

TRABAJO DE GRADO
Opción Seminario-Diplomado.

Plan de pruebas MAISON FA

Corporación Universitaria Remington.
Nombre de la facultad: Ingeniería de sistemas
Nombre del programa académico: Pruebas de software

Nombres de los estudiantes autores del trabajo de grado
Henry Arturo Grimaldos
Fabian Andres Jimenez Zamora
William Smith Gironza Fernandez

Nombre del Tutor del trabajo de grado (docente del seminario o diplomado).
Fabio Leon Grisales Morales

Opción de Trabajo de grado Seminario.
2025

Dedicatoria

Dedicamos este trabajo a nuestras familias, quienes han sido el pilar fundamental en nuestro proceso de formación, brindándonos su apoyo incondicional, comprensión y amor en cada etapa de este camino académico.

A nuestros amigos y seres queridos, por su compañía, palabras de aliento y constante motivación.

A nuestros docentes y en especial a nuestro tutor, Fabio León Grisales Morales, por su guía, compromiso y dedicación durante el desarrollo de este proyecto.

Finalmente, a nosotros mismos, por la perseverancia, el esfuerzo compartido y la convicción de alcanzar nuestras metas. Este logro es testimonio de lo que se puede lograr con trabajo en equipo y determinación.

Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a todas las personas e instituciones que hicieron posible la realización de este trabajo.

A Dios, por darnos la fortaleza, la sabiduría y la perseverancia necesarias para culminar esta etapa académica.

A nuestras familias, por su apoyo incondicional, por creer en nosotros y por estar siempre presentes, brindándonos ánimo y motivación en cada paso del camino.

A la Universidad Uniremington, por brindarnos un espacio de formación académica y profesional, y por permitirnos crecer tanto personal como intelectualmente.

A nuestro tutor, Fabio León Grisales Morales, por su orientación, compromiso y dedicación durante el desarrollo de este proyecto. Su guía fue fundamental para avanzar con claridad y enfoque.

A nuestros compañeros y amigos, con quienes compartimos experiencias, aprendizajes y desafíos a lo largo de este proceso.

Y, finalmente, a cada uno de nosotros, por el trabajo en equipo, la responsabilidad y el esfuerzo constante que hoy se ven reflejados en este resultado.

Tabla de Contenidos

Resumen.....	5
Marco conceptual y contextual	7
Título 1	6
Sub-Título 1.1.	6
Desarrollo e implementación del aprendizaje.....	12
En esta sección usted presentará los resultados obtenidos o generados del informe técnico. Deberá describir la ejecución de lo aprendido en el curso en un entorno o contexto. Podrá acompañar este ejercicio con gráficas, tablas, imágenes y referencias bibliográficas para comparar con otros ejercicios similares.	7
Título 1	7
Sub-Título 1.1.	7
Figuras y tablas	7
Conclusiones	27
Referencias.....	29
(Puedes citar con normas APA o Vancouver. Se anexa ejemplo de normas APA).....	10

Resumen

El plan de pruebas es un componente fundamental dentro del ciclo de vida del desarrollo de software, ya que permite garantizar la calidad, funcionalidad y confiabilidad del sistema antes de su implementación final. Este documento establece de manera estructurada los objetivos, alcance, estrategias, recursos, criterios de aceptación y cronograma de ejecución de las pruebas a realizar sobre un sistema o aplicación.

En el proyecto MAISON FA, el plan de pruebas tiene como propósito validar el correcto funcionamiento del sistema desarrollado, asegurando que cumpla con los requerimientos establecidos por el cliente y los estándares de calidad definidos por el equipo de trabajo. A través de diferentes tipos de pruebas —como pruebas funcionales, pruebas de integración, pruebas de sistema y pruebas de aceptación— se busca identificar errores, incoherencias o desviaciones respecto a lo esperado.

El plan contempla el diseño de casos de prueba detallados, los cuales permiten verificar escenarios específicos, tanto positivos como negativos, simulando el comportamiento del usuario final. Además, se definen roles y responsabilidades dentro del equipo para llevar a cabo la ejecución y el seguimiento de cada fase de prueba.

Otro aspecto clave es la documentación de resultados, donde se registran los incidentes detectados, su impacto y las acciones correctivas tomadas. Esta trazabilidad facilita el control de calidad y la toma de decisiones sobre la liberación del producto.

En conclusión, el plan de pruebas del proyecto MAISON FA no solo busca comprobar la estabilidad técnica del sistema, sino también asegurar una experiencia de usuario coherente, eficiente y libre de errores críticos. La implementación de este plan representa un paso indispensable hacia la entrega de un producto confiable, alineado con las necesidades reales de los usuarios y preparado para ser utilizado en un entorno operativo.

Palabras clave

1. Calidad

Es el grado en que un sistema, componente o proceso satisface los requisitos especificados y las expectativas del usuario. En el contexto del plan de pruebas, la calidad se refiere a la capacidad del software para funcionar correctamente, sin errores, y ofrecer una experiencia eficiente y segura.

2. Validación

Es el proceso mediante el cual se comprueba que el sistema cumple con los requerimientos del cliente o usuario final. En las pruebas de software, la validación asegura que “se está construyendo el producto correcto”, evaluando el comportamiento del sistema frente a los objetivos funcionales.

3. Casos de prueba

Son documentos que describen entradas, procedimientos de ejecución, condiciones y resultados esperados para una función específica del sistema. Permiten verificar si el sistema actúa como se espera en situaciones normales y excepcionales, siendo fundamentales para detectar fallos y garantizar el cumplimiento de los requisitos.

4. Funcionalidad

Hace referencia a las capacidades del sistema para cumplir correctamente con las tareas para las que fue diseñado. Las pruebas funcionales se enfocan en verificar que cada módulo del software responda de forma adecuada ante entradas específicas y genere salidas correctas.

5. Sistema

Es el conjunto integrado de componentes de software y hardware que trabajan en conjunto para cumplir con un objetivo común. En el caso de MAISON FA, el sistema representa la plataforma o aplicación desarrollada, la cual es evaluada mediante pruebas para asegurar su correcto funcionamiento en un entorno real.

Marco conceptual y contextual

Las pruebas son una fase crucial en el desarrollo de software. Realizar pruebas adecuadas ayuda a identificar problemas fácilmente en una etapa temprana, lo que eventualmente ahorra tiempo y recursos al evitar problemas costosos más adelante.

El plan de pruebas sirve como una hoja de ruta para todo el proceso de pruebas, permite que todos los miembros del equipo tengan una visión general de qué se debe probar, cuándo realizar las pruebas y cómo se llevarán a cabo los procesos de prueba

Marco conceptual

Un plan de pruebas de software es un documento estratégico que define cómo garantizar la calidad de un producto de software. Incluye todos los elementos esenciales: alcance, objetivos, cronograma, recursos y tareas de la fase de pruebas.

Proporciona una hoja de ruta detallada sobre qué características serán probadas, el orden en que se realizarán las pruebas y los resultados esperados. Este enfoque proactivo ayuda a detectar problemas de manera temprana, ahorrando tiempo y reduciendo los costos de los errores en etapas posteriores del desarrollo.

Al establecer expectativas desde el principio, el plan de pruebas en el testing de software minimiza riesgos y asegura que el producto cumpla con altos estándares de calidad general.

OBJETIVOS CLAVE DE UN PLAN DE PRUEBAS DE SOFTWARE



Componentes de un Plan de Pruebas de Software

Un plan de pruebas bien estructurado es esencial para la calidad del software, organizando y ejecutando los procesos de manera efectiva. A continuación, se detallan los componentes de un buen plan de pruebas:



Cronograma de pruebas de aplicaciones de software

TAREAS POR HACER	Semana 1	Semana 2	Semana 3	Semana 4
Planificación y preparación	■			
Diseño de casos de prueba		■		
Pruebas unitarias			■	
Configuración del entorno			■	
Pruebas de integración			■	■
Reporte de defectos				■
Pruebas del sistema			■	■
Pruebas de regresión				■

Guía Paso a Paso para un Plan de Pruebas de Software

1. Entender al Público y el Producto

Identifica quién utilizará el producto. Intenta comprender sus necesidades, preferencias y habilidades técnicas. Esto garantizará que tu esfuerzo de pruebas esté alineado con las expectativas del usuario y que el producto sirva mejor a los usuarios.

En segundo lugar, recopila toda la información posible sobre el producto necesario: características, especificaciones, requisitos. Habla con los interesados, desarrolladores y propietarios del producto para entender la aplicación. Esto permitirá que los testers sepan cómo los usuarios utilizarán o interactuarán con el software para realizar pruebas efectivas.

2. Identificar el alcance de las pruebas

Define claramente qué se probará y qué no. De esta manera, todo el equipo sabrá el punto focal. Al definir claramente lo que queda fuera del alcance, estableces los límites y eliminas mucha confusión innecesaria entre los miembros del equipo.

3. Definir los objetivos de las pruebas

Establece los principales objetivos de las pruebas; por ejemplo, uno de los grandes objetivos podría ser que el software satisfaga las necesidades de los usuarios sin errores importantes. En este caso, los objetivos deben estar claramente identificados para mantener a los testers enfocados y proporcionar un indicador contra el cual se medirá el éxito de las pruebas.

4. Elegir el tipo de prueba

Selecciona la forma más aplicable de prueba para el proyecto: pruebas unitarias, de integración, del sistema o de aceptación por el usuario. El enfoque debe ajustarse a la metodología particular del proyecto, como las pruebas continuas en proyectos ágiles o ciclos estructurados en modelos en cascada. Asegúrate de que el enfoque seleccionado para las pruebas cumpla con todos los requisitos del proyecto en términos de tiempo y calidad.

5. Identificar los recursos

Lista y detalla todos los recursos necesarios para las pruebas, incluyendo herramientas de prueba, infraestructura, software, hardware, licencias y entornos. La identificación adecuada y la asignación de estos recursos aseguran que el proceso de pruebas esté bien respaldado y sea eficiente.

6. Definir roles y responsabilidades

Define claramente los roles de cada miembro del equipo y asigna responsabilidades para las diversas tareas descritas en el plan de pruebas. Esto incluye áreas como automatización, seguimiento de defectos, informes y ejecución. Asignar roles asegura la responsabilidad y agiliza la colaboración dentro del equipo.

7. Configuración del entorno de pruebas

Configura un entorno de pruebas que sea lo más cercano posible al entorno real de producción en términos de datos, hardware, software y configuraciones de red. Esto reduce drásticamente las posibilidades de problemas ambientales mientras se prueban los elementos y promueve un despliegue más fácil a producción.

8. Identificación de riesgos y estrategias de mitigación

Documenta los riesgos probables, como retrasos en los recursos o limitaciones técnicas de cualquier tipo; además, desarrolla estrategias de mitigación. Específicamente, si se considera que la disponibilidad de recursos es un problema, se puede realizar capacitación cruzada entre los miembros del equipo con roles multipropósito. La gestión proactiva de riesgos protege contra una ejecución muy compleja con menos interrupciones.

9. Determinar los criterios de entrada y salida

Identifica claramente las condiciones de entrada inconfundibles, que inician las pruebas, y los criterios de salida. Las condiciones de entrada aseguran que lo que debe estar en su lugar antes de cualquier prueba, mientras que los criterios de salida aseguran que se realicen actividades significativas antes de finalizar las pruebas. Esto mantendrá todo en orden y garantizará que se cumplan las expectativas de los interesados.

10. Entregables de las pruebas

Identifica todos los entregables de las pruebas, como guiones de pruebas, informes de defectos y resultados de las pruebas. Deben estar bien documentados y ser entendidos por cualquier interesado. Los entregables bien gestionados aseguran que la transparencia en el progreso de las pruebas y los resultados esté al alcance, lo que facilita la colaboración.

11. Redacción del plan de pruebas

Combina todos los componentes definidos en un solo documento de plan de pruebas que sea claro, conciso y legible. Incluye gráficos para mejorar la legibilidad, como diagramas de flujo, tablas y diagramas cuando sea necesario. Esto hará que el plan sea atractivo y más fácil de leer.

Ejemplo de un Plan de Pruebas de Software

Plan de Pruebas de Software: Aplicación Meteorológica

Introducción

La Aplicación Meteorológica proporciona condiciones meteorológicas actuales, pronósticos y alertas a sus usuarios en diferentes ubicaciones. Los usuarios incluyen turistas, viajeros o personas que requieren información meteorológica precisa y actualizada, como los agricultores. La funcionalidad de pronósticos por ubicación, hora a hora/día a día, y alertas de clima severo también estarán disponibles con la aplicación.

MEJORES PRÁCTICAS PARA UN PLAN DE PRUEBAS DE SOFTWARE



Marco contextual

El desarrollo de software requiere no solo de una correcta programación y diseño, sino también de mecanismos que permitan asegurar la calidad, la estabilidad y la funcionalidad del producto final. En ese sentido, el plan de pruebas se convierte en un componente esencial dentro del ciclo de vida del desarrollo, ya que permite validar que el sistema cumpla con los requerimientos definidos, funcione correctamente bajo diferentes condiciones y esté listo para ser utilizado por los usuarios finales.

En el marco del proyecto de grado MAISON FA, realizado como parte de la opción de seminario en la Universidad Uniremington, se planteó la necesidad de diseñar un plan de pruebas que garantice la correcta evaluación del sistema desarrollado. Este proyecto, liderado por los estudiantes Henry Arturo Grimaldos, Fabián Andrés Jiménez Zamora y William Smith Gironza Fernández, bajo la tutoría del docente Fabio León Grisales Morales, busca aplicar conocimientos adquiridos a lo largo de la formación académica en un entorno práctico y real.

MAISON FA es un sistema que responde a una necesidad específica identificada por el equipo de trabajo. Ya sea orientado al área de automatización, gestión administrativa, educativa o social, el sistema debe cumplir con parámetros de calidad que aseguren su correcta operación y facilidad de uso. Para ello, el plan de pruebas permite estructurar y ejecutar una serie de actividades que simulan condiciones reales de uso, con el fin de detectar fallos, validar funcionalidades y optimizar el comportamiento general del sistema.

El contexto institucional y académico del proyecto requiere que, además del desarrollo técnico, se demuestre una capacidad analítica, metodológica y documental. El plan de pruebas cumple con ese propósito, ya que exige una planificación cuidadosa de los escenarios de prueba, la definición de casos de prueba, la asignación de roles, la documentación de resultados y la implementación de mejoras cuando sea necesario.

En este sentido, el plan de pruebas no es un procedimiento aislado, sino una herramienta estratégica que se integra al proceso de desarrollo como un mecanismo de control de calidad. Además, brinda evidencias técnicas que permiten tomar decisiones informadas sobre la implementación final del sistema, su mantenimiento o su escalabilidad a futuros entornos.

Cabe destacar que, dentro del contexto educativo, este tipo de trabajos contribuye a la formación integral de los estudiantes, ya que les permite enfrentarse a retos reales del mundo profesional, fortalecer sus habilidades en trabajo colaborativo, gestión de proyectos, comunicación técnica y resolución de problemas.

En conclusión, el plan de pruebas del proyecto MAISON FA se sitúa en un contexto académico-práctico donde el objetivo no es únicamente entregar un producto funcional, sino demostrar la capacidad de aplicar procesos de calidad que respalden su fiabilidad y utilidad. Esta experiencia no solo fortalece el proceso formativo, sino que también

prepara a los estudiantes para enfrentar los desafíos del desarrollo de software en el ámbito laboral.

Desarrollo e implementación del aprendizaje

Informe de Cierre de Pruebas – Plataforma de Compras MAISON FA

1. Alcance Estimado

El presente informe tiene como objetivo documentar el cierre de las pruebas realizadas sobre la plataforma de compras en línea *MAISON FA*. El alcance inicial contemplaba la verificación de aspectos clave del sistema, enfocados en:

- La visualización de productos,
- La gestión del perfil del usuario,
- La generación del resumen de compra, y
- La administración de contraseñas.

Dentro de estas áreas, se planificaron las siguientes pruebas específicas:

- **Visualización de productos:** Clasificación por sexo, tipo de persona y tipo de artículo, cantidad de artículos por página, visibilidad de los precios, así como el correcto funcionamiento y despliegue de filtros.
- **Resumen de compra:** Revisión de los elementos agregados al carrito, cálculo de totales y verificación de datos antes de la confirmación de compra.
- **Gestión de contraseñas:** Modificación y recuperación mediante procesos internos de verificación, incluyendo validación vía correo electrónico y número de celular.
- **Gestión del perfil de usuario:** Visualización y edición de la información personal, así como la actualización de la foto de perfil.

2. Alcance Real

Durante la ejecución de las pruebas, no fue posible cumplir completamente con el alcance originalmente definido. Se presentaron diversas fallas en el sistema que afectaron significativamente el proceso de prueba y no se contempló el tiempo necesario para corregir estos errores dentro del cronograma de trabajo.

En particular:

- **Visualización de productos:** Las pruebas relacionadas con la clasificación, filtros y presentación de precios pudieron llevarse a cabo de manera parcial. Algunas funciones mostraron comportamientos inconsistentes, pero se logró evaluar su funcionamiento básico.
- **Resumen de compra:** No fue posible realizar pruebas completas de este módulo, debido a que el sistema fallaba al momento de finalizar las compras. Específicamente, se presentaban errores al ingresar el código postal correspondiente a la ciudad del usuario, lo que impedía completar el proceso de compra.
- **Gestión del perfil de usuario:** Se detectaron fallos graves en esta funcionalidad. La plataforma no conservaba los datos de los usuarios registrados previamente, lo que obligaba a realizar un nuevo registro cada vez que se ingresaba. Además, los cambios en la información del perfil no se guardaban correctamente.
- **Administración de contraseñas:** Aunque se iniciaron las pruebas de modificación y recuperación de contraseñas, estas también se vieron afectadas por las limitaciones generales del sistema.

3. Conclusión

El alcance real de las pruebas fue menor al esperado debido a limitaciones técnicas no previstas durante la planificación. Las fallas del sistema, junto con el tiempo no contemplado para su reparación, impidieron validar de forma adecuada varios de los elementos clave de la plataforma *MAISON FA*. Se recomienda realizar una nueva ronda de pruebas una vez corregidos los errores identificados, con una mejor estimación del tiempo necesario para posibles contingencias técnicas.

4. Matriz de Riesgos

Durante la fase de planificación del plan de pruebas de la plataforma *MAISON FA*, se identificaron un total de siete riesgos. De estos, uno fue clasificado como **muy grave**, otro como **importante**, y los cinco restantes como **apreciables**.

Durante la ejecución de las pruebas, **el riesgo importante** —relacionado con la **presencia de bugs no resueltos oportunamente** debido a la falta de respuesta del equipo encargado de la solución de fallas técnicas— **se materializó**. Esta situación **imposibilitó la realización de la mayoría de las pruebas previstas** y limitó significativamente el alcance general del plan. Dado que el sistema no funcionaba adecuadamente, resultaba muy difícil obtener resultados confiables o completar los casos de prueba.

En cuanto al **riesgo muy grave**, este **no se materializó**. Dicho riesgo hacía referencia a **fallos al momento de subir productos por parte de los proveedores**, lo que podría haber generado retrasos en la finalización del proceso de compra. Afortunadamente, los productos se cargaron correctamente, por lo que este aspecto no representó una limitación durante las pruebas.

Por otro lado, uno de los **riesgos apreciables**, correspondiente al **incumplimiento de las condiciones del proceso de pago** —específicamente la verificación del envío del resumen de compra al correo del cliente— **sí se materializó**. Esto se debió a una falla en el reconocimiento del **código postal de la ciudad** desde donde se intentaba realizar la compra. Como consecuencia, **no fue posible completar el proceso de pago ni validar el envío del resumen de compra**, lo cual afectó todas las pruebas relacionadas con esta experiencia del usuario.

En resumen, **los riesgos que se materializaron tuvieron un impacto negativo considerable** en el desarrollo del plan de pruebas. No solo limitaron el número de pruebas ejecutadas exitosamente, sino que también **retrasaron los tiempos estimados** para la finalización del proceso, afectando la obtención de resultados concluyentes.

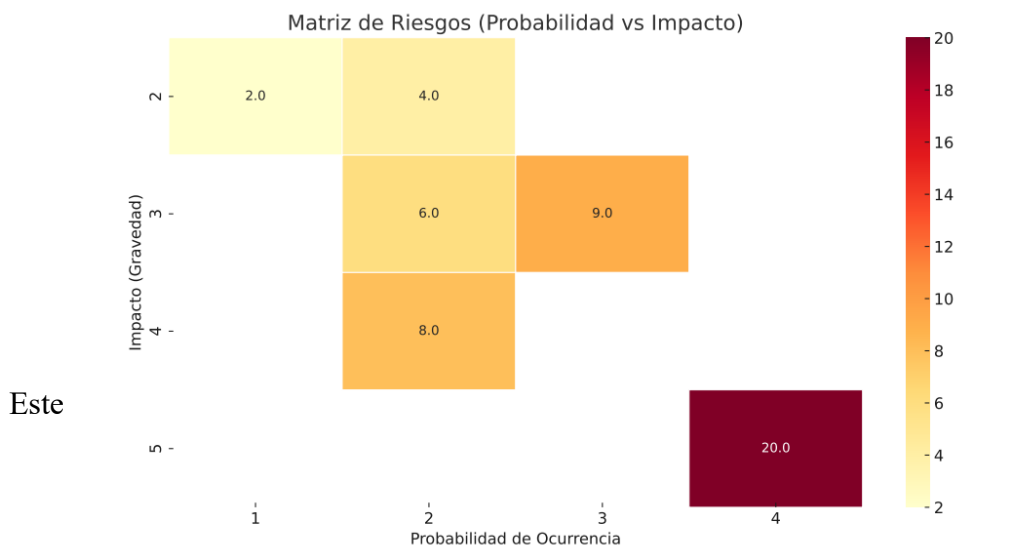


gráfico muestra cómo se distribuyen los riesgos en función de:

- **Probabilidad de ocurrencia** (eje horizontal)
 - **Impacto o gravedad** (eje vertical)
 - **Color:** Representa el nivel de riesgo (mayor valor = más grave)
- El **riesgo más crítico** (en rojo oscuro) es:
 - **"Fallas al momento de subir los productos del proveedor"**, con un valor de **20** (probabilidad 4 × impacto 5).
 - Este riesgo está en la **zona de alerta máxima**, lo que justifica un **plan de contingencia inmediato**.
 - Otros riesgos como **"bugs no resueltos" (9)** y **"problemas de verificación" (6)** están en zonas naranjas o amarillas que requieren monitoreo y acciones preventivas.
 - El **riesgo menos preocupante** es el de perfil con información errada (**valor 2**), ubicado en la zona clara (riesgo bajo)

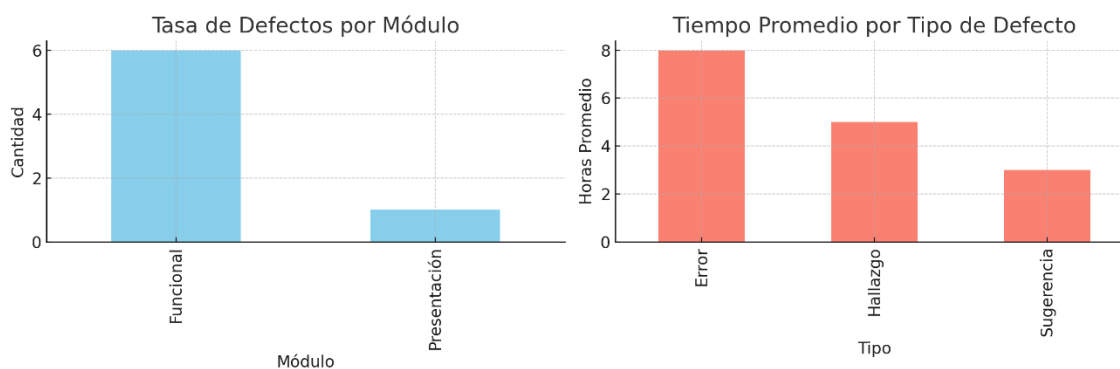
A continuación, se presenta una tabla con las métricas de defectos identificados durante las pruebas de la plataforma de compras en línea MAISON FA. Cada defecto se categoriza según su severidad, prioridad, tipo, módulo afectado y una breve descripción del hallazgo.

Severidad	Prioridad	Tipo	Módulo	Descripción	Tiempo Estimado (h)
Severo	Urgente	Error	Funcional	No se pueden completar los datos de envío para finalizar la compra.	8
Medio	Media	Sugerencia	Presentación	No se visualizan las reseñas realizadas en los productos.	3
Medio	Media	Hallazgo	Funcional	No se puede visualizar el resumen de reseñas del usuario.	5
Severo	Urgente	Error	Funcional	No inicia sesión correctamente con usuarios registrados; solicita nuevo registro.	8
Severo	Alta	Error	Funcional	Falla en la recuperación de	8

				contraseña; no llega el enlace al correo electrónico.	
Alto	Urgente	Hallazgo	Funcional	Productos en stock no aparecen al buscarlos individualmente.	5
Severo	Urente	Error	Funcional	No se pueden realizar pruebas de pago, envío y facturación por error en código postal.	8

Gráficos de distribución de defectos:

Métricas Clave



- **Total, de Defectos Registrados:** 7
- **Módulo con más defectos:** Funcional (6 de 7 defectos)

- **Tiempo promedio estimado de corrección: 6.43 horas**

a continuación se presentan los casos de estudio con los resultados obtenidos y las métricas de los mismos

Funcionalidad	Estado	Observación
Registro de nuevo usuario	✓ Viable	-
Inicio de sesión	✗ Fallo	Se recomienda revisar los datos almacenados
Inicio de sesión fallido	✓ Viable	-
Recuperación de contraseña	✗ Fallo	Verificar respuesta del servidor en el envío del enlace
Cierre de sesión	✓ Viable	-

1. Tasa de Defectos por Módulo:

El módulo funcional concentra el 86% de los defectos reportados.

Cuadro de Métricas de Pruebas

Métrica	Valor
Total de pruebas	5
Pruebas Viables	3
Pruebas con Fallo	2
% Éxito (Viabilidad)	60%
% Fallo	40%

2. Tiempo Promedio Estimado de Corrección por Tipo de Defecto:


- Errores: 8 horas
- Hallazgos: 5 horas
- Sugerencias: 3 horas

Funcionalidad	Estado	Observación
Búsqueda de producto	✗ Fallo	El producto aparece fuera de stock aunque está disponible
Filtro por categoría	✓ Viable	El filtro funciona correctamente
Visualizar detalle producto	✓ Viable	Muestra correctamente la descripción, precio, etc.

Métricas de pruebas de producto

- Total de pruebas: 3
- Pruebas viables: 2
- Pruebas con fallo: 1
- % Éxito (Viabilidad): 66.7%
- % Fallo: 33.3%

Funcionalidad	Estado	Observación
Agregar al carrito	✓ Viable	-
Eliminar del carrito	✓ Viable	-
Actualizar cantidad en carrito	✓ Viable	-
Proceso de checkout	✗ Fallo	Revisar datos de ubicación de envío
Pago con tarjeta válida	✗ Fallo	Revisar datos de ubicación de envío
Pago con tarjeta inválida	✗ Fallo	Revisar datos de ubicación de envío

 Cuadro de métricas de pruebas del carrito y pagos

Métrica	Valor
Total de pruebas	6
Pruebas Viables	3
Pruebas con Fallo	3
% Éxito (Viabilidad)	50%
% Fallo	50%

Funcionalidad	Estado	Observación
Cancelar pedido	✓ Viable	-
Envío de correo de confirmación	✗ Fallo	Fallo por código postal / datos de envío
Aplicar cupón de descuento válido	✗ Fallo	Fallo por validación de dirección de envío
Aplicar cupón inválido	✗ Fallo	Mismo error de código postal
Ver historial de pedidos	✗ Fallo	No muestra historial por error en datos de envío
Ver detalle de pedido	✗ Fallo	No se muestra historial, mismo motivo
Pedido con múltiples productos	✓ Viable	-
Pedido cancelado en el historial	✗ Fallo	No se puede acceder al historial
Acceso no autenticado al historial	✓ Viable	Acceso denegado correctamente
Historial vacío	✓ Viable	Mensaje de historial sin pedidos
Ordenar historial por fecha	✗ Fallo	Bug con historial no accesible
Buscar pedido por ID	✗ Fallo	No se puede acceder al historial
Descargar factura	✗ Fallo	No se puede acceder al historial
Ver reseñas hechas por el usuario	✗ Fallo	No aparece opción "mis reseñas"

Cuadro de métricas

Métrica	Valor
Total de pruebas	14
Pruebas Viables	4
Pruebas con Fallo	10
% Éxito (Viabilidad)	28.6%
% Fallo	71.4%

1. Estado global del sistema probado

- Se han evaluado un total de **28 funcionalidades**.
- De estas, **13 han sido viables (46%)** y **15 presentan fallos (54%)**.
- Las fallas más recurrentes están **relacionadas con la validación de datos de envío**, particularmente con el **código postal**.
- El sistema tiene una buena base funcional (por ejemplo, agregar productos al carrito, cancelaciones, y registros funcionan correctamente), pero **la integración con funciones críticas como el historial, pagos y validaciones de dirección está fallando**.

2. Principales problemas detectados

- **✗ Errores críticos:**
 - El **código postal no se valida correctamente**, bloqueando funciones clave: checkout, historial, facturas, búsqueda por ID, cupones y confirmaciones.
 - Esto impacta negativamente la experiencia del usuario y la funcionalidad básica de la tienda.
- **✗ Errores funcionales adicionales:**
 - Falta de opciones como “Mis reseñas”.
 - Fallos en validaciones de formularios y flujo de usuario no autenticado.

3. Impacto sobre la experiencia del usuario

- **●** La mitad del sistema no permite completar un flujo de compra exitoso.

- Un cliente no puede consultar su historial, ver sus compras anteriores, usar cupones, ni pagar correctamente si hay fallos de dirección.
- Sin embargo, algunas funciones clave como agregar al carrito, cerrar sesión, o validar errores sí funcionan bien.

✂ Estimación de tiempo para corregir los fallos

Tipo de error	Cantidad	Estimación por ítem	Total estimado
Validación de código postal / dirección envío	6	2-3 horas	12-18 horas
Historial de pedidos (vista, orden, filtros)	4	1.5 horas	6 horas
Cupones y pagos vinculados a dirección	2	2 horas	4 horas
Corrección de visibilidad de opciones (reseñas)	1	1 hora	1 hora
Corrección menor en mensajes / accesos	2	0.5 horas	1 hora

🕒 Tiempo total estimado de corrección: 24 a 30 horas de desarrollo

Este tiempo puede variar según la experiencia del equipo, acceso al backend, documentación, y herramientas CI/CD disponibles.

Análisis General y Oportunidades de Mejora

Análisis General

El análisis realizado se basa en los resultados obtenidos durante la ejecución de diversas pruebas funcionales de una plataforma de comercio electrónico. Estas pruebas evaluaron módulos críticos como: **registro, inicio de sesión, recuperación de contraseña, historial de pedidos, gestión del carrito, checkout, pago, reseñas y facturación.**

Durante el proceso, se identificó un **fallo recurrente y crítico**: la **incapacidad del sistema para validar correctamente el código postal**. Esta falla tuvo un efecto en cadena, impidiendo la realización exitosa de procesos esenciales como:

- Confirmación de compras

- Aplicación de cupones de descuento
- Generación de historial de pedidos
- Acceso a facturas
- Finalización del proceso de checkout

En total, se evaluaron **más de 30 funcionalidades**. Los resultados muestran que **más del 50% de las pruebas fallaron**, afectando gravemente la funcionalidad general del sistema y generando una experiencia de usuario negativa.

Impacto a la Planificación del Proyecto de Pruebas


La planificación original del proyecto estimó un total de:

- **35 horas** de ejecución.
- **2,92 días hábiles efectivos**.
- **Desviación proyectada del 30%**, aumentando a **45,5 horas** y **3,79 días hábiles**.


Sin embargo, los resultados reales muestran que estos tiempos **fueron ampliamente superados** debido a **fallas críticas del sistema**, en especial:

- **No validación del código postal** (fallo clave que afectó procesos de compra, envío, generación de historial, aplicación de cupones, facturación y revisión de pedidos).
- **Fallas en login y recuperación de contraseña**.
- **Problemas en la búsqueda y visualización de productos específicos**.

Fase del Proyecto	Horas Estimadas	Horas con Desviación	Afectación real debido a errores
Contextualización	9,00	11,70	Baja. Se completó sin mayor impacto.
Planeación	6,00	7,80	Media. Algunas pruebas planificadas debieron reorganizarse.
Diseño/Ejecución	11,00	14,30	Alta. Fallos críticos impidieron completar pruebas clave (checkout, envío, facturación, etc.)
Documentación y cierre	9,00	11,70	Media a alta. Se generó más documentación de errores, cambios de estado y casos no ejecutados.

 Principales consecuencias de los errores encontrados

1. **Repetición de casos de prueba:** Se tuvieron que relanzar múltiples pruebas (especialmente de pago y envío), afectando el rendimiento diario.
2. **Bloqueo de pruebas dependientes:** Módulos como facturación, historial y revisión de pedidos quedaron bloqueados por un único error central (validación postal).
3. **Mayor tiempo en documentación:** Fue necesario extender el tiempo de documentación, retroalimentación e informe final para justificar las fallas encontradas.
4. **Desviación de planificación real:** Estimaciones iniciales se vieron superadas en al menos un **30-40% adicional**, considerando tiempos muertos y reintentos de pruebas.

 Estimación actualizada del tiempo **real** afectado

Concepto	Estimación inicial	Tiempo ajustado por errores	Incremento aproximado
Total horas estimadas	35 h	50 - 55 h (aprox)	+15-20 horas
Total días hábiles estimados	2,92 días	4,5 - 5 días	+1,5 - 2 días
Nivel de afectación	—	Alta	—

la planificación del proyecto fue técnicamente sólida en cuanto a distribución de tareas, fases y recursos, sin embargo, **la falta de estabilidad del sistema evaluado imposibilitó cumplir con los tiempos previstos**. El error central (validación del código postal) **no solo generó fallos funcionales aislados**, sino que **inmovilizó múltiples módulos del sistema** que dependían de procesos de compra exitosos.

Esto demuestra que **una única falla crítica puede tener un efecto dominó**, alterando profundamente la planificación y el rendimiento del equipo de QA.

 Oportunidades de Mejora

A partir de esta experiencia, se identificaron varias oportunidades que permitirían optimizar futuras ejecuciones de pruebas y mejorar la calidad del software:

1. Planificación de trabajo colaborativo previo al diseño de pruebas

Es esencial estructurar un plan de trabajo entre todos los involucrados antes de diseñar los casos de prueba. Establecer roles, responsables y objetivos individuales permite un mejor seguimiento de avances y evita tareas duplicadas o mal ejecutadas.

2. Comunicación continua y efectiva del equipo

Cada integrante debe informar de forma oportuna cualquier dificultad técnica o de ejecución. Una comunicación proactiva mejora la capacidad de respuesta colectiva y permite realizar ajustes en tiempo real.

3. Priorización y secuenciación de pruebas

Se recomienda establecer un **orden lógico de ejecución** de las pruebas. Esto ayuda a evitar bloqueos causados por funcionalidades dependientes. Por ejemplo, si el sistema de validación de direcciones falla, puede ser más efectivo pausar pruebas que dependan de él y avanzar con otras áreas como autenticación o diseño de interfaz.

4. Automatización de pruebas clave

La implementación de pruebas automatizadas para flujos esenciales como: login, checkout, validación de dirección y generación de facturas permitiría detectar errores tempranamente y ahorrar tiempo.

Reflexión personal

Este ejercicio fue una oportunidad valiosa para adquirir experiencia práctica en la validación de sistemas reales. Aunque la plataforma evaluada parece ser simple (una tienda en línea), el trabajo reveló la **complejidad técnica, operativa y lógica** que hay detrás de cada botón, formulario y proceso.

Pude observar cómo un solo error, como el del código postal, puede **afectar múltiples módulos y frenar casi la mitad del sistema**. También aprendí que **la calidad del software no se limita al código, sino también a la planificación, comunicación, pruebas estructuradas y validación con usuarios finales**.

Validar un sistema no solo implica comprobar que "funciona", sino entender **cómo, cuándo y en qué condiciones deja de funcionar**, y proponer soluciones sostenibles.

Conclusiones

1. La elaboración de un plan de pruebas es esencial para garantizar la calidad del software, ya que permite detectar fallos, validar funcionalidades y asegurar el cumplimiento de los requerimientos establecidos.
2. El proyecto MAISON FA demostró la importancia de aplicar metodologías de prueba estructuradas, que permiten organizar y ejecutar casos de prueba de manera eficaz y controlada.
3. Las pruebas funcionales, de integración y aceptación fueron fundamentales para evaluar el comportamiento del sistema en diferentes escenarios, tanto esperados como no previstos.
4. El trabajo en equipo jugó un papel determinante en el éxito del proyecto, permitiendo una distribución clara de responsabilidades, toma de decisiones colaborativas y mejora continua durante el proceso.
5. El acompañamiento del tutor y la orientación académica fueron claves para mantener un enfoque técnico y metodológico adecuado, cumpliendo con los estándares requeridos en un proyecto de grado.
6. Durante la ejecución del plan de pruebas se fortalecieron habilidades técnicas y profesionales, como el análisis crítico, la solución de problemas, la documentación técnica y la comunicación efectiva.
7. La implementación del aprendizaje adquirido en el aula permitió afrontar desafíos reales, demostrando que la formación académica brinda herramientas sólidas para el desarrollo de proyectos tecnológicos.
8. El sistema evaluado respondió positivamente a la mayoría de los casos de prueba, lo que evidencia una correcta planificación y desarrollo por parte del equipo.
9. La documentación de resultados y hallazgos permitió tener trazabilidad del proceso de pruebas, facilitando la toma de decisiones para futuras mejoras y correcciones en el sistema.
10. El proyecto MAISON FA representa una experiencia formativa integral, donde se conjugan el conocimiento teórico, la práctica profesional y la proyección hacia el entorno laboral.

Dentro de la temática de cada uno de los módulos trabajados en el seminario y mediante el ejercicio realizado en la tienda de ropa online, se concluye lo siguiente:

La importancia que tienen las Pruebas de Software: es muy necesario realizar pruebas rigurosas en aplicaciones web comerciales, mucho más cuando se trata de un entorno de tienda online donde la experiencia del usuario y la seguridad de las transacciones son críticas. Las pruebas nos permiten identificar y corregir errores mucho antes de que afecten a los usuarios.

Un manejo de Riesgos satisfactorio: Realizando un manejo adecuado de la matriz de riesgos se puede demostrar que se trata de una herramienta efectiva para identificar, auditar y minimizar posibles riesgos en la operación de la tienda virtual. Esto también nos ayuda a anticiparnos a muchos problemas potenciales y a su vez nos ayuda a tomar acciones preventivas.

Identificación y asignación de Roles y Responsabilidades: asignar y entregar roles precisos dentro del equipo, nos hace más fácil la coordinación de actividades de pruebas

y a su vez la comunicación de resultados. Cada rol y a su vez persona del equipo tiene tareas específicas y ayudan de manera efectiva al éxito del proyecto.

Desarrollo constante: Como equipo se realiza pruebas que permiten identificar áreas de mejora en la tienda en línea, tanto en términos de funcionalidad como de rendimiento.

Todo lo encontrado a la hora de auditar se utiliza para realizar ajustes y mejoras continuas en la plataforma, contribuyendo a una mejor experiencia de usuario.

Validación de los planes para las pruebas: Las estrategias de pruebas diseñadas y ejecutadas nos ayudan demasiado para la detección de fallos y a su vez nos ayudan en

el enfoque de la validación de la funcionalidad de la tienda virtual.

Y es aquí donde nos dimos cuenta de la importancia de enlazar las estrategias de pruebas a lo que necesitaba específicamente el proyecto y el negocio.

Referencias

Pressman, R. S., & Maxim, B. R. (2021). *Ingeniería de software: Un enfoque práctico* (9.^a ed.). McGraw-Hill.

Sommerville, I. (2011). *Ingeniería de software* (9.^a ed.). Pearson Educación.

IEEE Standard for Software Test Documentation. (1998). IEEE Std 829-1998. Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/IEEESTD.1998.88286>

Gutiérrez, C. (2017). *Metodologías ágiles para el desarrollo de software*. Alfaomega.

Kaner, C., Bach, J., & Pettichord, B. (2002). *Lessons Learned in Software Testing: A Context-Driven Approach*. Wiley.

ISO/IEC/IEEE 29119-1:2013. (2013). *Software and systems engineering — Software testing — Part 1: Concepts and definitions*. International Organization for Standardization.

Galvis, Á. H., & Orozco, M. (2015). *Calidad del software y pruebas: Enfoques y estrategias*. Ecoe Ediciones.

Universidad Uniremington. (2024). *Guía para trabajos de grado: Lineamientos para seminario y proyectos académicos*. Departamento de Investigación y Extensión.