

**Proyecto:** Gestión de Clínica Veterinaria

**Versión:** 1.1

**Fecha:** 2025-07-25

**Autor:** Kristian David Castrillon Paz

**Revisión:**

- v1.0 (2025-07-20): Borrador inicial
- v1.1 (2025-07-25): Detalle de requisitos, flujos y métricas

## 1. Propósito del Documento

Este PRD define los requisitos funcionales y no funcionales, actores, flujos de usuario y métricas de éxito para la aplicación de escritorio **Gestión de Clínica Veterinaria**. Destinado a desarrolladores, QA, diseño UX/UI y equipo de producto.

## 2. Descripción General

### 2.1 Contexto

Las clínicas veterinarias suelen depender de procesos manuales en papel o en hojas de cálculo, lo que genera errores, duplicados y dificulta la consulta histórica de datos. Esta solución propone digitalizar y centralizar toda la información clínica.

### 2.2 Alcance del MVP

- Conexión y verificación de base de datos
- Menú principal con acceso a módulos CRUD
- Registro y consulta de Pacientes y Propietarios
- Gestión completa de Historia Clínica y submódulos
- Mensajes de confirmación y manejo de errores

### 3. Actores y Stakeholders

Actor	Responsabilidad
Recepcionista	Registrar pacientes y propietarios; iniciar proceso de atención
Veterinario	Completar anamnesis, exámenes, diagnósticos y tratamientos
Administrador TI	Configurar base de datos, manejar usuarios y respaldos
Propietario	Consultar información de su(s) mascota(s)
Equipo de QA	Validar cumplimiento de los requisitos y flujos
Equipo de Diseño UX	Asegurar usabilidad y accesibilidad de interfaces

### 4. Objetivos y KPIs

1. **Digitalización:** El 100 % de los registros deben hacerse en la app.
2. **Eficiencia:** Reducir el tiempo de registro de paciente a < 2 min.
3. **Disponibilidad:** 99.5 % uptime de conexión a BD.
4. **Precisión:** ≥ 99 % de entradas válidas sin errores de validación.
5. **Satisfacción:** Puntuación ≥ 4/5 en encuestas de usuario tras 1 mes de uso.

### 5. Requisitos Funcionales

#### RF-1 Conexión y Autenticación

- **RF-1.1** Al iniciar, mostrar pantalla Splash y verificar conexión JDBC.
- **RF-1.2** Si falla la conexión, mostrar mensaje de error detallado (`SQLException` o `Timeout`) y botón **Reintentar**.
- **RF-1.3** Permitir hasta 3 reintentos antes de cerrar la app con aviso.

## RF-2 Menú Principal

- **RF-2.1** Mostrar ícono de clínica + nombre en cabecera.
- **RF-2.2** Botones de acceso rápido a cada módulo (12 en total) dispuestos en 2 columnas.
- **RF-2.3** Botón **Cerrar Sesión** siempre visible en zona inferior.

## RF-3 Gestión de Pacientes

- **RF-3.1** Formulario con campos:
  - Nombre completo (texto, obligatorio)
  - Especie (dropdown: Canina, Felina, Otra)
  - Raza (texto libre)
  - Fecha de nacimiento (datepicker)
  - Sexo (radio: M / F)
  - Propietario (combo buscable por nombre o documento)
- **RF-3.2** Validaciones:
  - Campos no nulos
  - Fecha  $\leq$  fecha actual
  - Longitud mínima y máxima
- **RF-3.3** Mensajes de error inline y resumen al pie.
- **RF-3.4** Botones **Guardar** (persistir via **PacienteDAO**) y **Cancelar** (limpiar formulario).

## RF-4 Gestión de Propietarios

- **RF-4.1** Formulario con:
  - Nombre completo
  - Documento de identidad (número/alfanumérico)

- Teléfono (formato internacional)
- Dirección (texto)
- **RF-4.2** Mismas validaciones y botones que RF-3.

## RF-5 Historial Clínico

- **RF-5.1** Tabla con columnas: ID, Nombre Paciente, Especie, Propietario, Ver Historia.
- **RF-5.2** Fila de ejemplo + paginación futura.

## RF-6 Submódulos de Historia

Para cada pestaña (Anamnesis, Examen Físico, Complementarios, Diagnóstico Integral, Vacunación, Tratamiento, Procedimiento, Signos Anormales, Observaciones):

- **RF-6.x.1** Listado de registros con campos clave.
- **RF-6.x.2** Botones **Agregar**, **Editar**, **Eliminar**.
- **RF-6.x.3** Formulario modal o in-page con validaciones iguales a RF-3.

## RF-7 Confirmación y Manejo de Errores

- **RF-7.1** En cada operación CRUD mostrar pantalla de confirmación (“Éxito”) o mensaje de error con detalles.
- **RF-7.2** Botón **Volver al Menú** para retomar navegación general.

## 6. Requisitos No Funcionales

- **RNF-1 Performance:** Pantallas críticas (< 2 s) al cargar tablas y formularios.
- **RNF-2 Seguridad:** Sanitizar todos los inputs; uso de **PreparedStatement** para evitar SQL injection.
- **RNF-3 Usabilidad:** Títulos claros, mensajes legibles, navegación natural de izquierda a derecha / arriba a abajo.

- **RNF-4 Mantenibilidad:** Código MVC, separación de capas, DAOs reutilizables, comentarios claros.
- **RNF-5 Portabilidad:** Compatible con JRE 11+ en Windows, macOS y Linux.
- **RNF-6 Accesibilidad:** Contrast ratio  $\geq 4.5:1$ , soporte para teclado.

## 7. Flujos de Usuario

1. **Conexión inicial**  
Splash → Verificar BD → [Éxito → Menú] / [Error → Reintentar x3 → Cerrar]
2. **Registrar Paciente**  
Menú → Registrar Paciente → Completar formulario → Validar → Guardar → Confirmación → Menú
3. **Consultar Historia Clínica**  
Menú → Historia Clínica → Seleccionar paciente → Pestañas → CRUD submódulos → Confirmación/Error

## 8. UX/UI Wireframes

Referenciar los archivos [index.html](#) y [styles.css](#) de baja fidelidad. Pantallas diseñadas según especificación de grid 2×4, celdas 270×350 px y gutters 80 px. Diagramas de flujo y wireframes incluidas en [/wireframes/](#).

## 9. Métricas de Éxito

Métrica	Objetivo
Tiempo de búsqueda expediente	< 3 s
Tasa de éxito de guardado de datos	$\geq 99\%$
Tiempo promedio registro paciente	< 2 min
Disponibilidad de conexión JDBC	$\geq 99.5\%$ uptime

Satisfacción de usuario (post-mvp)  $\geq 4/5$  en  
encuesta

## 10. Suposiciones y Dependencias

- La base de datos SQL estará disponible y accesible en la red local.
- Existen tablas con esquemas normalizados con claves foráneas definidas.
- No se requiere multilocalización en MVP (solo español).
- Integración futura con módulo de facturación/agenda.

## 11. Glosario

- **DAO:** Data Access Object.
- **MVC:** Model-View-Controller.
- **JDBC:** Java Database Connectivity.
- **MVP:** Producto mínimo viable.