

## **Plan de Pruebas de Software - Sistema de Gestión para Clínica Veterinaria**

**Título:** Plan de Pruebas de Software - Sistema de Gestión para Clínica Veterinaria

**Proyecto:** Software de Gestión Integral para Clínica Veterinaria

**Autores:** Melissa Carol Ramírez Ortegón,

Ciro Montes

**Fecha:** 28/09/2025

**Versión:** 1.0

**SENA 2025**

---

## **Tabla de Contenidos**

1. Introducción
2. Objetivos de las Pruebas
3. Alcance de las Pruebas
4. Estrategia y Criterios de Pruebas
5. Entornos de Trabajo
6. Metodología y Procedimientos
7. Cronograma y Planificación
8. Artefactos de Pruebas
9. Herramientas de Pruebas
10. Riesgos y Contingencias
11. Conclusiones

---

## 1. Introducción

El presente plan de pruebas tiene como objetivo establecer la estrategia, metodología y recursos necesarios para validar el correcto funcionamiento del Sistema de Gestión para Clínica Veterinaria. Este sistema, desarrollado con arquitectura full-stack (Spring Boot + React), busca automatizar los procesos de gestión de pacientes, citas, inventario y reportes médicos.

Las pruebas se realizarán bajo los principios de calidad de software establecidos en la norma ISO/IEC 25000, asegurando que el producto final cumpla con los requisitos funcionales y no funcionales definidos en la matriz de trazabilidad.

## 2. Objetivos de las Pruebas

- Validar que el sistema cumple con los requisitos funcionales definidos en la [matriz de trazabilidad](#)
- Verificar el correcto funcionamiento de la autenticación y autorización por roles
- Garantizar la integridad de los datos en todos los módulos del sistema
- Validar la interfaz de usuario en diferentes dispositivos y navegadores
- Verificar el rendimiento del sistema bajo condiciones normales de operación
- Asegurar la compatibilidad entre el frontend (React) y backend (Spring Boot)

## 3. Alcance de las Pruebas

### Funcionalidades Existentes a Probar:

- **RF01:** Módulo de registro de pacientes (mascotas y dueños)
- **RF02:** Gestión de citas médicas (programar, modificar, cancelar)
- **INF01:** Autenticación de usuarios mediante contraseña y roles
- **BNF02:** Tiempo de respuesta del sistema < 2 segundos

### Funcionalidades Nuevas a Probar:

- **RF03:** Inventario de medicamentos y suministros
- **RF04:** Generación de reportes médicos en PDF

- **BNF03:** Interfaz intuitiva y accesible desde dispositivos móviles

#### **Exclusiones:**

- Pruebas de penetración de seguridad avanzada
- Pruebas de estrés a gran escala (>1000 usuarios concurrentes)
- Pruebas de compatibilidad con navegadores obsoletos

### **4. Estrategia y Criterios de Pruebas**

#### **Tipos de Pruebas a Implementar:**

##### **Pruebas Unitarias (Desarrollo):**

- **Backend:** Pruebas JUnit para servicios y controladores Spring Boot
- **Frontend:** Pruebas con Jest/React Testing Library para componentes React
- **Criterio de Aceptación:** 80% de cobertura de código

##### **Pruebas de Integración:**

- **API REST:** Pruebas de endpoints con Postman/Newman
- **Base de Datos:** Validación de constraints y relaciones JPA
- **Criterio de Aceptación:** 100% de endpoints funcionando correctamente

##### **Pruebas de Sistema:**

- **Flujos Complejos:** Pruebas de extremo a extremo (E2E) con Cypress
- **Roles de Usuario:** Validación de permisos y vistas por rol
- **Criterio de Aceptación:** Todos los casos de uso principales funcionando

##### **Pruebas de Aceptación:**

- **Validación con Usuarios Finales:** Pruebas con veterinarios y recepcionistas
- **Criterio de Aceptación:** Aprobación del cliente/profesor

### **5. Entornos de Trabajo**

#### **Hardware Requerido:**

- **Servidor de Pruebas:** 4GB RAM, 2 CPUs, 50GB almacenamiento

- **Estaciones de Pruebas:** 8GB RAM, navegadores Chrome/Firefox/Edge actualizados
- **Dispositivos Móviles:** Tablets y smartphones para pruebas responsive

#### **Software Requerido:**

- **Backend:** Java 17, Spring Boot 3.x, MySQL 8.0
- **Frontend:** Node.js 18+, React 18+, Vite
- **Herramientas:** Postman, Cypress, JUnit, Jest, Git

#### **Entornos Configurados:**

- **Desarrollo:** Local (localhost)
- **Pruebas:** Servidor dedicado (ambiente de staging)
- **Producción:** Servidor en la nube (post-aprobación)

### **6. Metodología y Procedimientos**

#### **Metodología Ágil con Enfoque TDD:**

- Desarrollo guiado por pruebas para componentes críticos
- Pruebas automatizadas en pipeline de CI/CD
- Retroalimentación continua durante el desarrollo

#### **Procedimientos de Ejecución:**

1. **Preparación:** Configuración de datos de prueba
2. **Ejecución:** Ejecución manual y automatizada de pruebas
3. **Registro:** Documentación de resultados en matriz de pruebas
4. **Reporte:** Generación de reportes de defectos
5. **Seguimiento:** Verificación de correcciones

### **7. Cronograma y Planificación**

Fase	Duración	Actividades	Responsables
Preparación	3 días	Configuración de entornos, diseño de casos de prueba	Equipo de pruebas
Pruebas Unitarias	5 días	Desarrollo y ejecución de pruebas unitarias	Desarrolladores
Pruebas de Integración	4 días	Pruebas de API y base de datos	Equipo de pruebas
Pruebas de Sistema	5 días	Pruebas E2E y validación de roles	Equipo de pruebas
Pruebas de Aceptación	3 días	Validación con usuarios finales	Cliente/Profesor
Reporte Final	2 días	Documentación de resultados	Líder de pruebas

**Total:** 22 días hábiles

## 8. Artefactos de Pruebas

### Matriz de Pruebas (Ejemplo):

ID Prueba	Módulo	Caso de Prueba	Precondiciones	Datos de Entrada	Resultado Esperado	Prioridad
PT-001	Autenticación	Login exitoso con credenci	Usuario registrado en BD	usuario: "vet01",	Redirección a dashboard	Alta

ID Prueba	Módulo	Caso de Prueba	Precondiciones	Datos de Entrada	Resultado Esperado	Prioridad
		ales válidas		password: "pass123"	según rol	
PT-002	Citas	Programar nueva cita	Veterinario autenticado	Datos de cita válidos	Cita creada en BD, notificación enviada	Alta
PT-003	Inventario	Alerta stock mínimo	Producto con stock = 3	Consulta de inventario	Mostrar alerta de stock bajo	Media

#### Formatos de Registro:

- **Formato de Caso de Prueba:** Incluye ID, descripción, pasos, resultado esperado/actual
- **Formato de Reporte de Defectos:** ID defecto, severidad, pasos para reproducir, estado
- **Bitácora de Pruebas:** Registro diario de actividades y hallazgos

## 9. Herramientas de Pruebas

#### Automatización:

- **Postman/Newman:** Pruebas de API REST automatizadas
- **Cypress:** Pruebas E2E del frontend

- **JUnit:** Pruebas unitarias del backend Spring Boot
- **Jest/React Testing Library:** Pruebas unitarias de componentes React

#### Gestión y Reportes:

- **Excel/Google Sheets:** Matriz de trazabilidad y casos de prueba
- **Git/GitHub:** Control de versiones y gestión de issues
- **ScreenRecorder:** Grabación de ejecución de pruebas para evidencia

### 10. Riesgos y Contingencias

#### Riesgos Identificados:

Riesgo	Probabilidad	Impacto	Plan de Mitigación
Cambios en requisitos durante pruebas	Media	Alto	Comunicación constante con stakeholders
Defectos críticos cerca de la fecha límite	Alta	Alto	Priorización de pruebas críticas
Problemas de compatibilidad con dispositivos	Media	Medio	Pruebas tempranas en múltiples dispositivos
Falta de datos de prueba realistas	Baja	Medio	Generación de datos sintéticos representativos

#### Plan de Contingencias:

- Reserva de tiempo del 20% para imprevistos
- Equipo de desarrollo disponible para correcciones urgentes
- Rollback a versión estable en caso de fallos críticos



## 11. Conclusiones

El presente plan de pruebas establece un marco estructurado para validar la calidad del Sistema de Gestión para Clínica Veterinaria. La estrategia combinada de pruebas manuales y automatizadas, junto con la definición clara de criterios de aceptación, asegurará que el producto final cumpla con las expectativas de los stakeholders.

La ejecución de este plan permitirá identificar y corregir defectos de manera temprana, reduciendo costos y tiempos de desarrollo, y entregando un producto de calidad que satisfaga las necesidades de la clínica veterinaria.

La documentación resultante de las pruebas servirá como evidencia del cumplimiento de los requisitos y como base para el mantenimiento futuro del sistema.

---

**Anexos(en cada enlace siguiente, verá los archivos correspondientes a cada anexo):**

- [Matriz de trazabilidad de requisitos completa](#)
- [Casos de prueba detallados por módulo](#)
- [Configuración de entornos de prueba](#)
- [Scripts de automatización de pruebas\(VER REPO\) BACKEND](#)
- [FRONET](#)(Repo Fronet)