

TRABAJO DE GRADO
Opción Seminario

PROYECTO DE GRADO
BUG MASTER

Corporación Universitaria Remington.
Facultad de Ingeniería de Sistemas
Ingeniería de Sistemas – Esp. Informática Educativa

Edwin Enrique Vega Orozco
Yefer Molina Molina.
Docente: Fabio León Grisales Morales
Opción de Trabajo de grado Seminario
Año 2024

Tabla de Contenidos

Resumen.....	3
Palabras claves	3
Marco conceptual y contextual	4
Tabla de figuras.....	5
Desarrollo del proyecto Bug Master	6
Alcances.	6
Riesgos.....	7
Tiempos.....	10
Desviaciones	11
Métricas	12
Oportunidades de mejora	13
Lecciones aprendidas	14
Conclusiones	16
Referencias.....	17

Resumen

El proyecto de prueba de software Bug Master se centra en cuatro historias de usuarios principales: revisión del producto, inicio de sesión, compra y contraseña.

El alcance real se debe en gran parte a limitaciones de la aplicación, como la falta de una base de datos persistente.

Se identificaron varios riesgos, incluidos retrasos en la carga de la página, problemas de seguridad con funciones clave y dificultad para interactuar con servicios de terceros.

Estos problemas se deben a factores como una mala optimización del código, la falta de medidas de seguridad estándar y una mala integración de las API externas.

El tiempo de ejecución de las pruebas se amplió de 30 a 35 horas debido al descubrimiento de más errores de los esperados, cambios en los requisitos y dificultades con el entorno de pruebas.

Las desviaciones importantes incluyen errores que no se corrigen con prontitud, equipos de control de calidad incompetentes y estimaciones inexactas.

En términos de métricas, el tiempo total de ejecución aumentó un 14%. De las 8 pruebas realizadas, 5 tuvieron éxito, 3 fallaron y un evento de bloqueo impidió realizar más pruebas.

La cobertura funcional alcanzó el 62,5%.

Se identificaron oportunidades de mejora a nivel de proceso, equipo e individual, incluidas revisiones de proyectos más exhaustivas, mejora de la documentación, implementación de pruebas automatizadas, mejora de la comunicación entre equipos y capacitación adicional en áreas específicas.

Las lecciones aprendidas resaltan la importancia de la seguridad, la atención al detalle, el conocimiento integral de las integraciones de terceros y la necesidad de una documentación clara.

A nivel de equipo, la atención se centra en mejorar la comunicación, revisiones de código más detalladas, pruebas de integración sólidas e implementar sistemas de monitoreo en tiempo real para procesos críticos.

En este texto debes resumir las ideas principales de tu trabajo de grado

Palabras clave

Pruebas, Seguridad, Riesgos, Desviaciones, Métricas

Marco conceptual y contextual

Bug Master es un proyecto de prueba de software realizado a una aplicación de comercio electrónico cuyo alcance se centró en probar cuatro historias de usuario principales relacionadas con la visualización de productos, inicio de sesión, proceso de compra y gestión de contraseñas.

En el proyecto utilizó un enfoque basado en pruebas de casos de uso o historias de usuario. Durante el proyecto hubo problemas con algunas funcionalidades del entorno de pruebas, lo que permitió que se utilizara un ambiente específico para las pruebas. En el desarrollo de la práctica se realizaron pruebas funcionales y de seguridad, y otros tipos de pruebas como pruebas de rendimiento y usabilidad y se utilizó una matriz de riesgos para identificar y evaluar los posibles problemas durante el proyecto. Para la elaboración del proyecto se emplearon métricas de tiempo, defectos, alcance y casos de prueba para evaluar el desempeño del proyecto; el equipo de QA, solo esta iniciando en el proceso de pruebas software lo que derivo que las pruebas solo se realizaron después de la fase de desarrollo, pero antes de la implementación final, se incluyen secciones sobre oportunidades de mejora y lecciones aprendidas, lo que indica un enfoque en la mejora continua del proceso de pruebas.

ÍNDICE DE FIGURAS

Figura	Descripción	Pag.
---------------	--------------------	-------------

Figura No. 1	UH-U-01 – Visualización de los productos	6
Figura No. 2	UH-U-04 – Inicio de sesión	6
Figura No. 3	UH-U-07 – Compra	6
Figura No. 4	UH-U-14 – Contraseña	6
Figura No. 5	Matriz de Riesgo	8
Figura No. 6	Calculadora de estimación	10
Figura No. 7	Desviaciones	11
Figura No. 8	Tiempo de pruebas	12
Figura No. 9	Impacto de los defectos	12
Figura No. 10	Pruebas de funcionalidad	13
Figura No. 11	Casos de prueba	13

**DESARROLLO DEL PROYECTO
BUG MASTER**

ALCANCES

Plan de pruebas

Alcance estimado

El alcance estimado dentro de nuestro plan de pruebas incluía las siguientes historias de usuario:

- UH-U-01 – Visualización de los productos

UH-U-01			
IDENTIFICADOR:	UH-U-01	TÍTULO:	Visualización de los productos
VALOR:	100	ESTIMACIÓN:	13
DEPENDENCIAS:	-	DESCRIPCIÓN:	Como cliente, quiero visualizar todos los productos sin la necesidad de registrarme para poder decidir si compro alguno o no.
CONFIRMACIONES:			
<ul style="list-style-type: none"> ~ Los productos se deben mostrar por sexo de la persona y por el tipo de artículo. ~ Se mostrarán un total de 10 productos por página. ~ Se pondrá el nombre, el precio actual y el anterior si está en oferta. ~ Se deben añadir los botones de "anterior" y "siguiente" para buscar más productos. 			

Figura No. 1 - UH-U-01 – Visualización de los productos

- UH-U-04 – Inicio de sesión

UH-U-04			
IDENTIFICADOR:	UH-U-04	TÍTULO:	Inicio de sesión
VALOR:	100	ESTIMACIÓN:	2
DEPENDENCIAS:	UH-U-03	DESCRIPCIÓN:	Como cliente, quiero iniciar o cerrar sesión, para poder acceder o salir del sistema respectivamente.
CONFIRMACIONES:			
<ul style="list-style-type: none"> ~ Al iniciar sesión, se conservarán los datos y se redirigirá a la página principal. ~ Al cerrar sesión, los datos de esa sesión se eliminarán. 			

Figura No.2 - UH-U-04 – Inicio de sesión

- UH-U-07 – Compra

UH-U-07			
IDENTIFICADOR:	UH-U-07	TÍTULO:	Compra
VALOR:	100	ESTIMACIÓN:	8
DEPENDENCIAS:	UH-U-03, UH-U-04 y UH-U-06	DESCRIPCIÓN:	Como cliente, quiero verificar mis datos de compra para poder quedarme tranquilo de que la compra se realiza correctamente.
CONFIRMACIONES:			
<ul style="list-style-type: none"> ~ Se puede verificar los datos personales y de envío. ~ Se puede añadir vales descuentos. 			

Figura No. 3 - UH-U-07 – Compra

- UH-U-14 – Contraseña

UH-U-14			
IDENTIFICADOR:	UH-U-14	TÍTULO:	Contraseña
VALOR:	90	ESTIMACIÓN:	5
DEPENDENCIAS:	-	DESCRIPCIÓN:	Como cliente, quiero recuperar mi contraseña en caso de olvido para poder iniciar sesión y garantizar la seguridad de los datos.
CONFIRMACIONES:			
<ul style="list-style-type: none"> ~ Se enviará una nueva contraseña al cliente y luego él la podrá modificar. 			

Figura No. 4 - UH-U-14 – Contraseña

Dentro del alcance estimado no se tuvieron en cuenta las siguientes historias de usuario.

UH-U-02 – Visualizar un producto, UH-U-03 Registro, UH-U-05 – Suscripción a un producto, UH-U-06 – Carrito, UH-U-08 –Confirmación de compra , UH-U-09 – Resumen, UH-U-10 – Mail, UH-U-11- Ver Perfil, UH-U-12 – Editar Perfil, UH-U-13 - Histórico

Alcance real.

Las historias de usuario estimadas para pruebas en el alcance iniciar se probaron de manera parcial.

- UH-U-01 – Visualización de los productos.
Se visualizo un total de 8 productos en la página inicial con botón de siguiente y la descripción correcta de cada producto.
- UH-U-04 – Inicio de sesión
Se registro correctamente un cliente dentro de la plataforma y al cerrar sesión los datos del cliente fueron eliminados del aplicativo.
- UH-U-07 – Compra
El aplicativo permitió agregar productos al carrito y redireccionarme a la ventana de pagos, actualizando de manera correcta los subtotales de los productos.
- UH-U-14 – Contraseña
El usuario registrado logro hacer cambio de contraseña dentro de una sesión iniciada.

Sustentación de diferencias.

Al momento de iniciar el proceso de pruebas con las historias de usuarios relacionadas, y con los casos de prueba delimitados por el alcance, se noto que el aplicativo no contaba con una base de datos que guardara la información que se registraba en tiempo real, esto ocasionaba que a la cerrar el navegador o apagar el dispositivo, lo avances que se tenían se perdían por completo.

El aplicativo olvidaba las credenciales de inicio de sesión cuando se cerraba el navegador, las pruebas de ingreso con credenciales invalidad quedan sin realizarse completamente.

El proceso de compra no se logro finalizar correctamente, inconvenientes con el envío de evidencias al correo electrónico y la selección de países dentro del registro de la compra.

Al no guardar los datos registrados de la sesión, quedó imposible probar el restablecimiento de la contraseña por olvido.

RIESGOS

Matriz de riesgo

MATRIZ DE RIESGOS					
RIESGO	Probabilidad (Ocurrencia)	Impacto (Gravedad)	Valor del Riesgo	Nivel de Riesgo	PLAN CONTINGENCIA
La probabilidad de que falle la plataforma de ecommerce la cual causaria que la aplicación web experimentar problemas técnicos que afecten la experiencia del usuario.	4	5	20	Muy grave	Mantener actualizaciones regulares del sistema, realizar pruebas de carga y tener soporte técnico disponible 24/7
La probabilidad de que la aplicación tenga problemas de usabilidad con Interfaz de usuario confusa o poco intuitiva que reduce las	3	3	9	Importante	Realizar pruebas de usabilidad, Implementar diseño responsivo, Recopilar y analizar feedback

Figura No. 5 – Matriz de Riesgo

Riesgos que se materializaron.

- Dentro del aplicativo se materializaron fallas en la plataforma al presentar demoras en el cargue la página principal.
- Se materializaron problemas de seguridad en diferentes funcionalidades del aplicativo, en los cuales podemos mencionar inicio de sesión, cambios de contraseña y proceso de compras.
- Se materializo el riesgo de problemas con la interacción del aplicativo con terceros, donde podemos destacar, inconvenientes con los servicios de correo electrónico y pasarela de pagos.

¿Qué ocasiono que se materializaran?

Demora en el cargue de la página principal:

- Optimización deficiente del código front-end
- Sobrecarga del servidor
- Consultas de base de datos ineficientes
- Carga excesiva de recursos (imágenes, scripts, etc.)
- Caché mal configurado

Inconvenientes con el inicio de sesión, cambios de contraseña y proceso de compras.

- Falta de implementación de prácticas de seguridad estándar
- Validación insuficiente de entradas de usuario
- Encriptación débil o ausente para datos sensibles
- Manejo inadecuado de sesiones

- Falta de pruebas de penetración y seguridad
- Configuraciones incorrectas en servidores o frameworks

Interacción del aplicativo con terceros.

- Integración incorrecta de APIs externas
- Falta de manejo de errores y excepciones en las comunicaciones
- Problemas de compatibilidad entre versiones de servicios
- Configuración incorrecta de firewalls o reglas de red
- Falta de pruebas end-to-end con los servicios reales
- Dependencia excesiva de servicios externos sin planes de contingencia

TIEMPOS

CALCULADORA DE ESTIMACIÓN				
Número de Horas a trabajar por día:	8,00	<<Por defecto esta en 8 horas, se modifica de acuerdo a lo pactado en el proyecto>>		
% Desviación	28%	<<Este valor es calculado de la hoja % Desviación>>		
<p>Nota: Si es necesario insertar celdas para adicionar actividades por favor tenga en cuenta que tiene que copiar las fórmulas y verificar que los totales incluyan todas las actividades.</p>				
Actividad	Total Horas Estimadas	Horas con % Desviación	Total Días Estimados	Total Días con % Desviación
Contextualización	5,00	6,40	0,625	0,8
Revisión de la documentación del proyecto	1,50	1,92	0,19	0,24
Análisis de requisitos e historias de usuario	2,00	2,56	0,25	0,32
Reuniones con el equipo de desarrollo para aclarar dudas	1,00	1,28	0,13	0,16
Familiarización con el entorno de pruebas	0,50	0,64	0,06	0,08
Planeación	7,50	9,60	0,9375	1,2
Definición de la estrategia de pruebas	1,50	1,92	0,19	0,24
Identificación y análisis de riesgos	1,50	1,92	0,19	0,24
Creación del plan de pruebas	2,00	2,56	0,25	0,32
Definición de casos de prueba de alto nivel	1,50	1,92	0,19	0,24
Estimación de recursos y cronograma	0,50	0,64	0,06	0,08
Revisión y aprobación del plan de pruebas	0,50	0,64	0,06	0,08
Diseño/ Ejecución	12,00	15,36	1,5	1,92
Diseño detallado de casos de prueba	3,00	3,84	0,38	0,48
Preparación de datos de prueba	1,00	1,28	0,13	0,16
Configuración del entorno de pruebas	1,00	1,28	0,13	0,16
Ejecución de pruebas funcionales	4,00	5,12	0,50	0,64
Ejecución de pruebas de rendimiento	1,50	1,92	0,19	0,24
Ejecución de pruebas de seguridad	1,50	1,92	0,19	0,24
Ejecución de pruebas de usabilidad	1,00	1,28	0,13	0,16
Reporte y seguimiento de defectos	1,00	1,28	0,13	0,16
Documentación, cierre y retrospectiva	5,50	7,04	0,6875	0,88
Documentación de resultados de pruebas	1,50	1,92	0,19	0,24
Análisis final de resultados	1,00	1,28	0,13	0,16
Preparación del informe final de pruebas	1,00	1,28	0,13	0,16
Reunión de cierre con el equipo del proyecto	1,00	1,28	0,13	0,16
Sesión de retrospectiva	0,50	0,64	0,06	0,08
Documentación de lecciones aprendidas	0,50	0,64	0,06	0,08
Total Tiempo Estimado de Pruebas	30,00	38,40	3,75	4,80

Figura No. 6 – Calculadora de estimación

Comparación tiempos estimados vs tiempos reales.

El tiempo estimado de desarrollo de la prueba se ha incrementado de 30 a 35 horas por los siguientes motivos:

- **Descubrimiento de más errores:** encontramos más errores de los esperados lo que requirió más tiempo para corregirlos y volver a realizar pruebas.
- **Cambios de requisitos:** los cambios de último momento en las especificaciones pueden aumentar el alcance de las pruebas.
- **Problemas con el entorno de pruebas:** las dificultades en el entorno de prueba o configuración lograron causar retrasos.
- **Documentación incompleta:** Especificaciones poco claras o incompletas permitieron ralentizar el proceso de pruebas.
- **Inexperiencia del equipo:** La falta de familiaridad con el sistema y las técnicas de prueba lograron aumentar el tiempo necesario.

- **Planificación optimista:** subestimamos el tiempo necesario para las pruebas en la planificación inicial.

Desviaciones

PORCENTAJE DE DESVIACIÓN POR IMPEDIMENTOS		
Impedimento	%	Justificación %
Ambiente NO disponible	2%	La falta de un entorno de pruebas puede detener por completo el proceso de pruebas, provocando retrasos importantes.
Data no disponible	2%	Pueden producirse retrasos, pero a menudo se pueden obtener datos de prueba alternativos.
Bugs sin resolución oportuna	6%	Esto puede bloquear muchos casos de prueba y requerir ciclos de prueba adicionales.
Disponibilidad de las personas	4%	Esto puede ralentizar el progreso, pero normalmente se puede reprogramar el trabajo.
Falta de capacidad y disponibilidad por parte de QA (Profesionales de Calidad)	4%	Afecta directamente la velocidad y la calidad de las pruebas.
Mala estimación y planeación	3%	Esto puede causar estrés innecesario y comprometer la calidad.
Cambios constantes en el alcance	3%	Es posible que se requiera un rediseño importante y trabajo adicional.
Desconocimiento del negocio	4%	Esto puede provocar malentendidos y errores en las pruebas.
Total % Desviación (Es el total que estimamos se puede desviar el proyecto, según experiencias o históricos), este porcentaje debe disminuir de proyecto a proyecto, para cumplir la promesa de disminuir el desperdicio y aumentar la disponibilidad.	28%	

Desviaciones que se activaron:

- **Bugs sin resolución oportuna:** al momento de iniciar con el proceso de pruebas se presentaron bugs bloqueantes que no permitían seguir con el proceso de pruebas hasta que fueran solucionados.
- **Falta de capacidad y disponibilidad por parte de QA:** al ser un equipo que solo este iniciado en el proceso de pruebas de software, se pierde esa experiencia y manejo de errores que se gana con el tiempo de trabajo.
- **Mala estimación y planeación:** al momento de diseñar los escenarios de casos de pruebas falta información para estimar y planear de mejor manera estos casos.
- **Desconocimiento del negocio:** ciertas fallas de conocimiento sobre el negocio permitieron que la desviación se activara. Al no solicitar toda la información desde un inicio, provocó conceptos equivocados dentro de los casos de pruebas.

METRICAS

- Métricas en tiempo:
Tiempo total de la ejecución de las pruebas.

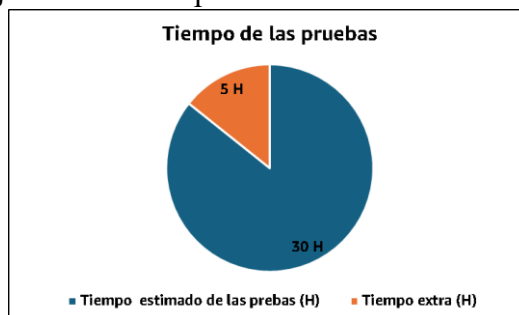


Figura No. 8 – Tiempo de pruebas

Se tuvo un incremento del 14% en el total de horas. Los errores de estimación y planeación lograron subir 5 horas en el tiempo real de las pruebas.

- Métricas de defectos
Impacto de los defectos encontrados.

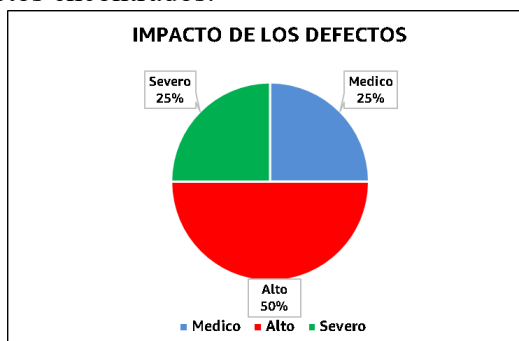


Figura No. 9 – Impacto de los defectos

El caso de prueba *Intento de inicio de sesión con credenciales inválidas*, ocasiono un impacto severo bloqueante que ocasiono inconvenientes para seguir probando los demás casos.

- Métricas del alcance
Cobertura de funcionalidades
De las funcionalidades del aplicativo: Catálogo de productos, Carrito de compras, Proceso de pago seguro, Registro y cuentas de usuario, Búsqueda y filtros de productos, Gestión de inventario, Seguimiento de pedidos y Servicio al cliente.

Se lograron probar las siguientes funcionalidades:

Catálogo de productos, Carrito de compras, Proceso de pago seguro, cuentas de usuario y Búsqueda y filtros de productos

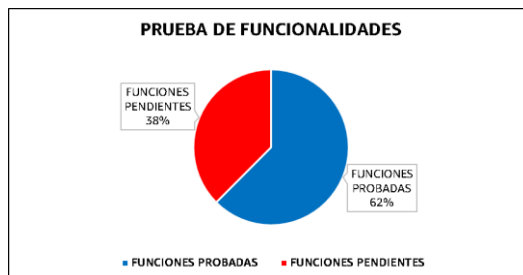


Figura No. 10 – Pruebas de funcionalidad

- Métricas de los casos de prueba.
Porcentaje de casos de prueba realizado con éxito y casos de pruebas fallidos.

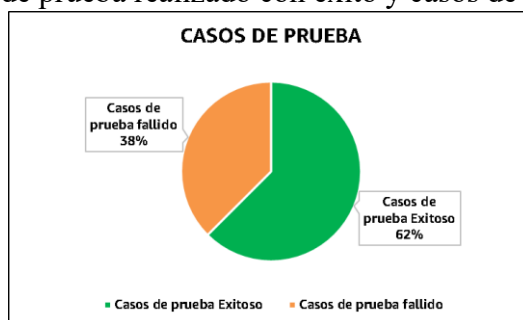


Figura No. 11 – Casos de prueba

De los 8 casos de pruebas realizados 5 fueron exitosos mientras que tres fueron fallidos. El primer caso de prueba fallido fue bloqueante impidiendo la secuencia en los casos siguientes.

OPORTUNIDADES DE MEJORA

A nivel de proceso:

- Realizar revisiones de diseño más rigurosas antes de la fase de desarrollo.
- Mejorar la documentación de requisitos y casos de prueba.
- Cree un proceso de prueba más completo, que incluya pruebas de integración y pruebas de un extremo a otro.
- Pruebas automatizadas de características importantes.
- Realice pruebas de carga y estrés para garantizar el rendimiento en diversas condiciones.
- Establecer procesos de control de calidad más estrictos.

A nivel de equipo:

- Mejore la comunicación entre desarrolladores, evaluadores y analistas de negocios.
- Realizar sesiones de capacitación conjuntas para que todos puedan comprender mejor las diferentes áreas del sistema.
- Implemente revisiones de código de igual a igual para detectar problemas lo antes posible.
- Fomentar una cultura de calidad dentro del equipo.
- Realizar retrospectivas periódicas para identificar y resolver problemas recurrentes.

A nivel individual:

- Proporcionar formación adicional en áreas específicas (como seguridad de pagos inicio de sesiones y conexiones con terceros).
- Fomentar la iniciativa para identificar y reportar problemas potenciales.
- Fomentar el aprendizaje continuo sobre las mejores prácticas de comercio electrónico.
- Promover la atención al detalle y el pensamiento crítico en todas las fases de desarrollo y prueba.
- Animar a cada miembro del equipo a sugerir mejoras al proceso en función de su experiencia.
- Cree métricas de calidad personales y proporcione comentarios periódicos.

LECCIONES APRENDIDAS

A nivel individual:

- Importancia de la seguridad: Cada desarrollador y evaluador debe profundizar sus conocimientos de seguridad, especialmente en las áreas de autenticación y datos confidenciales.
- Presta atención a los detalles: Se debe prestar especial atención a cada paso de los procesos más importantes; no dar nada por sentado
- Control integral: Es importante examinar más allá de los casos “felices” para abarcar una amplia gama de situaciones y obstáculos.
- Conocimiento integrado: Todos los involucrados deben comprender las complejidades de las integraciones y pasarelas de pago de terceros.
- Documentos claros: Es importante documentar en detalle los procesos y errores descubiertos para que sea más fácil corregirlos y prevenirlos en el futuro.

A nivel de equipo:

- Mejorar la comunicación: Es necesario que haya una comunicación más fluida entre desarrolladores, evaluadores y equipos de operaciones, especialmente en cuestiones de integración.

- Descripción general del código de emparejamiento: Realice revisiones de código más detalladas, especialmente para funciones importantes.
- Potentes pruebas de integración: Establecer un proceso de prueba de integración más sólido, especialmente para conexiones de terceros.
- Entorno de prueba real: Asegúrese de que su entorno de prueba refleje su entorno de producción lo más fielmente posible.
- Manejo de errores secuenciales: Desarrolle un enfoque coherente para el manejo de errores y el registro en su aplicación.
- Control de seguridad: hacer de las pruebas de seguridad una parte integral del proceso de desarrollo y prueba.
- Monitoreo activo: Implementar un sistema de monitoreo para detectar problemas en tiempo real, especialmente en procesos críticos como pagos.
- Gestión del cambio: optimice su proceso de gestión de cambios para comprender mejor el impacto de los cambios en las funciones críticas.
- Formación continua: Regularmente educar al equipo sobre las últimas tendencias en seguridad, autenticación y procesamiento de pagos.
- Plan de emergencia: Desarrollar y probar planes de contingencia para escenario críticos de falla funcional.

Conclusiones

El proyecto Bug Master proporcionó valiosas lecciones sobre la importancia de las pruebas integrales de software en aplicaciones de comercio electrónico.

Al evaluar cuatro historias de usuarios clave, se identificaron áreas importantes que requieren atención, como la seguridad de la autenticación, la gestión de datos confidenciales y la integración con servicios de terceros.

Las brechas de tiempo y alcance resaltan la necesidad de mejorar la planificación y estimación del proyecto, así como la importancia de contar con un equipo de control de calidad experimentado.

Los riesgos físicos, especialmente relacionados con el rendimiento y la seguridad, resaltan la necesidad de implementar prácticas de desarrollo más sólidas y realizar pruebas más completas.

Las métricas recopiladas mostraron una cobertura funcional del 62,5%, lo que demuestra que aún hay margen de mejora en la cobertura de las pruebas.

La tasa de éxito de los casos de prueba (5 de 8) muestra que, aunque se encontraron problemas graves, también se validó la funcionalidad principal.

Las oportunidades de mejora identificadas a nivel de proceso, equipo e individuo proporcionan una hoja de ruta clara para mejorar futuros proyectos piloto.

Esto incluye realizar revisiones de diseño más rigurosas, mejorar la documentación, automatizar las pruebas y fomentar una cultura de calidad dentro del equipo.

Las lecciones aprendidas resaltan la importancia de la seguridad, la atención al detalle y la necesidad de una comprensión integral de las integraciones de terceros.

También se enfatizó la necesidad de mejorar la comunicación entre equipos y establecer un seguimiento en tiempo real de las funciones críticas.

En conclusión, el proyecto Bug Master ha demostrado ser una valiosa experiencia de aprendizaje, que proporciona información importante para mejorar la calidad y la seguridad en el desarrollo de aplicaciones de comercio electrónico.

Implementar las lecciones aprendidas e identificar oportunidades de mejora será esencial para elevar los estándares de calidad en futuros proyectos de software.

Referencias

Montoya, E. S. (2013). *Prueba funcional del software: un proceso de verificación constante*. Instituto Tecnológico Metropolitano.

Ramos, J. (s/f). Los diferentes tipos de Pruebas de software. Programacionymas.com. Recuperado el 17 de agosto de 2024, de <https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software>

Quality-Stream [@QualityStream]. (s/f). 7 Principios de las Pruebas de Software | ISTQB examinable. Youtube. Recuperado el 17 de agosto de 2024, de <https://www.youtube.com/watch?app=desktop&v=nI28eoU6POQ>

Testing: Convertite en un experto en QA. (2017). Natsys.

Ramirez, O. A. A. (2018). *Cómo Ser Un Tester: Introducción a Las Pruebas de Software*. Independently Published.