



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Análisis y Cierre del Proceso de Pruebas de Software

Corporación Universitaria Remington.
Facultad de ingeniería de sistemas
Seminario de pruebas

Michael Steven Mina Muñoz, Nubia Vergara Meza, Jose Daniel Jimenez Diaz
Fabio Leon Grisales Morales
Seminario
2025

Agradecimientos

Este trabajo está dedicado al profesor del seminario, por su orientación, acompañamiento y disposición para compartir sus conocimientos a lo largo del proceso formativo. Su guía fue fundamental para el desarrollo de este proyecto y para fortalecer el aprendizaje en el área de pruebas de software.

De igual manera, este trabajo está dedicado a los compañeros de equipo, por el compromiso, la colaboración y el esfuerzo demostrado durante cada una de las etapas del proyecto. A pesar de las diferencias de horarios y puntos de vista, el trabajo en equipo permitió alcanzar los objetivos propuestos y enriquecer el proceso de aprendizaje.

Tabla de Contenidos

Resumen.....	4
Palabras Clave.....	5
Marco conceptual y contextual	6
Desarrollo e implementación del aprendizaje.....	7
Figuras y tablas	8
Figuras y tablas	9
Figuras y tablas	10
Figuras y tablas	11
Figuras y tablas	12
Figuras y tablas	13
Conclusiones	14

Resumen

El presente trabajo corresponde al informe de cierre del proceso de pruebas de software realizado como parte de la opción de grado Seminario–Diplomado. El objetivo principal del proyecto fue validar el correcto funcionamiento de un sistema de información mediante la ejecución de diferentes tipos de pruebas, tomando como base las historias de usuario (tabla 1) definidas durante la etapa de planeación.

En el plan de pruebas se estableció inicialmente la validación de catorce historias de usuario. Sin embargo, durante el desarrollo del proyecto se logró validar un total de cuatro historias, priorizando aquellas que tenían mayor impacto en el funcionamiento del sistema. Esta diferencia entre el alcance estimado y el alcance real se presentó principalmente por limitaciones de tiempo, ajustes realizados durante la ejecución, la disponibilidad de los integrantes del equipo y la falta de experiencia previa en la realización de pruebas de software.

Durante la ejecución del proyecto también se evidenció una diferencia importante entre los tiempos estimados y los tiempos reales. Aunque inicialmente se habían planeado 266 horas para el desarrollo de las actividades, el tiempo real invertido fue de aproximadamente 524 horas. Esta situación se debió a la subestimación de algunas tareas, a los reprocesos generados por los defectos encontrados, al proceso de aprendizaje en el uso de las herramientas de pruebas y al tiempo requerido para la elaboración de la documentación final.

A pesar de estas dificultades, el uso de herramientas como el plan de pruebas, el bugtracker (tabla 2) y la matriz de riesgos (tabla 3) permitió llevar un control más organizado del proceso. Finalmente, este proyecto permitió identificar lecciones aprendidas y oportunidades de mejora relacionadas con la planificación, la gestión del tiempo y el trabajo en equipo, las cuales serán de gran utilidad para futuros proyectos de pruebas de software.

Palabras clave

- **Pruebas de software**
Es una disciplina esencial dentro de la trayectoria evolutiva del software, cuya finalidad primordial radica en inspeccionar y confirmar que un producto o herramienta informática satisfaga las exigencias estipuladas y opere de forma predecible.
- **Historias de usuario**
Es una explicación formal, dentro de un contexto ágil, de una necesidad. Es una descripción corta y simple de quién es el usuario, qué quiere y por qué.
- **Gestión de riesgos**
Es el proceso sistemático de identificar, analizar, evaluar, tratar, monitorear y comunicar los riesgos asociados con cualquier proyecto, producto o proceso.
- **Bugtracker**
Es una herramienta de software esencial utilizada en la gestión de proyectos y el control de calidad y su propósito principal es registrar, clasificar, asignar, seguir y gestionar el ciclo de vida de cualquier defecto, error o bug encontrado en una aplicación o sistema de software.
- **Calidad del software.**
Es el grado en que un producto de software cumple con las necesidades y expectativas explícitas e implícitas del cliente y de los stakeholders

Marco conceptual y contextual

finalidad principal contribuir a la calidad de los sistemas de información mediante la verificación y validación de los requisitos definidos. Las pruebas de software permiten identificar fallos, errores o inconsistencias antes de que el sistema sea entregado al usuario final, reduciendo riesgos y mejorando la confiabilidad del producto.

Durante el desarrollo del seminario, se abordaron conceptos fundamentales relacionados con el aseguramiento de la calidad del software, tales como pruebas funcionales, pruebas de integración, pruebas de regresión, historias de usuario, casos de prueba, defectos y gestión de riesgos. Estos conceptos sirvieron como base teórica para la ejecución del proyecto y fueron aplicados de manera práctica a lo largo del proceso.

Las pruebas funcionales se enfocan en verificar que el sistema cumpla con los requisitos establecidos y que las funcionalidades respondan correctamente a las necesidades del usuario. Por otro lado, las pruebas de integración permiten validar la correcta interacción entre los diferentes módulos o componentes del sistema, asegurando que trabajen de forma conjunta sin generar fallos. Las pruebas de regresión, por su parte, se realizan para confirmar que los cambios o correcciones implementadas no afecten funcionalidades previamente validadas.

Este proyecto se desarrolló como parte del seminario–diplomado, permitiendo aplicar los conocimientos adquiridos en un escenario práctico. El trabajo se centró en la validación de un sistema de información, priorizando las historias de usuario de mayor impacto, lo cual permitió comprender la importancia de una adecuada planificación, documentación y ejecución del proceso de pruebas dentro del ciclo de vida del software.

Desarrollo e implementación del aprendizaje

Durante la ejecución del proyecto se aplicaron de manera progresiva los conocimientos adquiridos a lo largo del seminario–diplomado, siguiendo las diferentes etapas del proceso de pruebas de software: planeación, diseño, ejecución y cierre. Cada una de estas fases permitió reforzar los conceptos teóricos mediante su aplicación práctica.

En la etapa de planeación se definió el alcance del proyecto, se identificaron las historias de usuario a validar, se analizaron los posibles riesgos y se realizó una estimación de los tiempos necesarios con ayuda de la calculadora de estimación (tabla 4) para el desarrollo de las actividades. Esta fase fue fundamental para establecer una guía inicial del trabajo, aunque durante la ejecución se presentaron ajustes debido a las condiciones reales del proyecto.

Posteriormente, se llevó a cabo el diseño de los casos de prueba, los cuales fueron elaborados con base en las historias de usuario definidas en el plan de pruebas. En esta fase se contemplaron diferentes escenarios, tanto principales como alternos, con el fin de cubrir los posibles comportamientos del sistema. Este proceso permitió fortalecer la capacidad de análisis y comprensión de los requisitos.

Durante la fase de ejecución se validaron cuatro historias de usuario, priorizando aquellas consideradas más críticas para el funcionamiento del sistema. Los defectos encontrados fueron registrados en el bugtracker, lo que facilitó su clasificación, seguimiento y análisis. Esta etapa permitió evidenciar la importancia de la documentación adecuada de los errores y de la trazabilidad dentro del proceso de pruebas (tabla 5).

Finalmente, en la etapa de cierre se consolidaron los resultados obtenidos, se analizaron las métricas de tiempo (tabla 4), se revisó el cumplimiento del alcance, se identificaron las lecciones aprendidas y se definieron oportunidades de mejora. Esta fase permitió reflexionar sobre el proceso realizado y reconocer la importancia de la planificación, la comunicación y el trabajo colaborativo para el éxito de los proyectos de pruebas de software.

Tabla 1. Resumen de historias de usuario.

ID	Título de la Historia	Enfoque Principal
UH-U-01	Ver Productos	Navegación y búsqueda de ítems en el catálogo.
UH-U-02	Ver Detalles de Producto	Visualización de información detallada (precio, descripción).
UH-U-03	Registro	Creación de cuenta y validaciones de datos.
UH-U-04	Inicio de Sesión	Autenticación y persistencia de sesión.
UH-U-05	Suscripción a un producto	Gestión del estado de suscripción (alta/baja de intereses).
UH-U-06	Carrito	Adición/Eliminación de ítems y cálculo de subtotales.
UH-U-07	Compra	Resumen de compra, aplicación de vales de descuento.
UH-U-08	Pagos	Proceso de pago, stock y seguridad de precio.
UH-U-09	Ver Factura	Trazabilidad documental post-pago y visualización de detalles.
UH-U-10	Mail	Envío de documentación de compra por correo electrónico.
UH-U-11	Ver Perfil	Acceso seguro y verificación de la información personal.
UH-U-12	Editar Perfil	Modificación de datos y contraseña del usuario.
UH-U-13	Histórico	Historial de compras y agregación de dinero gastado.
UH-U-14	Contraseña	Recuperación de acceso por olvido de contraseña.

Tabla 2. Archivo de BUGS (Bugtracker)

Impacto: Efecto hacia el cliente	Naturaleza	Reporte
Medio	Funcional	<p>Al ingresar a la sección Proveedores (Suppliers), el sistema muestra el mensaje "La página que busca no ha sido encontrada" en lugar de mostrar la lista de proveedores o un mensaje contextual coherente con el módulo.</p> <p>Pasos:</p> <ol style="list-style-type: none"> 1. Ingresar al sitio demo de PrestaShop. 2. Ir al menú superior / footer y seleccionar Proveedores. 3. Observar el resultado.a.
Bajo	Ortografía	<p>Al cambiar el idioma del sitio varias palabras siguen apareciendo en inglés, como botones o mensajes internos.</p> <p>Pasos:</p> <ol style="list-style-type: none"> 1. Abrir la página principal. 2. Cambiar idioma en el footer. 3. Navegar por catálogo, carrito o checkout.
Alto	Hardware y/o Software	<p>Tras llenar toda la información del cliente en el checkout, al intentar proceder al pago, la página no carga la ventana para ingresar datos y el botón "Colocar pedido" no se habilita. El usuario queda bloqueado sin poder finalizar la compra.</p> <p>Pasos:</p> <ol style="list-style-type: none"> 1. Agregar un producto al carrito. 2. Ir al checkout. 3. Completar datos personales y dirección. 4. Intentar proceder al pago / colocar pedido.
Bajo	Funcional	<p>Reporte:</p> <p>La imagen promocional de "20% de descuento en Clothes" ubicada al final de la página tiene un botón "See more" que, al presionarlo, siempre redirige al inicio en lugar de llevar a la categoría correspondiente.</p> <p>Pasos para reproducir:</p> <p>Desplazarse al final de la página principal.</p> <p>Ubicar la imagen de "20% OFF Clothes".</p> <p>Presionar el botón "See more".</p>

Alto	Performance	<p>Google PageSpeed informa un LCP (Largest Contentful Paint) de 11.4 segundos en el percentil 75, lo cual representa un rendimiento extremadamente lento.</p> <p>Aunque el servidor muestra un TTFB aceptable, el navegador demora en renderizar el contenido principal.</p> <p>El problema está asociado al manejo de recursos front-end (CSS, JS, imágenes y estructura DOM).</p> <p>Pasos para reproducir:</p> <p>Ingresar a https://demo.prestashop.com/#/en/front</p>
Medio	Hardware y/o Software	<p>En la sección "Sobre nosotros", la imagen principal no carga. Aparece un espacio vacío o un ícono de imagen rota.</p> <p>Pasos:</p> <p>Ir al pie de página.</p> <p>Seleccionar la opción "Sobre Nosotros".</p> <p>Observar que la imagen principal no se carga.</p> <p>Mostrar correctamente la imagen asociada a la sección.</p>
Bajo	Hardware y/o Software	<p>En la ficha de cualquier producto, no es posible dejar reseñas. El formulario no aparece o no se habilita la opción de escribir una reseña.</p> <p>Pasos para reproducir:</p> <p>Abrir cualquier producto.</p> <p>Desplazarse hacia la sección de reseñas.</p> <p>Intentar escribir una opinión.</p> <p>Observar que la función no está disponible.</p>
Alto	Performance	<p>Al analizar la página en Google PageSpeed, se evidencian problemas críticos de rendimiento.</p> <p>Los valores combinados de LCP (Largest Contentful Paint) y TTFB (Time To First Byte) muestran:</p> <p>TTFB adecuado: el servidor responde rápido.</p> <p>LCP extremadamente alto (11.4 s): el navegador tarda demasiado en renderizar el contenido principal.</p> <p>Analizar la URL.</p> <p>Revisar métricas LCP y TTFB.</p> <p>Observar que TTFB es bueno, pero el LCP es excesivamente alto.</p>

Tabla 3. Matriz de riesgo.

RIESGO	PLAN CONTINGENCIA				
	Probabilidad (Ocurrencia)	Impacto (Gravedad)	Valor del Riesgo	Nivel de Riesgo	
UH-U-01	4	5	20	Muy grave	Pruebas E2E de Pago
UH-U-02	3	4	12	Importante	Pruebas de Conciliación
UH-U-03	4	5	20	Muy grave	Pruebas de Seguridad
UH-U-04	3	3	9	Importante	Pruebas de Notificación
UH-U-05	2	3	6	Apreciable	Pruebas de Administración
UH-U-06	4	4	16	Muy grave	Pruebas de Carga y Estrés
UH-U-07	3	3	9	Importante	Pruebas de Sincronización
UH-U-08	2	4	8	Apreciable	Pruebas de Validación de Entrada
UH-U-09	3	4	12	Importante	Pruebas de Integración Específica
UH-U-10	3	5	15	Muy grave	Pruebas de Cumplimiento Legal
UH-U-11	3	4	12	Importante	Pruebas de Cálculo
UH-U-12	2	5	10	Importante	Pruebas de Rollback
UH-U-13	3	2	6	Apreciable	Pruebas de Usabilidad de Búsqueda
UH-U-14	3	4	12	Importante	Pruebas de Permisos (RBAC)

Tabla 4. Calculadora de estimacion.

Actividad	Total Horas Estimadas	Horas con % Desviación	Total Días Estimados	Total Días con % Desviación
Contextualización	40,00	78,80	5	9,85
Entendimiento de la solución de software	16,00	31,52	2,00	3,94
Reunión de contextualización	8,00	15,76	1,00	1,97
Estimación a alto nivel (plan del plan)	8,00	15,76	1,00	1,97
Sustentación de la estimación a alto nivel	4,00	7,88	0,50	0,99
Crear proyecto en herramientas de apoyo	4,00	7,88	0,50	0,99
		0,00	0,00	0,00
Planeación	56,00	110,32	7	13,79
Elaboración del plan de calidad	32,00	63,04	4,00	7,88
Socializar y aprobar plan de calidad	16,00	31,52	2,00	3,94
Realizar calendario de actividades	8,00	15,76	1,00	1,97
		0,00	0,00	0,00
		0,00	0,00	0,00
		0,00	0,00	0,00
Diseño/Ejecución	144,00	283,68	18	35,46
Preparación de datos y ambiente (Final)	24,00	47,28	3,00	5,91
Ejecución de pruebas (Funcional, Integración, Seguridad)	64,00	126,08	8,00	15,76
Monitoreo y seguimiento de pruebas (Performance y Bugs)	24,00	47,28	3,00	5,91
Historia de Usuario 1	20,00	39,40	2,50	4,93
Regresión	12,00	23,64	1,50	2,96
		0,00	0,00	0,00
Documentación, cierre y retrospectiva	26,00	51,22	3,25	6,4025
Actualización documentación diaria	2,00	3,94	0,25	0,49
Creación informe de cierre	16,00	31,52	2,00	3,94
Socialización del informe de cierre	8,00	15,76	1,00	1,97
		0,00	0,00	0,00
		0,00	0,00	0,00
		0,00	0,00	0,00
Total Tiempo Estimado de Pruebas	266,00	524,02	33,25	65,50

Tabla 5. Ejemplo de caso de prueba.

Id	Nombre o Descripción	Objetivo	Precondiciones	Pasos	Resultado Esperado	Resultado Obtenido	Estado	Observaciones
CP1.001	Visualización correcta del catálogo y uso de paginación	Verificar que el catálogo carga 10 productos por página, muestra precios de oferta y que la navegación avanza/retrocede correctamente.	1. La base de datos contiene más de 20 productos de diversas categorías y al menos 5 con precios de oferta con su respectivo precio actual y precio anterior. 2. El entorno de visualización está operativo.	1. Abrir la página principal del catálogo. 2. Contar los productos visibles. 3. Identificar un producto en oferta y verificar que muestre dos precios. 4. Pulsar el botón "Siguiente". 5. Pulsar el botón	1. Deben visualizarse exactamente 10 productos. 2. El producto en oferta debe mostrar el precio actual y el anterior. 3. Al pulsar "Siguiente", se deben cargar 10 productos diferentes. 4. Al pulsar "Anterior", se debe	El catálogo muestra 10 productos. Los productos en oferta muestran correctamente el precio actual tachando el precio anterior. Los botones "Siguiente" y "Anterior" funcionan correctamente, cargando nuevos productos en menos de	Aprobado	El rendimiento de la paginación es óptimo, validando la mitigación del riesgo de latencia (Contingencia A - UH-U-01).
CP1.002	Aplicación correcta del filtro de categoría (Sexo/Tipo de Artículo).	Verificar que al aplicar filtros combinados (ej. "Hombre" y "Camisetas"), solo se muestren productos que cumplan ambas condiciones.	La base de datos contiene productos claramente clasificados.	1. Abrir el catálogo. 2. Aplicar el filtro de "Sexo" (ej. "Mujer"). 3. Aplicar el filtro de "Tipo de Artículo" (ej. "Pantalones"). 4. Contar los productos visibles. 5. Verificar la descripción/categoría	1. Solo se deben visualizar productos que coincidan con la clasificación "Mujer" y "Pantalones". 2. El número de productos visibles no debe superar el límite de 10 por página.	Tras aplicar los filtros, solo se muestran 6 productos que son "Pantalones de Mujer". El filtro aplica correctamente la lógica AND y no supera el límite de 10.	Aprobado	La lógica de clasificación múltiple es correcta, validando la integridad del dato
CP1.003	Manejo de error al cargar productos sin datos obligatorios (ej. sin precio).	Verificar que la interfaz maneja elegantemente (no crashea) el intento de mostrar productos con datos faltantes.	Configurar manualmente un producto en la BD para que no tenga asignado un precio actual o un nombre.	1. Abrir el catálogo. 2. Aplicar un filtro que muestre el producto con datos faltantes.	1. El sistema NO debe generar un error de tipo '500' o crashear la interfaz. 2. Idealmente, el producto debe mostrar un mensaje de "Precio no disponible" o simplemente no cargarse	El catálogo crashea y devuelve un error 500 al intentar cargar la página, ya que el front-end no tiene una regla para manejar el campo de precio nulo (null).	Fallido	Se debe implementar una validación null check en el front-end y back-end para campos obligatorios como el precio. Se reabre el ticket al desarrollador Back-end para aplicar la mitigación de integridad de datos.

Conclusiones

El desarrollo del proyecto permitió evidenciar la importancia de realizar una adecuada planificación y una correcta estimación de tiempos en el proceso de pruebas de software. La diferencia entre los tiempos estimados y los tiempos reales demuestra la necesidad de considerar factores como la experiencia del equipo, la complejidad de las actividades y los posibles riesgos que pueden surgir durante la ejecución.

Asimismo, el uso de herramientas como el plan de pruebas, el bugtracker y la matriz de riesgos facilitó el control y seguimiento de las actividades, permitiendo una mejor organización del trabajo. Finalmente, el trabajo en equipo, la comunicación constante y la gestión de riesgos se identifican como aspectos fundamentales para el logro de los objetivos y la mejora continua en futuros proyectos de pruebas de software.