí	6/9/2025
VacunaClostridial2	Inactivada	2ml	Intramuscular	Anual	Clostridiosis	Sí	6/9/2025
VacunaPorcina	Inactivada	2ml	Intramuscular	Anual	FiebreAftosa	Sí	6/9/2025
VacunaAves	Atenuada	0.5ml	Oral	Anual	Coccidiosis	Sí	6/9/2025



Crear Vacuna

Información Básica

Nombre *
• Campo obligatorio
Ej: Vacuna contra Aftosa
Este campo es obligatorio.

Tipo **Dosis**
Seleccionar tipo Ej: 2 mL

Ruta de Administración **Intervalo de Vacunación (días)**
Seleccionar ruta Ej: 180

Enfermedad Objetivo
Seleccionar enfermedad

Plan Nacional
 Plan Nacional

Cancelar Crear

Registrar vacunación aplicada

Seleccionar Vacunaciones.

Ingresar:

Animal

Vacuna

Fecha

Aprendiz que la realizó

Instructor responsable



Administrador
Panel de control

Panel y analítica

Gestión de Animales

Sanidad y Salud

Animales Enfermos

Enfermedades

Tratamientos

Medicamentos

Vacunas

Vacunaciones

Terrenos y Alimentación

Administración

Cerrar sesión

Vacunaciones

Buscar vacunaciones...

FECHA	ANIMAL	VACUNA	APRENDIZ	INSTRUCTOR	CREADO	ACCIONES
9/12/2025	00000040	VacunaParasitica	Carlos Lopez Final	Maria Garcia Final	10/12/2025	
9/12/2025	0004	VacunaAntiparasitaria	-	-	10/12/2025	
10/12/2025	MARGARITA	VacunaReproductiva	-	-	10/12/2025	
14/6/2020	MARGARITA	VacunaFiebreAftosa	-	-	10/12/2025	
6/12/2025	0004	VacunaBronquitis	-	-	10/12/2025	
21/1/2026	MARGARITA	VacunaFiebreAftosa	-	-	10/12/2025	
24/12/2025	0004	VacunaFiebreAftosa	Juan Pérez	Juan Pérez	10/12/2025	
9/12/2025	0004	VacunaAntiparasitaria	Admin Seed	Admin Seed	10/12/2025	
27/10/2025	Animal3	VacunaEspecial	Juan Pérez	Juan Pérez	28/10/2025	
10/10/2024	REC0040	VacunaEspecial	Instructor B	Veterinario 2	11/10/2024	

Página 1 de 5

Crear Vacunación

Información de Vacunación

Animal *
• Campo obligatorio
Seleccionar animal
⚠ Debe seleccionar animal

Vacuna *
• Campo obligatorio
Seleccionar vacuna
⚠ Debe seleccionar vacuna

Fecha de Vacunación *
• Campo obligatorio
11/12/2025

Aprendiz
Seleccionar aprendiz

Instructor
Seleccionar instructor

Cancelar Crear

5.4.7 Módulo de Mejoras Genéticas

Pasos:

Seleccionar Gestión de Animales > Mejoramiento Genético.



Registrar:

Animal

Técnica utilizada (IA, TE, etc.)

Resultados

Fecha

Detalles del evento

Actividad práctica:

Registrar un evento de inseminación artificial realizado en la finca.

The screenshot shows a web application interface for 'Mejoras Genéticas'. On the left is a sidebar menu with categories like 'Panel y analítica', 'Gestión de Animales', 'Razas', 'Mejoramiento Genético', 'Controles Sanitarios', 'Sanidad y Salud', 'Animales Enfermos', 'Enfermedades', 'Tratamientos', 'Medicamentos', and 'Vacunas'. The main content area displays a table with the following data:

TÉCNICA	DETALLES	RESULTADO	FECHA	ANIMAL	ACCIO
inseminación artificial	cruce	en espera	4/10/2025	fortaleza	👁
transferencia_de_embriones	paso_a_paso	mejoramiento_genetico	9/12/2025	0004	👁
transferencia_embriones	paso_paso	gyr	23/12/2025	00000040	👁
transferencia_de_embriones	paso_a_paso	holstein	9/12/2025	0004	👁
inseminacion	cruce	cruce_gyr_holstein	9/12/2025	LA VACA LOLA	👁
INSEMINACION ARTIFICIAL	TORO EDANK GYR SEXADO	SERVIDA	9/5/2022	0004	👁
inseminacion artificial	se utilizaron los bovinos mas c...	el ternero nació con optimas c...	9/12/2025	ANIMAL0121312JEB	👁
Cruzamiento	Cruzamiento	Plan a largo plazo	3/7/2025	REC0005	👁
Selección	Selección	Ajustes de genética	2/7/2025	Animal 4	👁
Inseminación artificial	Inseminación	Seguimiento	1/7/2025	Animal 3	👁

At the bottom right of the table area, it says 'Página 1 de 5' with navigation arrows.



Crear Mejora Genética

• Información de Mejora Genética

Fecha *
• Campo obligatorio
11/12/2025

Animal (opcional) *
• Campo obligatorio
ID del animal
⚠ Debe seleccionar animal (opcional)

Técnica del Evento Genético *
• Campo obligatorio
Técnica utilizada
Este campo es obligatorio.

Detalles *
• Campo obligatorio
Detalles de la mejora genética
Este campo es obligatorio.

Resultados *

Cancelar Crear

5.4.8 Módulo de Campos y Asignación de Pastoreo

Registrar un campo

Campos requeridos:

Nombre

Capacidad

Área

Manejo

Mediciones (pH, humedad, etc.)



Asignar animal a un campo

Seleccionar Terrenos y Alimentación > Ubicación de Animales.



Registrar:

Animal

Campo

Fecha inicio

Fecha fin El sistema calcula duración automáticamente.

ANIMAL	CAMPO	FECHA ASIGNACIÓN	FECHA RETIRO	NOTAS	CREADO	ACCIONES
00000040	Sector 37	10/12/2025	12/12/2025	-	10/12/2025	[Ver] [Editar] [Eliminar]
0004	Sector 40	9/12/2025	11/12/2025	-	10/12/2025	[Ver] [Editar] [Eliminar]
0004	Sector 21	9/12/2025	1/8/2027	con	10/12/2025	[Ver] [Editar] [Eliminar]
asfesa1	Corral 1	9/12/2025	-	-	10/12/2025	[Ver] [Editar] [Eliminar]
Animal3	Sector 40	27/10/2025	-	-	28/10/2025	[Ver] [Editar] [Eliminar]
Animal1	Sector 40	27/10/2025	-	-	28/10/2025	[Ver] [Editar] [Eliminar]
REC0039	potrero 4	14/10/2025	-	-	15/10/2025	[Ver] [Editar] [Eliminar]
REC0040	potrero 4	14/10/2025	-	-	15/10/2025	[Ver] [Editar] [Eliminar]
REC0040	Sector 40	30/9/2024	19/9/2025	Sector 40	1/10/2024	[Ver] [Editar] [Eliminar]
REC0039	Sector 39	31/8/2024	-	Sector 39	1/9/2024	[Ver] [Editar] [Eliminar]



Crear Asignación de animal

• Información de Asignación

Animal *
• Campo obligatorio
Seleccionar animal
⚠ Debe seleccionar animal

Campo *
• Campo obligatorio
Seleccionar campo
⚠ Debe seleccionar campo

Fecha de Asignación *
• Campo obligatorio
11/12/2025 📅

Fecha de Retiro
dd/mm/aaaa 📅

Notas
Observaciones adicionales (opcional)

Cancelar **Crear**

5.4.9 Módulo de Alimentación

Permite registrar:

Tipo de alimento

Fechas de siembra y cosecha

Manejo

Mediciones del cultivo



NOMBRE	DESCRIPCIÓN	CREADO	ACCIONES
Sin especificar	Placeholder autogenerado	13/9/2025	[Ocultar] [Editar] [Eliminar]
Concentrado	Mezcla	6/9/2025	[Ocultar] [Editar] [Eliminar]
Leguminosas	Cosecha manual	6/9/2025	[Ocultar] [Editar] [Eliminar]
GranoMix	Suministro diario	6/9/2025	[Ocultar] [Editar] [Eliminar]
Silaje	Fermentación	6/9/2025	[Ocultar] [Editar] [Eliminar]
PasturaSeca	Enfardado	6/9/2025	[Ocultar] [Editar] [Eliminar]
CéspedMezcla	Corte semanal	6/9/2025	[Ocultar] [Editar] [Eliminar]
Alfalfa	Corte quincenal	6/9/2025	[Ocultar] [Editar] [Eliminar]
Sorgo	Cosecha	6/9/2025	[Ocultar] [Editar] [Eliminar]
Maiz	Riego	6/9/2025	[Ocultar] [Editar] [Eliminar]

Crear Tipo de Alimento

Información Básica

Tipo de Alimento *

• Campo obligatorio

Ej: Pasto de corte

Este campo es obligatorio.

Manejos/Descripción

Descripción breve...

Detalles (opcional)

Fecha de Siembra: dd/mm/aaaa

Fecha de Cosecha: dd/mm/aaaa

Área (m²/ha):

Calibres:

Cancelar Crear

5.5 Orientaciones para Aprendices

Para aprendices pecuarios

Registrar datos reales de campo en tiempo real.

Comparar registros digitales vs. registros manuales.



Analizar estado de salud y ciclos productivos.

Consultar histórico de un animal antes de tomar decisiones.

Para aprendices ADSO

Identificar flujos CRUD completos.

Analizar validaciones implementadas.

Comprender interacción React → Axios → Flask → DB.

Revisar trazabilidad desde la historia de usuario hasta el endpoint.

5.6 Orientaciones para Instructores

Instructores Pecuarios

Usar el software como herramienta de apoyo en las prácticas.

Supervisar registros realizados por aprendices.

Validar que los datos ingresados correspondan a la realidad de la finca.

Planear actividades como:

Rutas de tratamiento

Rotación de potreros

Control de condición corporal

Instructores ADSO

Utilizar el sistema para enseñar:

API REST

Arquitectura cliente-servidor

Rutas protegidas



Validación y manejo de formularios

Diseñar actividades:

Crear nuevo módulo

Extender endpoint

Optimizar consulta

Mejorar interfaz

5.7 Buenas Prácticas de Uso

Mantener datos actualizados.

No compartir credenciales.

Registrar controles y tratamientos el mismo día que se realicen.

Seguir nomenclaturas recomendadas por la finca.

Respetar roles y permisos asignados.

6. Manual técnico: Estructura del código, diagramas, mantenimiento, lineamientos de actualización y seguridad.

El presente manual describe la arquitectura técnica, la organización del código, los procesos de mantenimiento, los lineamientos de actualización y los criterios de seguridad implementados en el sistema Finca Villa Luz. El proyecto está compuesto por un backend en Flask y un frontend en React, diseñados para gestionar animales, potreros, salud, genética, usuarios y analítica avanzada.

6.1 Arquitectura General del Sistema

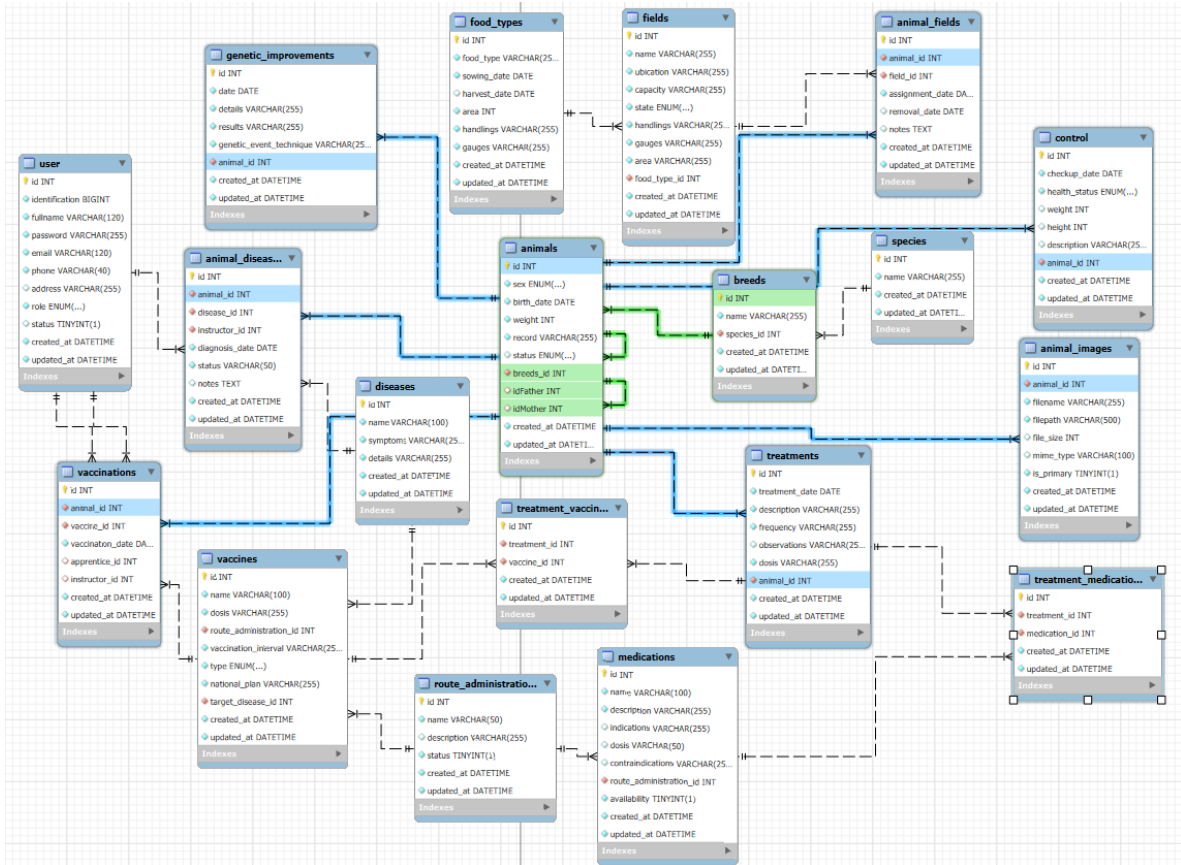
El sistema integra:

Frontend → Panel administrativo, formularios, tablas, acciones contextuales, dashboard analítico.



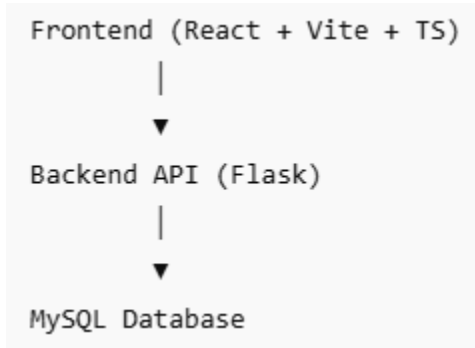
Backend → API RESTful, modelos SQLAlchemy, lógica de negocio, integridad referencial, seguridad.

Base de datos → MySQL.



Storage → Manejo de imágenes de animales.

Despliegue → VPS o contenedores (Docker/Coolify).



Incluye microcaché en endpoints críticos y caching en frontend.

6.2 Backend (Flask + SQLAlchemy + MySQL)

La arquitectura sigue un estilo modular, dividido por responsabilidades:

controllers/ → Controladores (capa de endpoints y lógica de flujo)

models/ → Modelos SQLAlchemy y relaciones

namespaces/ → Organización de módulos complejos (analytics, animales, usuario, etc.)

utils/ → Funciones auxiliares, API centralizada, respuestas estándar

templates/ → Plantillas Jinja2 vistas mínimas

api.py → Registro de rutas, blueprints y configuración central

app/init.py → Inicialización de Flask, DB, CORS, JWT, logs

migrations/ → Migraciones Alembic

instance/ → Configuraciones sensibles (base de datos, JWT secret)

certs/, static/, tools/, docs/ → Recursos del sistema



├── app/	
│ ├── __init__.py	→ Inicialización de Flask (DB, JWT, CORS, Logs)
│ ├── api.py	→ Registro central de rutas y Blueprints
│ └── config.py	→ Configuración (dev/prod)
│ └── models/	→ Modelos y relaciones <u>SQLAlchemy</u> (≈24 modelos)
│ ├── animals.py	
│ ├── breeds.py	
│ ├── species.py	
│ ├── fields.py	
│ ├── diseases.py	
│ ├── vaccines.py	
│ ├── treatments.py	
│ ├── users.py	
│ ├── analytics_models.py	
│ └── __init__.py	
│ └── namespaces/	→ “Módulos completos” agrupados como en frontend
│ ├── animals_namespace.py	→ CRUD + menús de acciones del animal
│ ├── analytics_namespace.py	→ 28 endpoints analíticos
│ ├── users_namespace.py	→ Usuarios, roles, seguridad
│ ├── fields_namespace.py	→ Potrereros + animal_count
│ ├── diseases_namespace.py	
│ ├── vaccines_namespace.py	
│ ├── treatments_namespace.py	
│ ├── uploads_namespace.py	→ Imágenes de animales (upload, delete)
│ └── __init__.py	
│ └── services/	→ Lógica reutilizable, igual que frontend/services
│ ├── base_service.py	→ Abstracción CRUD
│ ├── analytics_service.py	→ Motor SQL crudo optimizado
│ ├── image_service.py	→ Upload, resize y gestión de imágenes
│ ├── dependency_service.py	→ Integridad referencial (coincide con frontend)
│ └── auth_service.py	→ Seguridad, JWT, roles
│ └── utils/	→ Funciones auxiliares (idéntico al frontend/utils)
│ ├── responses.py	→ Respuestas estandarizadas JSON
│ ├── validators.py	→ Validaciones de negocio
│ ├── date_utils.py	→ Manejo de fechas
│ └── cache.py	→ Cache en memoria para analytics
│ └── templates/	→ Plantillas Jinja mínimas
│ └── static/	→ Imágenes y contenidos
├── migrations/	→ Alembic: Versionamiento de BD
├── instance/	→ Variables sensibles (DATABASE_URL, JWT_SECRET)
├── venv/	→ Entorno virtual
├── requirements.txt	
└── wsgi.py	→ Punto de entrada a producción

Responsable de:

CRUD general de entidades.

Módulos de salud, terrenos, genética, usuarios y trazabilidad.

Sistema de Analytics (33 KPIs + endpoints agregados).

Integridad referencial mediante validaciones y checks automáticos.

Seguridad, autenticación, permisos y auditoría.



Optimización y microcaché para endpoints pesados.

Integridad Referencial en el Backend

El backend evita operaciones peligrosas como eliminar entidades con dependencias activas. Reglas:

No eliminar especie con razas.

No eliminar raza con animales.

No eliminar campo con animales asignados.

No eliminar usuario con registros sanitarios.

No eliminar tratamientos con registros asociados.

Cada controlador implementa:

Verificación previa.

Respuesta descriptiva.

Sugerencia de acciones correctivas.

Lineamientos de Mantenimiento del Backend

Mantener controladores ligeros.

Mover lógica pesada a modelos o utils.

Toda entidad nueva debe incluir:

Modelo

Controlador

Migración

Validación

Registro en API

Utilizar solo respuestas JSON estandarizadas.

Cuidar claves y relaciones al extender modelos.



Implementar índices para KPIs o consultas analíticas.

Documentar cada endpoint nuevo.

Lineamientos de Actualización del Backend

Para agregar una nueva entidad:

Definir modelo en /models

Crear migración

Crear controlador CRUD

Registrar Blueprint

Crear validaciones de integridad referencial

Documentar endpoints

Para agregar nueva métrica de Analytics:

Crear consulta optimizada

Crear endpoint en namespace analytics

Implementar microcaché

Exponer datos en formato KPI

Lineamientos de Seguridad del Backend

Activar HTTPS en ambiente productivo.

Tokens JWT con expiración limitada.

Uso de cookies HttpOnly cuando aplique.

Validar payloads en cada endpoint.

Sanitizar uploads y limitar tamaños.

Configurar CORS solo con orígenes permitidos.

No exponer credenciales en el repositorio.



RespalDOS automáticos de la base de datos.

6.3 Frontend (React + Vite + TypeScript + Tailwind + React Query)

La arquitectura del frontend sigue un estilo modular, organizado por capas funcionales que separan la lógica de negocio, la vista, los servicios de datos y los componentes reutilizables.

components/ → Formularios, tablas, dashboards, menús de acciones

pages/ → Vistas completas del sistema

services/ → Conexión con API REST

hooks/ → Lógica reutilizable y React Query

utils/ → Búsqueda offline, fechas, formatos

layouts/ → Sidebar, navegación, roles

router/ → Ruteo y permisos

public/ → Archivos estáticos

main.tsx / App.tsx → Configuración inicial



```
src/
├── components/
│   ├── analytics/           → KPIs, gráficos, alertas y FieldCard
│   ├── common/             → AdminCRUDPage, modales, tablas
│   ├── dashboard/         → Menús, sidebar, layouts
│   └── forms/              → Formularios reutilizables
├── hooks/
│   ├── useAnalytics.ts     → Hook centralizado para consultas
│   └── useForeignKeySelect.ts → Selects dinámicos con filtros
├── pages/
│   ├── dashboard/
│   ├── analytics/         → DashboardExecutive, FieldsPage
│   └── admin/              → CRUD por entidad
├── services/
│   ├── api.ts              → Axios + cache dual (memoria + IndexedDB)
│   ├── baseService.ts     → CRUD reutilizable
│   ├── analyticsService.ts → 28 endpoints Analytics
│   ├── animalImageService.ts → Uploads corregidos ✓
│   └── (20+ servicios CRUD)
├── utils/
│   ├── globalSearch.ts    → Búsqueda global offline-first
│   ├── indexedDBCache.ts  → Cache persistente en navegador
│   ├── colors.ts          → Paleta del sistema
│   └── date.ts             → Conversión de fechas
```

- components/ → Componentes visuales reutilizables y UI especializada

Incluye:

Tablas CRUD, formularios, modales, menús de acciones

Tarjetas, gráficos, vistas del dashboard

Layouts, sidebar, navegación por roles Es la capa principal de construcción de interfaces.

- pages/ → Vistas completas que representan pantallas del sistema

Contiene:

Gestión de animales, potreros, razas, especies

Dashboard ejecutivo de Analytics

Módulos por rol (Admin, Instructor, Aprendiz) Cada página utiliza componentes y servicios para componer una interfaz funcional.



- services/ → Capa de servicios para consumir el backend (API REST)

Responsable de:

Definir endpoints

Enviar peticiones GET/POST/PUT/DELETE

Manejar errores de red

Estandarizar la comunicación con FlaskIncluye servicios por entidad (animales, campos, tratamientos, analytics, imágenes, usuarios).

hooks/ → Lógica reutilizable y manejo avanzado de datos

Incluye:

Integración con React Query para:

- caching • invalidación • revalidación

Hooks para selects de claves foráneas

Hooks para KPIs y consultas analíticas

utils/ → Funciones auxiliares, formateadores y herramientas del sistema

Contiene:

Búsqueda global offline (IndexedDB)

Conversión y normalización de fechas

Utilidades de texto, filtros y validaciones

Paletas de colores para gráficos e interfaces Sirve como apoyo para mantener el código limpio y reutilizable.

layouts/ (si se incluye dentro de components) → Estructura visual global

Define:

Estructura del dashboard

Header, sidebar, rutas protegidas



Renderizado condicionado por roles

router/ o configuración de rutas (según estructura usada)

Encargado de:

Definir rutas del frontend

Asignar vistas por rol

Controlar navegación y acceso autorizado

Redirigir según permisos

styles/ (si aplica) → Estilos globales y configuraciones

Incluye:

Variables de tema

Configuración Tailwind

Estilos compartidos

public/ → Archivos estáticos accesibles directamente

Ejemplos:

Íconos, logos, PWA manifest

Service worker para funcionalidades offline

main.tsx / App.tsx → Punto de entrada y configuración principal

Inicializa:

Router del sistema

Providers globales (React Query, temas, contexto de usuario)

Configuración inicial de la aplicación

Responsable de:

UI/UX moderna, responsiva y escalable.



PWA con búsqueda global offline-first.

Sistema de menús contextuales de acciones (Animals / Users).

Sistema completo de Analytics (Dashboard Ejecutivo, Potreros, Alertas).

Integridad referencial con advertencias antes de eliminar entidades.

Componentización, caching, hooks reutilizables.

6.4 Diagramas Técnicos

Arquitectura General

Frontend (React) → API REST (Flask) → PostgreSQL

Arquitectura del Backend

api.py → namespaces → controllers → models → DBLineamientos de Mantenimiento del Frontend

6.5 Mantenimiento del Sistema

Backend

Mantener controladores simples.

Crear servicios cuando la lógica sea compleja.

Documentar nuevos endpoints.

Mantener migraciones actualizadas.

Validar integridad referencial.

Frontend

Crear servicios por entidad.



Preferir hooks para no repetir lógica.

Mantener consistencia en estilos y formularios.

Tipar estructuras de datos con TypeScript.

6.6 Lineamientos de Actualización

Agregar nueva entidad

Backend:

Crear modelo.

Crear controlador.

Crear migración.

Registrar rutas.

Frontend:

Crear servicio.

Configurar CRUD con AdminCRUDPage.

Añadir formularios y validaciones.

Agregar nueva acción

Crear modal y formulario.

Añadir entrada al menú contextual.

Crear endpoint si aplica.

Agregar métricas de Analytics

Extender consulta en backend.

Crear endpoint.



Crear hook.

Crear componente KPI.

6.7 Lineamientos de Seguridad

Backend

HTTPS obligatorio.

JWT con expiración.

Cookies HttpOnly.

Validación de entrada.

Sanitización en uploads.

Frontend

No almacenar tokens en LocalStorage.

Validar permisos antes de cada acción.

No exponer información sensible.

7. Estrategia pedagógica de uso: Integración del software en la formación y actividades sugeridas.

La estrategia pedagógica de este software está en concordancia con el modelo pedagógico institucional del servicio nacional de aprendizaje MPI (SENA), el cual tiene como objetivo incrementar los niveles de pertinencia y calidad de los procesos formativos en la institución y promueve la innovación y la creatividad buscando diversas alternativas de solución a los problemas.

Además el MPI promueve la investigación, partiendo de la formulación de problemas, y apelando al trabajo colaborativo, la gestión de la información y el uso de las TIC, en el marco de la gestión del



conocimiento. Adicionalmente promueve la interdisciplinariedad; donde las diferentes disciplinas trabajan coherentemente en función de la resolución de problemas como lo plantea el desarrollo de este producto técnico pedagógico, al articular aprendices e instructores de las áreas pecuaria y de desarrollo de software, enfrentados a un problema real (la deficiencia de datos reales y confiables en la producción ganadera de la granja Villa Luz)

En ese mismo sentido el desarrollo y uso de este producto promueve el aprendizaje significativo, el cual tiene lugar cuando los aprendices perciben el objeto de conocimiento como importante para el logro de sus propios objetivos o resultados del aprendizaje, es decir una persona aprende realmente cuando percibe que el objeto de conocimiento está relacionado con el mantenimiento o enriquecimiento de sí mismo, es así que la mayor parte del aprendizaje significativo se logra mediante la práctica y uno de los modos más eficaces para promover el aprendizaje es enfrentar al aprendiz con problemas prácticos de carácter social, ético o productivo, problemas personales o de investigación, situación que se observa en el desarrollo y uso de este producto teniendo en cuenta la pregunta problema que se pretende resolver ¿Se puede mejorar la calidad de la formación en los programas pecuarios y de desarrollo de software del Centro de Gestión Agroempresarial del Oriente a través de la creación y la implementación de una herramienta de gestión digital?

Por otra parte las estrategias de aprendizaje deben ser eminentemente activas y deben estar caracterizadas porque a través de ellas: se aprenda a través de la experimentación y la práctica, se fortalezca tanto el trabajo autónomo como colaborativo, se desarrollen habilidades superiores de pensamiento y se apliquen en la resolución de problemas.

En ese orden de ideas las estrategias recomendadas para el desarrollo y uso del Software didáctico para la gestión de la información de la producción ganadera en la finca Villa Luz del centro de Gestión Agroempresarial del Oriente (CGAO) son:

Aprendizaje basado en proyectos (ABP)

Simulación

Exposición

Estudio de caso

Juego de roles

Mapa conceptual

línea de tiempo.

Seminario investigativo



8. Evidencias de validación: Pruebas piloto, retroalimentación y ajustes realizados. Debemos hacer dos una en Villa Luz y otra con aprendices.

Para verificar las validaciones se sugiere Ver el ANEXO. 1 Validación PG ADSO

9. Impacto esperado en la formación:

En una primera intervención se plantea impactar tres programas de formación de nivel tecnológico: Análisis y Desarrollo de Software, Producción Ganadera y Gestión de Empresas Pecuarias con 125 aprendices presentes en el centro de formación y en la granja Villa Luz. Seguidamente se propone trabajar con los aprendices de nivel técnico en los programas de programación de software, producción agropecuaria, mayordomía de empresas ganaderas y producción pecuaria, con un total de 220 aprendices incluyendo los programas de articulación con la media, también se espera poder trabajar en los programas de la estrategia CAMPESENA, logrando un efecto sobre 100 aprendices de programas de nivel complementario especial denominados como “emprendedores” en las áreas pecuaria especialmente en ganadería bovina y de esta manera expandir el rango de acción hasta los pequeños productores ganaderos

En cuanto al impacto técnico se espera que los para los aprendices del área de software, el desarrollo de este producto técnico pedagógico permita ver un ejemplo cercano de creación, desde una problemática real, además la herramienta y todo su código de desarrollo quedará disponible para su uso análisis y modificación.

Para los aprendices del área pecuaria el software se convierte en una herramienta de aprendizaje y de gestión, que permite mejorar los conocimientos y habilidades en la recolección y manejo de la información conllevando a la toma de decisiones basadas en datos confiables, competencias necesarias para desempeñarse en el mundo del trabajo con calidad y pertinencia.

10. Material complementario: Guías, tutoriales o recursos adicionales.

10.1 Guías oficiales y documentación técnica



Incluye enlaces confiables y vigentes que apoyen las tecnologías usadas en el proyecto.

Backend (Flask, SQLAlchemy, JWT, Alembic)

Flask Documentation: <https://flask.palletsprojects.com>

SQLAlchemy ORM y Core: <https://docs.sqlalchemy.org>

Alembic (migraciones): <https://alembic.sqlalchemy.org>

Flask-JWT-Extended: <https://flask-jwt-extended.readthedocs.io>

Frontend (React, Vite, TS, React Query, Chart.js)

React Docs (Beta — mejor explicada): <https://react.dev>

Vite Documentation: <https://vitejs.dev>

TypeScript: <https://www.typescriptlang.org/docs>

React Query (TanStack): <https://tanstack.com/query/latest/docs>

Chart.js: <https://www.chartjs.org/docs/latest>

Base de datos

MySQL Reference: <https://dev.mysql.com/doc>

Índices, claves primarias/foráneas: <https://dev.mysql.com/doc/refman/8.0/en/sql-indexes.html>

10.2 Tutoriales prácticos

Visual Studio Code + herramientas de IA

Documentación oficial de VS Code: <https://code.visualstudio.com/docs>

Introducción al uso de Git desde VS

Code: <https://code.visualstudio.com/docs/sourcecontrol/overview>



Docker y despliegue en Coolify

Docker Docs: <https://docs.docker.com/get-started>

Compose: <https://docs.docker.com/compose>

Coolify Docs (Deploy en un VPS): <https://docs.coolify.io>

Git y GitHub

Git Handbook: <https://guides.github.com/introduction/git-handbook>

GitHub Learning Lab: <https://skills.github.com>

10.3. Material complementario didáctico para instructores

Videos sugeridos (oficiales y didácticos)

Flask Mega-Tutorial (Miguel Grinberg): <https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>

React para principiantes (CódigoFacilito / Midudev)

Docker desde cero (Fazt / Víctor Robles / MoureDev)

Rutas de aprendizaje recomendadas

Frontend Developer Roadmap: <https://roadmap.sh/frontend>

Backend Developer Roadmap: <https://roadmap.sh/backend>

11. Propiedad intelectual y originalidad:

Este producto técnico pedagógico está en concordancia con la guía de propiedad intelectual versión 02 con código GIC-G-0008 del sistema integrado de gestión y autocontrol del Servicio Nacional de Aprendizaje SENA, en su numeral 8.2.9 mención especial al software como como obra protegible



por el derecho de autor, clasificado como un sistema de información de uso propietario bajo licencia MIT.

Así mismo el software es un desarrollo original realizado en el marco de procesos formativos y de innovación tecnológica del Servicio Nacional de Aprendizaje – SENA.

La autoría del software corresponde a:

- Miguel Ángel Tejedor Mendoza
- Eduar Javier Camacho Vanegas

El sistema se distribuye bajo la Licencia MIT, una licencia de software libre permisiva que permite su uso, copia, modificación y redistribución, tanto en contextos educativos como productivos, siempre que se conserve el aviso de autoría y la licencia original.

Esta licencia fue seleccionada por su compatibilidad con entornos educativos, su claridad jurídica y su alineación con los principios de transferencia tecnológica y apropiación del conocimiento promovidos por el SENA.

La licencia se encuentra debidamente documentada mediante:

- Archivo LICENSE incluido en los repositorios del backend y frontend
- Declaración de Propiedad Intelectual del proyecto.

Por otra parte El Software didáctico para la gestión de la información de la producción ganadera en la finca Villa Luz del centro de Gestión Agroempresarial del Oriente (CGAO) está pensado para que los aprendices e instructores del área de programación de software y de producción pecuaria o ganadera puedan usarlo como herramienta de aprendizaje y de gestión. También se plantea que este producto sea usado desde el punto de vista productivo para la gestión de fincas de pequeños productores ya que de acuerdo a la federación nacional de ganaderos FEDEGAN este segmento representa el 80% del total de los productores ganaderos del país los cuales tienen menos de 25 hectáreas y 21 animales, a su vez la gestión se hace desde dos escenarios el primero centrados en el animal y un escenario general o de hato lo que permite la toma de decisiones basadas en datos confiables.

12. Anexos: Capturas de pantalla, diagramas, bitácora de desarrollo, versionado, entre otros.



12.1 Manejo de Versiones

Para el manejo de versiones se utiliza la herramienta github de la cual se incluyen pantallazos del historial de cabios en GitHub Desktop

Pantallazo del Frontend

The screenshot displays the GitHub Desktop interface. On the left, the 'Changes' tab shows a list of commit messages, with 'Refactor CRUD components and optimize hooks usage' selected. The main area shows a diff view for the file 'src/components/common/AdminCRUDPage.tsx'. The diff highlights changes in imports and state management, such as removing unnecessary imports and using state setters.

```
Removed unnecessary memo imports and unused code, improved hook dependencies, and replaced default exports of grouped models with named exports. Enhanced performance and maintainability by using usememo and usecallback where appropriate, and cleaned up unused icon imports. Also fixed some effect dependencies and improved event handling in image upload components.

84 changed files
src/components/common/AdminCRUDPage.tsx
src/components/.../AdminCRUDPage.tsx
src/components/.../CRUDModels.tsx
src/components/.../CRUDPagination.tsx
src/components/.../CRUDSearch.tsx
src/components/.../CRUDToolBar.tsx
src/.../OptimizedAdminCRUDPage.tsx
src/components/common/.../index.ts
src/components/.../ForeignKeyHelpers.tsx
src/components/.../ForeignKeyLink.tsx
src/components/.../GenericModal.tsx
src/components/.../HelpTooltip.tsx
src/components/.../ImageManager.tsx
src/components/.../ModalStyles.tsx
src/componen.../PWAAUpdateHandler.tsx
src/componen.../AnimalActionsMenu.tsx
src/componen.../AnimalHistoryModal.tsx
src/componen.../BreedActionsMenu.tsx
src/compon.../DescendantsTreeModal.tsx
src/components/.../FieldActionsMenu.tsx
src/components/.../GeneticTreeModal.tsx
src/compon.../OptimizedDashboard.tsx
src/components/.../RoleBasedSidebar.tsx
src/components/.../UserActionsMenu.tsx
src/components/dash.../AnimalCard.tsx
src/componen.../AnimalImageGallery.tsx
src/compon.../AnimalImagePreUpload.tsx
src/compon.../AnimalImageUpload.tsx
src/componen.../DashboardLayout.tsx
```

Pantallazo del Backend



The screenshot shows a Git commit diff for the commit titled "Fix cache warmup logic and improve API authorizations". The commit message is: "Explicitly disables cache warmup in testing mode and refines the check for CACHE_WARMUP_ENABLED in app initialization. Refactors API authorizations to use ordereddict for consistent ordering and maintainability. Removes duplicate /docs/ route and updates related test and output files." The diff shows changes in several files, including `app_init.py`, `app.py`, `pytest_unassign_output_2.py`, `pytest_unassign_output_3.py`, `pytest_unassign_output_4.py`, `pytest_unassign_output_5.py`, and `test_unassign_animal.py`. The code changes include adding a cache warmup check, refactoring API authorizations, and updating tests.

12.2 Anexos:

Anexo 2 Enlaces a git y Kanban: <https://cautious-tuba-5a8.notion.site/Proyecto-Finca-Villa-Luz-2c4931baaafe80708671db5360bd1b0d>

Anexo 3 Especificacion_Requisitos_VillaLuz.pdf

Anexo 4 Historias de usuario finca villa luz .pdf

Anexo 5 Modelo Entidad Relacion.mwb. para poder visulizar se requiere descargar mysqlworkbeach <https://dev.mysql.com/downloads/windows/installer/8.0.html>

Anexo 6 Flask Microservicios.docx

Anexo 7 Guia del frontend.md. Para poder visualizar se requiere descargar vs code <https://code.visualstudio.com/Download>

Anexo 8 Guia backend.md. Para poder visualizar se requiere descargar vs code <https://code.visualstudio.com/Download>

Anexo 9 Manejo de versiones con GitHub.docx



13. Archivo ejecutable o instalador del software: Preferiblemente acompañado de un video demostrativo.

Anexo 10 Carpeta comprimida con el backend y frontend y guía para sincronizar con el repositorio oficial en Github