



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Desarrollo e implementación de pruebas de software para tienda online

Corporación Universitaria Remington.

Faculta de Ingenierías

Ingeniería de sistemas y Tecnología en desarrollo de software

David Stiven Muñoz, Santiago Colorado Martínez y Sebastian Castrillon

Fabio León Grisales Morales

Seminario de pruebas de software

2024

Tabla de Contenidos

Resumen.....	3
Marco conceptual y contextual	4
Marco conceptual.....	4
Marco contextual	4
Desarrollo e implementación del aprendizaje.....	5
Planteamiento del problema.	5
Planeación	5
Ilustración I.....	6
Ilustración 2.	9
Ilustración 3.	10
Desarrollo de las pruebas.....	10
Ilustración 4.	11
Conclusiones.....	12
Bibliografía.....	13

Resumen

El siguiente trabajo se enfoca en la implementación de pruebas de software en una tienda web, desarrollando distintos aspectos esenciales para asegurar la calidad del producto final. Se comienza con un análisis exhaustivo de riesgos, identificando posibles amenazas como la seguridad de los datos del cliente y la estabilidad del sistema en diferentes circunstancias, lo cual prosiguió a una estrategia hacia la mitigación de estos riesgos, priorizando la seguridad y la capacidad de escalabilidad.

Se definió el alcance del proyecto, centrándose en la funcionalidad esencial del sitio web: la navegación del catálogo, la gestión de carritos de compra y la pasarela de pago. Se estableció una estrategia de pruebas funcionales estimando los tiempos necesarios considerando la complejidad del funcionamiento, la disponibilidad de recursos y el calendario de pruebas.

Con el flujo de pruebas se cubrió todas las etapas del desarrollo asegurándose de contar con precondiciones claras como configuración de entorno de pruebas replicando el entorno de producción, definir roles en el equipo y la revisión y aprobación de los casos de prueba.

En los tipos de pruebas, se implementaron pruebas funcionales que abarcaron los aspectos definidos en el alcance y pruebas de integración para evaluar la capacidad del software cuando entre en interacción con los distintos módulos y sistema.

El enfoque de este trabajo es implementar pruebas de software en una tienda web mejorando la experiencia del usuario final y fortaleciendo la confianza en la estabilidad y fácil manejo del sistema, fundamental en entornos comerciales competitivos y exigentes, donde la calidad del software es fundamental en el éxito del negocio.

Marco conceptual y contextual

Marco conceptual

Las pruebas de software son un proceso sistemático destinado para evaluar y verificar la funcionalidad de una aplicación o sistema de software. El objetivo principal es identificar defectos y asegurar que el software cumple con los requisitos especificados, ofreciendo un producto de alta calidad y fiable al usuario final. Las pruebas de software abarcan diversas metodologías y tipos, incluyendo pruebas funcionales, de integración, de rendimiento, de usabilidad y de seguridad.

En el ciclo de vida del desarrollo de software, las pruebas juegan un papel crucial para garantizar que el producto final esté libre de errores y sea funcionalmente completo. Las pruebas ayudan a prevenir problemas en etapas tempranas del desarrollo, reduciendo así costos y esfuerzos adicionales en fases posteriores. La calidad del software es esencial para mantener la satisfacción del cliente y la integridad de la marca.

Marco contextual

El comercio electrónico ha transformado la manera en que los consumidores interactúan con las empresas. En el sector de la moda y los accesorios, las tiendas online han crecido exponencialmente, ofreciendo una alternativa conveniente a las tiendas físicas. Esta evolución exige que las plataformas de comercio electrónico sean robustas, seguras y ofrezcan una experiencia de usuario excepcional para mantener la competitividad.

El mercado de la moda online enfrenta desafíos como la alta competencia, las expectativas cambiantes de los consumidores, y la necesidad de mantener la seguridad y

privacidad de los datos. Sin embargo, también ofrece oportunidades significativas, como la capacidad de llegar a un público global, la personalización de ofertas y la integración de nuevas tecnologías para mejorar la experiencia del cliente.

La aplicación operará en un entorno web, accesible desde diversos dispositivos como computadoras, tablets y smartphones. Es esencial que sea responsive y compatible con diferentes navegadores.

Desarrollo e implementación del aprendizaje

Planteamiento del problema.

El principal problema radica en garantizar que la aplicación web de la tienda online funcione de manera correcta y eficiente, proporcionando una experiencia de usuario positiva y cumpliendo con los estándares de seguridad y funcionalidad requeridos. Esto implica abordar varios subproblemas específicos:

- Integración de Pasarelas de Pago
- Gestión de Catálogo de Productos
- Notificaciones y Comunicación con Usuarios
- Usabilidad y Experiencia de Usuario
- Conciliación Financiera

Planeación

El primer paso que se llevó a cabo fue la elaboración de un plan de pruebas detallado. Este plan era fundamental para garantizar la calidad y el éxito del proyecto. Contábamos con un total de 14 historias de usuarios, cada una describiendo una funcionalidad específica que debía ser probada y validada.

Se inició el proceso con la creación de una matriz de riesgo. Esta herramienta fue crucial para identificar y evaluar los posibles riesgos asociados con cada historia de usuario. La matriz de riesgo nos permitió:

- Priorizar las historias de usuarios
- Planificar las actividades de prueba
- Definir medidas preventivas y correctivas

Ilustración I

Matriz de Riesgo

MATRIZ DE RIESGOS					
RIESGO	Probabilidad (Ocurrencia)	Impacto (Gravedad)	Valor del Riesgo	Nivel de Riesgo	PLAN CONTINGENCIA
Errores en la visualización del catálogo	2	4	8	Apreciable	Realizar revisiones de código centradas en la UI antes de cada despliegue.
Si hay fallos en la integración con pasarelas de pago, las transacciones no podrán completarse	3	5	15	Muy grave	Colaborar con los proveedores de pasarelas de pago para solucionar problemas rápidamente.
Las inconsistencias en los datos del catálogo podrían resultar en información incorrecta de los productos.	3	3	9	Importante	Implementar pruebas de validación de datos automatizadas para verificar la consistencia del catálogo.
La inoperabilidad del Ambiente Demo impedirá la realización de pruebas.	2	5	10	Importante	Mantener respaldos actualizados del entorno de demostración.
La falta de personal podría llevar a tiempos de respuesta más largos.	2	3	6	Apreciable	Desarrollar planes de capacitación cruzada para que el personal pueda cubrir múltiples roles.
Los errores en el cálculo y la conciliación de ventas provocarán desajustes en los balances financieros.	2	5	10	Importante	Implementar pruebas automatizadas para la lógica de cálculo y conciliación.
Los errores en la gestión de inventario causarán que los niveles de stock no se actualicen adecuadamente.	3	3	9	Importante	Ejecutar una revisión manual del inventario y corregir discrepancias.
Los problemas en el carrito de compras evitarán que los usuarios puedan finalizar sus compras.	4	3	12	Importante	Realizar pruebas de funcionalidad del carrito de compras regularmente y monitorear su desempeño.
Los errores en el registro de nuevos usuarios impedirán la creación correcta de cuentas nuevas.	2	3	6	Apreciable	Realizar pruebas de funcionalidad y usabilidad regularmente.

Después de realizar la matriz de riesgo, procedimos a la elaboración del alcance de las pruebas. Este paso fue crucial para definir qué historias de usuario se incluirían en el proceso de pruebas y cuáles no.

Se plantearon y seleccionaron las historias de usuario que serían sometidas a pruebas. La selección se basó en varios criterios, incluyendo:

- **Resultados de la Matriz de Riesgo:** Historias con mayor riesgo recibieron prioridad, asegurando que se abordaran potenciales problemas antes de que afectaran la calidad del producto.
- **Relevancia Funcional:** Se tuvieron en cuenta las funcionalidades que, según nuestro criterio, eran más relevantes para asegurar la calidad del producto. Estas funcionalidades críticas fueron priorizadas para garantizar que el producto final cumpliera con las expectativas y requerimientos del usuario.

El alcance de las pruebas tuvo como objetivo principal asegurar la calidad del producto final. Esto se logró mediante:

- **Enfoque en Funcionalidades Críticas:** Asegurar que las funcionalidades más importantes fueran rigurosamente probadas para detectar y corregir cualquier problema antes de la implementación final.
- **Optimización de Recursos:** Al priorizar las historias de usuario más relevantes y de mayor riesgo, se optimizaron los recursos disponibles, enfocándose en áreas que tendrían el mayor impacto en la calidad del producto.

La elaboración del alcance de las pruebas fue un paso fundamental en el proceso de pruebas del proyecto. Definir claramente qué historias de usuario serían probadas y cuáles no, nos permitió enfocar nuestros esfuerzos en las áreas de mayor relevancia y riesgo. Este enfoque garantizó que se mantuviera la calidad del producto, cumpliendo con los estándares esperados y las necesidades del usuario final.

Con el alcance de las pruebas ya determinado, procedimos a elaborar un cuadro detallado que estableciera los tipos de pruebas a realizar y las técnicas a utilizar, teniendo en cuenta las funcionalidades seleccionadas.

Para garantizar una cobertura adecuada y exhaustiva, se decidió utilizar la técnica de caja negra. Esta técnica nos permitió poner a prueba los casos de uso planteados sin necesidad de un conocimiento profundo del funcionamiento interno de la aplicación.

La elección de la técnica de caja negra se basó en varias ventajas clave:

- **Independencia del Conocimiento Interno:** No requiere un conocimiento detallado del código o la estructura interna de la aplicación, lo que facilita la identificación de defectos desde la perspectiva del usuario.
- **Eficiencia en la Prueba de Casos de Uso:** Permite centrarse en la validación de entradas y salidas, asegurando que la aplicación responde correctamente a los diferentes escenarios de uso previstos.

Dentro de las técnicas de caja negra, se seleccionó específicamente la técnica funcional. Esta técnica se encarga de validar que las características del aplicativo cumplen con los requisitos y expectativas definidos. La técnica funcional fue elegida por las siguientes razones:

- **Enfoque en la Funcionalidad:** Se asegura de que cada función de la aplicación opere conforme a los requisitos especificados, verificando que las entradas correctas produzcan las salidas esperadas.

- **Detección de Desviaciones:** Identifica cualquier desviación de la funcionalidad esperada, ayudando a garantizar que el producto final cumpla con las expectativas del usuario.

Ilustración 2.

Cuadro de Estrategia.

Tipo	Nivele	Técnicas	Tipos Prueba	Criterios de entrada	Criterios de salida
Analítica	Integración sistema	Caja negra	Funcional	<ol style="list-style-type: none"> 1. Requisitos y especificaciones del sistema 2. Ambiente de prueba configurado 3. Casos de prueba definidos 4. Datos de prueba preparados 5. Acceso a las herramientas de prueba 	<ol style="list-style-type: none"> 1. Ejecución completa de casos de prueba 2. Resultados documentados 3. Incidentes registrados y gestionados 4. Requisitos cumplidos 5. Pruebas de regresión realizadas 6. Aprobación del equipo de calidad

Ya con todo lo anterior se procedió a la elaboración de una estimación del proyecto para poder determinar los tiempos que se le dedicarían a la realización de cada prueba.

La estimación del proyecto involucró varias etapas clave:

Se desglosaron todas las actividades necesarias para la ejecución de las pruebas, considerando cada historia de usuario y las funcionalidades seleccionadas.

- **Asignación de Tiempos:** Para cada actividad identificada, se asignaron tiempos estimados, tomando en cuenta la complejidad y el esfuerzo requerido. Esta asignación se basó en experiencias previas y en la dificultad relativa de cada prueba.
- **Revisión y Ajuste:** Las estimaciones iniciales fueron revisadas y ajustadas en colaboración con el equipo de pruebas, asegurando que fueran realistas y alcanzables.

Ilustración 3.

Estimación del proyecto .

Diseño/Ejecución	32,00	35,20	4	4,4
Visualización del catálogo	2,00	2,20	0,25	0,28
Integración con Pasarelas de Pago	5,00	5,50	0,63	0,69
Datos del Catálogo	2,00	2,20	0,25	0,28
Cálculo y Conciliación de Ventas	3,00	3,30	0,38	0,41
Envío de Mensajes a los Usuarios	3,00	3,30	0,38	0,41
Gestión de Inventario	2,00	2,20	0,25	0,28
Inicio de Sesión de los Usuarios	3,00	3,30	0,38	0,41
Suscripción a Productos	3,00	3,30	0,38	0,41
Carrito de Compras	5,00	5,50	0,63	0,69
Registro de Nuevos Usuarios:	4,00	4,40	0,50	0,55
		0,00	0,00	0,00
Documentación, cierre y retrospectiva	12,00	13,20	1,5	1,65
Actualización documentación diaria	2,00	2,20	0,25	0,28
Creación informe de cierre	4,00	4,40	0,50	0,55
Socialización del informe de cierre	2,00	2,20	0,25	0,28
Creación SUITE (Base de conocimiento)	3,00	3,30	0,38	0,41
Retrospectiva del proyecto	1,00	1,10	0,13	0,14
		0,00	0,00	0,00
Total Tiempo Estimado de Pruebas	70,00	77,00	8,75	9,63

Desarrollo de las pruebas.

Con toda la planificación realizada, se procedió a establecer todos los casos de prueba para las historias de usuario seleccionadas en una tabla detallada. Esta tabla fue fundamental para organizar y documentar el proceso de pruebas.

- **Funcionalidad a probar:** Se especificó claramente qué funcionalidad de la aplicación sería objeto de prueba, basándonos en las historias de usuario seleccionadas.
- **Objetivo:** Se definió el objetivo de cada prueba, describiendo qué se buscaba validar o verificar con la ejecución del caso de prueba.
- **Pasos:** Se detallaron los pasos específicos que el equipo de pruebas seguiría para llevar a cabo cada prueba. Esta sección incluyó una secuencia clara y ordenada de acciones necesarias para realizar la prueba de manera efectiva.
- **Resultado esperado:** Para cada caso de prueba, se estableció cuál sería el resultado esperado si la funcionalidad operaba correctamente. Esto incluyó descripciones precisas de las salidas esperadas para las entradas dadas.
- **Resultado obtenido:** Se documentó el resultado real obtenido al realizar la prueba. Esta comparación entre el resultado esperado y el resultado obtenido permitió identificar cualquier discrepancia y tomar acciones correctivas cuando fuese necesario.

Ilustración 4.

Tabla de casos de prueba.

Plantilla de Casos de Pruebas de Software

Id	Nombre o Descripción	Objetivo	Precondiciones	Pasos	Resultado Esperado	Información para el Seguimiento		Estado	Observaciones
						Resultado Obtenido			
UHU-07 CP001	Visualización de Datos de compra	Verificar que se muestren correctamente los datos de los artículos o artículos que se están comprando (datos personales, datos de envío, descuentos).	Estar logeado y haber agregado uno o varios artículos al carro de compra.	1) Ingresar a la página y logearse. 2) Ingresar al módulo de catálogo 3) Agregar un producto al carro de compra 4) Ingresar al carro de compras	Se debe visualizar los datos del producto que se va a adquirir, además de los datos de envío y si se tienen descuentos el valor.	La visualización de la información es correcta			
UHU-07 CP002	Agregar Vales de descuento	Verificar que los vales de descuento puedan ser agregados a la compra	Estar logeado y haber agregado uno o varios artículos al carro de compra	1) Ingresar a la página y logearse. 2) Ingresar al módulo de catálogo 3) Agregar un producto al carro de compra 4) Ingresar al carro de compras 5) Seleccionar la opción agregar vale	Se debería de agregar correctamente el vale de descuento y el valor debería actualizarse	No se visualiza la opción de agregar vales de descuento			Se finalizó toda la operación de compra y no se mostró la opción de agregar vales
UHU-07 CP003	Modificar datos	Verificar que en caso de haber un dato erróneo o desactualizado el usuario lo pueda corregir	Estar logeado y haber agregado uno o varios artículos al carro de compra	1) Ingresar al módulo de catálogo 2) Agregar un producto al carro de compra 3) Ingresar al carro de compras 4) Seleccionar el dato a modificar 5) Editar la información	La actualización de los datos debería de reflejarse en la información de la compra	Los Datos se pueden corregir o actualizar			
UHU-08 CP001	Confirmación de compra exitosa	Verificar que al momento de finalizar una compra, se no muestre un mensaje de confirmación de la transacción.	Estar logeado y haber agregado uno o varios artículos al carro de compra.	1) Ingresar a la página y logearse. 2) Ingresar al módulo de catálogo 3) Agregar un producto al carro de compra 4) Ingresar al carro de compras 5) Llenar el formulario de compra 6) Si se tienen vales de descuentos agregarlos 7) Seleccionar medio de pago 8) Seleccionar la opción finalizar compra	Se visualizará en la pantalla un mensaje confirmando si la transacción fue un éxito.	Al finalizar la compra aparece una notificación de confirmación			El mensaje se visualiza claramente
UHU-08 CP002	Compra no exitosa	Verificar que al momento de finalizar una compra y se tengan los datos del medio de pago	Estar logeado y haber agregado uno o varios artículos al carro de compra.	1) Ingresar a la página y logearse. 2) Ingresar al módulo de catálogo 3) Agregar un producto al carro de compra	Se visualizará en la pantalla un mensaje confirmando si la	Aparece una notificación			

La ejecución de los casos de prueba fue esencial para garantizar la calidad y funcionalidad de la aplicación web de la tienda online. Estas pruebas permitieron identificar y corregir fallos tempranos, validar que las funcionalidades cumplieran con los requisitos y mejorar la calidad del producto final.

La mayoría de los casos de prueba fueron exitosos, lo que indica que las funcionalidades principales de la aplicación operan conforme a los requisitos específicos. Esto incluye:

- La visualización del catálogo
- La gestión del carrito de compras
- La integración con la pasarela de pago

En resumen, las pruebas realizadas aseguraron que la aplicación web cumpliera con los estándares esperados de calidad y funcionalidad, proporcionando una mejor experiencia de usuario.

Conclusiones

La implementación de un plan de pruebas bien estructurado resultó fundamental para garantizar la funcionalidad y estabilidad de la aplicación web. Este enfoque sistemático permitió

identificar y corregir defectos de manera efectiva, asegurando que el producto final cumpliera con los requisitos de calidad y ofreciera una experiencia de usuario fluida y segura.

La elaboración de una matriz de riesgo fue crucial para identificar y priorizar posibles fallos en el sistema. Este análisis permitió enfocar los esfuerzos en áreas de mayor riesgo, como la integración de pasarelas de pago y la gestión del catálogo de productos, minimizando así la probabilidad de que problemas críticos afectaran la operación de la tienda en línea.

La adopción de diversas metodologías de pruebas, en especial pruebas funcionales, aseguró una cobertura completa y exhaustiva de la aplicación. Esto garantizó no solo la funcionalidad correcta de la aplicación, sino también su seguridad y eficiencia, aspectos vitales en el contexto del comercio electrónico.

Las pruebas exhaustivas de funcionalidades críticas, como la visualización del catálogo, la gestión del carrito de compras, y la integración de pasarelas de pago, aseguraron que la aplicación proporcionara una experiencia de usuario intuitiva y libre de problemas. En el competitivo mercado del comercio electrónico, una experiencia de usuario positiva es crucial para atraer y retener clientes.

Bibliografía

Fundamentos de la Prueba de Software: conceptos, justificación y alcance.

(n.d.). SG Buzz. <https://sg.com.mx/revista/4/fundamentos-prueba-software-conceptos-justificacion-y-alcance>

Paz, J. M. (2016). Análisis del proceso de pruebas de calidad de software.

Ingeniería Solidaria/Revista Ingeniería Solidaria, 12(20), 163–176.

<https://doi.org/10.16925/in.v12i20.1482>

Alonso, D. B., Rodríguez, M. I. A., & Alonso, A. S. (2007). *Pruebas de software y junit. un análisis en profundidad y ejemplos prácticos*. PRENTICE HALL.

Aristegui, J. L. (2010). Los Casos de Prueba en la Prueba del Software.

LáMpsakos, 3, 27. <https://doi.org/10.21501/21454086.785>

Morales, E. C., Rivera, M. E. R., & Lizama, E. R. (2017). Desarrollo de un Modelo de Pruebas Funcionales de Software Basado en la Herramienta SELENIUM.

Industrial Data, 20(1), 139. <https://doi.org/10.15381/idata.v20i1.13498>

M, E. S. (2019). *Análisis y comparación de las propuestas recientes para diseñar casos de prueba desde los casos de uso orientados a verificar los aspectos funcionales del software*. <https://repositorio.unal.edu.co/handle/unal/9280>

Palacio, L. G. (2009). Generating functional testing case method in software development. *Revista Ingenierías Universidad De Medellín/Revista Ingenierías Universidad De Medellín*, 8(15), 29–36.

<https://dialnet.unirioja.es/descarga/articulo/4845741.pdf>

Mascheroni, M. A., & Irrazábal, E. (2018). Problemas que afectan a la calidad de software en entrega continua y pruebas continuas. *XXIV Congreso Argentino De Ciencias De La Computación (La Plata, 2018)*.

http://sedici.unlp.edu.ar/bitstream/handle/10915/73270/Documento_completo.pdf?sequence=1