**📘 VetCare - Sistema de Gestión Veterinaria Completo**

**📑 Índice**

1. **Introducción**
2. **Tecnologías Implementadas**
   * Frontend
   * Backend
3. **Funcionalidades Principales**
   * Dashboard del Cliente
   * Gestión de Mascotas
   * Sistema de Citas
   * Exámenes Médicos
   * Historial Médico
   * Gestión de Pagos
   * Perfil de Usuario
4. **Sistema de Autenticación**
5. **Diseño y Experiencia de Usuario**
6. **Arquitectura Técnica**
   * Estructura del Proyecto
   * Descripción Detallada de Archivos
   * Base de Datos Relacional
   * Framework de Seguridad
7. **Instalación y Configuración**
   * Requisitos del Sistema
   * Configuración del Backend
   * Configuración del Frontend
   * Configuración de Base de Datos
8. **Scripts de Desarrollo**
9. **Estado del Proyecto**
10. **Conclusión**

**1. Introducción**

VetCare es un **sistema full-stack de gestión veterinaria** que permite a los clientes administrar integralmente el cuidado de sus mascotas. Incluye un frontend moderno, un backend robusto y una base de datos relacional optimizada.

**2. Tecnologías Implementadas**

**🔹 Frontend**

* **React 19** + **TypeScript** + **Vite**
* **React Router DOM** – Enrutamiento dinámico
* **Context API** – Estado global
* **CSS Utilitario** + **PostCSS**

**🔹 Backend**

* **Node.js** + **Express** + **TypeScript**
* **MySQL** – Base de datos relacional
* **JWT** – Autenticación segura
* **Zod** – Validación de esquemas
* **bcrypt** – Hash de contraseñas

**3. Funcionalidades Principales**

**🏠 Dashboard del Cliente**

* Resumen con estadísticas en tiempo real
* Contadores de mascotas, citas y pagos
* Acciones rápidas y notificaciones
* Menú lateral responsivo con indicadores

**🐾 Gestión de Mascotas**

* Registro con validación en tiempo real
* Vista en tarjetas con CRUD completo
* Modal interactivo y confirmaciones

**📅 Sistema de Citas**

* Reserva de citas con calendario
* Estados: pendiente, confirmada, completada, cancelada
* Historial de citas y precios dinámicos

**📋 Exámenes Médicos**

* Solicitud de exámenes con validaciones
* Tipos de exámenes variados
* Estados con colores distintivos

**📄 Historial Médico**

* Vista en línea de tiempo y por mascota
* Búsquedas y filtros combinados
* Exportación de datos a CSV

**💳 Gestión de Pagos**

* Resumen financiero con estados visuales
* Historial de pagos con métodos soportados
* Seguimiento de pagos fallidos

**👤 Perfil de Usuario**

* Datos personales y de cuenta
* Edición con validación
* Modal interactivo con confirmación

**4. Sistema de Autenticación**

* **Login** con JWT y persistencia de sesión
* **Registro** con validación y encriptación
* **Protección de rutas** con middleware
* **Context global** para autenticación

**5. Diseño y Experiencia de Usuario**

* **Paleta cálida**: naranja, terracota y verde esmeralda
* **Componentes modernos**: tarjetas, botones, iconos
* **Responsive design** adaptable a móviles y desktop
* **UX intuitiva**: breadcrumbs, tooltips, feedback visual

**6. Arquitectura Técnica**

**🏗️ Estructura del Proyecto**

Incluye backend en Node.js + Express, frontend en React + Vite, documentación técnica, scripts SQL y configuraciones.  
*(Se mantiene la estructura completa con carpetas y archivos del backend y frontend)*

**📊 Base de Datos Relacional**

* **usuarios** – Roles, autenticación y sesiones
* **pacientes** – Registro médico de mascotas
* **servicios** – Catálogo con precios dinámicos
* **citas** – Agendamiento con estados y auditoría
* **examenes** – Resultados y seguimiento
* **pagos** – Transacciones y métodos de pago

**🔒 Seguridad**

* Autenticación JWT + refresh tokens
* Validación de datos con Zod
* Hash de contraseñas con bcrypt
* Middleware de autorización granular
* Rate limiting con Redis
* Auditoría y logging avanzado

**7. Instalación y Configuración**

**🔹 Requisitos**

* Node.js 18+
* MySQL 8.0+
* npm 9+ o yarn 3+
* Redis 6+ (opcional)

**🔹 Backend**

cd backend

npm install

npm run build

npm start

**🔹 Frontend**

cd frontend

npm install

npm run dev

**🔹 Base de Datos**

1. Crear BD vetcare
2. Ejecutar migraciones y seed
3. Configurar .env
4. Optimizar índices

**8. Scripts de Desarrollo**

**Frontend**

* npm run dev – Servidor desarrollo
* npm run build – Build producción
* npm run lint – Linter

**Backend**

* npm run dev – Servidor desarrollo
* npm run build – Compilar TypeScript
* npm start – Servidor producción

**9. Estado del Proyecto**

**✅ Completado**

* Autenticación JWT
* CRUD de mascotas
* Sistema de citas
* Exámenes médicos
* Historial médico con exportación
* Pagos
* Perfil de usuario
* API RESTful

**🔮 Mejoras Futuras**

* Panel de administración
* Notificaciones SMS/Email
* Integración con pasarelas de pago
* App móvil
* Reportes avanzados

**10. Conclusión**

**VetCare** es un sistema completo de gestión veterinaria, diseñado con **mejores prácticas en arquitectura full-stack**, seguridad robusta y una experiencia de usuario moderna. Está listo para despliegue en producción y puede escalar con nuevas funcionalidades.

Interfaz de usuario gráfica, Aplicación, Sitio web

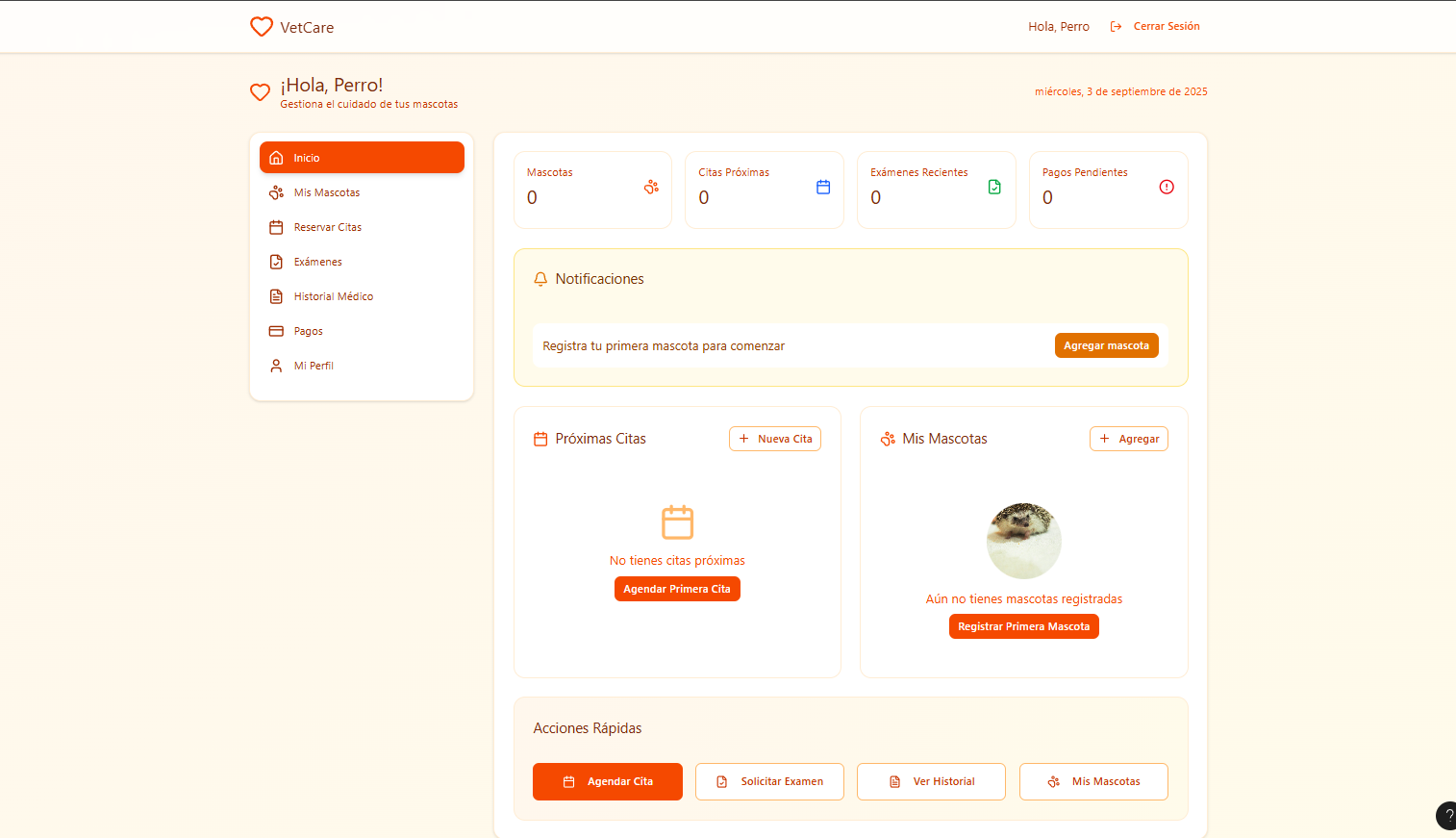
El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, PowerPoint

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, PowerPoint

El contenido generado por IA puede ser incorrecto.



Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.