

# Backlog Recovery — User Stories

## Spring PetClinic

---

### 1. Introducción

Este documento presenta la recuperación de historias de usuario (Backlog Recovery) a partir del análisis del código fuente del proyecto Spring PetClinic. Las historias fueron identificadas mediante revisión directa de controladores, entidades y flujo funcional del sistema. Cada historia está asociada a un archivo específico como evidencia técnica.

### 2. Actores del Sistema

| Actor         | Descripción  |
|---------------|--|
| Recepcionista | Gestiona dueños, mascotas y visitas médicas.       |
| Administrador | Consulta listado de veterinarios y especialidades. |

Observación: El sistema no implementa autenticación ni control de roles. Todas las operaciones son públicas.

### 3. User Stories Recuperadas

#### US-01 – Registrar dueño de mascota

Como recepcionista quiero registrar un nuevo dueño en el sistema para asociar correctamente sus mascotas.

Fuente técnica: OwnerController.java

#### US-02 – Buscar dueños por apellido

Como recepcionista quiero buscar dueños por apellido para acceder rápidamente a su información.

Fuente técnica: OwnerController.java

### **US-03 – Ver información de un dueño**

Como recepcionista quiero visualizar la información detallada de un dueño para consultar mascotas y visitas.

Fuente técnica: OwnerController.java

### **US-04 – Agregar mascota a un dueño**

Como recepcionista quiero agregar una mascota a un dueño existente para mantener actualizada la información.

Fuente técnica: PetController.java

### **US-05 – Actualizar información de una mascota**

Como recepcionista quiero modificar los datos de una mascota para asegurar que la información sea correcta.

Fuente técnica: PetController.java

### **US-06 – Registrar visita médica**

Como recepcionista quiero registrar una visita médica para una mascota para llevar control clínico.

Fuente técnica: VisitController.java

### **US-07 – Visualizar veterinarios**

Como administrador quiero visualizar el listado de veterinarios y sus especialidades para controlar el personal.

Fuente técnica: VetController.java

### **US-08 – Actualizar información de un dueño**

Como recepcionista quiero actualizar los datos de un dueño existente para mantener información vigente.

Fuente técnica: OwnerController.java

### **US-09 – Visualizar información de una mascota**

Como recepcionista quiero visualizar los datos de una mascota para consultar su tipo y visitas asociadas.

Fuente técnica: OwnerController.java

### **US-10 – Visualizar historial de visitas**

Como recepcionista quiero consultar el historial de visitas de una mascota para revisar antecedentes médicos.

Fuente técnica: VisitController.java / OwnerController.java

## 4. Observaciones Funcionales

Durante el análisis del código se identificaron las siguientes limitaciones:

- No existen funcionalidades de eliminación (Delete) para dueños o mascotas.
- No existe edición de visitas médicas.
- No existe gestión completa CRUD para veterinarios.
- No existe autenticación ni autorización.

## 5. Resumen del Backlog

| Dominio | Cantidad de Historias |
|---------|-----------------------|
| Owner   | 4                     |
| Pet     | 3                     |
| Visit   | 2                     |
| Vet     | 1                     |

Total de historias recuperadas: 10

## 6. Conclusión

El análisis permitió recuperar 10 historias de usuario asociadas a los principales dominios del sistema. El sistema cubre operaciones de registro, consulta y actualización básica, pero presenta limitaciones en eliminación de datos, gestión avanzada de veterinarios y control de acceso. Este documento sirve como base para futuras mejoras y planificación evolutiva.