Universidad Centroamericana José Simeón Cañas

Facultad de Ingeniería y Arquitectura

Departamento de Electrónica e informática

Ingeniería de Software

Plan de pruebas – VetiCare

Catedrático:

Ing. Nelson Giovanni Castro Rodas

Ing. Luisa Daniela Arévalo Rodas

Nombre de integrantes:

Alison Aracely Argueta Flores 00076422 sección 01

Natalia Melissa Avelar Menjivar 00135922 sección 01

Diego Eduardo Castro Quintanilla 00117322 sección 01

Josué Fernando Gómez Guardado 00103722 sección 01

Christian Joel López Ortega 00179320 sección 01

Fecha: 23 de noviembre de 2025

Casos de Prueba por Historia de Usuario – VetiCare

Historia de Usuario HU-1: Registro con validaciones y redirección automática

CP-001: Validar campos obligatorios y formato básico en el formulario de registro

Tipo: Funcional
Prioridad: Alta

Precondiciones

- El sistema frontend y backend están disponibles.
- El usuario no ha iniciado sesión.
- La vista de registro es accesible desde la ruta correspondiente.

Pasos de ejecución

1. Navegar a la página de registro.
2. Dejar vacíos los campos obligatorios (por ejemplo: nombre completo, correo, teléfono, contraseña).
3. Intentar enviar el formulario.
4. Completar solo algunos campos con formatos incorrectos (por ejemplo, correo sin @, teléfono sin guion 00000000, DUI sin guion 00000000).
5. Intentar enviar nuevamente el formulario.

Resultado esperado

- El sistema muestra mensajes de error indicando que los campos obligatorios están vacíos.
- Los mensajes indican claramente cuál es el problema (por ejemplo: “El campo correo es requerido”, “Ingresa un correo electrónico válido”).
- El sistema valida el formato de los campos (correo, DUI, teléfono) y marca como inválidos los que no cumplen el patrón establecido.

CP-002: Validar mensajes de validación en tiempo real mientras se escribe

Tipo: Funcional / UX
Prioridad: Alta

Precondiciones

- Vista de registro cargada correctamente.

Pasos de ejecución

1. Colocar el cursor en el campo “Correo electrónico”.
2. Escribir un texto sin formato de correo (por ejemplo: correo_invalido).
3. Pasar al siguiente campo (blur) o seguir escribiendo en el mismo.
4. Observar el mensaje de validación que aparece debajo del campo.
5. Corregir el valor hasta que sea un correo válido (por ejemplo: usuario@test.com).
6. Repetir el proceso con otros campos con validación de formato (por ejemplo: teléfono 0000-0000, DUI 00000000-0, nombre sin números).

Resultado esperado

- El sistema muestra mensajes de error en tiempo real cuando el valor del campo no cumple el formato requerido.
- Al corregir el valor del campo, el mensaje de error desaparece sin necesidad de recargar la página ni reenviar el formulario.
- El usuario puede identificar fácilmente qué campo tiene el error y cómo corregirlo.

CP-003: Validar que no se permita enviar el formulario con errores o campos vacíos

Tipo: Funcional / Validación
Prioridad: Alta

Precondiciones

- Vista de registro cargada correctamente.

Pasos de ejecución

1. Ingresar datos válidos en algunos campos y dejar otros campos obligatorios vacíos.
2. Ingresar intencionalmente formatos incorrectos en al menos un campo (por ejemplo, teléfono sin guion).
3. Hacer clic en el botón “Registrarse”.
4. Abrir DevTools en la pestaña Network y verificar si se envía o no la petición al endpoint de registro.

Resultado esperado

- El botón de registro no envía la petición al backend mientras haya errores de formato o campos obligatorios vacíos.
- El usuario permanece en la misma página de registro.

- Los campos con errores se muestran claramente marcados y con su mensaje correspondiente.
- En Network no se observa request de tipo POST al endpoint de registro cuando existen errores.

Historia de Usuario HU-2 – Mensajes amigables, avatares y emojis para el propietario

CP-004: Validar mensajes claros cuando el dueño no tiene mascotas ni citas

Tipo: Funcional
Prioridad: Alta

Precondiciones

- El usuario tiene una cuenta creada como dueño.
- El usuario puede iniciar sesión correctamente.
- La cuenta no tiene mascotas registradas ni citas creadas.

Pasos de ejecución

1. Ingresar al sistema con un usuario dueño.
2. Observar el mensaje que se muestra al no tener citas registradas.
3. Acceder a la sección "Mis mascotas" desde el menú principal.
4. Observar el mensaje que se muestra al no tener mascotas registradas.
5. Acceder a la sección "Citas" desde el menú.
6. Observar el mensaje que se muestra al no tener citas registradas.

Resultado esperado

- Al iniciar sesión, el sistema muestra un mensaje claro indicando que no hay citas disponibles.
- La sección "Mis mascotas" muestra un mensaje claro indicando que no existen mascotas registradas.
- El mensaje es amigable, comprensible y sin errores visibles.
- La sección "Citas" muestra un mensaje informativo indicando que no existen citas registradas.
- No deben aparecer mensajes técnicos, errores ni valores nulos.

CP-005: Validar creación de cita con mensaje de éxito

Tipo: Funcional
Prioridad: Alta

Precondiciones

- El usuario ha iniciado sesión como propietario.
- Tiene al menos una mascota registrada.

Pasos de ejecución

1. Ingresar al sistema con un usuario dueño.
2. Acceder a la sección “Citas”.
3. Presionar el botón para crear una nueva cita.
4. Completar los campos del formulario.
5. Guardar la cita.
6. Verificar que la cita aparece en la tabla.

Resultado esperado

- El sistema muestra un mensaje de éxito al crear la cita.
- La cita aparece en la tabla de historial con los datos ingresados.

CP-006: Validar cambio de foto de perfil

Tipo: Funcional
Prioridad: Media

Precondiciones

- El usuario ha iniciado sesión.

Pasos de ejecución

1. Ingresar al sistema como dueño.
2. Ir a la sección “Perfil”.
3. Presionar el botón “Cambiar foto”.
4. Seleccionar una nueva foto de perfil.
5. Guardar los cambios.
6. Verificar que la foto se actualiza.

Resultado esperado

- Se muestra un mensaje confirmando que la foto fue actualizada.
- La nueva foto se visualiza inmediatamente en el perfil.

CP-007: Validar desactivación de mascota

Tipo: Funcional
Prioridad: Media

Precondiciones

- El usuario es de tipo dueño.
- Tiene al menos una mascota registrada.

Pasos de ejecución

1. Ingresar al sistema como dueño.
2. Ir a la sección “Mis mascotas”.
3. Presionar el botón de desactivar en una mascota.
4. Confirmar la desactivación.

Resultado esperado

- El sistema solicita una confirmación clara.
- La mascota desactivada deja de mostrarse en la tabla de “Mis mascotas”.

Criterios de aceptación cubiertos: AC1, AC2, AC3 y AC4

Historia de Usuario HU-3 – Rediseñar las tablas del propietario de mascotas

CP-008: Validar estructura visual de la tabla "Mis mascotas"

Tipo: Funcional
Prioridad: Media

Precondiciones

- El usuario ha iniciado sesión como propietario.
- El usuario cuenta con al menos una mascota registrada.

Pasos de ejecución

1. Ingresar al sistema como propietario.
2. Acceder a la sección “Mis mascotas”.
3. Observar el título, icono y texto descriptivo.
4. Verificar columnas y estructura visual de la tabla.

Resultado esperado

- La pantalla muestra un encabezado claro y visual.
- La tabla contiene las columnas: nombre, especie, raza, emoji, historial médico y acciones.
- Los iconos de modificar y eliminar se muestran correctamente.
- Los datos en las filas están centrados y se leen fácilmente.

CP-009: Validar edición de mascota desde la tabla

Tipo: Funcional
Prioridad: Alta

Precondiciones

- El usuario es propietario.
- Debe tener al menos una mascota registrada.

Pasos de ejecución

1. Ingresar al sistema como dueño.
2. Ir a la sección “Mis mascotas”.
3. Presionar el botón de editar en una mascota.
4. Modificar los campos disponibles.
5. Guardar los cambios.

Resultado esperado

- El formulario se muestra con los datos actuales.
- Al presionar el botón actualizar, los datos se guardan correctamente y se reflejan en la tabla.

CP-010: Validar visualización de historial médico

Tipo: Funcional
Prioridad: Media

Precondiciones

- El usuario es propietario.
- Debe contar con una mascota que haya asistido a una cita con un veterinario.

Pasos de ejecución

1. Ingresar al sistema como dueño.
2. Ir a “Mis mascotas”.
3. Presionar el botón de historial.
4. Revisar la información mostrada.

Resultado esperado

- Si la mascota tiene historial, se muestran sus citas.
- Si no, se muestra un mensaje claro y amigable.

Criterios de aceptación cubiertos: AC1, AC2 y AC3

HU 4 – Corrección de errores de visualización, botones e iconos para interfaz de veterinario

CP-011: Validar visibilidad del botón (X) en el modal de creación de expediente

Tipo: Funcional / UI
Prioridad: Alta

Precondiciones

- El usuario debe estar loggeado como veterinario.
- El modal de “Nuevo expediente” debe poder abrirse desde la sección correspondiente.

Pasos

1. Navegar a la sección **Expedientes**.
2. Hacer clic en **Nuevo expediente**.
3. Verificar la visibilidad del botón **X** en la esquina superior derecha.

Resultado esperado

- La “X” es claramente visible con color contrastante.
- La “X” no está pegada a bordes o oculta detrás de elementos.

Criterios de aceptación cubiertos: AC1

CP-012: Validar funcionamiento del botón (X) al cerrar modal

Tipo: Funcional
Prioridad: Alta

Precondiciones

- Modal de “Nuevo expediente” abierto.

Pasos de ejecución

1. Hacer clic en la “X” del modal.
2. Verificar comportamiento.

Resultado esperado

- El modal se cierra inmediatamente.
- No se guarda ningún cambio.
- No se genera ningún error en consola.

Criterios de aceptación cubiertos: AC2

CP-013: Validar que todos los botones de la vista del veterinario respondan correctamente

Tipo: Funcional
Prioridad: Media

Precondiciones

- Veterinario logueado.
- Secciones funcionales (Citas, Mascotas, Expedientes).

Pasos de ejecución

1. Navegar por todas las pestañas del veterinario.
2. Probar botones: guardar, crear, cancelar, buscar, etc.
3. Observar si alguno no reacciona.

Resultado esperado

- Todos los botones ejecutan su acción correctamente.
- No hay botones congelados o visualmente rotos.

Criterios de aceptación cubiertos: AC3

HU-5: Manejo de datos vacíos o incompletos de gráficas e información de empleados

CP-014: Validar búsqueda de limpieza en las listas de citas, veterinarios, dueños, mascotas y administradores en la sesión de administrador

Tipo: Funcional / UI
Prioridad: Alta

Precondiciones

- El usuario tiene las credenciales de una cuenta de superadministrador o administrador
- El usuario puede iniciar sesión correctamente
- Existen al menos 2 cuentas de administrador y veterinarios registradas
- Existen al menos 2 mascotas registradas
- Existen al menos 2 citas registradas

Pasos de ejecución:

1. Ingresar sesión con éxito en el sistema con una cuenta de superadministrador
2. La aplicación redirige al dashboard del negocio automáticamente
3. Navegar a las secciones de “Citas”, “Veterinarios”, “Dueños”, “Mascotas” o “administrador” que se encuentran en el menú lateral
4. Escribir en el campo de búsqueda de la pestaña un nombre existente
5. Verificar que la tabla se filtra correctamente
6. Presionar el botón X al lado del buscador
7. Verificar que se borraron los datos ingresados
8. Verificar que la tabla muestre todos los datos nuevamente

Resultado esperado:

- La búsqueda filtra correctamente los resultados
- Funciona el botón de limpieza del buscador y realiza la acción
- No se superpone el botón de limpieza con el otro de buscador

Criterios de aceptación verificados: AC1

CP-015: Validar mensajes de error en formularios de diferentes vistas del administrador y mensajes de confirmar a acciones importantes

Tipo: Funcional / UI
Prioridad: Alta

Precondiciones

- El usuario tiene las credenciales de una cuenta de superadministrador o administrador
- El usuario puede iniciar sesión correctamente
- Existen al menos una cuenta de administrador y de veterinario registrada
- Existen al menos una mascota registrada
- Existen al menos una cita registrada

Pasos de ejecución:

1. Ingresar sesión con éxito en el sistema con una cuenta de superadministrador
2. La aplicación redirige al dashboard del negocio automáticamente
3. Navegar a las secciones de “Citas”, “Veterinarios”, “Dueños”, “Mascotas” o “administrador” que se encuentran en el menú lateral
4. Presionar en la acción de editar datos
5. Verificar que se abre el modal con los datos necesarios
6. Colocar datos erróneos
7. Presionar enviar
8. Colocar datos correctos
9. Presionar botón de enviar
10. Verificar que se actualizara con éxito los datos

Resultado esperado:

- Al colocar datos inválidos brinda mensaje de error
- Funciona correctamente el botón de envío
- Se actualizan los datos en la tabla

Criterios de aceptación verificados: AC2, AC3 y AC4

HU-6: Manejo de datos vacíos o incompletos de gráficas e información de empleados

CP-016: Validar dashboard con datos vacíos

Tipo: Funcional / UI
Prioridad: Media

Precondiciones

- El usuario tiene las credenciales de una cuenta de superadministrador o administrador

- El usuario puede iniciar sesión correctamente
- No existe previamente datos o cuentas registradas de veterinarios, dueños, mascotas y/o citas

Pasos de ejecución:

1. Ingresar sesión con éxito en el sistema con una cuenta de superadministrador
2. La aplicación redirige al dashboard del negocio automáticamente
3. Verificar que la información del dashboard sea coherente
4. Revisar cards, gráficas y tablas con mensajes correctos

Resultado esperado:

- El Dashboard no muestra mensajes de error técnicos
- Dashboard carga mensajes correctamente y que son amigables con el usuario
- Cuando se ingresan datos se actualiza el dashboard automáticamente con dichos datos

Criterios de aceptación verificados: AC3

CP-017: Validar mensajes de datos vacíos en todas las pestañas de administrador

Tipo: Funcional / UI
Prioridad: Media

Precondiciones

- El usuario tiene las credenciales de una cuenta de superadministrador o administrador
- El usuario puede iniciar sesión correctamente
- No existe previamente datos o cuentas registradas de veterinarios, dueños, mascotas y/o citas

Pasos de ejecución:

11. Ingresar sesión con éxito en el sistema con una cuenta de superadministrador
12. La aplicación redirige al dashboard del negocio automáticamente
13. Navegar a las secciones “Citas”, “Veterinarios”, “Dueños” o “Mascotas” que se encuentran en el menú lateral
14. Observar y verificar mensaje en cada vista

Resultado esperado:

- Se muestra en cada pestaña que un mensaje haciendo saber que no existen datos ingresados en ningún caso
- Ningún mensaje tiene nomenclaturas técnicas y poco amigables para el usuario
- Cada vista carga de una manera correcta y se cargan todas sus partes

Criterios de aceptación verificados: AC1, AC2 y AC4

HU 7 – Corrección de creación de expedientes médicos

CP-018: Validar existencia del botón "Guardar/Crear Expediente"

Tipo: Funcional / UI
Prioridad: Alta

Precondiciones

- Modal de registro de mascota abierto.

Pasos

1. Abrir el modal de registro de mascota.
2. Buscar el botón "Guardar" o "Crear expediente".

Resultado esperado

- El botón está visible y claramente etiquetado.

Criterios de aceptación: AC1

CP-019: Validar envío correcto de datos al endpoint

Tipo: Integración / Funcional
Prioridad: Alta

Precondiciones

- Conexión al backend activa.
- Campos obligatorios ingresados.

Pasos

1. Llenar el formulario con datos válidos.
2. Hacer clic en “Guardar”.
3. Revisar en las DevTools → pestaña Network.

Resultado esperado

- La request se envía al endpoint correcto:
- Código HTTP 200.

Criterios de aceptación: AC2

CP-020: cierre automático del modal y adición del nuevo expediente a la lista de expedientes

Tipo: Funcional / UX
Prioridad: Alta

Precondiciones

- Datos enviados correctamente.

Pasos

1. Crear expediente con datos válidos.
2. Verificar si el modal se cierra.
3. Verificar el nuevo expediente en la lista de expedientes

Resultado esperado

- El modal se cierra automáticamente.
- El nuevo expediente es claramente visible en la lista de expedientes

Criterios de aceptación: AC3

CP-021: Validar mensaje de error si faltan campos obligatorios

Tipo: Funcional / Validación
Prioridad: Alta

Precondiciones

- Modal abierto.

Pasos

1. Dejar campos obligatorios vacíos.
2. Hacer clic en “Guardar”.

Resultado esperado

- Se muestra un mensaje claro:
  “Faltan campos obligatorios”.
- No se realiza request.

Criterios de aceptación: AC4

HU 8 – Corrección en listado de expedientes médicos

CP-022: Cargar correctamente la lista de expedientes

Tipo: Integración / Funcional
Prioridad: Alta

Precondiciones

- Veterinario logueado.
- Endpoint activo.

Pasos

1. Ir a la pestaña **Expedientes**.
2. Observar carga inicial.

Resultado esperado

- Se muestra la tabla completa con:
  - ID mascota
  - Nombre
  - Fecha creación

Criterios de aceptación: AC1, AC2

CP-0024: Validar mensaje cuando no existan expedientes

Tipo: Funcional / UX
Prioridad: Alta

Precondiciones

- La base de datos no contiene registros.

Pasos

1. Ir a la pestaña **Expedientes**.

Resultado esperado

- Mensaje: “No se encontraron expedientes registrados”.

Criterios de aceptación: AC4

HU 9 – Corrección del funcionamiento de buscadores y endpoint de citas del día

CP-025: Cargar citas del día sin errores

Tipo: Funcional
Prioridad: Alta

Precondiciones

- Veterinario logueado.
- Endpoint de citas activo.

Pasos

1. Ir a la pestaña **Citas**.
2. Verificar si aparece error.

Resultado esperado

- Aparece un sweet alert que dice “Sin citas”
- No aparece “Cannot read properties of null”.
- No aparece “Fecha inválida, use formato DD-MM-YYYY”.
- Se muestran las citas del día (si existen).

Criterios de aceptación: AC1

CP-026: Filtrar citas por nombre del dueño

Tipo: Funcional
Prioridad: Alta

Precondiciones

- Lista cargada.

Pasos

1. Escribir el nombre del dueño en el buscador.
2. Revisar tabla.

Resultado esperado

- Solo se muestran coincidencias con el nombre ingresado.

Criterios de aceptación: AC2

HU 12 – Limpieza del archivo main.go

CP-027: Prueba Funcional – Inicialización centralizada

Tipo: Funcional
Prioridad: Alta

Precondiciones

- El proyecto compila sin errores.
- Las funciones InitApp(), InitDB(), BuildDeps() y InitRouter() existen y están exportadas.

Pasos

3. Ejecutar la aplicación con go run main.go.
4. Verificar que se llama primero a InitApp().
5. Verificar que se inicializa la base de datos mediante InitDB().
6. Validar que las dependencias se construyen mediante BuildDeps(db).
7. Confirmar que el router se inicializa con InitRouter(controllers)..

Resultado esperado

- El programa ejecuta la secuencia exacta: App → DB → Dependencias → Router
- No se observan llamadas de creación de controladores dentro del main..
- El sistema inicia sin panics ni errores de dependencias.

Criterios de aceptación: AC1, AC2 y AC3.

HU 13 – Refactorización del enrutamiento

CP-028: Prueba Funcional – Registro centralizado de rutas

Tipo: Funcional
Prioridad: Alta

Precondiciones

- Existe una función única InitRouter().

Pasos

1. Ejecutar la aplicación.
2. Revisar logs del servidor o usar herramienta como Postman para inspeccionar endpoints.
3. Confirmar que todas las rutas provienen exclusivamente de las llamadas:
   - UserController.RegisterRoutes()
   - AdminController.RegisterPublicRoutes()
   - AdminController.RegisterProtectedRoutes()
   - AppointmentController.RegisterRoutes()
   - PetController.RegisterRoutes()
   - AdminTypeController.RegisterRoutes()
   - UserRoleController.RegisterRoutes()
   - SpeciesController.RegisterRoutes()

Resultado esperado

- Todas las rutas se cargan desde InitRouter().
- No existen rutas registradas desde main.go...
- No se observan rutas huérfanas o no registradas.

Criterios de aceptación: AC1, AC2

HU 13 – Refactorización del enrutamiento

CP-029: Prueba Funcional – Registro centralizado de rutas

Tipo: Funcional
Prioridad: Alta

Precondiciones

- Existe una función única InitRouter().

Pasos

4. Ejecutar la aplicación.
5. Revisar logs del servidor o usar herramienta como Postman para inspeccionar endpoints.
6. Confirmar que todas las rutas provienen exclusivamente de las llamadas:
    - UserController.RegisterRoutes()
    - AdminController.RegisterPublicRoutes()
    - AdminController.RegisterProtectedRoutes()
    - AppointmentController.RegisterRoutes()
    - PetController.RegisterRoutes()
    - AdminTypeController.RegisterRoutes()
    - UserRoleController.RegisterRoutes()
    - SpeciesController.RegisterRoutes()

Resultado esperado

- Todas las rutas se cargan desde InitRouter().
- No existen rutas registradas desde main.go...
- No se observan rutas huérfanas o no registradas.

Criterios de aceptación: AC1, AC2

HU 14 – Refactorización del enrutamiento

CP-030: Prueba Funcional – Solicitud de restablecimiento exitosa

Tipo: Funcional
Prioridad: Alta

Precondiciones

- Usuario registrado con correo activo.
- Servidor SMTP funcionando.

Pasos

1. Enviar POST a /forgot-password con correo registrado.
2. Verificar respuesta del servidor.
3. Revisar bandeja de correo del usuario

Resultado esperado

- Se genera un token temporal de restablecimiento.
- El usuario recibe un email con un enlace válido.
- La API responde con estado 200.

Criterios de aceptación: AC3

HU 15 – Separar lógica de negocio del controlador de User

CP-31: Prueba Funcional – Controlador sin lógica de negocio

Tipo: Funcional
Prioridad: Alta

Precondiciones

- Existe un UserService.
- El controlador debe delegar toda la lógica

Pasos

1. Abrir el archivo UserController.
2. Revisar cada handler (CreateUser, Login, UpdateUser, etc.).
3. Validar que no existan consultas a BD, lógica de cálculo o validaciones profundas.

Resultado esperado

- Cada handler solo:

    - valida input
    - delega al servicio
    - retorna la respuesta
- El servicio contiene toda la lógica de negocio.

Criterios de aceptación: AC1, AC2

HU 16 – Creación de EmailService

CP-032: Prueba Funcional – Controlador sin lógica de negocio

Tipo: Funcional
Prioridad: Alta

Precondiciones

- SMTPClient funcional.
- EmailService implementado.

Pasos

1. Crear una instancia de EmailService.
2. Ejecutar SendMail() con parámetros válidos.
3. Revisar bandeja del destinatario.

Resultado esperado

- El correo se envía correctamente
- No se filtran detalles del SMTP al exterio.
- No se lanza panic ni error inesperado

Criterios de aceptación: AC1, AC2