

18/11/2025
Grupo: 303



INFORME EJECUTIVO

FINAL DE TESTING

Testers:
Natasha Carrion
Branhyan Bernaschina
Gwyneth Kolesnick
Matias Peluffo
Carolina Rodríguez
Kevin Rodríguez

Resumen Ejecutivo



Este informe presenta los resultados del proyecto de testing del sistema de gestión veterinaria "Guau Guau", una aplicación web diseñada para la administración de clientes, mascotas y registro de vacunación. El sistema requería validación completa de funcionalidad y calidad antes de su implementación.

OBJETIVO:

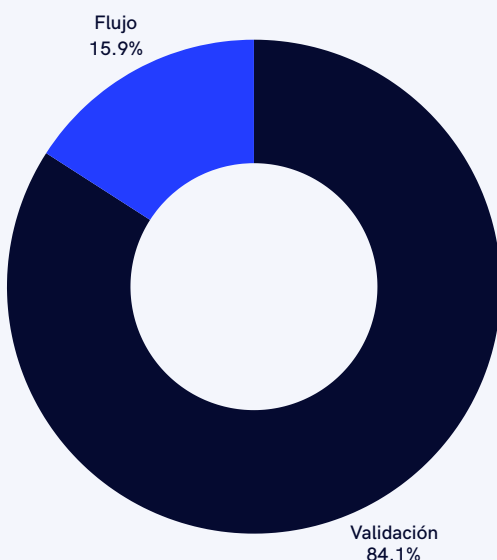
Evaluar el estado actual del sistema mediante testing exhaustivo, identificar defectos críticos, validar cumplimiento de requerimientos funcionales y no funcionales, y proporcionar recomendaciones de mejora priorizadas para elevar la calidad del producto.

ALCANCE:

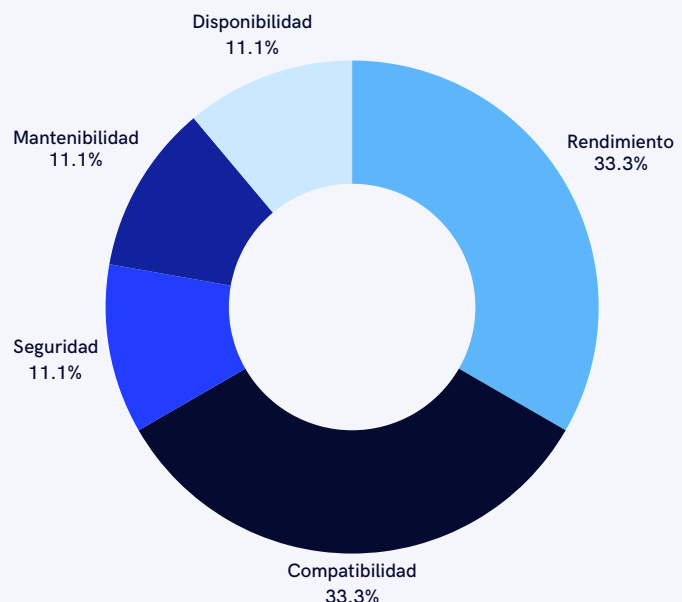
Se diseñaron y ejecutaron 100+ casos de prueba distribuidos en dos ciclos completos sobre las versiones 1.0 y 2.0 del sistema.

Distribución de casos:

Pruebas funcionales



Pruebas no funcionales



ESTRUCTURA DEL DOCUMENTO:



Este informe detalla la metodología aplicada en cuatro fases, los resultados obtenidos en cada ciclo de testing, el análisis de defectos y mejoras identificadas, las observaciones del proceso de trabajo, y las recomendaciones priorizadas para el desarrollo futuro del sistema.

Resultados del testing:



Mejora de Calidad:



El sistema evolucionó de 43% a 74% de casos exitosos entre versiones, representando un incremento de 31 puntos porcentuales en calidad medida. Las pruebas fallidas disminuyeron de 26 a 18, mientras que los casos bloqueados se redujeron de 18 a 8. Este progreso demuestra la efectividad del ciclo de testing iterativo, donde se ejecutaron 58

pruebas de confirmación y 42 de regresión para validar correcciones y asegurar que no se introdujeran nuevos defectos.

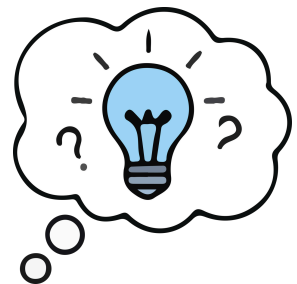
Defectos e Incidencias:

Se identificaron y documentaron 27 defectos con trazabilidad completa: 46% severidad alta, 23% media y 31% baja. Adicionalmente, se registraron 2 no conformidades y 3 observaciones. Esta clasificación detallada permitió priorizar correcciones críticas y optimizar los recursos del equipo de desarrollo, asegurando que los problemas de mayor impacto fueran resueltos en etapas tempranas.



Ideas de Mejora:

Se propusieron 47 mejoras priorizadas: 17% prioridad alta, 51% media y 32% baja. De las 7 mejoras identificadas en la versión 1.0, se implementaron 3 completamente y 2 parcialmente. Este análisis proactivo identificó oportunidades de optimización que mejoran la experiencia del usuario y la eficiencia operativa del sistema.



Documentación generada:



Se generaron más de 300 páginas de documentación técnica incluyendo casos de prueba, informes de ejecución, análisis de defectos y conclusiones por fase.

Metodología

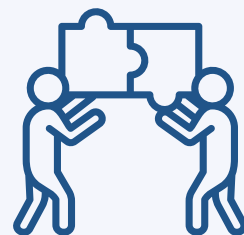
Enfoque estructurado en 4 fases



El proyecto se ejecutó mediante una metodología de testing estructurada en cuatro fases secuenciales, este enfoque permitió mantener el control del proceso, asegurar la calidad de entregables y facilitar la coordinación de un equipo de 6 testers con horarios diversos.

Coordinación del Equipo:

Se implementó un sistema de trabajo asíncrono con responsabilidades individuales definidas, donde cada miembro ejecutaba sus tareas de forma independiente y luego las integraba al equipo en reuniones de sincronización. Cada etapa duró 2 semanas con deadlines definidos y tuvimos 2 reuniones por semana.



Distribución de pruebas

El testing funcional se distribuyó según la etapa por tipo de prueba o por módulo, el no funcional en un principio se centralizó para más eficiencia al ser diseñados y ejecutados por la misma persona, luego se repartieron entre todo el equipo en la siguiente fase.

Técnicas de Diseño de Casos

Se aplicaron técnicas formales de testing para maximizar cobertura con eficiencia y se utilizó un sistema de trazabilidad con códigos estandarizados que permite dar seguimiento y clasificar los diferentes tipos de casos de prueba.



Herramientas y proceso de trabajo

Implementación de Jira

El equipo decidió utilizar Jira como herramienta central de gestión, esto debido a que es una herramienta vital en el ámbito laboral actual, y tiene muchas facilidades para manejar un proyecto de tamaño considerable, así que la configuramos con funciones tales como:



Flujo de trabajo personalizado

Estados definidos para casos de prueba (Realizado exitosamente, fallido o bloqueado), incidencias (Solucionada, Solucionada parcialmente, No solucionada) e Ideas de mejora (Implementada, Implementada parcialmente y no implementada)

Implementación de etiquetas para mejor seguimiento

Clasificación mediante etiquetas por módulo (Clientes, Mascotas, Vacunas, Dashboard, Login, General), categorías de prueba (Funcional - Categoría, No Funcional - Categoría,) y técnica de diseño (Participación de Equivalencias, Casos de Uso, Requisitos funcionales y no funcionales)

Trazabilidad completa

Vinculación directa entre casos de prueba, incidencias detectadas y requerimientos del sistema. También establecieron incidencias por versión para hacer seguimiento de incidencias solucionadas por versión y nuevas incidencias.

Asignación y seguimiento:

Distribución de casos e incidencias por miembro del equipo con seguimiento en tiempo real del avance. Esto nos permitió tener una mejor gestión de las tareas y los casos de prueba que hacía cada integrante del equipo.

Herramientas de apoyo:



Trello:

Utilizamos Trello en 2 tableros. El primero para gestión de coordinación con diagrama de Gantt mediante Planyway, distribuyendo las tareas de planificación, testing, documentación y evaluación de las 4 fases por miembro del equipo. El segundo tablero era exclusivo de testing para hacer reportes preliminares antes de ser publicados en Jira, permitiendo revisión rápida antes de documentación formal.

Comunicación: Webex (reuniones en horario de clases), Discord (reuniones fuera de horario), WhatsApp (coordinación rápida y resolución de bloqueos).

Documentación: Canvas (entregables formales del curso), Google Docs (escritura colaborativa en tiempo real), XMind (mapas mentales para visualización de ideas y estructuración de documentación).

Conclusiones

Generales

Calidad del Software

Durante el periodo de pruebas el programa ha sido estable, el sistema es robusto y se mantiene funcionando ante errores. Esta estabilidad se mantuvo consistente incluso con la identificación de 27 defectos, sin presentar caídas críticas o pérdida de funcionalidad completa del sistema.

La evolución entre versiones evidenció mejora significativa medible: la V1.0 presentó 43% de casos exitosos, mientras que la V2.0 alcanzó 74%, representando un incremento del 31% tras el proceso de corrección y retesting.

El principal punto de atención identificado es la dependencia de localStorage para persistencia de datos. Si bien funciona correctamente en ambiente local, se recomienda implementar base de datos backend para garantizar resistencia y confiabilidad de la información en ambiente productivo.

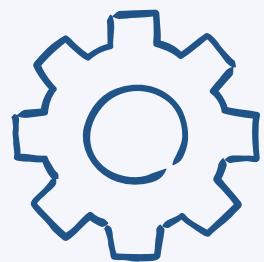


Funcionalidad

Las pruebas realizadas demostraron que el programa cumple con los requerimientos establecidos. El sistema implementa correctamente las funcionalidades principales: autenticación de usuarios, gestión de clientes, registro de mascotas, y control de vacunación.

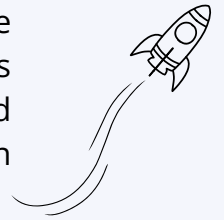
El programa funciona actualmente de manera local utilizando localStorage, por lo tanto los datos se pueden perder fácilmente ante limpieza de caché o cambio de navegador. Esta arquitectura es adecuada para desarrollo y testing, pero limita su aplicación en ambiente productivo.

Se recomienda implementar base de datos backend para garantizar persistencia de datos, habilitar acceso multi-usuario, y permitir escalabilidad del sistema a futuro.



Rendimiento

El rendimiento general es excelente, con tiempos de carga promedio de 0.065 segundos que cumplen ampliamente con el límite de 2 segundos especificado. Las mediciones se realizaron en condiciones de red simuladas (3G, 4G lento, 4G rápido) para evaluar comportamiento en diferentes escenarios.



Es importante considerar que el programa está desplegado localmente, por lo que las mediciones reflejan performance en ambiente controlado. En ambiente productivo con backend real, los tiempos dependerán de latencia de red y carga del servidor.

Accesibilidad

El sistema cumple con aspectos de accesibilidad WCAG 2.1 Level AA evaluados: navegación completa por teclado con orden lógico de tabulación y elementos interactivos claramente identificables.

Registro de Nuevo Cliente

Nombre: 1

Apellido: 2

Correo Electrónico: 3

Dirección: 4

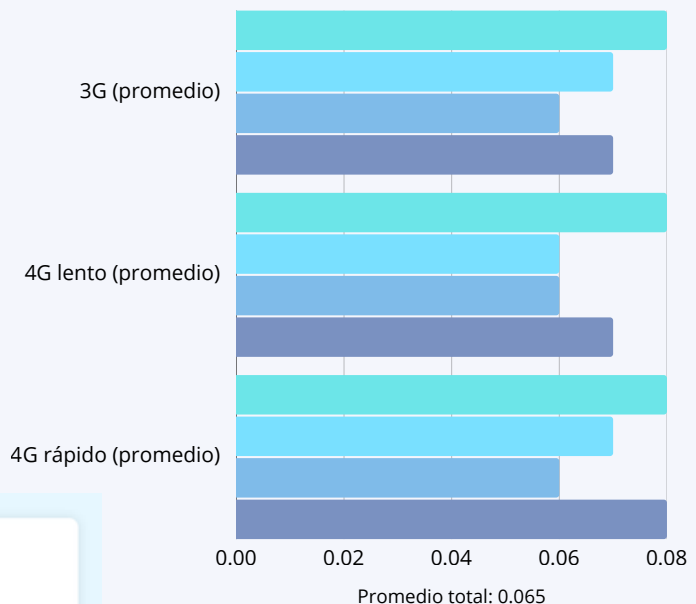
Ingresar Mascota 5

Registar Mascota 6

Velocidad de carga (LCP)

Promedio en segundos

- Registro nuevo cliente
- Registro nueva mascota
- Login
- Registro nuevo vacuna



Las pruebas de contraste de color demuestran cumplimiento destacado: texto normal con ratio 7.45:1 y títulos con ratio 12.63:1, superando ampliamente los requisitos mínimos de 4.5:1 y 3:1 respectivamente. Los mensajes de error mantienen legibilidad adecuada sobre sus fondos.

MANEJO DE ERRORES

El programa cuenta con advertencias ante campos vacíos y validación de formato en correo electrónico. Sin embargo, no verifica que los límites de caracteres sean sobrepasados ni realiza verificaciones correctas en fechas (permitiendo, por ejemplo, fechas futuras en campo de nacimiento).

Esto no supone un problema al estar desplegado localmente, pero a la hora de conectar con una base de datos real, al no realizar comprobaciones de límites, va a generar errores si los datos introducidos superan el límite establecido por las columnas de la base de datos. Se recomienda implementar estas validaciones antes del despliegue productivo.

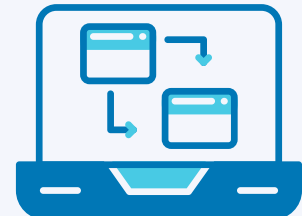


USABILIDAD

El sistema presenta una interfaz intuitiva con diseño limpio y consistente en todos los formularios. Los campos están correctamente ordenados siguiendo un flujo lógico de arriba hacia abajo, y los botones cuentan con texto descriptivo claro que facilita su comprensión. La navegación es sencilla, permitiendo moverse entre secciones sin confusión. El diseño responde correctamente a cambios de tamaño de ventana, manteniendo legibilidad en diferentes resoluciones. Los elementos visuales como etiquetas y mensajes son claros, reduciendo la necesidad de documentación adicional para usar el sistema.

COMPATIBILIDAD

El sistema fue verificado exitosamente en los principales navegadores web: Google Chrome, Microsoft Edge y Opera GX. Todas las funcionalidades (login, registro de clientes, mascotas, vacunas y dashboard) operan correctamente sin errores de JavaScript en consola.



Los estilos se aplican de manera consistente, la interfaz se visualiza correctamente, y los formularios validan datos apropiadamente en los tres navegadores evaluados. La compatibilidad cross-browser garantiza accesibilidad amplia independientemente del navegador preferido del usuario.

SEGURIDAD



La V2.0 implementó sistema de autenticación básico que valida credenciales correctamente y rechaza usuarios inválidos, protegiendo el acceso a formularios de registro. El sistema detecta campos vacíos y credenciales incorrectas, mostrando mensajes de error apropiados.

Las pruebas se realizaron con credenciales predefinidas para testing (usuario: estudiante, contraseña: 12345). Dado el alcance de prototipo

frontend, las contraseñas se almacenan sin cifrado. Para despliegue productivo, se recomienda implementar registro de usuarios, ocultación de contraseñas y gestión de sesiones segura.

Resultados Clave



100
CASOS DE PRUEBA
EJECUTADOS



27
INCIDENCIAS
ENCONTRADAS



47
IDEAS DE MEJORA



0.065S
TIEMPO DE CARGA
PROMEDIO
(ES EN LOCAL)



7.06:1
CONTRASTE TEXTO



3
NAVEGADORES
TESTEADOS



100%
COBERTURA DE
REQUERIMIENTOS
APLICABLES



2 VERSIONES
TESTEADAS
(V1.0 Y V2.0)



+72% CASOS
EXITOSOS
V1.0 → V2.0
(42 → 74)



8
SEMANAS DE
EJECUCIÓN



+300 PÁGINAS DE
DOCUMENTACIÓN



8 TIPOS DE PRUEBAS
APLICADAS



TRAZABILIDAD
COMPLETA:
REQUERIMIENTO →
CASO → INCIDENCIA
→ CORRECCIÓN →
VALIDACIÓN



10
INCIDENCIAS
RESUELTAS
DE V1.0



5 DE 7 MEJORAS
IMPLEMENTADAS
(TOTAL O
PARCIALMENTE)
ENTRE VERSIONES

Cobertura de Pruebas

Se alcanzó 85.7% de cobertura total de requerimientos (12/14 evaluables), con validación completa de los 6 módulos funcionales del sistema. La limitación en cobertura se debe a restricciones técnicas del ambiente de testing, no a falta de diseño de casos.



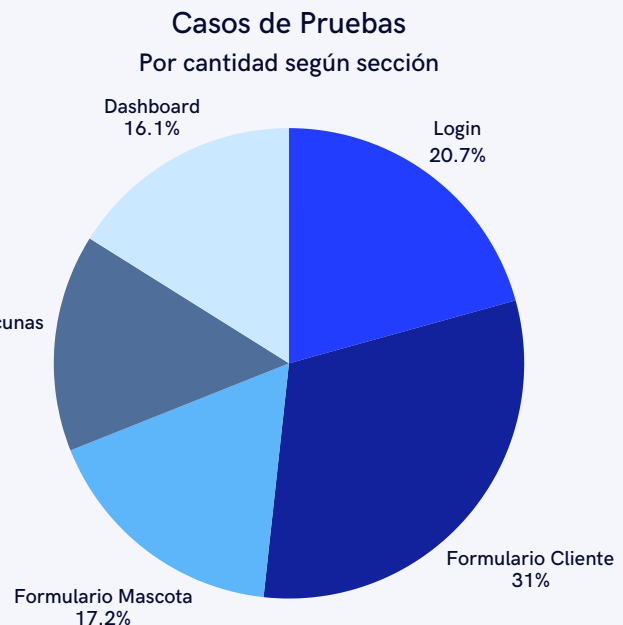
COBERTURA POR REQUERIMIENTOS:

- Requerimientos Funcionales: 100% cubiertos (6/6)
- Requerimientos No Funcionales: 75% cubiertos (6/8)

Los dos requerimientos no funcionales bloqueados fueron RNF04 (Disponibilidad - requiere infraestructura de servidor) y RNF08 (Escalabilidad - no aplicable en ambiente local). Ambas limitaciones quedaron documentadas para evaluación en ambiente productivo.

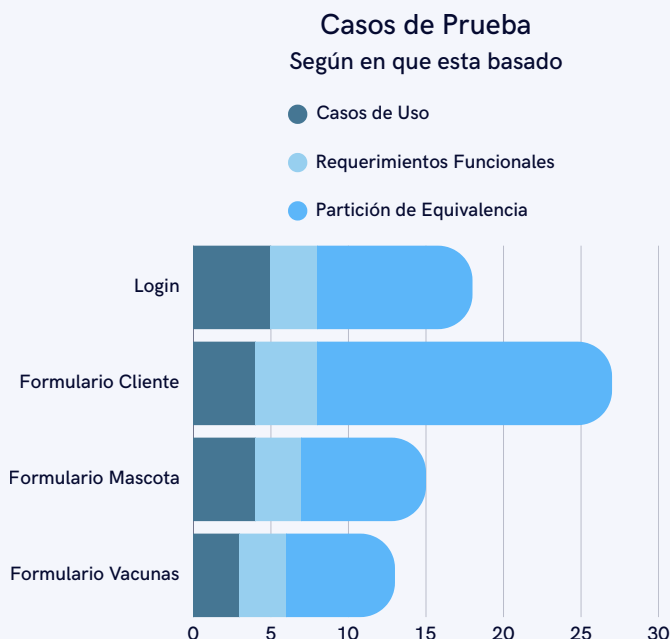
COBERTURA POR MÓDULOS:

Se validaron el 100% de módulos funcionales disponibles: Registro de Clientes (33 casos), Login (16 casos), Registro de Mascotas (16 casos), Dashboard (14 casos), Registro de Vacunas (13 casos) y funcionalidades generales (13 casos).



ESTRATEGIA DE TESTING:

La cobertura se logró combinando múltiples técnicas: 50% partición de equivalencias, 18.2% casos de uso, 14.8% requerimientos funcionales, 14.8% requerimientos no funcionales y 2.3% pruebas específicas de dashboard. Esta distribución aseguró validación exhaustiva de funcionalidad, usabilidad, rendimiento, compatibilidad y accesibilidad dentro de las posibilidades del ambiente de testing.



Ideas de mejora

A lo largo del proceso de pruebas se identificaron aspectos que, a pesar de que no impidieron el funcionamiento del sistema, sí limitan su uso en un entorno real. Estas ideas permiten proponer mejoras que apuntan a reforzar la seguridad, el almacenamiento de datos y la experiencia de usuario. Las más destacadas fueron:

Gestión de datos y persistencia

Reemplazar el uso de localStorage por una base de datos real con respaldo automático. Esto permitirá guardar la información de forma segura, compartirla entre distintos dispositivos y protegerla ante fallos del sistema.



Sistema de notificaciones y agenda

Incorporar alertas automáticas para vencimiento de vacunas y un sistema de turnos vinculado a cada mascota, con posibilidad de sincronizar citas en calendarios externos. Esto mejorará la organización y reducirá olvidos.

Mejoras en Dashboard y visualización

Mejorar el panel principal para que muestre nombres en lugar de IDs, incluir contadores de mascotas por cliente y un buscador más completo. Esto hará la navegación más sencilla y la información más comprensible.



Función de editar y eliminar

Habilitar la edición y eliminación de registros de clientes, mascotas y vacunas. Actualmente solo se pueden crear registros, lo que dificulta corregir errores o actualizar datos.



Historial médico completo

Ampliar la ficha veterinaria con observaciones médicas, tratamientos, alergias y opción de adjuntar archivos (radiografías, análisis). También sería útil generar reportes exportables en PDF o Excel..}



Experiencia de usuario e interfaz

Optimizar la interfaz con navegación más clara, opción de cerrar sesión desde cualquier pantalla, modo oscuro y diseño adaptable a celulares. También añadir autocompletado en campos frecuentes para mayor rapidez.

Retrospectiva

FORTALEZAS DEL PROCESO:

Planificación y asignación estratégica:

Definición temprana de roles basada en fortalezas individuales maximizó eficiencia y ambiente de trabajo efectivo.

Trabajo asíncrono efectivo:

Responsabilidad individual + 2 reuniones semanales permitieron coordinar 6 personas con horarios diversos sin perder cohesión.

Adaptabilidad: Flexibilidad al migrar Trello→Jira, agregar casos no previstos, y ajustar distribución de pruebas sin comprometer calidad.

Documentación exhaustiva: 300+ páginas con visualizaciones crearon base de conocimiento reutilizable y material profesional para portfolios.

Buena disposición: Actitud colaborativa constante, disposición a aprender herramientas nuevas y ajustarse a cambios.

DESAFÍOS ENFRENTADOS Y RESOLUCIÓN:

Curva de aprendizaje de Jira: Sin experiencia previa en la herramienta. → Capacitación incremental primera semana: configuración y explicación en paralelo hasta adopción uniforme.

Diseño inicial de RNF: Requerimientos no funcionales poco estructurados en ESRE. → Asignación a miembro con criterio técnico + diseño iterativo durante ejecución = 13 casos RNF, 75% cobertura.

Coordinación asíncrona: 6 personas con disponibilidades diferentes generaba riesgo de bloqueos. → Responsabilidad individual clara + reuniones semanales + WhatsApp proactivo + visibilidad en Jira.

Visiones diferentes sobre documento: Densidad vs. legibilidad tradicional. → División clara: contenido/estructura vs. diseño visual. Comunicación previa evitó retrabajos.

APRENDIZAJES CLAVE:

Jira vale la pena: Ayuda mucho a generar estadísticas desde etiquetas/campos, y a gestionar casos. Diferencia entre gestión informal y profesional.

Planificación flexible: Roles y deadlines claros dan dirección, pero capacidad de ajustar sin perder norte es crítica.

Plantillas estandarizadas: Estructuras similares para informes y casos ahorran tiempo masivamente.

Responsabilidad individual funciona: Cuando cada miembro se compromete y cumple, todos dan lo mejor de sí.

RNF requieren especialización: Más difíciles que funcionales - requieren traducir conceptos técnicos a casos validables.

RECOMENDACIONES FUTURAS:

Pruebas de seguridad profundas: Incluir penetración, manejo de sesiones, protección contra ataques, cifrado desde planificación.

Mantener planificación temprana: Asignación por fortalezas en Fase 1 con deadlines realistas.

Automatización de regresión:

Selenium/Cypress para liberar tiempo de ejecución repetitiva.

Documentación visual rica: Mantener gráficos y visualizaciones para comunicación profesional.

Ambiente cercano a producción:

Backend/servidor real para validación completa de RNF.

Agradecimientos

Queremos expresar nuestro agradecimiento a todo el equipo de Jóvenes a Programar, especialmente a Carlos y Ylva por estar presentes en todas nuestras dudas. El esfuerzo, cooperación y planificación entre los integrantes del equipo fue fundamental para la finalización del proyecto.

Conclusión final

El proyecto Veterinaria Guau Guau ha finalizado exitosamente, cumpliendo con los objetivos de validación exhaustiva del sistema y generación de documentación profesional completa.

Durante las 8 semanas de ejecución se diseñaron y ejecutaron 100+ casos de prueba en dos ciclos completos, logrando 85.7% de cobertura de requerimientos evaluables. La evolución del sistema fue medible y significativa: de 43% a 74% de casos exitosos entre versiones, demostrando la efectividad del testing iterativo.

Se identificaron 27 defectos con trazabilidad completa, de los cuales 10 fueron resueltos y validados en la segunda versión. Adicionalmente, se propusieron 47 mejoras priorizadas que aportan valor para el desarrollo futuro del sistema.

El software cumple con los requisitos funcionales especificados y presenta calidad aceptable para ambiente de desarrollo. Las funcionalidades core (autenticación, gestión de clientes, mascotas y vacunación) operan correctamente y el sistema demuestra estabilidad ante errores de usuario.

Sin embargo, la arquitectura actual basada en localStorage limita su aplicación en producción. Para despliegue real se requiere implementar base de datos backend, gestión de usuarios concurrentes, pruebas de seguridad profundas, y las mejoras de alta prioridad identificadas.

El proceso de testing permitió validar la calidad del sistema dentro de las limitaciones del ambiente disponible, documentar hallazgos con trazabilidad completa, y proporcionar información clara para las decisiones de desarrollo futuro.