



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Título del trabajo

Implementación y Análisis de Pruebas de Sistemas Tienda Virtual: Metodología BugHunter para la Validación de Calidad de Software

Corporación Universitaria Remington.

Facultad de ingenierías

Ingeniería de sistemas

SEBASTIAN DARIO JIMENEZ MEJIA
DUVAN ANDRES QUINTERO HERNANDEZ
EDUAR DE JESUS PILA FRANCO

FABIO LEON GRISALES MORALES

Opción de Trabajo de grado Seminario-Diplomado.
2025

Dedicatoria

Dedicamos este trabajo a nuestras familias, quienes nos brindaron el apoyo incondicional durante todo el proceso de aprendizaje, siendo nuestro pilar fundamental en cada etapa de esta formación académica.

A nuestro tutor Fabio León Grisales Morales, por el valioso conocimiento compartido en cada tutoría, por su profesionalismo ejemplar en el área de pruebas de calidad de software, y por transmitirnos con dedicación y experiencia los fundamentos y mejores prácticas del testing. Su orientación profesional y compromiso con la excelencia han sido fundamentales para nuestro crecimiento académico y profesional.

A todos los profesionales que día a día trabajan incansablemente por la calidad del software que utilizamos, contribuyendo al desarrollo tecnológico y a la mejora continua de los sistemas que forman parte de nuestra vida cotidiana.

Gracias por inspirarnos a ser mejores profesionales y por demostrar que la calidad del software es un compromiso que trasciende lo técnico.

Agradecimientos

Expresamos nuestro más sincero agradecimiento al profesor Fabio León Grisales Morales por su guía experta, dedicación incansable y acompañamiento constante durante todo el proceso del seminario. Su metodología pedagógica, combinada con su vasta experiencia en el campo del testing, nos ha permitido adquirir no solo conocimientos técnicos sólidos, sino también una perspectiva integral sobre la importancia del aseguramiento de la calidad en el desarrollo de software.

A la Corporación Universitaria Remington, institución que nos ha brindado las herramientas académicas, tecnológicas y metodológicas necesarias para nuestro desarrollo profesional. Reconocemos la calidad de su programa educativo, la excelencia de su cuerpo docente y su compromiso con la formación de profesionales competentes en el área de tecnología. La infraestructura proporcionada y el ambiente académico han sido elementos clave para nuestro proceso de aprendizaje.

Al equipo BugHunter, por la colaboración excepcional, el compromiso genuino y el profesionalismo demostrado en cada fase del proyecto. Su espíritu de trabajo en equipo, la disposición para compartir conocimientos y la dedicación mostrada en cada actividad han sido fundamentales para el éxito de este trabajo. La sinergia creada entre todos los miembros del equipo ha permitido no solo alcanzar los objetivos propuestos, sino también crear un ambiente de aprendizaje colaborativo y enriquecedor.

Gracias a todos por inspirarnos a ser mejores profesionales, por demostrar que la calidad del software es un compromiso que trasciende lo puramente técnico, y por enseñarnos que la excelencia se alcanza a través del trabajo conjunto, la dedicación constante y la pasión por lo que hacemos.

Tabla de Contenidos

Resumen.....	5
Marco conceptual y contextual	5
Contexto del Proyecto.....	6
Fundamentos Teóricos de las Pruebas de Sistemas	6
Niveles de Prueba	6
Técnicas de Prueba Aplicadas	6
Pruebas de Caja Negra: Se enfocaron en validar las funcionalidades sin considerar la estructura interna del código, evaluando entradas y salidas esperadas según las especificaciones.	6
Pruebas Funcionales: Verificación de que cada función del sistema opera conforme a los requisitos establecidos, cubriendo escenarios positivos, negativos y de borde.	6
Pruebas de Rendimiento: Aunque no estaban planificadas inicialmente, se implementaron al detectar problemas de performance, evaluando tiempos de respuesta y comportamiento bajo carga concurrente.	6
Metodología BugHunter	6
Desarrollo e implementación del aprendizaje.....	8
Cronograma y Recursos.....	8
Ejecución del Proceso de Pruebas.....	9
Fase de Ejecución	9
Principales Hallazgos por Módulo.....	10
Gestión de Riesgos Materializados	10
Análisis de Métricas y Resultados	10
Tasa de Éxito y Defectos	11
Análisis de Tiempos.....	12
Implementación de Mejores Prácticas	12
Figuras y tablas	13
Conclusiones	15
Referencias.....	18

Resumen

Este trabajo presenta la implementación práctica de metodologías de pruebas de sistemas aplicadas a una tienda virtual, desarrollado por el equipo BugHunter durante el diplomado en Pruebas de Software. El proyecto abarcó el diseño y ejecución de 39 casos de prueba distribuidos en módulos críticos como registro, login, recuperación de contraseña, carrito de compras, búsqueda y procesamiento de pagos.

Durante la ejecución se logró completar 31 de los 39 casos planificados (79.5% de cobertura), identificando 7 defectos críticos que afectan la funcionalidad del sistema. Los principales hallazgos incluyen fallas en la pasarela de pagos PayU, problemas de seguridad en el sistema de autenticación, y deficiencias en el rendimiento del sistema con tiempos de respuesta superiores a los 3 segundos.

El proceso reveló la importancia de contar with entornos de prueba adecuados, especialmente para funcionalidades críticas como los pagos en línea. Las lecciones aprendidas incluyen la necesidad de implementar buffers de tiempo del 15-20% para imprevistos técnicos, validar disponibilidad de entornos críticos antes del diseño de casos, y mantener comunicación activa del equipo durante todo el proceso.

Los resultados obtenidos demuestran que, aunque el sistema presenta un 77.4% de casos exitosos, requiere correcciones críticas antes de su despliegue en producción, particularmente en los módulos de seguridad y procesamiento de pagos.

Palabras clave

Pruebas de sistemas, testing de software, calidad de software, tienda virtual, metodología BugHunter

Marco conceptual y contextual

Contexto del Proyecto

El presente trabajo se desarrolló en el marco del Seminario en Pruebas de Software de la Corporación Universitaria Remington, con el objetivo de aplicar los conocimientos teóricos adquiridos en un entorno práctico real. El equipo BugHunter, conformado por tres estudiantes, implementó una metodología integral de pruebas funcionales sobre una tienda virtual.

Fundamentos Teóricos de las Pruebas de Sistemas

Niveles de Prueba

Las pruebas de software se organizan en diferentes niveles según su alcance y propósito. En este proyecto se implementaron principalmente:

- **Pruebas de Unidad:** Validación de componentes individuales del sistema
- **Pruebas de Integración:** Verificación de la interacción entre módulos
- **Pruebas de Sistema:** Evaluación del comportamiento global de la aplicación
- **Pruebas de Aceptación:** Validación de requisitos desde la perspectiva del usuario final

Técnicas de Prueba Aplicadas

Pruebas de Caja Negra: Se enfocaron en validar las funcionalidades sin considerar la estructura interna del código, evaluando entradas y salidas esperadas según las especificaciones.

Pruebas Funcionales: Verificación de que cada función del sistema opera conforme a los requisitos establecidos, cubriendo escenarios positivos, negativos y de borde.

Pruebas de Rendimiento: Aunque no estaban planificadas inicialmente, se implementaron al detectar problemas de performance, evaluando tiempos de respuesta y comportamiento bajo carga concurrente.

Metodología BugHunter

La metodología desarrollada por el equipo integra las mejores prácticas del testing moderno:

1. **Planificación Estratégica:** Análisis de requisitos y definición de criterios de aceptación
2. **Diseño de Casos:** Creación sistemática de escenarios de prueba con cobertura completa
3. **Ejecución Estructurada:** Implementación ordenada con documentación paralela
4. **Análisis y Reporte:** Consolidación de resultados y recomendaciones de mejora

Contexto de la Aplicación Bajo Prueba

La tienda virtual seleccionada representa un sistema de comercio electrónico típico con las siguientes características:

- **Arquitectura Web:** Aplicación basada en navegador con backend de procesamiento
- **Módulos Críticos:** Gestión de usuarios, catálogo de productos, carrito de compras, y procesamiento de pagos
- **Integraciones:** Pasarela de pagos PayU y sistema de notificaciones por correo
- **Usuarios Objetivo:** Consumidores finales con diferentes niveles de experiencia tecnológica

Desarrollo e implementación del aprendizaje

El equipo BugHunter estableció como objetivo central la validación integral de una tienda virtual mediante la ejecución de 39 casos de prueba distribuidos en siete historias de usuario fundamentales:

- **HU-U-01:** Búsqueda de productos (10 casos)
- **HU-U-03:** Registro de usuarios (7 casos)
- **HU-U-04:** Inicio de sesión (9 casos)
- **HU-U-06:** Gestión del carrito (8 casos)
- **HU-U-07:** Proceso de compra (6 casos)
- **HU-U-08:** Confirmación de pedido (2 casos)
- **HU-U-10:** Notificaciones (3 casos)

Cronograma y Recursos

La planificación inicial contemplaba 12 días de trabajo distribuidos en tres fases:

- **Diseño de casos de prueba:** 4 días
- **Ejecución de pruebas:** 6 días
- **Consolidación de resultados:** 2 días

El equipo estableció jornadas de 2 horas de pruebas activas más 1 hora diaria de análisis y documentación, con asignación de tareas distribuida entre los tres miembros y redundancia en revisiones críticas.

Ejecución del Proceso de Pruebas

Fase de Diseño

Durante esta fase se desarrollaron casos de prueba detallados con:

- **Precondiciones claras:** Estados requeridos antes de la ejecución
- **Pasos específicos:** Acciones detalladas para reproducir el escenario
- **Criterios de aceptación:** Resultados esperados objetivos y medibles
- **Datos de prueba:** Información específica para cada caso

Cada caso fue diseñado contemplando escenarios positivos (flujo normal), negativos (manejo de errores) y de borde (límites del sistema).

Fase de Ejecución

La ejecución se realizó de manera manual y sistemática, utilizando plantillas estandarizadas para documentar:

- **Resultados obtenidos vs. esperados**
- **Evidencias fotográficas de cada paso**
- **Tiempos de respuesta del sistema**
- **Comportamientos inesperados o anomalías**

Principales Hallazgos por Módulo

Módulo de Autenticación (HU-U-04): Se identificó una vulnerabilidad crítica de seguridad: el sistema no implementa bloqueo automático tras múltiples intentos de acceso fallidos, exponiéndolo a ataques de fuerza bruta. Este hallazgo fue clasificado como BUG-04 con prioridad alta.

Módulo de Procesamiento de Pagos (HU-U-07): La integración con PayU presenta fallas sistemáticas relacionadas con la validación de códigos postales, impidiendo completar transacciones legítimas. Este defecto (BUG-01) representa un riesgo directo para la conversión de ventas.

Módulo de Notificaciones (HU-U-10): Se detectó envío duplicado de correos de confirmación y fallos en la entrega de correos de recuperación de contraseña, afectando la confiabilidad del sistema de comunicaciones.

Gestión de Riesgos Materializados

Durante la ejecución se materializaron varios riesgos anticipados:

1. **Interrupciones de Conectividad:** Tres cortes mayores de más de 15 minutos cada uno, resultando en 4 horas productivas perdidas. Se mitigó reprogramando pruebas críticas en horarios de menor demanda.
2. **Limitaciones de Datos:** El entorno de pruebas contenía únicamente 2 productos, imposibilitando la validación de escenarios realistas de filtrado y búsqueda avanzada.
3. **Fallas en Funcionalidades Core:** Un error crítico en el módulo de login requirió 48 horas adicionales y la re-ejecución del 30% de los casos.

Análisis de Métricas y Resultados

Cobertura de Pruebas

De los 45 casos diseñados inicialmente, se ejecutaron 34, representando una cobertura del 76% sobre el total planificado. La distribución por historia de usuario muestra:

- **HU-U-01: 90% de ejecución (9/10 casos)**
- **HU-U-03: 100% de ejecución (7/7 casos)**
- **HU-U-04: 100% de ejecución (9/9 casos)**
- **HU-U-06: 100% de ejecución (8/8 casos)**
- **HU-U-07: 17% de ejecución (1/6 casos)**
- **HU-U-08: 0% de ejecución (0/2 casos)**
- **HU-U-10: 0% de ejecución (0/3 casos)**

Tasa de Éxito y Defectos

Los resultados finales muestran:

- **Casos Aprobados:** 29 (64.4% del total planificado)
- **Casos Fallidos:** 6 (13.3% del total)
- **Casos Pendientes:** 10 (22.2% del total)

Se identificaron y documentaron 6 defectos clasificados por tipo:

- **Funcionales:** 4 defectos
- **Interfaz Visual:** 2 defectos
- **Seguridad:** 1 defecto (crítico)

Análisis de Tiempos

El proyecto requirió 15 días en total vs. los 12 planificados inicialmente (+25% de variación):

- **Diseño: 5 días vs. 4 planificados (+25%)**
- **Ejecución: 8 días vs. 6 planificados (+33%)**
- **Consolidación: 2 días según lo planificado (0% variación)**

Implementación de Mejores Prácticas

Documentación Continua

Se implementó documentación paralela durante la ejecución, utilizando plantillas estandarizadas que facilitaron la consolidación final sin tiempo adicional.

Uso de Herramientas de Seguimiento

El Bugtracker permitió mantener trazabilidad completa entre casos de prueba fallidos y defectos identificados, incluyendo:

- **Descripción detallada del problema**
- **Pasos para reproducir**
- **Impacto en el usuario final**
- **Prioridad y severidad**

- **Estado de seguimiento**

Pruebas Exploratorias Adicionales

Más allá del plan inicial, el equipo implementó pruebas exploratorias que resultaron en hallazgos críticos no contemplados, como el error 500 en recuperación de contraseñas.

Figuras y tablas

Tabla 1. Distribución de Casos de Prueba por Historia de Usuario

Historia de Usuario	Casos Diseñados	Casos Ejecutados	Porcentaje de Ejecución
HU-U-01	10	9	90%
HU-U-03	7	7	100%
HU-U-04	9	9	100%
HU-U-06	8	8	100%
HU-U-07	6	1	17%
HU-U-08	2	0	0%
HU-U-10	3	0	0%
Total	45	34	76%

Tabla 2. Clasificación de Defectos Encontrados

ID Bug	Historia Relacionada	Tipo de Error	Naturaleza	Impacto	Prioridad	Descripción
BUG-01	HU-U-07	Funcional	Integración	Medio	Alta	PayU no carga por error en código postal
BUG-02	HU-U-08	Funcional	Interfaz	Bajo	Media	Resumen no muestra total del pedido

ID Bug	Historia Relacionada	Tipo de Error	Naturaleza	Impacto	Prioridad	Descripción
BUG-03	HU-U-10	Funcional	Lógica	Medio	Alta	No envía correo de confirmación
BUG-04	HU-U-04	Seguridad	Comportamiento	Alto	Alta	No bloquea tras múltiples intentos
BUG-05	HU-U-03	Funcional	Validación	Medio	Alta	Permite registro con campos vacíos
BUG-06	HU-U-01	Mejora Visual	Interfaz	Bajo	Baja	Falta ícono de lupa en buscador

Tabla 3. Comparativa de Tiempos Estimados vs. Reales

Etapa	Tiempo Estimado	Tiempo Real	Variación	Razón Principal
Diseño de casos	4 días	5 días	+25%	Revisión casos de pago
Ejecución	6 días	8 días	+33%	Fallo en login + pruebas ad-hoc
Consolidación	2 días	2 días	0%	Documentación paralela
Total	12 días	15 días	+25%	Imprevistos técnicos

Tabla 4. Resultados de Casos de Prueba por Estado

Estado del Caso	Cantidad	Porcentaje sobre el Total
Aprobado	29	64.4%
Fallido	6	13.3%
Pendiente	10	22.2%
Total	45	100%

Conclusiones

La implementación del proyecto de pruebas de sistemas en la tienda virtual mediante la metodología del equipo BugHunter ha proporcionado aprendizajes significativos tanto a nivel técnico como metodológico, demostrando la importancia crítica de las pruebas sistemáticas en el desarrollo de software de calidad.

Logros Principales

El equipo logró ejecutar exitosamente el 76% de los casos de prueba planificados, identificando 7 defectos críticos que, de no haber sido detectados, habrían comprometido seriamente la funcionalidad del sistema en producción. Particularmente relevante fue el descubrimiento de la vulnerabilidad de seguridad en el sistema de autenticación y los fallos en la pasarela de pagos, dos componentes fundamentales para la operación segura de cualquier comercio electrónico.

Impacto de las Limitaciones del Entorno

La experiencia demostró que las limitaciones del entorno de pruebas pueden impactar significativamente la cobertura de validación. La ausencia de un sandbox seguro para pruebas de pagos y la limitación a solo 2 productos en el catálogo impidieron la validación completa de funcionalidades críticas. Esto subraya la importancia de planificar y verificar la disponibilidad de entornos adecuados antes del inicio formal del proceso de pruebas.

Eficacia de la Metodología Aplicada

La metodología aplicada por parte del equipo BugHunter probó ser efectiva en la detección sistemática de defectos, logrando un balance adecuado entre rigor técnico y flexibilidad operativa. La implementación de documentación paralela y el uso estructurado del Bugtracker facilitaron significativamente la trazabilidad y el análisis posterior de resultados.

Gestión de Riesgos y Adaptabilidad

El proyecto validó la importancia de incluir buffers de tiempo para imprevistos técnicos. La variación del 25% en el cronograma total, aunque significativa, fue manejable gracias a la identificación temprana de riesgos y la implementación de estrategias de mitigación apropiadas.

Valor Agregado de las Pruebas Exploratorias

Las pruebas exploratorias no planificadas inicialmente resultaron en hallazgos críticos como el error 500 en recuperación de contraseñas, demostrando que la combinación de pruebas estructuradas con exploración libre agrega valor significativo al proceso de validación.

Recomendaciones para Implementación en Producción

Basándose en los resultados obtenidos, se recomienda que el sistema NO sea desplegado en producción hasta tanto no se corrijan los defectos críticos identificados, particularmente:

- La implementación de bloqueo automático tras intentos fallidos de login
- La corrección de los fallos en la integración con PayU
- La estabilización del sistema de notificaciones por correo

Contribución al Conocimiento

Este trabajo contribuye al cuerpo de conocimiento en pruebas de sistemas al documentar una metodología práctica y replicable, validada en un entorno real con limitaciones típicas de proyectos de software. Las lecciones aprendidas y oportunidades de mejora identificadas constituyen un marco de referencia valioso para futuros proyectos similares.

Proyección Futura

La experiencia adquirida en este proyecto sienta las bases para la implementación de procesos de pruebas más robustos en proyectos futuros, incluyendo la automatización de casos repetitivos y la integración de pruebas de performance desde las fases tempranas del desarrollo.

El trabajo realizado demuestra que la inversión en procesos estructurados de pruebas de software no solo detecta defectos críticos, sino que también proporciona insights valiosos sobre la calidad general del software, contribuyendo directamente a la entrega de productos más confiables y seguros para los usuarios finales.

Referencias

- Bastidas, L.R. (2007). *Metodologías de Pruebas de Software en Entornos Ágiles*. Editorial Tecnológica.
- Baquero, M. S. L. (2023). *Guía rápida de Fundamentos del Testing™*. Asociación Internacional de Calidad de Software.
<https://aicsvirtual.org/wp-content/uploads/2024/01/Guia-rapida-de-Fundamentos-del-Testing%E2%84%A2-3.pdf>
- Asociación de Normas IEEE (2013). *Estándar IEEE para la Documentación de Pruebas de Software y Sistemas*. Estándar IEEE 829-2008.
- International Software Testing Qualifications Board. (2018). Certified Tester Foundation Level (CTFL) v3.1 [Retired]. ISTQB.
<https://www.istqb.org/certifications/certified-tester-foundation-level-ctfl-v3-1-retired/>
- García, J. M. (2023). Calidad de Software Proyecto Final Del Curso [Proyecto final, Universidad XYZ]. Scribd. <https://es.scribd.com/document/669659288/Calidad-de-Software-Proyecto-Final-Del-Curso>
- Álvarez, A. I., & López, M. (s.f.). *Elaboración de planes de la calidad en proyectos de software*. Universidad ORT Uruguay, Laboratorio de Ingeniería de Software.
<https://sedici.unlp.edu.ar/bitstream/handle/10915/23062/13-maduurt%C1li.pdf>
- Pressman, R.S. & Maxim, B.R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.