**Plan de Pruebas de Software**

***[Mi Almacén]***

***Fecha: [01/09/2024]***

**Historial de Versiones**

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Información del Proyecto**

| Empresa / Organización | Capstone |
| --- | --- |
| Proyecto | Mi Almacén |
| Fecha de preparación | 01-09-2024 |
| Cliente | Almacén Don Goyo |
| Patrocinador principal | Almacén Don Goyo |
| Gerente / Líder de Proyecto | Simón Carrasco |
| Gerente / Líder de Pruebas de Software |  |

**Aprobaciones**

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| **Benjamín Muñoz** | **Scrum Master** |  |  |  |
| **Simón Carrasco** | **Product Owner** |  |  |  |
| **Almendra Pizarro** | **Team Developer** |  |  |  |
|  |  |  |  |  |

**Resumen Ejecutivo**

Resumen de todo el contenido del plan de Pruebas de Software, describe cuál es su propósito, establece si es un plan maestro o un plan detallado, identifica el alcance del plan de pruebas en relación con el plan de Proyecto de Software, restricciones (por ejemplo de recursos o presupuesto), alcance del esfuerzo de pruebas entre otros aspectos.

El plan de pruebas para el "Proyecto Mi Almacén" tiene como propósito asegurar que todas las funcionalidades desarrolladas cumplan con los requerimientos establecidos y que la aplicación sea estable, eficiente, y fácil de usar. Este plan servirá como una guía detallada para llevar a cabo las pruebas necesarias en cada fase del proyecto, identificando posibles defectos y asegurando que se corrijan antes del despliegue final. Este es un plan de pruebas detallado que cubre todas las fases del ciclo de pruebas, desde las pruebas unitarias hasta las pruebas de integración, regresión y aceptación. El plan se alinea con la metodología Scrum utilizada en el proyecto, asegurando que se realicen pruebas en cada sprint.

El plan de pruebas abarca todas las funcionalidades clave del sistema, incluyendo:

* **Sistema de autenticación de usuarios**: Verificar que el registro, inicio de sesión, y gestión de usuarios sean seguros y funcionales.
* **Gestión de productos (CRUD)**: Asegurar que la adición, modificación, eliminación, y visualización de productos funcione correctamente.
* **Integración de pagos electrónicos mediante Transbank**: Probar que la integración con Transbank permita transacciones seguras y confiables.
* **Generación de informes**: Validar que la generación de informes de ventas e inventario sea precisa y útil para los usuarios finales.

**Restricciones**:

El proyecto está limitado por un presupuesto ajustado y un equipo de desarrollo pequeño, lo que requiere una priorización cuidadosa de las tareas y un enfoque ágil en las pruebas. Además, dado que el proyecto debe completarse dentro de un marco de tiempo específico (durante el período académico), las pruebas deberán realizarse en paralelo con el desarrollo.

**Alcance de las Pruebas**

**Elementos de Pruebas**

Listado de todos los módulos, componentes o elementos que se van a probar. Si es de alto nivel, se listan las áreas funcionales (módulos o procesos que cubre el Testing), por otro lado, si es de un nivel detallado se listan los programas, unidades o módulos.

* **Autenticación de Usuarios:** Probar el proceso de registro, inicio de sesión, y gestión de usuarios, asegurando la seguridad y la correcta funcionalidad.
* **Gestión de Productos:** Verificar las funcionalidades de Crear, Leer, Actualizar, y Eliminar (CRUD) productos, asegurando que los cambios se reflejen correctamente en la base de datos.
* **Integración con Transbank:** Probar la funcionalidad de pagos electrónicos, asegurando que las transacciones se procesen correctamente y de manera segura.
* **Generación de Informes:** Asegurar que los informes de ventas e inventario se generen correctamente y contengan información precisa y útil.

**Nuevas Funcionalidades a Probar**

Es un listado de lo que se va a probar “Desde el Punto de vista del Usuario”. No es una descripción técnica del software sino sus características y funcionalidades. Se incluyen tanto las que son nuevas como las que se están modificando.

* **Funcionalidades CRUD para la gestión de productos:** Asegurarse de que los usuarios puedan agregar, editar, eliminar y visualizar productos de manera eficiente y sin errores.
* **Integración con la API de Transbank:** Probar que la integración permita realizar pagos electrónicos sin fallos y que se registren correctamente en el sistema.
* **Generación de Informes de Ventas e Inventario:** Verificar que los informes sean generados correctamente, mostrando datos precisos y útiles para la toma de decisiones.

**Pruebas de Regresión**

Listado de las funcionalidades no directamente involucradas en el desarrollo, pero cuyos componentes están siendo afectados y por ende deben probarse para asegurar que continúan funcionando adecuadamente. Al igual que en el punto anterior, se describen desde el punto de vista del usuario.

Probar que las nuevas funcionalidades no afecten negativamente el proceso de autenticación y asegurar que las modificaciones no introduzcan errores en las funcionalidades previamente desarrolladas y probadas.

**Enfoque de Pruebas (Estrategia)**

La Estrategia de Pruebas puede definirse como un documento aparte, o puede ser incluido dentro del Plan de Pruebas según su extensión. Aquí pueden definirse los tipos de pruebas a realizar (funcionales, de desempeño, de interfaces, no funcionales, etc.), requerimientos especiales de las pruebas, configuraciones a probar, subconjuntos de datos a considerar, nivel de pruebas de regresión, entre otros aspectos.

La estrategia de pruebas se basará en un enfoque ágil, alineado con la metodología Scrum adoptada para el desarrollo del proyecto. Se realizarán pruebas unitarias y de integración en cada sprint, asegurando que cada incremento de funcionalidad sea probado exhaustivamente antes de avanzar al siguiente.

**Pruebas Funcionales:** Se realizarán pruebas funcionales para verificar que cada componente del sistema cumple con los requisitos especificados.

**Pruebas de Rendimiento:** A medida que las funcionalidades críticas se implementen, se realizarán pruebas de rendimiento para asegurar que el sistema pueda manejar la carga esperada.

**Pruebas de Seguridad:** Dado que la página maneja pagos electrónicos, se realizarán pruebas de seguridad para asegurar que los datos sensibles estén protegidos.

**Pruebas de Regresión:** Después de cada sprint, se ejecutarán pruebas de regresión para asegurarse de que las nuevas funcionalidades no introduzcan defectos en las áreas existentes del sistema.

**Criterios de Aceptación o Rechazo**

**Criterios de Aceptación o Rechazo**

Son los criterios que serán considerados para dar por completado el Plan de Pruebas de Software, por ejemplo: Completar 100% de pruebas unitarias, cierto porcentaje de casos exitosos, cobertura de todos los componentes y líneas de código, porcentaje de defectos corregidos, entre otros.

**Criterios de Aceptación:**

Gestión de Usuarios:

**Inicio de Sesión**: El acceso debe ser exitoso en al menos el 95% de los intentos con credenciales correctas.

**Error de Validación:** El sistema debe manejar correctamente los errores de validación en el 100% de los casos.

**Credenciales Incorrectas:** El sistema debe notificar correctamente en el 100% de los intentos cuando las credenciales son incorrectas.

**Recuperación de contraseña:** El correo de recuperación debe ser enviado exitosamente en al menos el 98% de las solicitudes.

Gestión de Inventario:

**Agregar Producto:** El producto debe ser añadido correctamente en al menos el 99% de los intentos.

**Editar Producto:** Las ediciones deben reflejarse correctamente en al menos el 98% de los casos.

**Eliminar Producto:** La eliminación debe ser exitosa y el inventario actualizado en al menos el 99% de los casos.

**Filtrar Productos:** Los filtros deben funcionar correctamente en al menos el 95% de los casos.

**Criterios de Rechazo:**

Gestión de Usuarios:

Si el acceso exitoso con credenciales correctas es menor al 95%, se considera inaceptable.

Si los errores de validación o credenciales incorrectas no se manejan correctamente en más del 1% de los casos, se rechaza la funcionalidad.

Gestión de Inventario:

Si la adición, edición o eliminación de productos no es correcta en más del 2% de los casos, la funcionalidad es rechazada.

Procesamiento de Pagos Electrónicos:

Si las transacciones fallan en más del 1% de los intentos, se considera inaceptable.

Si los recibos no se envían correctamente en cualquier caso de pago exitoso, se rechaza la funcionalidad.

**Recursos**

**Requerimientos de Entornos – Hardware**

Pc

**Requerimientos de Entornos – Software**

Usaremos el lenguaje de código PHP, con la aplicación laravel, el hosting cpanel para las bases de datos y el alojamiento web.

**Herramientas de Pruebas Requeridas**

**Metodologías:**

**Scrum:** La metodología ágil que guía el proceso de pruebas, integrando las pruebas dentro de cada sprint para asegurar una mejora continua y retroalimentación rápida.

**Entrenamiento**

**Capacitación en el Sistema o Aplicación:**

**Conocimiento Funcional**: Entrenamiento en el funcionamiento del sistema "Mi Almacén", incluyendo la gestión de usuarios, inventarios, pagos electrónicos y generación de informes.

**Procedimientos Operativos:** Formación sobre cómo realizar operaciones clave dentro de la aplicación, como la adición y eliminación de productos, la configuración de alertas, y la integración con Transbank.

**Capacitación en Metodologías:**

**Scrum:** Asegurarse de que todo el equipo esté familiarizado con la metodología Scrum, comprendiendo su rol en el ciclo de desarrollo y pruebas, y cómo colaborar eficazmente en sprints.

**Planificación y Organización**

**Procedimientos para las Pruebas**

El enfoque del proyecto está basado en la metodología Scrum, que implica un desarrollo iterativo e incremental. Las pruebas se realizarán en cada sprint, con revisiones continuas y mejoras basadas en el feedback obtenido. El equipo colaborará estrechamente para garantizar que cada incremento de funcionalidad sea probado exhaustivamente.

**Matriz de Responsabilidades**

**Product Owner:** Simón Eduardo Carrasco Gutiérrez. Responsabilidades: Definir la visión del proyecto, priorizar las funcionalidades en el Product Backlog y asegurarse de que el equipo Scrum entienda los requisitos del producto.

**Scrum Master:** Benjamín Rodrigo Muñoz Pardo. Responsabilidades: Facilitar las reuniones Scrum, eliminar impedimentos, asegurar que el equipo siga las prácticas Scrum y ayudar a maximizar la productividad del equipo​.

**Team Scrum:** Almendra Aileen Pizarro Velásquez. Responsabilidades: Desarrollar las funcionalidades del producto, participar en la planificación de sprints, colaborar en las revisiones de sprint y en la mejora continua del proceso

 **Dependencias y Riesgos**

En el proceso de pruebas de software, es fundamental identificar los riesgos que pueden afectar el desarrollo y la calidad de las pruebas. Algunos de los riesgos más comunes incluyen:

* **Dependencias con Desarrollos**: Riesgos asociados a la necesidad de que ciertas partes del software estén listas antes de poder realizar las pruebas.
* **Dependencias con otros Proyectos**: Interacciones con otros proyectos que podrían retrasar o modificar el plan de pruebas.
* **Disponibilidad de Recursos**: Posible falta de acceso a personal, herramientas o entornos de prueba necesarios para ejecutar las pruebas de manera efectiva.
* **Restricciones de Tiempo**: Limitaciones de tiempo que pueden reducir la cobertura de las pruebas o apresurar su ejecución, afectando la calidad.
* **Premisas Incorrectas**: Suposiciones iniciales que luego resultan ser incorrectas, lo que puede cambiar el alcance de las pruebas.

Cada riesgo se clasifica según su **probabilidad** e **impacto**, y se deben diseñar dos tipos de planes para abordarlos:

* **Plan de Mitigación**: Estrategias para reducir la probabilidad de que el riesgo ocurra.
* **Plan de Contingencia**: Acciones a tomar si el riesgo se materializa y no puede ser mitigado, asegurando que el impacto sea manejable.

Este análisis permite estar preparados ante posibles inconvenientes durante el proceso de pruebas, asegurando una respuesta rápida y adecuada.