



## **Plan de pruebas para el Sitio Web del Albergue Huellitas Villa**

**Integrantes:**

**Sebastian Oswaldo Velandia Botello**

**Docente:**

**Fanny Casadiego Chiquillo**

**Facultad de Ingeniería y Arquitecturas**

**Universidad de Pamplona**

**Villa del Rosario**

**2025-2**



## 1. Introducción

### 1.1 Propósito

El propósito de este documento es describir la estrategia de pruebas para el proyecto Sitio Web del Albergue Huellitas Villa.

El plan tiene como objetivo garantizar que el sistema cumpla con los requerimientos funcionales y no funcionales definidos, asegurando su calidad, estabilidad y confiabilidad antes de su entrega final.

### 1.2 Alcance

Este plan de pruebas cubre las actividades necesarias para validar el correcto funcionamiento del sistema web que gestiona los procesos del albergue Huellitas Villa.

El sistema incluye los siguientes módulos:

- **Módulo Usuario:** registro, inicio/cierre de sesión, recuperación de contraseña y actualización de perfil.
- **Módulo Donaciones:** registro y confirmación de donaciones, y registro de historial.
- **Módulo Administración:** gestión de animales (crear, editar, eliminar) y usuarios (excepto administradores).
- **Módulo General:** navegación del sitio, enlaces y visualización de información del albergue.

El alcance incluye pruebas funcionales, de integración, interfaz, base de datos y seguridad. No contempla pruebas de producción ni de integración con pasarelas de pago reales.

### 1.3 Objetivos de Pruebas

- Verificar que las funcionalidades principales del sistema operen correctamente.
- Validar que el software cumpla con los requisitos funcionales y no funcionales especificados.
- Detectar y corregir errores antes de la entrega del producto final.
- Evaluar la usabilidad, seguridad y rendimiento del sitio web.



- Garantizar que los datos registrados se almacenen y gestionen adecuadamente en la base de datos PostgreSQL.

## 2. Estrategia de Pruebas

### 2.1 Enfoque de Pruebas

Para el proyecto Huellitas Villa se aplicará una estrategia de pruebas mixta, combinando técnicas de caja negra y caja gris.

El enfoque general se basará en validar las funcionalidades visibles al usuario y el comportamiento del sistema frente a diferentes entradas y condiciones.

Las pruebas planificadas incluyen:

- Pruebas unitarias:** realizadas por los desarrolladores para verificar el correcto funcionamiento de métodos individuales del backend (controladores, servicios y repositorios).

- Pruebas de integración:** para asegurar la correcta comunicación entre módulos, especialmente entre el frontend HTML/JS y el backend en Spring Boot con PostgreSQL.

- Pruebas funcionales:** centradas en validar el cumplimiento de los requerimientos del usuario final (login, registro, donaciones, gestión de animales, etc.).

- Pruebas de interfaz:** verifican la correcta visualización y navegación de las páginas del sitio web.

- Pruebas de seguridad:** revisión de manejo de sesiones, bloqueo de intentos fallidos de inicio de sesión y control de roles (usuario y administrador).

- Pruebas de aceptación:** realizadas con la participación del cliente para confirmar que el sistema cumple con las expectativas establecidas.

### 2.2 Recursos de Pruebas

#### Personal de pruebas:

- Tester principal: Sebastián (estudiante responsable del proyecto).
- Usuario de prueba: Compañeros de curso que simulan los roles de usuarios y administradores.
- Supervisor: Docente de la asignatura (validación final).



### Ambiente de pruebas:

- Frontend: Navegadores web (Google Chrome, Mozilla Firefox).
- Backend: Spring Boot, Java 21.
- Base de datos: PostgreSQL 16.
- Entorno local: Visual Studio Code y PgAdmin 4.
- Servidor de pruebas: Implementación local (localhost:8080).

### Herramientas de pruebas:

- Postman: para verificar endpoints y respuestas del backend.
- Browser DevTools: para validar errores de red, consola y diseño responsive.
- Excel/Google Sheets: para registrar resultados y defectos encontrados.

## 2.3 Planificación de Pruebas

Tipo de prueba	Inicio	Fin estimado	Responsable	Objetivo
Pruebas unitarias	08/11/2025	09/11/2025	Sebastián	Validar componentes individuales del backend.
Pruebas de integración	09/11/2025	10/11/2025	Sebastian	Verificar comunicación entre módulos (login, donaciones, gestión).
Pruebas funcionales	10/11/2025	11/11/2025	Fernando	Confirmar que las funciones cumplen con los requerimientos.
Pruebas de interfaz	11/11/2025	12/11/2025	Fernando	Evaluar diseño, botones, formularios y enlaces.
Pruebas de aceptación	14/11/2025	14/11/2025	Cliente	Validación final del sistema completo.



## 2.4 Criterios de Aceptación

El sistema se considerará aprobado si cumple con los siguientes criterios:

- El 100 % de los casos de prueba críticos son satisfactorios.
- No se detectan errores graves que impidan el uso normal del sistema.
- El 95 % de las funcionalidades secundarias funcionan sin defectos.
- La interfaz mantiene coherencia en todas las páginas y dispositivos.
- Las credenciales, roles y sesiones funcionan correctamente.
- Los datos registrados se guardan correctamente en la base de datos.

## 3. Casos de Prueba

### 3.1 Identificación de Casos de Prueba Funcionales

A continuación, se presentan los casos de prueba funcionales identificados para las principales funcionalidades del sistema:

ID	Módulo	Nombre del Caso de Prueba	Tipo de Prueba	Prioridad	Estado
CP-01	Usuario	Registro de nuevo usuario	Funcional / Integración	Alta	Pendiente
CP-02	Usuario	Inicio de sesión con credenciales válidas	Funcional / Seguridad	Alta	Pendiente
CP-03	Usuario	Bloqueo tras tres intentos fallidos de inicio sesión	Seguridad / Interfaz	Media	Pendiente
CP-04	Usuario	Edición del perfil del usuario	Funcional / Integración	Media	Pendiente
CP-05	Donaciones	Registro de donación	Funcional / Integración	Alta	Pendiente
CP-06	Donaciones	Visualización del mensaje "Donación exitosa"	Interfaz / Validación	Media	Pendiente
CP-07	Administración	Agregar nuevo animal	Funcional / Integración	Alta	Pendiente
CP-08	Administración	Editar o eliminar registro de animal	Funcional / Integración	Alta	Pendiente



CP-09	Seguridad	Validar persistencia de sesión	Seguridad / Integración	Alta	Pendiente
CP-10	General	Navegación entre páginas	Interfaz / Usabilidad	Media	Pendiente

### 3.2 Descripción de Casos de Prueba Funcionales

A continuación, se detallan los principales casos de prueba, sus entradas, pasos, resultados esperados y criterios de aprobación:

#### Caso de Prueba CP-01: Registro de nuevo usuario

- Objetivo: Validar que un usuario pueda registrarse correctamente.
- Entradas: Nombre, correo, contraseña y confirmación de contraseña.
- Pasos de ejecución:
  - 1.Acceder a la página usuario.html.
  - 2.Completar los campos del formulario de registro.
  - 3.Presionar el botón “Registrarse”.
- Resultado esperado: El sistema muestra el mensaje “Registro exitoso” y almacena la información en la base de datos.
- Criterio de aceptación: El usuario queda registrado y puede iniciar sesión posteriormente.

#### Caso de Prueba CP-02: Inicio de sesión con credenciales válidas

- Objetivo: Verificar que el login funcione con datos correctos.
- Entradas: Correo y contraseña válidos.
- Pasos de ejecución:
  - 1.Acceder a usuario.html.
  - 2.Ingresar credenciales válidas.
  - 3.Pulsar “Iniciar sesión”.



- Resultado esperado: El sistema inicia sesión y redirige a index.html.

- Criterio de aceptación: La sesión queda activa en el localStorage.

### **Caso de Prueba CP-03: Bloqueo tras tres intentos fallidos**

- Objetivo: Verificar el bloqueo temporal del usuario después de tres intentos fallidos.

- Entradas: Correo válido y contraseñas incorrectas.

- Pasos de ejecución:

- 1.Intentar iniciar sesión tres veces seguidas con contraseña incorrecta.

- 2.Esperar la cuenta regresiva del bloqueo.

- Resultado esperado: El sistema muestra el mensaje “Usuario bloqueado, intente nuevamente en X segundos”.

- Criterio de aceptación: El botón de inicio de sesión queda deshabilitado durante el bloqueo.

### **Caso de Prueba CP-04: Edición del perfil del usuario**

- Objetivo: Comprobar que el usuario logueado pueda modificar su información personal correctamente.

- Entradas: Nuevos valores para nombre, correo, teléfono o fecha de nacimiento.

- Pasos de ejecución:

- 1.Iniciar sesión con credenciales válidas.

- 2.Acceder a perfil.html.

- 3.Editar los campos deseados y presionar “Guardar cambios”.

- Resultado esperado: El sistema actualiza los datos del usuario en la base de datos y muestra el mensaje “Perfil actualizado correctamente”.

- Criterio de aceptación: Los nuevos datos se reflejan al recargar la página o volver a iniciar sesión.

### **Caso de Prueba CP-05: Registro de donación**

- Objetivo: Comprobar que el usuario pueda registrar una donación correctamente.





- Entradas: Monto, método de pago y descripción opcional.
- Pasos de ejecución:
  1. Acceder a donaciones.html.
  2. Seleccionar el método de pago.
  3. Completar el formulario y confirmar la donación.
- Resultado esperado: El sistema muestra el mensaje “Donación exitosa” y guarda el registro en la base de datos.
- Criterio de aceptación: La información queda almacenada en la tabla donaciones.

#### **Caso de Prueba CP-06: Visualización del mensaje “Donación exitosa”**

- Objetivo: Verificar que, al completar una donación, el sistema muestre correctamente el mensaje de confirmación.
- Entradas: Donación registrada correctamente.
- Pasos de ejecución:
  - 1.Ingresar al módulo de donaciones.
  - 2.Llenar los campos requeridos y enviar el formulario.
- Resultado esperado: Aparece el mensaje “Donación exitosa” y los datos quedan registrados.
- Criterio de aceptación: El mensaje aparece sin errores y el registro se guarda en la base de datos.

#### **Caso de Prueba CP-07: Agregar nuevo animal (Administrador)**

- Objetivo: Validar la creación de un nuevo registro en el módulo de administración.
- Entradas: Nombre, edad, especie, descripción y foto.
- Pasos de ejecución:
  - 1.Acceder como administrador a admin.html.
  - 2.Llenar el formulario “Agregar Animal”.





3. Pulsar “Guardar”.

- Resultado esperado: El nuevo animal se agrega al listado de la base de datos.
- Criterio de aceptación: El registro aparece correctamente en la tabla y puede editarse o eliminarse.

### **Caso de Prueba CP-08: Editar o eliminar registro de animal**

- Objetivo: Comprobar que el administrador pueda actualizar o eliminar animales registrados.
- Entradas: Datos del animal existente (ID, nombre, especie, descripción).

• Pasos de ejecución:

1. Iniciar sesión como administrador.
2. Acceder a admin.html.
3. Seleccionar un animal existente.
4. Editar sus datos y guardar o presionar “Eliminar”.

• Resultado esperado:

- En edición: se actualiza la información correctamente.
- En eliminación: el registro desaparece del listado.
- Criterio de aceptación: Los cambios se reflejan en la base de datos sin errores.

### **Caso de Prueba CP-09: Validar persistencia de sesión**

• Objetivo: Confirmar que la sesión del usuario permanezca activa mientras no se cierre manualmente.

• Entradas: Inicio de sesión exitoso.

• Pasos de ejecución:

1. Iniciar sesión.
2. Cerrar el navegador y volver a abrirlo.
3. Acceder nuevamente al sitio.



- Resultado esperado: El sistema detecta la sesión activa mediante localStorage y muestra el estado “Usuario logueado”.

- Criterio de aceptación: El usuario no necesita volver a iniciar sesión hasta cerrar la sesión manualmente.

### **Caso de Prueba CP-10: Navegación entre páginas**

- Objetivo: Verificar que todos los enlaces del sitio funcionen correctamente y redirijan a las páginas esperadas.

- Entradas: Clic en los botones del menú principal (Inicio, Donaciones, Contacto, etc.).

- Pasos de ejecución:

- 1.Acceder al sitio web.

- 2.Hacer clic en cada enlace del menú y botones de navegación.

- Resultado esperado: Todas las páginas cargan correctamente y muestran el contenido correspondiente.

- Criterio de aceptación: Ningún enlace genera errores ni rutas rotas.



### 3.3 Identificación de Casos de Prueba no Funcionales

A continuación, se presentan los casos de prueba no funcionales identificados para las principales funcionalidades del sistema:

ID	Módulo	Nombre del Caso de Prueba	Tipo de Prueba	Prioridad	Estado
NF-01	Usabilidad	Evaluación de claridad y comprensión de la interfaz	Usabilidad	Alta	Pendiente
NF-02	Rendimiento	Medición del tiempo de carga de páginas	Rendimiento	Alta	Pendiente
NF-03	Seguridad	Acceso no autorizado sin sesión activa	Seguridad	Alta	Pendiente
NF-04	Seguridad	Manipulación del localStorage para obtener privilegios	Seguridad	Alta	Pendiente
NF-05	Compatibilidad	Prueba en navegadores diferentes (Chrome, Edge, Firefox)	Compatibilidad	Media	Pendiente
NF-06	Estabilidad	Verificación de sesión persistente durante la navegación	Estabilidad	Media	Pendiente
NF-07	Mantenibilidad	Cambio de datos en perfil sin afectar otros módulos	Mantenibilidad	Media	Pendiente
NF-08	Portabilidad	Verificación de funcionamiento en tamaño móvil	Portabilidad	Media	Pendiente
NF-09	Estabilidad UI	Comportamiento del sistema al actualizar la página	Estabilidad	Baja	Pendiente
NF-10	Procedimientos	Validación del flujo completo de navegación	Procedimientos	Alta	Pendiente



### 3.4 Descripción Detallada de Casos de Prueba No Funcionales

#### Caso de Prueba NF-01: Claridad y comprensión de la interfaz

- Objetivo: Evaluar si los usuarios comprenden fácilmente los elementos visuales, botones, formularios y mensajes del sistema.
- Entradas: Interacción libre del usuario con la interfaz.
- Pasos de ejecución:
  1. Acceder al sitio web.
  2. Navegar entre todas las páginas.
  3. Observar botones, formularios y mensajes emergentes.
  4. Registrar si el usuario identifica correctamente la función de cada elemento.
- Resultado esperado: La interfaz debe ser clara y fácil de interpretar por cualquier usuario sin necesidad de instrucciones externas.
- Criterio de aceptación: Al menos el 90% de los elementos deben ser comprendidos sin confusión.
- Métrica aplicada: Puntuación de accesibilidad (Lighthouse) — valor mínimo aceptado: 80%.

#### Caso de Prueba NF-02: Tiempo de carga de páginas

- Objetivo: Medir el tiempo de carga de las principales páginas del sistema.
- Entradas: URL de cada página del sitio.
- Pasos de ejecución:
  1. Abrir la herramienta Lighthouse en Google Chrome.
  2. Ejecutar auditoría de Performance.
  3. Registrar el tiempo de carga del sitio y de cada vista.
  4. Comparar resultados con el límite establecido.
- Resultado esperado: Las páginas deben cargar en menos de 3 segundos.
- Criterio de aceptación: Tiempo de carga  $\leq 3$  segundos en la primera visita.
- Métrica aplicada: LCP, FCP y TBT — LCP < 2.5s, FCP < 1.5s, TBT < 300ms.



### Caso de Prueba NF-03: Acceso no autorizado

- Objetivo: Verificar que un usuario sin sesión activa no pueda acceder a páginas restringidas.
- Entradas: URLs protegidas del sistema (perfil, donaciones, admin).
- Pasos de ejecución:
  1. Cerrar sesión del sistema.
  2. Intentar acceder directamente a /perfil.html, /donaciones.html o /admin.html escribiendo la URL.
  3. Observar el comportamiento del sistema.
- Resultado esperado: El sistema debe redirigir automáticamente al login.
- Criterio de aceptación: Ninguna página restringida debe cargarse sin sesión activa.
- Métrica aplicada: Tasa de intentos de acceso no autorizado detectados (0 accesos permitidos).

### Caso de Prueba NF-04: Manipulación del localStorage

- Objetivo: Verificar que modificar manualmente el rol o datos en localStorage no otorgue privilegios indebidos.
- Entradas: Modificación del valor “rol” o “usuarioActivo”.
- Pasos de ejecución:
  1. Cerrar sesión.
  2. Abrir DevTools → Application → LocalStorage.
  3. Cambiar manualmente el valor “rol” a “ADMIN”.
  4. Intentar acceder a /admin.html.
- Resultado esperado: El sistema debe bloquear el acceso y redirigir al login.
- Criterio de aceptación: Ninguna manipulación del localStorage debe conceder roles privilegiados.
- Métrica aplicada: Detección de vulnerabilidades — severidad aceptada “Baja” o inexistente.



### Caso NF-05: Compatibilidad – Prueba en distintos navegadores

- Objetivo: Verificar que el sistema funcione correctamente en Chrome, Edge y Firefox.
- Entradas: Navegación estándar del usuario.
- Pasos de ejecución:
  1. Abrir el sitio en Chrome, Edge y Firefox.
  2. Navegar por todas las páginas.
  3. Probar botones, formularios y enlaces.
- Resultado esperado: El sistema debe comportarse igual en todos los navegadores.
- Criterio de aceptación: No deben existir diferencias funcionales o visuales entre navegadores.
- Métrica aplicada: Porcentaje de compatibilidad = navegadores funcionales / probados × 100 (mínimo 100%).

### Caso NF-06: Persistencia de sesión

- Objetivo: Verificar que la sesión del usuario permanezca activa durante la navegación continua.
- Entradas: Sesión activa en localStorage.
- Pasos de ejecución:
  1. Iniciar sesión en el sistema.
  2. Navegar durante 10 minutos por todas las páginas.
  3. Verificar si la sesión permanece activa.
- Resultado esperado: La sesión no debe cerrarse automáticamente.
- Criterio de aceptación: La sesión debe permanecer activa hasta que el usuario cierre sesión manualmente.
- Métrica aplicada: Tasa de persistencia de sesión (debe mantenerse en 100% de las pruebas).

### Caso NF-07: Modificar datos sin afectar módulos

- Objetivo: Comprobar que los cambios realizados en el perfil del usuario no afecten otras funcionalidades del sistema.
- Entradas: Datos modificados en perfil.



- Pasos de ejecución:

1. Iniciar sesión.
2. Editar datos del perfil.
3. Navegar a páginas como donaciones y animales.
4. Observar si aparece algún error.

- Resultado esperado: El cambio de datos no debe generar fallos en otros módulos.

- Criterio de aceptación: No se deben presentar errores tras la modificación del perfil.

- Métrica aplicada: Índice de fallos recurrentes tras modificación (debe ser 0%).

### **Caso NF-08: Visualización en dispositivo móvil**

- Objetivo: Evaluar la visualización del sistema en tamaños de pantalla móvil.

- Entradas: Modo “Responsive” en DevTools.

- Pasos de ejecución:

1. Abrir el sitio en Google Chrome.
2. Activar DevTools → Toggle Device Toolbar.
3. Seleccionar un tamaño móvil (iPhone XR o Galaxy S20).
4. Navegar por las páginas.

- Resultado esperado: La interfaz debe ajustarse correctamente al tamaño móvil.

- Criterio de aceptación: Todos los elementos deben ser visibles y accesibles.

- Métrica aplicada: Puntuación móvil (Lighthouse Mobile) — valor mínimo aceptado: 75%.

### **Caso NF-09: Comportamiento al recargar**

- Objetivo: Verificar que el sistema mantenga el estado correcto al actualizar la página.

- Entradas: Sesión activa.

- Pasos de ejecución:

1. Iniciar sesión.
2. Navegar a otra página.
3. Presionar F5 para recargar.





- Resultado esperado: El sistema debe conservar la sesión y no mostrar errores.
- Criterio de aceptación: La sesión no debe reiniciarse tras recargar la página.
- Métrica aplicada: Consistencia de interfaz después de recarga (debe mantenerse al 100%).

### Caso NF-10: Flujo completo

- Objetivo: Evaluar la consistencia del sistema en un flujo completo (inicio → acción → cierre).
- Entradas: Acciones completas del usuario.
- Pasos de ejecución:
  1. Iniciar sesión.
  2. Realizar una acción: donar, editar perfil o registrar animal.
  3. Cerrar sesión.

- Resultado esperado: El sistema debe completar el flujo sin errores.
- Criterio de aceptación: Ningún paso del flujo debe presentar fallos.
- Métrica aplicada: Porcentaje de pasos completados sin error (debe ser 100%).

## 4. Ejecución de Pruebas

### 4.1 Procedimientos de Ejecución

Las pruebas se llevarán a cabo de manera secuencial siguiendo el orden establecido en la planificación.

Cada prueba será ejecutada en el entorno local, registrando los resultados obtenidos y cualquier incidencia detectada.

#### Pasos generales del procedimiento:

##### 1.Preparación del entorno:

- o Iniciar el servidor de Spring Boot.
- o Verificar la conexión con la base de datos PostgreSQL.
- o Asegurar que las páginas HTML y archivos JavaScript estén actualizados y funcionales.



## 2.Ejecución de los casos de prueba:

- o Seguir los pasos detallados en el punto 3.2 para cada caso.
- o Registrar los resultados obtenidos (éxito, error o falla parcial).
- o Capturar evidencias mediante capturas de pantalla o registros de consola en caso de error.

## 3.Evaluación de resultados:

- o Comparar los resultados obtenidos con los resultados esperados.
- o Clasificar los defectos encontrados según su gravedad (crítico, medio o leve).

## 4.Revisión y corrección:

- o En caso de defectos, reportarlos y documentarlos.
- o Realizar las correcciones en el código o configuración correspondiente.
- o Repetir la prueba para confirmar la solución del error.

## 5.Validación final:

- o Realizar una última ronda de pruebas funcionales y de aceptación para garantizar que el sistema esté estable y cumpla con los criterios establecidos.



## 4.2 Registro de Defectos

Todos los errores o defectos identificados durante las pruebas serán registrados en un formato de control que incluye la siguiente información (La siguiente tabla muestre ejemplos previstos):

ID Defecto	Caso de Prueba Relacionado	Descripción del Error	Gravedad	Estado	Responsable	Acción Tomada
DF-01	CP-01	El formulario de registro no valida correctamente el formato del correo electrónico.	Medio	Corregido	Sebastián	Se ajustó la expresión regular de validación.
DF-02	CP-03	El contador del bloqueo no se reinicia después de la espera.	Bajo	Pendiente	Sebastián	Revisar el temporizador de JavaScript.
DF-03	CP-07	Al agregar un nuevo animal no se actualiza la lista automáticamente.	Medio	Corregido	Sebastián	Se agregó recarga dinámica de tabla tras inserción.
DF-04	CP-09	La sesión no se elimina correctamente al cerrar sesión.	Alto	En revisión	Sebastián	Revisar limpieza de localStorage y backend.



## 5. Riesgos y Contingencias

### 5.1 Riesgos

Durante la ejecución de las pruebas del sistema Huellitas Villa, pueden presentarse varios riesgos que afecten la validez, continuidad o calidad del proceso de validación del software.

A continuación, se listan los principales riesgos posibles:

ID	Riesgo	Descripción	Probabilidad	Impacto	Nivel
R-01	Fallos de conexión con la base de datos	El servidor local puede perder conexión con PostgreSQL, impidiendo el registro o consulta de datos.	Media	Alta	Alto
R-02	Datos inconsistentes	Errores al ingresar datos pueden provocar inconsistencias o duplicados durante las pruebas.	Alta	Media	Alto
R-03	Fallas del entorno de desarrollo	Posibles errores en Spring Boot, Postman o el navegador pueden retrasar la ejecución de pruebas.	Media	Media	Medio
R-04	Tiempo insuficiente para repetir pruebas	La planificación ajustada puede limitar el tiempo para repetir casos después de corregir errores.	Alta	Media	Alto
R-05	Cambios de última hora en el código	Modificaciones al backend o frontend durante las pruebas pueden invalidar resultados anteriores.	Media	Alta	Alto
R-06	Errores humanos	Falta de atención o errores al registrar resultados pueden alterar la precisión del informe final.	Media	Media	Medio
R-07	Incompatibilidad del navegador	Algunas pruebas pueden variar dependiendo del navegador o su versión.	Baja	Media	Bajo

### 5.2 Estrategias de Mitigación

Para reducir el impacto de los riesgos posibles, se plantean las siguientes estrategias de mitigación:

- R-01: Mantener una copia de respaldo de la base de datos y verificar la conexión antes de cada sesión de pruebas.



- R-02: Realizar pruebas con datos controlados y limpiar la base de datos antes de cada ejecución.
- R-03: Utilizar versiones estables de software y reiniciar los entornos de prueba en caso de fallo.
- R-04: Establecer tiempos de margen entre cada ciclo de pruebas para posibles repeticiones.
- R-05: Congelar el código durante la fase de pruebas (no realizar cambios en producción).
- R-06: Registrar resultados inmediatamente y revisar los informes con cuidado.
- R-07: Estandarizar las pruebas en un solo navegador (por ejemplo, Google Chrome) para mantener consistencia.

## 6. Entregables

### 6.1 Informes de Pruebas

Durante la ejecución del plan de pruebas se generarán diversos documentos que evidencian la correcta validación del sistema. Estos entregables permitirán mantener un control detallado del proceso y documentar los resultados obtenidos.

#### Informes principales:

##### 1.Registro de Casos de Prueba:

Documento (en Excel o Word) que incluye los casos de prueba con sus resultados, observaciones, y evidencias (capturas de pantalla o registros).

- Contenido: ID del caso, descripción, resultado esperado, resultado obtenido, estado (éxito, fallo, pendiente).

##### 2.Registro de Defectos:

Archivo complementario donde se documentan los errores detectados, su gravedad, causa y solución aplicada.

- Contenido: ID del defecto, descripción, módulo afectado, responsable, estado y fecha de corrección.

##### 3.Informe de Resultados de Pruebas:

Documento resumen (que servirá como Informe de ejecución del plan de pruebas) donde se presentan los resultados finales, las métricas alcanzadas, defectos corregidos y conclusiones.



- o Incluye gráficos o tablas comparativas entre resultados esperados y reales.

#### 4.Evidencias de Prueba:

Capturas de pantalla o videos que muestren el funcionamiento de los módulos probados, así como los errores y sus correcciones.

#### 5.Checklist de Validación Final:

Formato donde se verifican los criterios de aceptación definidos en el punto 2.4, marcando qué pruebas cumplen o no los requisitos.

### 6.2 Documentación Actualizada

Toda la documentación de pruebas será almacenada y actualizada en la carpeta del proyecto, dentro del repositorio de GitHub.

El mantenimiento de esta documentación garantizará la trazabilidad de las pruebas y la disponibilidad de información para futuras versiones del sistema.

#### Lineamientos:

- La documentación se mantendrá sincronizada en el repositorio bajo la carpeta /Pruebas.
- Cada modificación o actualización deberá registrarse con fecha y versión.
- Al finalizar la ejecución de pruebas, se generará una versión final validada por el responsable del proyecto.
- Los cambios en el sistema que afecten funcionalidades deberán ir acompañados de la actualización de sus respectivos casos de prueba.

### 7. Métricas de Pruebas

El presente plan de pruebas establece una serie de métricas que permitirán evaluar objetivamente la calidad del software, la cobertura alcanzada por las pruebas y la eficiencia del proceso de verificación. Estas métricas se aplicarán durante la ejecución del plan y servirán como base para el análisis posterior en el Informe de Ejecución de Pruebas.



## 7.1 Métricas de Cobertura

### Cobertura de Casos de Prueba

Permite conocer qué porcentaje de casos de prueba fueron ejecutados respecto al total planificado.

#### Fórmula:

$$(\text{Casos ejecutados} / \text{Casos planificados}) \times 100$$

### Cobertura de Requisitos

Mide qué proporción de requisitos funcionales y no funcionales cuentan con al menos un caso de prueba asociado.

#### Fórmula:

$$(\text{Requisitos cubiertos} / \text{Requisitos totales}) \times 100$$

## 7.2 Métricas de Defectos

### Densidad de Defectos

Número de defectos encontrados por módulo o funcionalidad.

#### Fórmula:

$$(\text{Defectos totales} / \text{Tamaño del módulo})$$

### Tasa de Defectos Detectados

Indica cuántos defectos se encuentran durante una hora de ejecución de pruebas.

#### Fórmula:

$$(\text{Defectos detectados} / \text{Horas de prueba})$$





## Distribución de Defectos por Severidad

Clasificación de los defectos detectados según su impacto:

- Crítico
- Alto
- Medio
- Bajo

## 7.3 Métricas de Esfuerzo

### Tiempo Promedio por Caso de Prueba

Tiempo invertido en ejecutar un caso de prueba.

#### Fórmula:

(Tiempo total de ejecución / Casos ejecutados)

### Retrabajo por Defectos

Tiempo dedicado a corregir defectos y volver a ejecutar los casos afectados.

## 7.4 Métricas de Rendimiento del Sistema

### Tiempo de Respuesta

Evalúa cuánto tarda el sistema en cargar páginas o procesar una acción.

Se medirá mediante herramientas como:

- Google Lighthouse
- GTmetrix
- Chrome Performance Tools

### Uso de Recursos

Evalúa el consumo de memoria, CPU y red mientras se interactúa con el sistema.



## 7.5 Métricas de Calidad General

### Porcentaje de Pruebas Exitosas

$(\text{Casos aprobados} / \text{Casos ejecutados}) \times 100$

### Índice de Fallos Recurrentes

Número de fallos repetidos tras correcciones (mide estabilidad y mantenibilidad del software).

## 8. Aprobación

### 8.1 Aprobación del Cliente

Una vez completadas las pruebas y verificados los resultados, se realizará una sesión de revisión con el cliente.

El proceso de aprobación consistirá en:

1. Presentación del informe de resultados: donde se evidencien los casos de prueba ejecutados, defectos corregidos y evidencias gráficas.
2. Demostración funcional: ejecución en vivo del sistema mostrando las funcionalidades principales (login, registro, donaciones, gestión de animales y cierre de sesión).
3. Validación de criterios de aceptación: confirmación de que el software cumple los requisitos establecidos en el alcance inicial.
4. Firma o visto bueno: el cliente confirmará la aprobación del sistema una vez todos los puntos estén verificados y sin defectos críticos.

Aprobado por:

- Nombre del Cliente: \_\_\_\_\_
- Firma: \_\_\_\_\_
- Fecha: \_\_\_\_\_



## 8. Historial de Revisiones

Versión	Descripción del Cambio	Autor	Fecha
1.0	Creación del documento inicial del Plan de Pruebas	Sebastián	07/11/2025