



**Tecnicatura Universitaria en Programación
2025
UTN – FRT**

Trabajo Final Integrador

SOFTVET

Sistema de Gestión para Veterinarias

Desarrolladores	N° de Legajo
Valdez Gramajo, David	61181
Celiz, Mariano	61134
Pastrana, Angel	61105
Joaquin, Diaz	61726

Contenido

1. Resumen.....	7
2. Estado del arte	8
Sistemas de Gestión Específicos para Veterinaria	8
Tendencias de Digitalización en PyMEs del Sector Salud	8
Tecnologías Relevantes para el Proyecto	8
Casos de Éxito Regionales	9
Análisis del Mercado	9
Panorama Global y Soluciones Específicas	9
ERP Genéricos y Tendencias de PyMEs	10
El Enfoque de Soluciones a Medida.....	10
Tema a Desarrollar	11
Planteamiento del Problema:.....	11
3. Acta de constitución del proyecto	12
1. Justificación	12
2. Objetivo General	12
3. Objetivos Específicos	13
4. Alcance y Limitaciones	14
Alcance:.....	14
Limitaciones:	14
5. Entregables.....	14
6. Criterios de Éxito	15
7. Cronograma Inicial (en Sprints)	15
4. Alcance – Exclusiones – Supuestos – Restricciones	16
Alcance del Proyecto	16
Exclusiones del Proyecto	17
Supuestos del Proyecto	18
Restricciones del Proyecto	18
5. Requerimientos	20
Requerimientos Funcionales	20
1 Gestión de Roles.....	20
2 Gestión de Veterinario	20
3 Gestión de Empleados.....	21

4 Gestión de Mascotas	21
5 Gestión de Turnos	22
6 Gestión de Clientes.....	22
7 Gestión de Razas	23
8 Gestión de Especies.....	23
9 Gestión de Historia Clínica.....	24
10 Gestión de Detalles de Historias Clínicas	24
11 Gestión de Stock.....	25
12 Gestión de Sucursales	25
13 Lectura de Sucursales	26
14 Gestión de Categorías	26
15 Gestión de Productos	27
16 Gestión de Compras	27
17 Gestión de Detalles de Compras	28
18 Lectura de Compras.....	28
19 Lectura de Detalles de Compras	29
20 Gestión de Proveedores	29
21 Lectura de Proveedores.....	30
22 Gestión de Ventas	31
23 Gestión de Detalles de Ventas.....	32
Requerimientos No Funcionales	33
Introducción	33
1.0 Rendimiento	33
2.0 Seguridad	34
3.0 Fiabilidad/Disponibilidad	35
4.0 Usabilidad	36
5.0 Mantenibilidad.....	37
6. Historias de usuario y sus criterios de aceptación	38
Gestión de Roles	38
Gestión de Veterinarios.....	38
Gestión de empleados.....	39
Gestión de mascotas	40
Gestión de turnos.....	41
Gestión de clientes.....	42
Gestión de razas.....	43

Gestión de especies.....	44
Gestion de historias clinica.....	45
Gestion de detalle de historias clinicas	46
Gestion de stock.....	47
Gestion de sucursales.....	48
Lectura de sucursales	49
Gestión de categorías.....	49
Gestion de productos.....	50
Gestion de compras	51
Gestion de detalles de compras	52
Lectura de compras.....	53
Lectura de detalles de compra	53
Gestion de proveedores.....	54
Lectura de proveedores	54
Gestion de ventas.....	55
Gestion de ventas(recepcionista).....	55
Gestion de detalles de ventas	56
Gestion de detalles de ventas (recepcionista).....	57
7. Diagramas	59
1. Diagrama de Entidad-Relación (DER)	59
2. Modelo Relacional.....	61
8. Prototipos de interfaz.....	63
Página Principal (Dashboard o Home).....	64
Menú de Navegación	65
Modal de Gestión (Crear/Editar/Configurar).....	66
Vista de Detalles (Ver un elemento específico).....	67
9. Arquitectura del sistema	68
1. Patrón Arquitectónico (MVC / Cliente-Servidor)	68
2. Diagrama de Arquitectura (Nivel Alto)	68
3. Descripción de Componentes (El Stack Tecnológico)	68
A. Capa de Cliente (Frontend - React)	68
B. Capa de Servidor (Backend - Node.js & Express).....	69
C. Capa de Datos (Database - MySQL)	69
4. Flujo de Información (Cómo se conectan).....	70

10. Plan de pruebas	70
1. Introducción y Objetivos	70
2. Alcance de las Pruebas (Scope)	70
3. Entorno de Pruebas	71
4. Estrategia y Herramientas	71
6. Casos de Prueba (Muestra Representativa)	72
11. Manual de usuario	73
12. Manual técnico	73
Información para mantenimiento y evolución del sistema	73
Arquitectura, configuración y procedimientos de actualización	73
Arquitectura general	73
Configuración del entorno de desarrollo	74
Procedimientos de actualización	74
Secciones mínimas	74
Requisitos del sistema:	74
Diseño del sistema:	76
Estructura principal del Frontend:	76
Estructura principal del backend:	77
Estructura de base de datos	78
Api:	79
Empleados	79
Categorías	79
Clientes	80
Compras	80
Detalle Historia Clínica	81
Detalle de Compras	81
Detalle de Ventas	82
Empleados	83
Especies	84
Historia Clínica	84
Mascotas	85
Productos	86
Proveedores	86
Razas	87
Roles	87

Stock.....	88
Sucursales	88
Turnos	89
Ventas	89
13. Conclusion	91

Resumen

El presente proyecto consiste en la implementación de un **Sistema de Gestión Web integral**, desarrollado para **una clínica o red de clínicas veterinarias**.

La iniciativa surge a partir de la necesidad crítica de **modernizar y optimizar la gestión de pacientes, la historia clínica, la agenda de turnos y el inventario**. Hasta el momento, muchas de las operaciones clave de la clínica, como el registro de citas, la elaboración de fichas médicas y el control de *stock* de medicamentos, se manejan mediante métodos tradicionales (papel, hojas de cálculo), lo cual genera ineficiencias, riesgos de pérdida de información y errores operativos.

La solución propuesta es un **Sistema de Gestión Web** que permite centralizar y digitalizar las operaciones clave de la empresa, específicamente en las siguientes áreas:

1. **Gestión Médica:** Registro inmutable y acceso rápido a la Historia Clínica de cada mascota.
2. **Gestión de Operaciones:** Control de Turnos (evitando sobretornos y validando disponibilidad de veterinarios).
3. **Administración:** Control preciso del Inventario por sucursal, registro de Compras, Ventas y generación de reportes automáticos.

El proyecto se estructura en fases que incluyen el **Análisis de Requerimientos** (identificación de reglas de negocio específicas), el **Diseño del Sistema** (prototipos de interfaz de usuario y de la Historia Clínica), el **Desarrollo e Implementación** del *software* (cumpliendo con los requerimientos de rendimiento 1.5s) y la **Capacitación del Personal** (Administradores, Veterinarios y Recepcionistas).

El éxito del proyecto se reflejará en la **centralización de la información de pacientes**, la **reducción del tiempo invertido en tareas de recepción**, la **eliminación de errores de stock y facturación** y una **mejora en la toma de decisiones médicas y administrativas** a través de informes en tiempo real.

Estado del arte

En el ámbito de la digitalización y modernización de la gestión empresarial, existen numerosas herramientas y plataformas tecnológicas que abordan las necesidades específicas de las **clínicas y hospitales veterinarios**. A continuación, se analizan las tendencias actuales en este sector y su relación con las características del presente proyecto.

Sistemas de Gestión Específicos para Veterinaria

El mercado cuenta con soluciones de gestión de clínica (*Practice Management Software* o PMS) como **Cloudvet** o **ezyVet**. Estas plataformas, al igual que los grandes ERP en otros sectores, ofrecen funcionalidades avanzadas como la gestión de laboratorios, el historial clínico detallado con diagramas y la facturación electrónica avanzada. Sin embargo, su implementación suele estar orientada a **grandes hospitales veterinarios o cadenas de clínicas** debido a sus **altos costos y complejidad** de configuración inicial. Esto representa una barrera para las **Pequeñas y Medianas Clínicas (PyMEs)** que buscan soluciones más **personalizadas y accesibles**.

Tendencias de Digitalización en PyMEs del Sector Salud

Para empresas de menor tamaño, la tendencia de digitalización se centra en sistemas que ofrecen una alternativa **modular y personalizable**. Mientras que algunas recurren a sistemas ERP genéricos como **Odoo** (que permite gestionar inventarios y recursos humanos), la mayoría de las clínicas más pequeñas optan por **desarrollos propios y a medida**. Este enfoque es preferido porque permite una adaptación exacta a sus procesos internos específicos (ej. el flujo de atención de un veterinario) y ayuda a **reducir los costos de implementación**.

Tecnologías Relevantes para el Proyecto

El proyecto se diferencia por su enfoque en la creación de una solución a medida que se beneficia de las mejores prácticas observadas, mientras utiliza una pila tecnológica **moderna, robusta y escalable**.

- **Tecnología Central (Stack):** Se utiliza **React** para el desarrollo del *frontend* y **Express** con **Node.js** para el *backend*. La información se

gestiona en una **base de datos relacional MySQL** , lo que asegura la integridad y centralización de datos clave como la Historia Clínica.

- **Diseño y Usabilidad:** Para garantizar una experiencia de usuario (UX) óptima y un diseño *responsive* , se emplearon *frameworks* de diseño como **TailwindCSS y Bootstrap**.
- **Seguridad y Mantenimiento:** El enfoque modular del proyecto se alinea con las tendencias de software moderno, facilitando su mantenimiento. Además, la implementación de estándares de seguridad como **Json Web Token (JWT)** y **BCrypt** garantiza la confiabilidad del sistema y la protección de las credenciales de los empleados.

Casos de Éxito Regionales

A nivel local, muchas PyMEs han optado por soluciones personalizadas similares para enfrentar sus desafíos de digitalización. Por ejemplo, en el sector de la salud y servicios profesionales, la implementación de sistemas internos desarrollados con tecnologías web modernas ha permitido a las empresas **centralizar datos y optimizar procesos administrativos**, logrando **reducir costos y tiempos operativos significativamente**

Análisis del Mercado

El mercado de sistemas de gestión empresarial para el **sector de clínicas veterinarias** se caracteriza por una amplia oferta de herramientas tecnológicas diseñadas para optimizar la atención al paciente y mejorar la productividad administrativa.

Panorama Global y Soluciones Específicas

Entre las soluciones más destacadas se encuentran plataformas internacionales de Gestión de Práctica Veterinaria (*Veterinary Practice Management Software* o VPMS) como **ezyVet** o **Cloudvet**. Estas plataformas ofrecen funcionalidades específicas para la gestión médica, la planificación de tratamientos y la supervisión detallada de pacientes. Sin embargo, estas herramientas suelen ser **costosas** y están orientadas principalmente a **grandes hospitales o cadenas veterinarias**, lo que limita significativamente su accesibilidad y justificación de costos para organizaciones de menor tamaño.

ERP Genéricos y Tendencias de PyMEs

Por otro lado, sistemas ERP genéricos como **Odoo** han ganado popularidad debido a su flexibilidad y capacidad de personalización. Estas soluciones permiten integrar la gestión de inventarios (especialmente importante para el control de *stock* de medicamentos y alimentos), finanzas y recursos humanos en una misma plataforma, ofreciendo una alternativa más económica y adaptable para pequeñas y medianas empresas (PyMEs).

El Enfoque de Soluciones a Medida

A nivel local y en clínicas de menor escala, muchas empresas optan por desarrollar **soluciones personalizadas** que se ajusten con precisión a sus necesidades operativas específicas. Este enfoque es crucial en el sector veterinario, donde los flujos de trabajo (desde la reserva de turnos hasta la actualización de la historia clínica) pueden variar significativamente. Un desarrollo a medida permite una **mayor alineación con los procesos internos** de la clínica, evitando costos asociados a funcionalidades que nunca se utilizarán y reduciendo los costos de implementación.

El presente proyecto se inserta en este último contexto, buscando combinar las mejores prácticas del mercado (centralización de datos, automatización) con un **enfoque personalizado**. La implementación de un sistema de gestión web a medida para la veterinaria no solo optimizará sus procesos internos (agenda, stock, ventas), sino que también **mejorará la precisión y la calidad de la atención médica** a través de la Historia Clínica digitalizada.

Tema a Desarrollar

Planteamiento del Problema:

La clínica veterinaria enfrenta un desafío considerable en la gestión y administración de sus operaciones. La dependencia de métodos tradicionales (hojas de cálculo, agendas de papel y ficheros físicos) para el manejo de información esencial conlleva una serie de deficiencias significativas que impactan la eficiencia operativa y, fundamentalmente, la calidad de la atención al paciente.

Entre las principales deficiencias identificadas, se destacan:

- **Pérdida de la Trazabilidad de la Historia Clínica:** La información médica de las mascotas no está centralizada, lo que dificulta el acceso rápido a diagnósticos y tratamientos anteriores por parte del veterinario, introduciendo riesgos en la atención.
- **Gestión Ineficiente de la Agenda:** La administración manual de los turnos conlleva a errores frecuentes como los **sobretornos**, generando frustración en los clientes y demoras en la atención médica.
- **Errores en Inventario y Rentabilidad:** El control manual del **stock de medicamentos y alimentos** por sucursal es propenso a errores, resultando en desabastecimiento o pérdidas por productos caducados, y afectando directamente la rentabilidad del negocio.
- **Tiempo Excesivo en Tareas Administrativas:** El personal de recepción invierte demasiado tiempo en buscar fichas, registrar ventas manualmente y consolidar reportes, lo que reduce su enfoque en la atención al cliente.
- **Falta de Información Estratégica:** Las dificultades en la **generación de reportes de ventas y estadísticas** (RF 17.0.0) complican la toma de decisiones estratégicas sobre qué servicios o productos son más rentables.

La solución propuesta consiste en la implementación de un **Sistema de Gestión Web** que digitalizará la ficha clínica, optimizará el flujo de trabajo de turnos (evitando sobretornos) y garantizará el control de inventario en tiempo real. Esta herramienta permitirá a la clínica

superar sus desafíos, optimizando su operatividad y fortaleciendo la calidad de su servicio médico.

Acta de constitución del proyecto

1. Justificación

El sistema de Gestión de Veterinaria “SOFTVET” busca agilizar la atención y la gestión de una Veterinaria, actualmente está enfocado en llevar el registro de los pacientes mediante Historia Clínica, ventas y gestión de productos, actividad de los clientes, turnos para peluquería canina, control de personal.

Actualmente, la gestión se está realizando en papel (historias clínicas, facturación a mano, turnos para peluquería presenciales con agenda programada), lo cual se pierden al menos un promedio de 8 turnos al mes por errores de agenda manual, también esto genera demoras, confusiones y sobrecarga administrativa.

Con esta solución se pretende reducir un 70% de tiempo en la gestión administrativa y minimizar errores en la facturación.

2. Objetivo General

Desarrollar un sitio web, el cual tendrá una sección informativa y publicitaria sobre el local, permitir la reserva de turnos para peluquería canina, además tendrá una sección administrativa para poder gestionar el inventario, los clientes y las historias clínicas, que a lo cual solo accederán los empleados de la veterinaria.

3. Objetivos Específicos

- Desarrollar una landing page la cual debe mostrar datos de ubicación, contacto, horarios y servicios principales.
- Implementar un módulo de autenticación de empleados para el acceso a la sección administrativa.
- Implementar un módulo para gestionar inventario que solo puedan ingresar los empleados.
- Implementar un módulo para que los empleados puedan gestionar las historias clínicas.
- Cumplir con seguridad avanzada para proteger los datos de los clientes con historias clínicas y las ventas.
- Implementar un módulo para gestionar médicos veterinarios, clientes, productos, turnos y empleados administrativos.

4. Alcance y Limitaciones

Alcance:

- Gestión de Clientes, Médicos Veterinarios, Productos, Ventas, Compras de reesctock, Proveedores y Turnos.
- Reserva, modificación y cancelación de turnos de Peluquería.
- Notificaciones automáticas mediante un Chat Bot vía WhatsApp.

Limitaciones:

- En esta versión no se incluirá pago en línea mediante Billeteras Virtuales.
- No se contempla integración con sistemas externos de historia clínica.
- No se permitirán pagos que no sean en efectivo.

5. Entregables

- Documento de requisitos funcionales.
- Diseño de Base de datos y diagramas UML.
- Prototipos de interfaz de usuario (mockups).
- Código fuente en repositorio (GitHub).
- Manual de Usuario y Manual Técnico.

6. Criterios de Éxito

- El sistema debe responder a las solicitudes en menos de 2 segundos el 98% del tiempo.
- La aplicación debe ser compatible con Navegadores Chrome, Firefox y Edge.
- El usuario debe poder reservar su turno en menos de 3 minutos.
- La interfaz debe ser clara y amigable, validada mediante pruebas con usuarios de prueba.

7. Cronograma Inicial (en Sprints)

- ***Sprint 1 (Semanas 1-2): Relevamiento de Requisitos y diseño Inicial.***
- ***Sprint 2 (Semanas 3): Implementación de Módulos de Gestión de Usuarios Administrativos y Médicos Veterinarios.***
- ***Sprint 3 (Semanas 4): Implementación de Modulo de turnos y disponibilidad horaria.***
- ***Sprint 4 (Semana 5): Ajustes, pruebas, test con usuarios reales, validaciones finales.***
- ***Sprint 5 (Semana 6): Documentación, presentación y Entrega Final.***

Alcance – Exclusiones – Supuestos – Restricciones

Alcance del Proyecto

El proyecto abarca el desarrollo de un Sistema de Gestión Web integral, diseñado para centralizar todas las operaciones de la clínica veterinaria, desde la atención al paciente hasta la administración financiera del *stock* y las ventas.

- **Gestión de Pacientes:**

Registro de Clientes con DNI/correo electrónico único. Una mascota debe estar asociada a una raza y un cliente. Gestión completa de Especies y Razas, donde una raza está vinculada obligatoriamente a una especie.

- **Gestión de Citas y Clínica:**

Módulo de **Turnos** asociado a un Cliente, Mascota y Veterinario. Validación de **No Sobretornos** (superpuestos en fecha y hora para el mismo veterinario). Registro y actualización de la **Historia Clínica** y observaciones en cada atención.

- **Gestión de Personal:**

Gestión de Empleados y Roles, asegurando que el DNI del empleado sea único. Cada usuario debe tener obligatoriamente un rol asignado.

- **Stock y Transacciones:**

Gestión de **Productos** asociados a Categoría y, opcionalmente, a Especie. Control de **Stock por Sucursal**. Registro de Compras, Proveedores y Ventas. La compra actualiza automáticamente el stock.

Exclusiones del Proyecto

Los siguientes elementos han sido definidos como fuera de los límites del proyecto en su versión inicial, basándose en la información de los RF y un alcance razonable para una primera fase:

- **Facturación con Medios de Pago Diversos:**

Las funcionalidades de Ventas y Compras solo contemplan el registro de transacciones. **No se incluye la integración con pasarelas de pago** (Tarjetas de crédito/débito, QR, *links* de pago).

- **Contabilidad Avanzada y Liquidación:**

El sistema incluye módulos de compras, ventas y reportes básicos (estadísticas), pero **no se incluye un módulo ERP completo** para la gestión de activos fijos, contabilidad general, impuestos, o liquidación de sueldos (nómina).

- **Integración de Comunicación Externa:**

El sistema se centra en la gestión interna. **No se contempla el envío automatizado** de correos de confirmación de turno, recordatorios por SMS o la gestión de marketing hacia clientes.

- **Gestión de Infraestructura de *Hardware*:**

No incluye la provisión, configuración o mantenimiento del equipo de *hardware* del lado del cliente, solo se especifican los requisitos mínimos para el terminal de usuario.

Supuestos del Proyecto

Estos factores se asumen como condición de verdad, y su incumplimiento constituye un riesgo significativo para la finalización del proyecto:

- **Disponibilidad de Roles Clave para Validación:**

Se asume la disponibilidad periódica de usuarios con rol de **Administrador** (para la configuración inicial de roles y sucursales) y de **Veterinario** (para validar el flujo crítico de la Historia Clínica).

- **Estabilidad del Precio de Productos:**

Se asume que la **actualización de los precios de productos** se realizará de manera periódica y manual en la tabla Productos, y no mediante integración automatizada con proveedores.

- **Conformidad con Requisitos de Seguridad:**

Se asume que el entorno de despliegue (producción) permitirá la instalación y configuración del protocolo de seguridad exigido (SSL/TLS) y la arquitectura de almacenamiento de credenciales.

- **Disponibilidad de Datos Históricos:**

Se asume que la migración inicial de datos (clientes, mascotas, productos) será posible mediante la provisión de información estructurada por parte de la veterinaria.

Restricciones del Proyecto

El proyecto debe adherirse a las siguientes limitaciones obligatorias de rendimiento, seguridad y reglas de negocio, definidas en los RNF:

- **Rendimiento:** La carga de la Ficha Clínica no puede exceder los **1.5s**. La validación de sobretornos debe ser **200ms**. La actualización del Stock debe ser **1s**
- **Seguridad Crítica:** El acceso y modificación de la Historia Clínica es **exclusivo para el Veterinario**. Se exige el registro de cualquier cambio en registros médicos mediante un **log de auditoría inmutable**
- **Fiabilidad y Recuperación:** El sistema debe tener una **disponibilidad mínima del 99.5 %**. Se exige una **copia de seguridad completa diaria** de la base de datos. Recuperación total ante fallo (RTO) en **4 horas**.
- **Regla de Negocio (Stock):** No se permite la finalización de una venta o transacción si el producto asociado **no cuenta con stock suficiente** en la sucursal correspondiente.
- **Usabilidad y Flujo:** La interfaz debe ser **responsive** y compatible con tabletas. El registro de un nuevo turno debe completarse en **máximo 4 pasos**.

Requerimientos

Requerimientos Funcionales

1 Gestión de Roles

Número de requisito	RF 1.0.0		
Nombre de requisito	Gestión de Roles		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de administrador crear, modificar, eliminar y visualizar los diferentes tipos de roles de usuario , con el fin de mantener actualizada la estructura de permisos del sistema.		

2 Gestión de Veterinario

Número de requisito	RF 2.0.0		
Nombre de requisito	Gestión de Veterinario		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de administrador visualizar, crear, modificar y eliminar los registros de veterinarios , con el fin de mantener actualizada la información del personal profesional.		

3 Gestión de Empleados

Número de requisito	RF 3.0.0		
Nombre de requisito	Gestión de Empleados		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de administrador visualizar, crear, modificar y eliminar los registros de empleados , así como asociar un rol a cada uno de ellos, con el fin de administrar correctamente los accesos y responsabilidades dentro del sistema.		

4 Gestión de Mascotas

Número de requisito	RF 4.0.0		
Nombre de requisito	Gestión de Mascotas		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de recepcionista visualizar, crear, modificar y eliminar los registros de mascotas . Cada mascota debe estar asociada a una raza y a un cliente propietario , con el objetivo de mantener organizada y actualizada la información de los animales atendidos.		

5 Gestión de Turnos

Número de requisito	RF 5.0.0		
Nombre de requisito	Gestión de Turnos		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de repcionista visualizar, crear, modificar y eliminar los turnos registrados. Además, el sistema debe validar que no se generen sobretornos (turnos superpuestos en fecha y hora) y permitir la asignación de un estado válido para cada turno: Pendiente, Confirmado, Cancelado o Atendido.		

6 Gestión de Clientes

Número de requisito	RF 6.0.0		
Nombre de requisito	Gestión de Clientes		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de repcionista visualizar, crear, modificar y eliminar registros de clientes . Al momento de crear un nuevo cliente, el sistema debe validar que el número de DNI sea único , para evitar duplicidad en los registros.		

7 Gestión de Razas

Número de requisito	RF 7.0.0		
Nombre de requisito	Gestión de Razas		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de recepcionista visualizar, crear, modificar y eliminar los registros de razas , con el fin de mantener actualizada la información necesaria para la correcta identificación de las mascotas dentro del sistema.		

8 Gestión de Especies

Número de requisito	RF 8.0.0		
Nombre de requisito	Gestión de Especies		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir al usuario con rol de recepcionista visualizar, crear, modificar y eliminar registros de especies , con el fin de mantener actualizada la información utilizada en la clasificación de las mascotas dentro del sistema.		

9 Gestión de Historia Clínica

Número de requisito	RF 9.0.0		
Nombre de requisito	Gestión de Historia Clínica		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir a los usuarios con rol de recepcionista y veterinario visualizar, crear, modificar y eliminar registros de historias clínicas . Cada historia clínica debe estar asociada a una mascota y contener la información relevante de sus atenciones médicas, tratamientos y observaciones.		

10 Gestión de Detalles de Historias Clínicas

Número de requisito	RF 9.0.1		
Nombre de requisito	Gestión de Detalles de Historias Clínicas		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	El sistema debe permitir a los usuarios con rol de recepcionista y veterinario visualizar, crear, modificar y eliminar los detalles de las historias clínicas , los cuales contienen información específica de cada atención, Estos detalles deben estar asociados a una historia clínica existente .		

11 Gestión de Stock

Número de requisito	RF 10.0.0		
Nombre de requisito	Gestión de Stock		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir a los usuarios con rol de repcionista y administrador visualizar, crear, modificar y eliminar registros de stock asociados a productos existentes.</p> <p>Cada registro de stock deberá incluir la cantidad disponible, fecha de ingreso, observaciones y la sucursal correspondiente.</p> <p>El sistema debe mantener actualizada la cantidad total disponible de cada producto según los movimientos de stock, garantizando una gestión de inventario.</p>		

12 Gestión de Sucursales

Número de requisito	RF 11.0.0		
Nombre de requisito	Gestión de Sucursales		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir al usuario con rol de administrador visualizar, registrar, modificar y eliminar sucursales.</p> <p>Cada sucursal deberá contener información básica como nombre, dirección, teléfono.</p> <p>El sistema deberá garantizar la integridad de los datos y evitar la duplicación de sucursales con el mismo nombre o dirección.</p>		

13 Lectura de Sucursales

Número de requisito	RF 11.0.1		
Nombre de requisito	Lectura de Sucursales		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir al usuario con rol de recepcionista visualizar la información de las sucursales registradas.</p> <p>El usuario podrá consultar datos como el nombre, dirección, teléfono de cada sucursal.</p> <p>El acceso será de solo lectura, sin la posibilidad de crear, modificar o eliminar registros.</p>		

14 Gestión de Categorías

Número de requisito	RF 12.0.0		
Nombre de requisito	Gestión de Categorías		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir a los usuarios con rol de recepcionista y administrador visualizar, registrar, modificar y eliminar categorías de productos.</p> <p>Cada categoría deberá incluir información básica como nombre, garantizando que no existan duplicados.</p> <p>Esta funcionalidad permitirá organizar adecuadamente los productos dentro del sistema y facilitar su gestión.</p>		

15 Gestión de Productos

Número de requisito	RF 13.0.0		
Nombre de requisito	Gestión de Productos		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir a los usuarios con rol de recepcionista y administrador visualizar, registrar, modificar y eliminar productos.</p> <p>Cada producto debe incluir información como nombre, código, precio y categoría asociada, asegurando la integridad y la correcta clasificación dentro del sistema.</p> <p>Esta funcionalidad permitirá mantener actualizado el catálogo de productos y facilitar su gestión en relación con el stock.</p>		

16 Gestión de Compras

Número de requisito	RF 14.0.0		
Nombre de requisito	Gestión de Compras		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir al usuario con rol de administrador visualizar, registrar(solo con efectivo), modificar y eliminar compras de productos.</p> <p>Cada compra deberá incluir información como total, fecha y hora, id de empleado, asegurando un control preciso sobre el inventario y la gestión financiera de la clínica.</p>		

17 Gestión de Detalles de Compras

Número de requisito	RF 14.0.1		
Nombre de requisito	Gestión de Detalles de Compras		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir al usuario con rol de administrador visualizar, crear, modificar y eliminar los detalles de cada compra realizada.</p> <p>Cada detalle debe estar asociado a una compra existente e incluir información como cantidad, precio unitario, sub total, id de producto y id de compra, asegurando un registro preciso de cada adquisición y su impacto en el stock.</p>		

18 Lectura de Compras

Número de requisito	RF 14.0.2		
Nombre de requisito	Lectura de Compras		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir al usuario con rol de recepcionista visualizar las compras realizadas.</p> <p>El acceso será solo de lectura, mostrando información como total, fecha y hora, id de empleado, sin permitir crear, modificar o eliminar registros.</p>		

19 Lectura de Detalles de Compras

Número de requisito	RF 14.0.3		
Nombre de requisito	Lectura de Detalles de Compras		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir al usuario con rol de recepcionista visualizar los detalles de las compras efectuadas, incluyendo información de cantidad, precio unitario, sub total, id de producto y id de compra.</p> <p>El acceso será solo de lectura, sin permitir crear, modificar o eliminar registros.</p>		

20 Gestión de Proveedores

Número de requisito	RF 15.0.0		
Nombre de requisito	Gestión de Proveedores		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir a los usuarios con rol de administrador visualizar, registrar, modificar y eliminar proveedores.</p> <p>Cada proveedor debe incluir información como nombre, dirección, teléfono y mail.</p> <p>Esta funcionalidad permitirá mantener actualizado el catálogo de proveedores y asegurar la correcta gestión de compras y stock.</p>		

21 Lectura de Proveedores

Número de requisito	RF 15.0.1		
Nombre de requisito	Lectura de Proveedores		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir al usuario con rol de recepcionista visualizar los proveedores registrados en el sistema.</p> <p>El acceso será solo de lectura, mostrando información como nombre, dirección, teléfono y mail.</p> <p>No se permitirá crear, modificar ni eliminar proveedores.</p>		

Número de requisito	RF 16.0.0		
Nombre de requisito	Gestión de Ventas		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir:</p> <ul style="list-style-type: none"> Al usuario con rol de administrador: visualizar, registrar(solo con efectivo), modificar y eliminar ventas. Al usuario con rol de recepcionista: visualizar y registrar ventas(solo con efectivo), pero no eliminar registros. <p>Cada venta deberá incluir información como fecha y hora, cliente, id empleado, sucursal y total</p> <p>Esta funcionalidad permitirá controlar correctamente las transacciones y mantener actualizado el stock de productos.</p>		

Número de requisito	RF 16.0.1		
Nombre de requisito	Gestión de Detalles de Ventas		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción del requisito	<p>El sistema debe permitir:</p> <ul style="list-style-type: none"> Al usuario con rol de administrador: visualizar, registrar, modificar y eliminar los detalles de cada venta, incluyendo venta, producto, cantidad, precio unitario y subtotal. Cada detalle debe estar asociado a una venta registrada, garantizando la integridad de la información y la correcta actualización del stock. Al usuario con rol de recepcionista: visualizar y registrar los detalles de ventas, pero no eliminar registros. 		

Requerimientos No Funcionales

Introducción

El presente documento detalla los Requerimientos No Funcionales (RNF) que definen los criterios de calidad, eficiencia, seguridad y operatividad del Sistema de Gestión de Veterinaria. Estos requerimientos son esenciales para asegurar que el sistema cumpla con las expectativas operacionales de los usuarios (Administrador, Veterinario y Recepcionista).

1.0 Rendimiento

ID	Requerimiento No Funcional (RNF)	Métrica / Criterio de Aceptación	Justificación (RF Asociado / Módulo)
RNF-1.1	La visualización de la ficha clínica completa de una mascota debe cargarse en un tiempo máximo.	≤ 1.5 segundos en el 95% de los casos.	Flujo de trabajo del Veterinario (historiaClinica.js).
RNF-1.2	El sistema debe soportar una cantidad mínima de usuarios realizando transacciones intensivas simultáneamente.	Mínimo de 15 usuarios concurrentes.	Operación durante horas pico (Turnos, Ventas).
RNF-1.3	La generación de reportes de ventas (RF 17.0.0) para un rango de 30 días debe tener un tiempo máximo de procesamiento.	≤ 5 segundos.	RF 17.0.0 (Gestión de Estadísticas).
RNF-1.4	La validación de sobretornos (RF 5.0.0) debe ejecutarse de forma ágil.	≤ 200 milisegundos.	RF 5.0.0 (Gestión de Turnos) / Eficiencia de la Recepcionista.

RNF-1.5	La actualización del Stock al registrar una venta o una compra (RF 16.0.0, RF 14.0.0) debe ser una operación rápida.	≤ 1 segundo.	RF 10.0.0 / Garantizar la integridad del inventario.
----------------	---	-------------------	--

2.0 Seguridad

ID	Requerimiento No Funcional (RNF)	Métrica / Criterio de Aceptación	Justificación (RF Asociado / Módulo)
RNF-2.1	El acceso a los módulos de gestión de empleados y roles debe ser restringido por rol.	Exclusivo para el rol de Administrador .	RF 1.0.0, RF 2.0.0, RF 3.0.0 (Gestión de Personal y Roles).
RNF-2.2	Los permisos de escritura (Crear/Modificar) sobre la Historia Clínica deben ser exclusivos.	Exclusivo para el rol de Veterinario .	RF 9.0.0, RF 9.0.1 (Integridad de la Historia Clínica).
RNF-2.3	Las comunicaciones con el servidor que manejan información sensible de clientes y empleados deben estar cifradas.	Uso de protocolo SSL/TLS .	RF 6.0.0, RF 3.0.0 / Protección de datos personales.
RNF-2.4	El sistema debe utilizar un mecanismo de almacenamiento seguro para las credenciales de los empleados.	Uso de un algoritmo de <i>hashing</i> robusto (Ej: BCrypt o Argon2) para contraseñas.	RF 3.0.0 (Gestión de Empleados).
RNF-2.5	El sistema debe registrar cualquier cambio realizado en los registros médicos.	Implementación de un log de auditoría inmutable	RF 9.0.0, RF 9.0.1 (Auditoría Médica).

		para modificaciones en Historia Clínica.	
RNF-2.6	El sistema debe exigir el cumplimiento de una política de contraseñas.	Mínimo 8 caracteres, mayúsculas, minúsculas, números y símbolos.	Seguridad general de acceso.

3.0 Fiabilidad/Disponibilidad

ID	Requerimiento No Funcional (RNF)	Métrica / Criterio de Aceptación	Justificación (RF Asociado / Módulo)
RNF-3.1	El sistema debe mantener su operatividad durante el horario laboral.	Disponibilidad mínima del 99.5% (24/7 o en horario de atención).	Operación crítica de la clínica.
RNF-3.2	Se debe implementar un procedimiento de resguardo de datos.	Copia de seguridad completa de la base de datos diariamente .	Recuperación ante fallos.
RNF-3.3	El sistema debe recuperarse completamente ante una falla total (RTO).	Recuperación total en menos de 4 horas .	Continuidad del negocio (Turnos, Ventas).
RNF-3.4	El subsistema de Gestión de Turnos y Clientes (RF 5.0.0, RF 6.0.0) debe tener la máxima prioridad de restauración.	Prioridad 1 en el Plan de Recuperación de Desastres.	Flujos de trabajo críticos de la Recepcionista.

4.0 Usabilidad

ID	Requerimiento No Funcional (RNF)	Métrica / Criterio de Aceptación	Justificación (RF Asociado / Módulo)
RNF-4.1	La interfaz de gestión de Turnos (RF 5.0.0) debe minimizar los pasos para programar una cita.	No requerir más de 4 pasos para el registro de un nuevo turno.	RF 5.0.0 / Optimización de la Recepcionista.
RNF-4.2	El diseño de la interfaz debe ser adaptable.	Interfaz responsive y compatible con tabletas.	Uso del Veterinario en áreas de consulta.
RNF-4.3	Se debe proporcionar retroalimentación clara al usuario ante errores de entrada.	Mensajes de error claros que sugieran la acción correctiva.	Experiencia de usuario (Clientes, Mascotas).
RNF-4.4	El proceso inicial de creación de un nuevo Cliente (RF 6.0.0) debe ser rápido.	Máximo de 3 campos obligatorios en la primera pantalla de registro.	RF 6.0.0 / Agilizar el <i>check-in</i> de nuevos clientes.
RNF-4.5	La interfaz de Registro de Ventas (RF 16.0.0) debe incorporar una ayuda visual para encontrar productos.	Inclusión de función de búsqueda predictiva de productos.	RF 16.0.0 / Acelerar la transacción de venta.

5.0 Mantenibilidad

ID	Requerimiento No Funcional (RNF)	Métrica / Criterio de Aceptación	Justificación (RF Asociado / Módulo)
RNF-5.1	El código fuente debe adherirse a un conjunto de reglas para garantizar su uniformidad y legibilidad.	Uso obligatorio de un estándar de codificación (Ej: ESLint o Prettier).	Facilitar la incorporación de nuevos desarrolladores.
RNF-5.2	La arquitectura del sistema debe ser modular.	La modificación en un módulo (Ej: compras.js) no debe impactar la funcionalidad de otros módulos principales (Ej: ventas.js).	Baja cohesión/Alto acoplamiento.
RNF-5.3	El sistema debe mantenerse actualizado con las tecnologías base.	Compatibilidad con la última versión estable de Node.js y el motor de base de datos utilizado.	Portabilidad y seguridad futura.
RNF-5.4	La lógica de cálculo de estadísticas debe ser independiente de la lógica transaccional.	Módulo de Estadísticas (RF 17.0.0) desacoplado de Ventas y Stock.	Permitir modificaciones de reportes sin afectar la operación diaria.

Historias de usuario y sus criterios de aceptación

HU1	Gestion de Roles
Descripción	Como administrador del sistema , quiero gestionar los roles de usuario (crear, editar, eliminar y ver) para mantener actualizados los permisos y accesos dentro de la aplicación .
Criterios de aceptación	<ul style="list-style-type: none"> ☑ El administrador puede crear un nuevo rol, ingresando nombre ☑ El sistema valida que el nombre del rol no esté duplicado. ☑ El administrador puede eliminar un rol, previa confirmación mediante mensaje de advertencia. ☑ El sistema muestra un listado de todos los roles registrados (únicamente solo para administrador). ☑ Los cambios realizados se guardan correctamente en la base de datos. ☑ Solo los usuarios con rol de administrador pueden acceder a esta funcionalidad.

HU2	Gestion de Veterinarios
Descripción	Como administrador del sistema , quiero gestionar la información de los veterinarios (crear, editar, eliminar y visualizar) para mantener actualizada la base de datos del personal profesional .
Criterios de aceptación	<ul style="list-style-type: none"> ☑ El administrador puede registrar un nuevo veterinario, ingresando usuario, contraseña, nombre, dni, dirección, teléfono, mail, id de rol. ☑ El sistema valida los campos obligatorios antes de guardar. ☑ El administrador puede editar los datos de un veterinario existente. ☑ El administrador puede eliminar un veterinario, previa confirmación mediante mensaje de advertencia. ☑ El sistema permite visualizar un listado de todos los veterinarios registrados. ☑ Los cambios se guardan correctamente en la base de datos y se reflejan en la interfaz.

	<p>❑ Solo los usuarios con rol de administrador tienen acceso a esta funcionalidad.</p>
--	--

HU3	Gestion de empleados
Descripción	<p>Como <u>administrador</u>, quiero <u>gestionar los empleados (crear, editar, eliminar, ver y asignar roles)</u> para <u>mantener actualizada la información del personal y controlar sus permisos dentro del sistema</u>.</p>
Criterios de aceptación	<p>❑ El administrador puede crear un nuevo empleado, ingresando usuario, contraseña, nombre, dni, dirección, teléfono, mail, id de rol.</p> <p>❑ El sistema valida los campos obligatorios antes de guardar.</p> <p>❑ El administrador puede editar la información de un empleado existente, incluyendo su rol.</p> <p>❑ El administrador puede eliminar un empleado, previa confirmación mediante mensaje de advertencia.</p> <p>❑ El sistema permite visualizar una lista completa de empleados registrados junto con su rol asignado.</p> <p>❑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.</p> <p>❑ Solo los usuarios con rol de administrador pueden acceder a esta funcionalidad.</p>

HU4	Gestión de mascotas
Descripción	Como repcionista del sistema , quiero gestionar las mascotas (crear, editar, eliminar y visualizar) para mantener actualizada la información de los animales y su vínculo con los clientes propietarios .
Criterios de aceptación	<ul style="list-style-type: none"> ☑ El repcionista puede crear una nueva mascota, ingresando nombre, edad, sexo, id de raza, id de cliente y id de historia clínica ☑ El sistema valida que se asigne una raza y un propietario antes de guardar. ☑ El repcionista puede editar los datos de una mascota existente. ☑ El repcionista puede eliminar una mascota, previa confirmación mediante mensaje de advertencia. ☑ El sistema permite visualizar un listado completo de mascotas, mostrando nombre, raza y propietario asociado. ☑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz. ☑ Solo los usuarios con rol de repcionista o administrador pueden acceder a esta funcionalidad.

HU5	Gestion de turnos
Descripción	Como recepcionista del sistema , quiero gestionar los turnos (crear, editar, eliminar y visualizar) para organizar las citas de los clientes evitando sobretornos y manteniendo actualizada la información del estado de cada turno .
Criterios de aceptación	<ul style="list-style-type: none"> ☑ El recepcionista puede crear un nuevo turno, asignando fecha y hora, motivo, estado, id de cliente, id de mascota, id de empleado. ☑ El sistema valida que no existan sobretornos, es decir, que no haya otro turno con el mismo veterinario, fecha y hora. ☑ El recepcionista puede editar un turno existente, modificando datos o su estado. ☑ El recepcionista puede eliminar un turno, previa confirmación mediante mensaje de advertencia. ☑ El sistema permite visualizar una lista de turnos, mostrando cliente, mascota, veterinario, fecha, hora y estado. ☑ Los estados posibles de un turno son: <i>Pendiente, Confirmado, Cancelado y Atendido</i>. ☑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz. ☑ Solo los usuarios con rol de recepcionista o superior (por ejemplo, administrador) pueden acceder a esta funcionalidad.

HU6	Gestion de clientes
Descripción	Como repcionista , quiero gestionar los clientes (crear, editar, eliminar y visualizar) para mantener actualizada la información de los propietarios y evitar registros duplicados .
Criterios de aceptación	<ul style="list-style-type: none"> ☐ El repcionista puede registrar un nuevo cliente, ingresando nombre, DNI, teléfono, correo electrónico y dirección. ☐ El sistema valida que el DNI no se encuentre registrado previamente antes de guardar. ☐ El repcionista puede editar los datos de un cliente existente. ☐ El repcionista puede eliminar un cliente, previa confirmación mediante mensaje de advertencia. ☐ El sistema permite visualizar un listado completo de clientes registrados. ☐ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz. ☐ Solo los usuarios con rol de repcionista o superior (por ejemplo, administrador) pueden acceder a esta funcionalidad.

HU7	Gestion de razas
Descripción	Como recepcionista , quiero gestionar las razas (crear, editar, eliminar y visualizar) para mantener actualizada la base de datos y facilitar el registro correcto de las mascotas .
Criterios de aceptación	<ul style="list-style-type: none"> ☑ El recepcionista puede crear una nueva raza, ingresando nombre y especie correspondiente. ☑ El sistema valida que no existan razas duplicadas dentro de la misma especie. ☑ El recepcionista puede editar el nombre o especie de una raza existente. ☑ El recepcionista puede eliminar una raza, previa confirmación mediante mensaje de advertencia. ☑ El sistema permite visualizar un listado de todas las razas registradas, mostrando su especie asociada. ☑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz. ☑ Solo los usuarios con rol de recepcionista o superior (por ejemplo, administrador) pueden acceder a esta funcionalidad.

HU8	Gestión de especies
Descripción	Como Recepcionista , quiero gestionar las especies (crear, editar, eliminar y visualizar) para mantener actualizada la base de datos y asegurar la correcta clasificación de las mascotas registradas .
Criterios de aceptación	<ul style="list-style-type: none"> ☐ El recepcionista puede crear una nueva especie, ingresando su nombre. ☐ El sistema valida que el nombre de la especie no esté duplicado antes de guardar. ☐ El recepcionista puede editar el nombre de una especie existente. ☐ El recepcionista puede eliminar una especie, previa confirmación mediante mensaje de advertencia. ☐ El sistema permite visualizar un listado completo de especies registradas. ☐ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz. ☐ Solo los usuarios con rol de Recepcionista o superior (por ejemplo, administrador) pueden acceder a esta funcionalidad.

HU9	Gestion de historias clinica
Descripción	Como veterinario o recepcionista del sistema , quiero gestionar las historias clínicas (crear, editar, eliminar y visualizar) para mantener un registro completo y actualizado de las atenciones y tratamientos de cada mascota .
Criterios de aceptación	<ul style="list-style-type: none"> ☑ El usuario (recepcionista o veterinario) puede crear una nueva historia clínica, asociándola a una mascota existente. ☑ La historia clínica debe incluir datos como fecha y observaciones generales ☑ El sistema valida que la historia esté asociada a una mascota registrada antes de guardar. ☑ El usuario puede editar la información de una historia clínica existente. ☑ El usuario puede eliminar una historia clínica, previa confirmación mediante mensaje de advertencia. ☑ El sistema permite visualizar un listado de historias clínicas, mostrando mascota, fecha y veterinario responsable. ☑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.

	<p>❑ Solo los usuarios con rol de repcionista o veterinario pueden acceder a esta funcionalidad.</p>
--	---

HU10	Gestion de detalle de historias clinicas
Descripción	<p>Como veterinario o repcionista del sistema, quiero gestionar los detalles de las historias clínicas (crear, editar, eliminar y visualizar) para registrar con la evolución médica de las mascotas y mantener un historial completo de sus atenciones.</p>
Criterios de aceptación	<p>❑ El usuario (repcionista o veterinario) puede crear un nuevo detalle, asociándolo a una historia clínica existente.</p> <p>❑ El detalle debe incluir información como observaciones, id de historia clínica, id de sucursal y id de venta</p> <p>❑ El sistema valida que el detalle esté vinculado a una historia clínica registrada antes de guardar.</p> <p>❑ El usuario puede editar la información de un detalle existente.</p> <p>❑ El usuario puede eliminar un detalle, previa confirmación mediante mensaje de advertencia.</p> <p>❑ El sistema permite visualizar un listado de detalles asociados a una historia clínica específica.</p>

	<p>❑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.</p> <p>❑ Solo los usuarios con rol de veterinario o repcionista pueden acceder a esta funcionalidad.</p>
--	---

HU11	Gestion de stock
Descripción	Como repcionista o administrador del sistema , quiero gestionar el stock de productos (crear, editar, eliminar y visualizar) para mantener actualizada la cantidad disponible de productos asegurar una correcta administración del inventario .
Criterios de aceptación	<p>❑ El usuario (repcionista o administrador) puede registrar un nuevo movimiento de stock, seleccionando el producto correspondiente e indicando cantidad, fecha de ingreso, y observaciones, así como la sucursal asociada.</p> <p>❑ El sistema valida los campos obligatorios (producto, cantidad y sucursal) antes de guardar.</p> <p>❑ El usuario puede editar un registro de stock existente, modificando la cantidad, fecha de ingreso u observaciones.</p> <p>❑ El usuario puede eliminar un registro de stock, previa confirmación mediante mensaje de advertencia.</p>

	<p>☑ El sistema permite visualizar un listado completo del stock, mostrando el nombre del producto, código, categoría, cantidad disponible, fecha de ingreso y sucursal.</p> <p>☑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz de manera inmediata.</p> <p>☑ El sistema actualiza automáticamente la cantidad total disponible de cada producto cuando se registren movimientos de ingreso o egreso de stock (por ejemplo, uso en tratamientos o compras de reposición).</p> <p>☑ Solo los usuarios con rol de repcionista o administrador pueden acceder a esta funcionalidad.</p>
--	--

HU12	Gestion de sucursales
Descripción	<p>Como usuario con rol de administrador, quiero poder visualizar, crear, modificar y eliminar sucursales, para mantener actualizada la información de las diferentes sedes del sistema y asegurar una correcta organización administrativa.</p>
Criterios de aceptación	<p>☑ El administrador puede registrar una nueva sucursal completando los campos requeridos (nombre, dirección, teléfono).</p> <p>☑ El sistema valida que no exista otra sucursal con el mismo nombre o dirección antes de guardar.</p> <p>☑ El administrador puede editar los datos de una sucursal existente.</p> <p>☑ El administrador puede eliminar una sucursal, previa confirmación mediante mensaje de advertencia.</p>

	<p>❑ El sistema muestra un listado actualizado de todas las sucursales registradas.</p> <p>❑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.</p>
--	--

HU13	Lectura de sucursales
Descripción	<p><u>Como</u> usuario con rol de <u>repcionista</u>, <u>quiero</u> poder <u>ver la lista de sucursales registradas en el sistema</u>, <u>para</u> conocer la información de contacto y ubicación de cada sede disponible.</p>
Criterios de aceptación	<p>❑ El <u>repcionista</u> puede acceder a una sección del sistema que muestre el listado de sucursales.</p> <p>❑ Cada sucursal mostrará su nombre, dirección, teléfono y correo electrónico.</p> <p>❑ El sistema debe obtener los datos de la base de datos y mostrarlos de forma actualizada.</p> <p>❑ El <u>repcionista</u> no podrá crear, editar ni eliminar sucursales.</p> <p>❑ La interfaz debe permitir realizar búsquedas o filtrado por nombre o ubicación (opcional).</p>

HU14	Gestión de categorías
Descripción	<p><u>Como</u> usuario con rol de <u>repcionista o administrador</u>, <u>quiero</u> poder <u>ver, crear, editar y eliminar categorías de productos</u>, <u>para</u> mantener una clasificación organizada que facilite la administración y búsqueda de artículos en el sistema.</p>
Criterios de aceptación	<p>❑ El usuario puede registrar una nueva categoría completando los campos requeridos (nombre).</p>

	<p>❑ El sistema valida que no exista otra categoría con el mismo nombre antes de guardarla.</p> <p>❑ El usuario puede editar los datos de una categoría existente.</p> <p>❑ El usuario puede eliminar una categoría previa confirmación mediante un mensaje de advertencia.</p> <p>❑ El sistema muestra un listado actualizado de todas las categorías registradas.</p> <p>❑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.</p> <p>❑ Solo los usuarios con rol de repcionista o administrador pueden acceder a esta funcionalidad.</p>
--	---

HU15	Gestion de productos
Descripción	<p>Como usuario con rol de repcionista o administrador, quiero poder ver, crear, editar y eliminar productos, para mantener actualizado y asegurar que los productos estén</p>

	correctamente clasificados y disponibles para su gestión en el sistema.
Criterios de aceptación	<p>☐ El usuario puede registrar un nuevo producto completando los campos requeridos (nombre, código, precio y categoría).</p> <p>☐ El sistema valida que no exista otro producto con el mismo código antes de guardarlo.</p> <p>☐ El usuario puede editar los datos de un producto existente.</p> <p>☐ El usuario puede eliminar un producto, previa confirmación mediante mensaje de advertencia.</p> <p>☐ El sistema permite visualizar un listado completo de productos, mostrando nombre, código, precio y categoría asociada.</p> <p>☐ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.</p> <p>☐ Solo los usuarios con rol de recepcionista o administrador pueden acceder a esta funcionalidad.</p>

HU16	Gestion de compras
Descripción	<p>Como usuario con rol de administrador, quiero poder ver, crear, editar y eliminar compras, para mantener actualizado el registro de adquisiciones y garantizar que el stock refleje correctamente los productos disponibles.</p>
Criterios de aceptación	<p>☐ El administrador puede registrar una nueva compra ingresando total, fecha y hora, id de empleado.</p> <p>☐ El sistema valida que los campos obligatorios estén completos antes de guardar la compra.</p> <p>☐ El administrador puede editar los datos de una compra existente.</p> <p>☐ El administrador puede eliminar una compra, previa confirmación mediante mensaje de advertencia.</p> <p>☐ El sistema muestra un listado completo de compras registradas, indicando producto, cantidad, precio, proveedor y fecha.</p>

	<p>☐ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.</p> <p>☐ Las compras registradas actualizan automáticamente la cantidad disponible en el stock correspondiente.</p> <p>☐ Solo los usuarios con rol de administrador pueden acceder a esta funcionalidad.</p>
--	---

HU17	Gestion de detalles de compras
Descripción	<p>Como usuario con rol de administrador, quiero poder ver, crear, editar y eliminar los detalles de las compras, para mantener un registro completo y preciso de todas las adquisiciones realizadas y actualizar correctamente el inventario.</p>
Criterios de aceptación	<p>☐ El administrador puede registrar un nuevo detalle de compra asociándolo a una compra existente.</p> <p>☐ Cada detalle debe cantidad, precio unitario, sub total, id de producto y id de compra.</p> <p>☐ El sistema valida que todos los campos obligatorios estén completos antes de guardar.</p> <p>☐ El administrador puede editar los datos de un detalle de compra existente.</p> <p>☐ El administrador puede eliminar un detalle de compra, previa confirmación mediante mensaje de advertencia.</p> <p>☐ El sistema muestra un listado de todos los detalles asociados a cada compra, incluyendo producto, cantidad, precio y subtotal.</p> <p>☐ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en el stock y en los reportes.</p> <p>☐ Solo los usuarios con rol de administrador pueden acceder a esta funcionalidad.</p>

HU18	Lectura de compras
Descripción	Como usuario con rol de recepcionista , quiero poder ver las compras registradas en el sistema , para consultar información sobre adquisiciones y controlar el stock de productos sin modificar los registros existentes.
Criterios de aceptación	<p>☑ El recepcionista puede acceder a una sección del sistema que muestre el listado de compras total, fecha y hora, id de empleado</p> <p>☑ El sistema obtiene los datos de la base de datos y los muestra de forma actualizada.</p> <p>☑ El recepcionista no puede crear, editar ni eliminar compras.</p> <p>☑ La interfaz permite realizar búsquedas o filtrado por fecha, proveedor o producto (opcional).</p> <p>☑ Solo los usuarios con rol de recepcionista pueden acceder a esta funcionalidad.</p>

HU19	Lectura de detalles de compra
Descripción	Como usuario con rol de recepcionista , quiero poder ver los detalles de cada compra registrada en el sistema , para conocer la composición de las adquisiciones y verificar información de productos y cantidades sin modificar los registros.
Criterios de aceptación	<p>☑ El recepcionista puede acceder a una sección que muestre los detalles de cada compra registrada.</p> <p>☑ Cada detalle incluye cantidad, precio unitario, sub total, id de producto y id de compra</p> <p>☑ El sistema obtiene los datos de la base de datos y los muestra de forma actualizada.</p> <p>☑ El recepcionista no puede crear, editar ni eliminar detalles de compra.</p>

	<p>❑ La interfaz permite filtrar o buscar por producto, compra o proveedor (opcional).</p> <p>❑ Solo los usuarios con rol de repcionista pueden acceder a esta funcionalidad.</p>
--	---

HU20	Gestion de proveedores
Descripción	<p>Como usuario con rol de administrador, quiero poder ver, crear, editar y eliminar proveedores, para mantener actualizada la información de contacto de los proveedores y garantizar un correcto proceso de compras e inventario.</p>
Criterios de aceptación	<p>❑ El usuario puede registrar un nuevo proveedor completando los campos obligatorios nombre, dirección, teléfono y mail.</p> <p>❑ El sistema valida que no exista otro proveedor con el mismo nombre antes de guardarlo.</p> <p>❑ El usuario puede editar la información de un proveedor existente.</p> <p>❑ El usuario puede eliminar un proveedor, previa confirmación mediante mensaje de advertencia.</p> <p>❑ El sistema permite visualizar un listado completo de proveedores registrados, mostrando todos los datos relevantes.</p> <p>❑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en la interfaz.</p> <p>❑ Solo los usuarios con rol de administrador o repcionista pueden acceder a esta funcionalidad.</p>

HU21	Lectura de proveedores
Descripción	<p>Como usuario con rol de repcionista, quiero poder ver los proveedores registrados, para consultar información de contacto y ubicación sin modificar los registros.</p>

Criterios de aceptación	<p>☐ El usuario puede acceder a una sección que muestre el listado de proveedores registrados.</p> <p>☐ Cada proveedor muestra nombre, dirección, teléfono y mail.</p> <p>☐ El sistema obtiene los datos de la base de datos y los muestra de forma actualizada.</p> <p>☐ El usuario no puede crear, editar ni eliminar proveedores.</p> <p>☐ La interfaz permite búsqueda o filtrado por nombre (opcional).</p> <p>☐ Solo los usuarios con rol de recepcionista o recepcionista pueden acceder a esta funcionalidad.</p>
-------------------------	--

HU22	Gestion de ventas
Descripción	Como usuario con rol de administrador , quiero poder ver, crear, editar y eliminar ventas , para mantener un registro completo de las transacciones y garantizar la correcta actualización del stock.
Criterios de aceptación	<p>☐ El administrador puede registrar una nueva venta completando fecha y hora, cliente, id empleado, sucursal y total</p> <p>☐ El sistema valida los campos obligatorios antes de guardar.</p> <p>☐ El administrador puede editar una venta existente.</p> <p>☐ El administrador puede eliminar una venta, previa confirmación.</p> <p>☐ El sistema muestra un listado completo de ventas con toda la información relevante.</p> <p>☐ Los cambios realizados se guardan correctamente en la base de datos y actualizan el stock.</p>

HU23	Gestion de ventas(recepcionista)
Descripción	Como usuario con rol de recepcionista , quiero poder ver las ventas registradas y crear nuevas ,

	para registrar transacciones y consultar información sin eliminar registros existentes.
Criterios de aceptación	<p>☐ El recepcionista puede acceder a un listado de ventas con fecha y hora, cliente, id_empleado, sucursal y total</p> <p>☐ El recepcionista puede registrar nuevas ventas completando los campos obligatorios.</p> <p>☐ El sistema valida los campos antes de guardar la venta.</p> <p>☐ El recepcionista no puede eliminar ventas existentes.</p> <p>☐ Los cambios se guardan correctamente en la base de datos y actualizan el stock.</p>

HU24	Gestion de detalles de ventas
Descripción	Como usuario con rol de administrador , quiero poder ver, crear, editar y eliminar los detalles de cada venta , para mantener un registro completo de las transacciones y asegurar la correcta actualización del inventario.
Criterios de aceptación	<p>☐ El administrador puede registrar un nuevo detalle de venta asociándolo a una venta existente.</p> <p>☐ Cada detalle incluye venta, producto, cantidad, precio unitario y sub total.</p> <p>☐ El sistema valida que todos los campos obligatorios estén completos antes de guardar.</p> <p>☐ El administrador puede editar los datos de un detalle de venta existente.</p> <p>☐ El administrador puede eliminar un detalle de venta, previa confirmación.</p>

	<p>☐ El sistema muestra un listado de todos los detalles asociados a cada venta, incluyendo producto, cantidad, precio y subtotal.</p> <p>☐ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en el stock y reportes.</p> <p>☐ Solo los usuarios con rol de administrador pueden acceder a esta funcionalidad.</p>
--	--

HU24	Gestion de detalles de ventas (repcionista)
Descripción	<p><u>Como</u> usuario con rol de <u>repcionista</u>, <u>quiero</u> poder <u>ver, crear, editar y los detalles de cada venta</u>. <u>No quiero poder eliminar.</u> <u>para</u> mantener un registro completo de las transacciones y asegurar la correcta actualización del inventario.</p>
Criterios de aceptación	<p>☐ El recepcionista puede registrar un nuevo detalle de venta asociándolo a una venta existente.</p> <p>☐ Cada detalle incluye producto, cantidad, precio unitario, subtotal y observaciones.</p> <p>☐ El sistema valida que todos los campos obligatorios estén completos antes de guardar.</p> <p>☐ El recepcionista puede editar los datos de un detalle de venta existente.</p>

	<ul style="list-style-type: none">❑ El recepcionista NO puede eliminar un detalle de venta.❑ El sistema muestra un listado de todos los detalles asociados a cada venta, incluyendo producto, cantidad, precio y subtotal.❑ Los cambios realizados se guardan correctamente en la base de datos y se reflejan en el stock y reportes.❑ Solo los usuarios con rol de recepcionista pueden acceder a esta funcionalidad.
--	--

Diagramas

1. Diagrama de Entidad-Relación (DER)

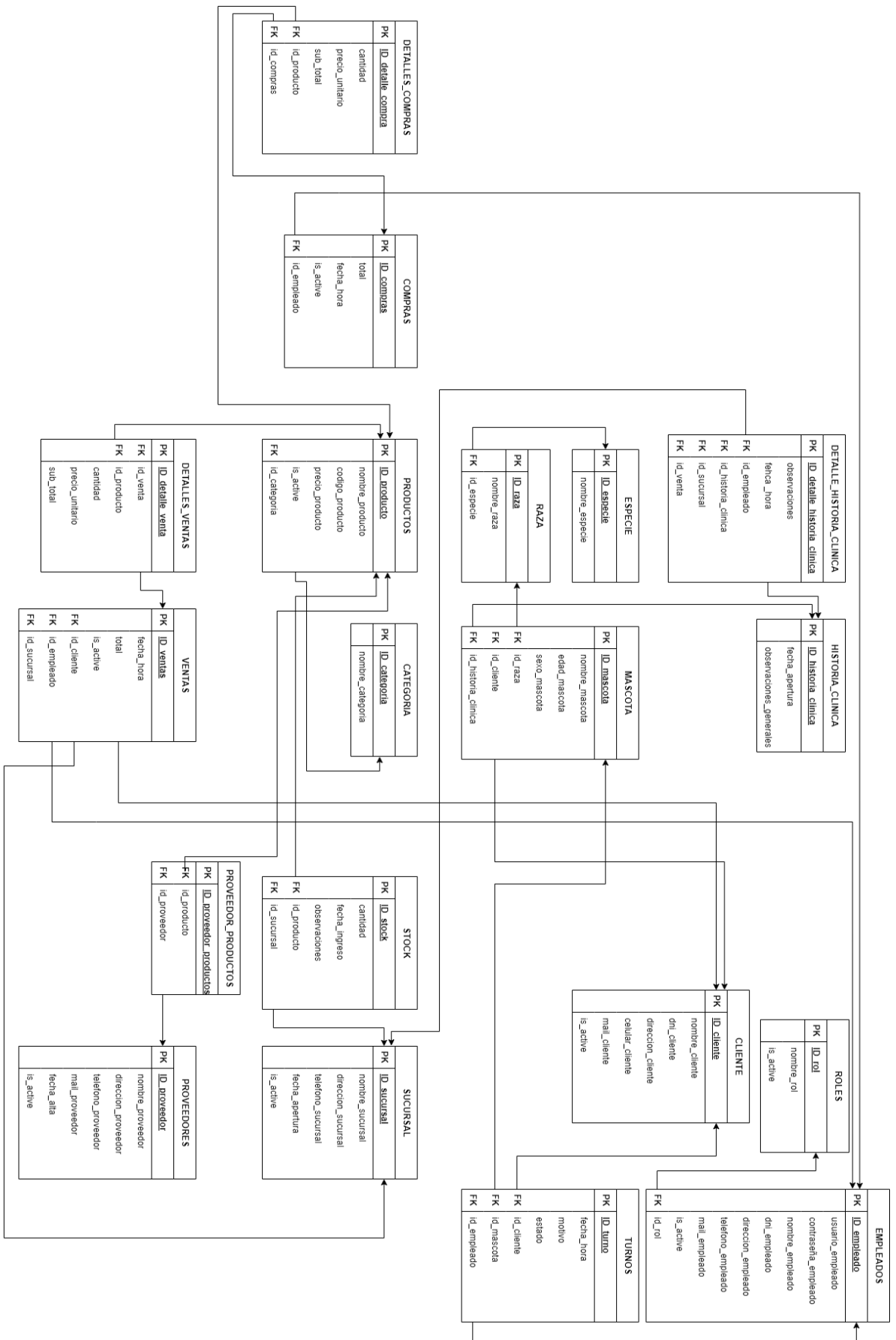
El Diagrama de Entidad-Relación (DER) es una herramienta de modelado conceptual que permite visualizar la estructura lógica de la información del sistema. Su objetivo principal es identificar las **entidades** (objetos reales o abstractos de interés para el sistema), los **atributos** que las describen y las **relaciones** que existen entre ellas.

Este diagrama facilita la comunicación entre los desarrolladores y los interesados (stakeholders) del proyecto, asegurando que los requerimientos de información estén correctamente representados antes de pasar a la implementación técnica. En este proyecto, el DER ilustra cómo interactúan los actores y los datos principales del flujo de negocio.

2. Modelo Relacional

El Modelo Relacional representa la estructura técnica de la base de datos tal como será implementada en el Sistema de Gestión de Bases de Datos (SGBD). En esta etapa, el modelo conceptual se transforma en un conjunto de **tablas** normalizadas.

Este diagrama detalla las columnas (campos), los tipos de datos, las **claves primarias (PK)** que identifican registros únicos y las **claves foráneas (FK)** que establecen la integridad referencial entre las tablas. Su propósito es servir de guía técnica para la creación del esquema de la base de datos y la construcción de las consultas SQL.



Prototipos de interfaz

En esta sección se presentan los prototipos de interfaz de usuario (UI) diseñados **inicialmente** para el sistema. Estos mockups representan la visualización conceptual de las pantallas principales y sirvieron como **referencia base** para la maquetación del frontend.

El objetivo principal de estos prototipos fue validar la experiencia de usuario (UX), proponer una disposición de los elementos y establecer un flujo de navegación lógico. Cabe destacar que **estas vistas no constituyen el diseño final estático**. Al tratarse de una maquetación preliminar basada en referencias visuales, el diseño **evolucionó durante la etapa de desarrollo**. Se realizaron modificaciones y ajustes sobre la marcha para optimizar la usabilidad, resolver desafíos técnicos emergentes y adaptar la interfaz a las necesidades funcionales reales que surgieron durante la implementación.

Página Principal (Dashboard o Home)

Figura 1. Vista de la Página Principal (Dashboard) "Esta pantalla constituye el punto de entrada centralizado al sistema tras la autenticación. Su propósito es ofrecer al usuario una visión panorámica del estado actual de la aplicación, accesos directos a las funcionalidades más utilizadas y un resumen de la actividad reciente, facilitando así la toma de decisiones rápidas y la navegación hacia módulos específicos.



Menú de Navegación

Figura 2. Estructura del Menú de Navegación Principal "Se ilustra el componente de navegación primario del sistema [puedes añadir aquí: (barra lateral / menú superior)]. Este elemento define la arquitectura de la información y la jerarquía de los módulos disponibles. Su diseño prioriza la claridad y la accesibilidad, permitiendo al usuario orientarse dentro de la aplicación y desplazarse ágilmente entre las distintas secciones funcionales.



Modal de Gestión (Crear/Editar/Configurar)

Figura 3. Ventana Modal para Gestión de Registros "La imagen muestra la implementación de una ventana modal (pop-up) diseñada para realizar tareas de gestión puntuales, como la creación de un nuevo registro. Este patrón de interfaz permite al usuario completar una acción transaccional sin perder el contexto visual de la pantalla subyacente, agilizando el flujo de trabajo y manteniendo el foco en la tarea actual.

The screenshot displays the SOFT VET application interface. At the top, a purple header bar contains the 'SOFT VET' logo, a user profile icon, and the text 'Nombre Usuario'. On the left, a light blue sidebar lists navigation options: 'Orden al día', 'Caja', 'Clientes', 'Compras', 'Estadísticas', 'Productos', 'Proveedores', 'Ventas', 'Mi Cuenta', and 'Salir'. The main content area is light purple and features two floating action buttons: 'Balance de Caja' and 'Quirófano'. A modal window titled 'Crear Clientes' is centered on the screen. It has a yellow border and a green header. Inside the modal, the title 'Nuevo Cliente' is underlined. Below it are five input fields: 'Nombre y Apellido', 'DNI', 'Direccion', 'Teléfono Celular', and 'Mail'. At the bottom of the modal are two buttons: 'Guardar' (blue) and 'Cancelar' (red). The footer of the application is purple and includes the 'SOFT VET' logo, links for 'Terminos y Condiciones - Políticas de Privacidad' and 'SoftVet', and a 'Contactos' link.

Vista de Detalles (Ver un elemento específico)

Figura 4. Vista Detallada de Entidad "Esta interfaz se despliega al seleccionar un elemento específico desde un listado general. El objetivo de la vista de detalles es presentar de forma exhaustiva y estructurada toda la información granular asociada a dicha entidad (atributos, historial, datos relacionados), permitiendo una revisión profunda y focalizada del registro seleccionado.

The screenshot displays the 'SOFT VET' web application interface. The main content area shows a modal window titled 'Productos' (Products) with a search bar and a table of products. The table has columns for 'Nombre' (Name), 'Descripción' (Description), 'Stock', and 'Acciones' (Actions). The first row shows 'Royal Canin Adult...' with a stock of 15. The 'Acciones' column contains buttons for 'Ver' (View), 'Actualizar' (Update), and 'Eliminar' (Delete). The right sidebar features a 'Balance de Caja' (Cash Balance) button and a 'Quirófano' (Operating Room) button with a count of 1. The bottom footer includes the 'SOFT VET' logo, links for 'Terminos y Condiciones - Políticas de Privacidad' (Terms and Conditions - Privacy Policies), and 'Contactos' (Contacts).

Nombre	Descripción	Stock	Acciones
Royal Canin Adult...	Alimento balanceado...	15	Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar
			Ver Actualizar Eliminar

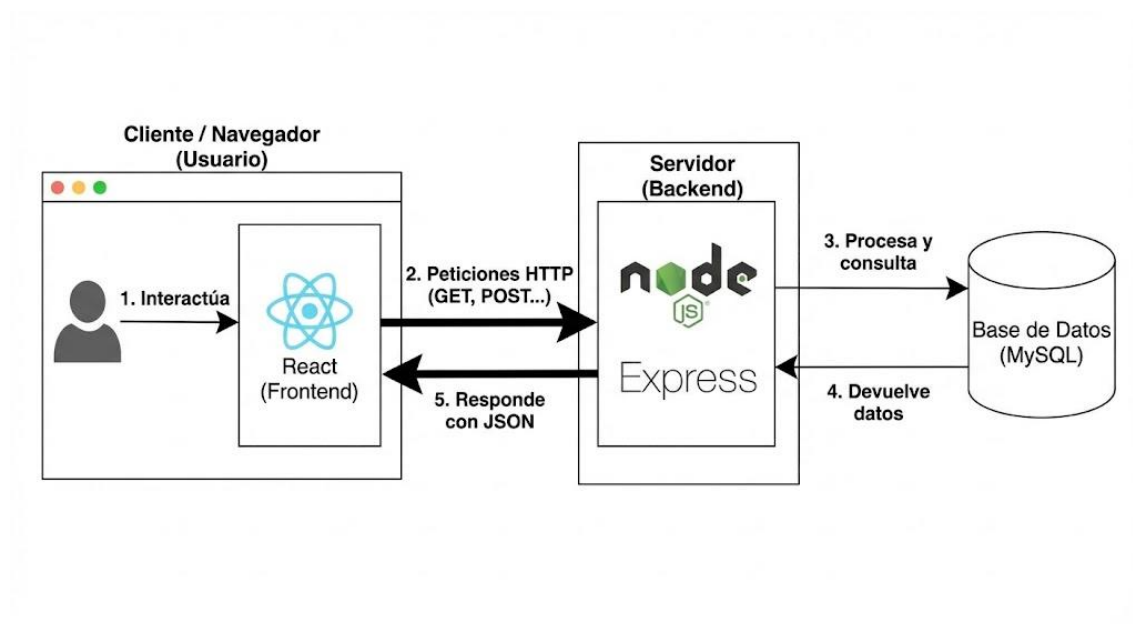
Arquitectura del sistema

1. Patrón Arquitectónico (MVC / Cliente-Servidor)

"El sistema se ha desarrollado bajo una arquitectura de **Aplicación de Página Única (SPA)** siguiendo el modelo **Cliente-Servidor**. Esta estructura desacopla completamente la lógica de presentación (Frontend) de la lógica de negocio y acceso a datos (Backend), permitiendo que ambas partes evolucionen de manera independiente."

2. Diagrama de Arquitectura (Nivel Alto)

1. **Navegador (Usuario)** interactúa con React.
2. **React** envía peticiones HTTP (GET, POST...) a Node/Express.
3. **Express** procesa y consulta a MySQL.
4. **MySQL** devuelve datos a Express.
5. **Express** responde con JSON a React.



3. Descripción de Componentes (El Stack Tecnológico)

Aquí se detalla la función de cada tecnología que mencionaste:

A. Capa de Cliente (Frontend - React)

"La interfaz de usuario está construida con **React.js**. Esta capa es responsable de todo lo que el usuario ve y con lo que interactúa."

- **Renderizado Dinámico:** Se encarga de actualizar el DOM de manera eficiente sin necesidad de recargar la página completa, mejorando la experiencia de usuario (UX).
- **Consumo de API:** Actúa como cliente HTTP, enviando solicitudes asíncronas al servidor para obtener o persistir información."

B. Capa de Servidor (Backend - Node.js & Express)

"El núcleo lógico del sistema se ejecuta sobre **Node.js**, utilizando el framework **Express**. Esta capa actúa como una **API RESTful** que centraliza la reglas de negocio.

- **Enrutamiento:** Express gestiona las rutas (endpoints) que reciben las peticiones del frontend.
- **Intermediario:** Procesa los datos entrantes, realiza validaciones de seguridad y coordina la comunicación con la base de datos antes de enviar una respuesta al cliente."

Interacción con la Base de Datos (Driver) "La comunicación entre el servidor Node.js y la base de datos MySQL se gestiona a través de la librería **mysql2**. Este cliente fue seleccionado por su alto rendimiento y compatibilidad con el protocolo estándar de MySQL.

Su implementación permite:

- **Seguridad:** Uso de **sentencias preparadas (prepared statements)**, lo que mitiga vulnerabilidades críticas como la inyección SQL al separar la estructura de la consulta de los datos.
- **Eficiencia:** Gestión de conexiones mediante un **Connection Pool** (grupo de conexiones), optimizando el uso de recursos del servidor y reduciendo la latencia en aplicaciones con múltiples usuarios concurrentes."

C. Capa de Datos (Database - MySQL)

"Para la persistencia de la información se utiliza **MySQL**, un sistema de gestión de bases de datos relacional (RDBMS).

- **Integridad Referencial:** Almacena los datos en tablas estructuradas y normalizadas, asegurando la consistencia y las relaciones entre las distintas entidades del negocio (como Usuarios, Productos, Pedidos, etc.)."

4. Flujo de Información (Cómo se conectan)

"La comunicación entre el Frontend (React) y el Backend (Express) se realiza a través del protocolo **HTTP**, utilizando **JSON** como formato estándar de intercambio de datos. El servidor no mantiene estado de la interfaz (stateless), lo que significa que cada petición contiene toda la información necesaria para ser procesada, garantizando una arquitectura escalable y modular."

Plan de pruebas

1. Introducción y Objetivos

"El propósito de este Plan de Pruebas es validar la funcionalidad, fiabilidad y desempeño del sistema SOFTVET. Se busca identificar y corregir errores antes del despliegue final, asegurando que los módulos críticos (autenticación, gestión de clientes/pacientes, historial médico) cumplan con los requerimientos especificados."

2. Alcance de las Pruebas (Scope)

"El alcance de las pruebas abarca las siguientes áreas funcionales:

- **Módulo de Autenticación:** Login, Registro de personal, Recuperación de credenciales.
- **Gestión de Usuarios y Roles:** CRUD completo de administradores y veterinarios.
- **Funcionalidades del Negocio:** Registro de mascotas, gestión de historias clínicas y agendamiento de turnos.
- **Validaciones:** Verificación de formularios en el Frontend y manejo de errores.
- **API REST:** Respuestas del servidor y códigos de estado HTTP (200, 404, 500).

Exclusiones:

No se realizaron pruebas de carga masiva (stress testing) ni pruebas de seguridad avanzada (pentesting) para esta versión del prototipo."

3. Entorno de Pruebas

"Las pruebas fueron ejecutadas bajo la siguiente configuración técnica:

- **Navegadores:** Google Chrome (v120+), Mozilla Firefox.
- **Sistema Operativo:** Windows 10/11 (Entorno Local).
- **Base de Datos:** MySQL Server.
- **Servidor de Aplicación:** Node.js v18+."

4. Estrategia y Herramientas

"Se aplicó una metodología de pruebas manuales y verificación de servicios:

- **Pruebas de API:** Se utilizó **Postman** para verificar los endpoints del Backend, asegurando que la estructura JSON de respuesta y los códigos de estado fueran correctos.
- **Pruebas de Interfaz:** Se realizaron pruebas de usabilidad y flujo de navegación directamente en el navegador (Pruebas de Caja Negra)."

6. Casos de Prueba (Muestra Representativa)

ID	Caso de Prueba	Pasos Realizados	Resultado Esperado	Estado
CP-01	Login Correcto	Ingresar credenciales válidas (Vet/Admin) y dar clic en "Entrar".	El sistema redirige al Dashboard principal.	Exitoso
CP-02	Login Incorrecto	Ingresar una contraseña errónea intencionalmente.	El sistema muestra alerta roja: "Credenciales inválidas".	Exitoso
CP-03	Campos Vacíos	Intentar guardar un formulario (ej: Nueva Mascota) dejando campos obligatorios en blanco.	El sistema impide el envío y resalta los campos faltantes.	Exitoso
CP-04	Registrar Mascota	Completar ficha de paciente y guardar.	Aparece mensaje "Guardado con éxito" y se actualiza el listado.	Exitoso
CP-05	Conexión API	Verificar carga de listados (Clientes/Turnos).	Los datos se recuperan correctamente de la Base de Datos.	Exitoso

Manual de usuario

Manual técnico

Información para mantenimiento y evolución del sistema

Este sistema está diseñado para permitir una fácil actualización y ampliación de funcionalidades.

El documento describe la estructura tecnológica utilizada, los componentes principales, los puntos de integración y los procedimientos necesarios para mantener el proyecto operativo y actualizado.

Incluye lineamientos para:

- Modificación y ampliación de módulos.
- Corrección de errores.
- Ajustes de seguridad.
- Optimización de rendimiento.
- Actualización de dependencias.

Arquitectura, configuración y procedimientos de actualización

Arquitectura general

Frontend: desarrollado en React, empleando componentes funcionales, Hooks, React Router para navegación y Axios para comunicación con la API.

Backend: desarrollado con Node.js y express.js Gestiona autenticación, validaciones y la lógica de negocio.

API REST: comunicación entre frontend y backend mediante peticiones HTTP con respuestas en formato JSON.

Base de Datos: MYSQL utilizada para almacenar todos los registros del sistema.

Estructura Cliente–Servidor: separación completa entre interfaz y servicios.

Configuración del entorno de desarrollo

- 1- Clonar repositorio
- 2- Instalar dependencias del frontend en /frontend:
 - a. npm install
- 3- Configurar archivo .env con
 - a. Variables de entornos necesarias
- 4- Iniciar el proyecto en modo desarrollo
 - a. npm run dev
- 5- Ejecutar backend en modo desarrollo
 - a. Node --watch index.js

Procedimientos de actualización

- 1- Actualizar repositorio local con los cambios recientes.
- 2- Crear rama para el desarrollo de nuevas funciones o correcciones.
- 3- Realizar modificaciones necesarias en frontend, backend o base de datos.
- 4- Ejecutar pruebas locales y validar funcionamiento.
- 5- Generar el build (en caso de ser necesario) del frontend:
 - a. Npm run build
- 6- Subir el build al servidor o servicio de hosting.
- 7- Actualizar el backend y aplicar migraciones de base de datos si corresponden.
- 8- Verificar funcionamiento general del sistema después de la actualización.

Secciones mínimas

Requisitos del sistema:

Node.js: versión recomendada (18.x o superior)

React: versión estable incluida en el proyecto.

Dependencias principales:

Frontend:

```
"axios": "^1.12.2",  
"bootstrap": "^5.3.8",  
"bootstrap-icons": "^1.13.1",  
"lucide-react": "^0.552.0",  
"react": "^19.1.1",
```

```
"react-bootstrap": "^2.10.10",  
"react-bootstrap-icons": "^1.11.6",  
"react-dom": "^19.1.1",  
"react-icons": "^5.5.0",  
"react-router-dom": "^7.9.4",  
"sweetalert2": "^11.26.3",  
"zustand": "^5.0.8"
```

Backend:

```
"bcrypt": "^6.0.0",  
"cookie-parser": "^1.4.7",  
"cors": "^2.8.5",  
"dotenv": "^17.2.3",  
"express": "^5.1.0",  
"jsonwebtoken": "^9.0.2",  
"morgan": "^1.10.1",  
"mysql2": "^3.15.1"
```

Diseño del sistema:

Estructura principal del Frontend:

```
/src
  /assets
  /components
  /endpoints
  /hooks
  /pages
  /routers
  /validations
  /zustand
```

/src

Directorio raíz donde se encuentra todo el código fuente del frontend.

/assets

Contiene imágenes, íconos, estilos y otros recursos estáticos utilizados en la interfaz del sistema.

/components

Incluye los componentes reutilizables de interfaz, como formularios, tablas, botones, modales y elementos visuales que se utilizan en distintas partes del sistema.

/endpoints

Carpeta donde se centralizan todas las URLs o funciones relacionadas a los endpoints de la API, permitiendo una gestión organizada de las rutas de comunicación con el backend.

/hooks

Contiene hooks personalizados utilizados para encapsular lógica reutilizable (manejo de formularios, consultas, validaciones, etc.).

/pages

Agrupar las pantallas principales del sistema, como Empleados, Mascotas, Clientes, Turnos, Ventas, Dashboard, entre otras.

/routers

Define la configuración de navegación del sistema, gestionando las rutas que conectan cada página y controlando el acceso según permisos.

/validations

Incluye funciones, esquemas o utilidades para validar los datos ingresados en formularios o procesos específicos del sistema.

/zustand

Carpeta destinada al manejo del estado global del sistema utilizando **Zustand**, donde se crean y administran los stores que controlan datos compartidos entre componentes (auth, usuario, configuraciones, etc.).

Estructura principal del backend:

`/config`

`/controllers`

`/middlewares`

`/routers`

`/service`

`/validations`

/config

Contiene los archivos de configuración del sistema, como conexión a la base de datos, variables de entorno, configuración de seguridad y parámetros generales utilizados por el servidor.

/controllers

Incluye los controladores responsables de gestionar la lógica de cada módulo del sistema. Aquí se procesan las solicitudes recibidas desde los routers, se coordinan llamadas a servicios y se generan las respuestas para la API.

/middlewares

Agrupar funciones que se ejecutan antes o durante el manejo de una petición, como validaciones de autenticación, permisos, manejo de errores, logs, protección de rutas o filtrado de datos.

/routers

Define las rutas de la API, asociando cada endpoint con su respectivo controlador y middleware. Es la capa que organiza las URL del sistema y su comportamiento.

/service

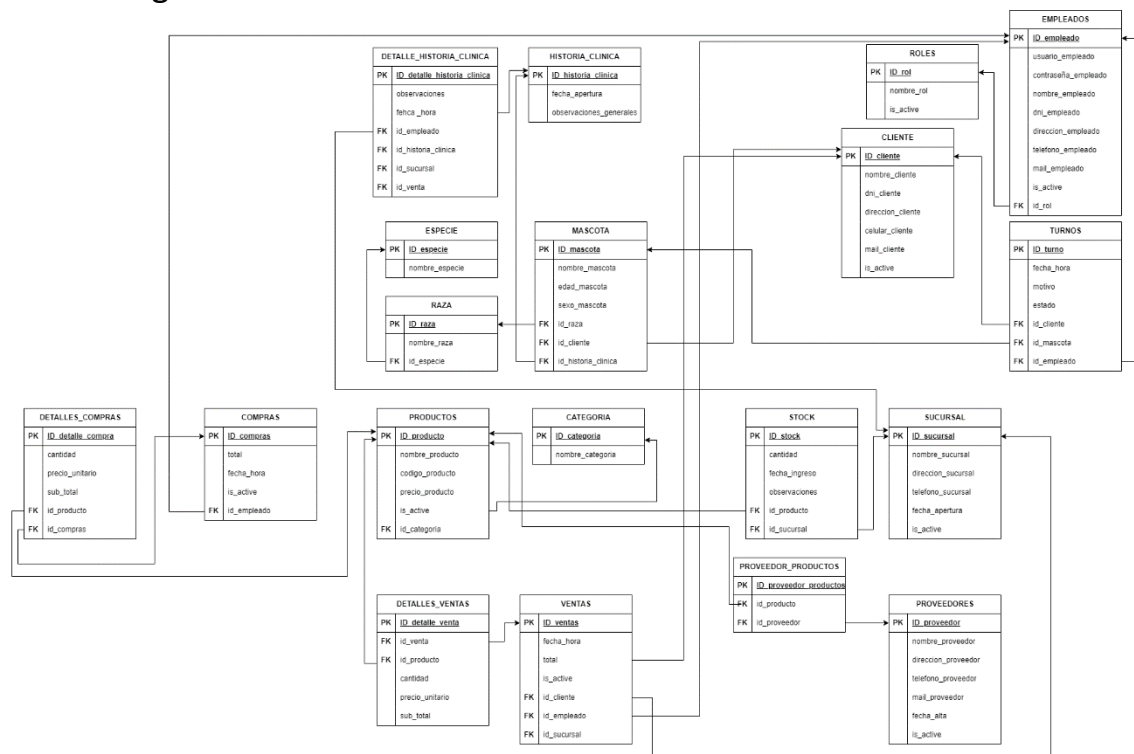
Contiene la lógica de negocio y operaciones principales del sistema, como consultas a la base de datos, procesamiento de información o interacción con otros servicios. Los controladores se apoyan en esta capa para mantener el código limpio y organizado.

/validations

Incluye funciones o esquemas de validación utilizados para verificar datos entrantes (por ejemplo, en formularios o peticiones), garantizando que cumplan con el formato y los requisitos antes de ser procesados por el sistema.

Estructura de base de datos

Diagrama entidad relacion



Api:

La comunicación con el backend se realiza mediante endpoints REST.

Empleados

GET	/empleados	Obtiene todos los empleados
POST	/empleados/crear	Crea un nuevo empleado
PUT	/empleados/editar/:id	Edita un empleado
POST	/empleados/authenticar	Autentifica un empleado
POST	/empleado/info	Obtiene información de empleado
GET	/empleados/ver	Obtiene todos los empleados
GET	/empleados/ver/:id	Obtiene un empleado por id
GET	/empleados/logout	Cierra sesión de un empleado

Categorías

Método	Endpoint	Descripción
GET	/categorias/ver	Obtiene el listado de todas las categorías
GET	/categorias/ver/:id	Obtiene una categoría específica por su ID
POST	/categorias/crear	Crea una nueva categoría en el sistema
PUT	/categorias/editar/:id	Edita los datos de una categoría existente
DELETE	/categorias/eliminar/:id	Elimina una categoría del sistema

Cientes

Método	Endpoint	Descripción
GET	/clientes/ver	Obtiene el listado de todos los clientes
GET	/clientes/ver/:id	Obtiene un cliente específico por su ID
GET	/clientes/buscar	Realiza una búsqueda de clientes (filtros)
POST	/clientes/crear	Registra un nuevo cliente en el sistema
PUT	/clientes/editar/:id	Edita los datos de un cliente existente
DELETE	/clientes/eliminar/:id	Elimina un cliente del sistema

Compras

Método	Endpoint	Descripción
GET	/compras/ver	Obtiene el listado de todas las compras
GET	/compras/ver/:id	Obtiene una compra específica por su ID
POST	/compras/crear	Registra una nueva compra (Admin)
PUT	/compras/editar/:id	Edita una compra existente (Admin)

DELETE	/compras/eliminar/:id	Elimina una compra del sistema (Admin)
---------------	-----------------------	--

Detalle Historia Clínica

Método	Endpoint	Descripción
GET	/detalleHistoria/ver	Obtiene todos los detalles de historias clínicas
GET	/detalleHistoria/ver/:id	Obtiene un detalle específico por su ID
GET	/detalleHistoria/verDetalle/:id	Consulta vista extendida de un detalle
POST	/detalleHistoria/crear	Agrega un nuevo detalle a una historia clínica
PUT	/detalleHistoria/editar/:id	Modifica un detalle existente
DELETE	/detalleHistoria/eliminar/:id	Elimina un detalle de historia clínica

Detalle de Compras

Método	Endpoint	Descripción
GET	/detalleCompra/ver	Obtiene todos los detalles de las compras

GET	/detalleCompra/ver/:id_detalle_compra	Obtiene un detalle de compra por ID
POST	/detalleCompra/crear	Crea un nuevo detalle de compra (Admin)
PUT	/detalleCompra/editar/:id_detalle_compra	Edita un detalle de compra (Admin)
DELETE	/detalleCompra/eliminar/:id_detalle_compra	Elimina un detalle de compra (Admin)

Detalle de Ventas

Método	Endpoint	Descripción
GET	/detalleVenta/ver	Obtiene el listado de detalles de ventas
POST	/detalleVenta/crear	Crea un nuevo detalle de venta
PUT	/detalleVenta/editar/:id	Edita un detalle de venta existente
DELETE	/detalleVenta/borrar/:id	Elimina un detalle de venta (Admin)

Empleados

Método	Endpoint	Descripción
GET	/empleados/ver	Obtiene el listado de todos los empleados
GET	/empleados/ver/:id	Obtiene un empleado específico por su ID
POST	/empleados/crear	Registra un nuevo empleado (Admin)
PUT	/empleados/editar/:id	Edita los datos de un empleado (Admin)
DELETE	/empleados/eliminar/:id	Elimina un empleado del sistema (Admin)
POST	/empleados/autenticar	Inicia sesión y genera token (Login)
POST	/empleados/info	Obtiene info del empleado autenticado
GET	/empleados/logout	Cierra la sesión del empleado

Especies

Método	Endpoint	Descripción
GET	/especies/ver	Obtiene el listado de todas las especies
GET	/especies/ver/:id	Obtiene una especie específica por su ID
POST	/especies/crear	Crea una nueva especie
PUT	/especies/editar/:id	Edita una especie existente
DELETE	/especies/eliminar/:id	Elimina una especie del sistema

Historia Clínica

Método	Endpoint	Descripción
GET	/historiaClinica/ver	Obtiene todas las historias clínicas
GET	/historiaClinica/ver/:id	Obtiene una historia clínica por ID
POST	/historiaClinica/crear	Crea una nueva historia clínica
PUT	/historiaClinica/editar/:id	Edita una historia clínica existente
DELETE	/historiaClinica/eliminar/:id	Elimina una historia clínica

Mascotas

Método	Endpoint	Descripción
GET	/mascotas/ver	Obtiene el listado de todas las mascotas
GET	/mascotas/cliente/:id_cliente	Obtiene las mascotas de un cliente específico
GET	/mascotas/ver/:id	Obtiene una mascota específica por su ID
POST	/mascotas/crear	Registra una nueva mascota
PUT	/mascotas/editar/:id	Edita los datos de una mascota
DELETE	/mascotas/eliminar/:id	Elimina (o da de baja) una mascota
PUT	/mascotas/activar/:id	Reactiva una mascota dada de baja

Productos

Método	Endpoint	Descripción
GET	/productos/ver	Obtiene el listado de productos
GET	/productos/ver/:id	Obtiene un producto por su ID
POST	/productos/crear	Crea un nuevo producto
PUT	/productos/editar/:id	Edita un producto existente
DELETE	/productos/eliminar/:id	Elimina un producto del sistema

Proveedores

Método	Endpoint	Descripción
GET	/proveedores/ver	Obtiene el listado de proveedores
GET	/proveedores/ver/:id	Obtiene un proveedor por su ID
POST	/proveedores/crear	Registra un nuevo proveedor (Admin)
PUT	/proveedores/editar/:id	Edita datos de un proveedor (Admin)
DELETE	/proveedores/eliminar/:id	Elimina un proveedor (Admin)

Razas

Método	Endpoint	Descripción
GET	/razas/ver	Obtiene el listado de todas las razas
GET	/razas/ver/:id	Obtiene una raza por su ID
GET	/razas/porEspecie/:id_especie	Filtra las razas según la especie
POST	/razas/crear	Crea una nueva raza
PUT	/razas/editar/:id	Edita una raza existente
DELETE	/razas/eliminar/:id	Elimina una raza del sistema

Roles

Método	Endpoint	Descripción
GET	/roles/ver	Obtiene el listado de roles del sistema
GET	/roles/ver/:id	Obtiene un rol específico por su ID
POST	/roles/crear	Crea un nuevo rol de usuario
PUT	/roles/editar/:id	Edita un rol existente
DELETE	/roles/eliminar/:id	Elimina un rol del sistema

Stock

Método	Endpoint	Descripción
GET	/stock/ver	Obtiene el estado general del stock
GET	/stock/ver/producto/:id_producto	Consulta stock de un producto específico
GET	/stock/verStock/:id_stock	Obtiene registro de stock por ID único
POST	/stock/crear	Crea un nuevo registro de stock
PUT	/stock/editar/:id_stock	Modifica un registro de stock
DELETE	/stock/borrar/:id_stock	Elimina un registro de stock

Sucursales

Método	Endpoint	Descripción
GET	/sucursales/ver	Obtiene el listado de sucursales
GET	/sucursales/ver/:id	Obtiene una sucursal por su ID
POST	/sucursales/crear	Crea una nueva sucursal (Admin)

PUT	/sucursales/editar/:id	Edita datos de una sucursal (Admin)
DELETE	/sucursales/eliminar/:id	Elimina una sucursal (Admin)

Turnos

Método	Endpoint	Descripción
GET	/turnos/ver	Obtiene el listado de todos los turnos
GET	/turnos/horariosDisponibles	Consulta horarios libres para agenda
GET	/turnos/ver/:id	Obtiene un turno específico por ID
POST	/turnos/crear	Agenda un nuevo turno
PUT	/turnos/editar/:id	Reprograma o edita un turno
DELETE	/turnos/eliminar/:id	Cancela o elimina un turno

Ventas

Método	Endpoint	Descripción
GET	/ventas/ver	Obtiene el historial de ventas
POST	/ventas/crear	Registra una nueva venta

PUT	/ventas/editar/:id_venta	Edita una venta existente
PUT	/ventas/borrar/:id_venta	Marca venta como borrada (Admin)
PUT	/ventas/activar/:id_venta	Reactiva una venta anulada (Admin)

Conclusion

El desarrollo e implementación del sistema **SOFTVET** ha logrado cumplir satisfactoriamente con el objetivo general de modernizar y optimizar la gestión integral de la clínica veterinaria. Se ha transitado exitosamente de un modelo de gestión basado en procesos manuales y soporte en papel a una solución digital centralizada.

Desde una perspectiva técnica, la elección de una arquitectura **Cliente-Servidor** utilizando el stack tecnológico de **React, Node.js y MySQL** demostró ser robusta y escalable. Esta estructura no solo garantiza el cumplimiento de los requisitos no funcionales de rendimiento (tiempos de respuesta inferiores a 1.5s) y disponibilidad, sino que también facilita el mantenimiento futuro del código gracias a su modularidad.

A nivel operativo, el sistema resuelve las problemáticas críticas identificadas en el análisis inicial: * **Integridad de Datos:** La digitalización de la Historia Clínica y el uso de auditorías aseguran que la información médica sea accesible, segura e inmutable. * **Eficiencia Administrativa:** La validación automática de turnos ha eliminado el riesgo de "sobretornos", y la actualización automática del stock tras cada venta o compra minimiza los errores de inventario.

En conclusión, SOFTVET no solo provee una herramienta de software funcional, sino que proyecta una reducción estimada del 70% en la carga administrativa, permitiendo que el personal de la veterinaria enfoque sus esfuerzos en la calidad de atención al paciente en lugar de la burocracia operativa.

Equipo de desarrollo:

Desarrolladores	Correo
Valdez Gramajo, David	davidvaldezgramajo@gmail.com
Celiz, Mariano	marianoceliz94@gmail.com
Pastrana, Angel	angelpastrana0333@gmail.com
Joaquin, Diaz	joaquindiaz2103@gmail.com

