**Plan de Pruebas de Software**

***[2CL1CO]***

***Fecha: [dd/mm/aaa]***

**Tabla de contenido**

Historial de Versiones 4

Información del Proyecto 4

Aprobaciones 4

Resumen Ejecutivo 5

Alcance de las Pruebas 5

Elementos de Pruebas 5

Nuevas Funcionalidades a Probar 6

Pruebas de Regresión 6

Funcionalidades a No Probar 7

Enfoque de Pruebas (Estrategia) 7

Criterios de Aceptación o Rechazo 8

Criterios de Aceptación o Rechazo 8

Criterios de Suspensión 8

Criterios de Reanudación 9

Entregables 9

Recursos 10

Requerimientos de Entornos – Hardware 10

Requerimientos de Entornos – Software 10

Herramientas de Pruebas Requeridas 11

Personal 11

Entrenamiento 12

Planificación y Organización 12

Procedimientos para las Pruebas 12

Matriz de Responsabilidades 13

Cronograma 13

Premisas 14

Dependencias y Riesgos 14

Referencias 15

Glosario 15

**Historial de Versiones**

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

**Información del Proyecto**

| Empresa / Organización |  |
| --- | --- |
| Proyecto |  |
| Fecha de preparación |  |
| Cliente |  |
| Patrocinador principal |  |
| Gerente / Líder de Proyecto |  |
| Gerente / Líder de Pruebas de Software |  |

**Aprobaciones**

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

***Resumen Ejecutivo***

*Este Plan de Pruebas de Software está diseñado para guiar y estructurar todas las actividades de pruebas en el desarrollo de nuestro sistema de gestión de inventario y ventas. Se centra en asegurar que las funcionalidades críticas, como la gestión de inventario, el registro de ventas, y las notificaciones de productos, operen sin fallos y cumplan con las expectativas del usuario.*

*El propósito principal del plan es verificar que el software no solo sea funcional, sino también seguro y eficiente, alineándose con las metodologías ágiles que estamos utilizando en el proyecto. Se ha identificado el alcance del plan, abarcando las funcionalidades clave y las áreas que podrían verse afectadas por los nuevos desarrollos, sin dejar de lado las pruebas de regresión para asegurar que no se generen errores imprevistos.*

*Sin embargo, debido a algunas restricciones, como el presupuesto limitado y la falta de pruebas de estrés a gran escala, se ha priorizado la funcionalidad básica y la experiencia del usuario. Las pruebas abarcarán componentes tanto nuevos como preexistentes, utilizando un enfoque mixto de pruebas manuales y automatizadas.*

*El equipo cuenta con un entorno de pruebas completo y un conjunto de herramientas que facilitan la detección y corrección de defectos. Además, se ha previsto un plan de entrenamiento para garantizar que todos los miembros estén capacitados en las herramientas y metodologías necesarias.*

***Alcance de las Pruebas***

***Elementos de Pruebas***

*El plan de pruebas cubre los siguientes módulos y componentes, basados en las actividades realizadas y descritas en los reportes diarios del equipo:*

1. ***Autenticación y Seguridad****:*
   * ***Roles y Permisos****: Configuración y validación de roles, implementación de encriptación de contraseñas y pruebas de acceso seguro realizadas por Bastián Navarrete.*
   * ***Gestión de Usuarios****: Configuración de permisos específicos para cada usuario y pruebas de autenticación.*
2. ***Base de Datos y Gestión de Inventario****:*
   * ***Tablas de Productos y Categorías****: Configuración y verificación de la estructura de datos, incluyendo validación y relación entre tablas, desarrolladas por Danny Betancourt.*
   * ***Formularios de Productos****: Diseño y validación de formularios para crear, actualizar y eliminar productos del inventario.*
3. ***Interfaz de Usuario (UI)****:*
   * ***Prototipos de Interfaz****: Creación de diseños y mockups para la gestión de ventas y productos, con pruebas de botones y validaciones visuales, realizadas por Leonardo Aguilera.*
   * ***Experiencia del Usuario****: Ajustes estéticos y pruebas de usabilidad para asegurar que la interfaz sea intuitiva y funcional.*
4. ***Notificaciones y Alertas****:*
   * ***Alertas de Productos****: Implementación de alertas para productos próximos a vencer y configuración del tiempo de alerta, incluida en las tareas de sprints avanzados.*

***Nuevas Funcionalidades a Probar***

***Gestión de Inventario****:*

* ***Agregar y Eliminar Productos****: Desde el punto de vista del usuario, se podrá agregar y eliminar productos del inventario de forma intuitiva.*
* ***Actualización de Inventario****: Se debe verificar la actualización automática del inventario cuando un producto es vendido o eliminado.*

***Gestión de Ventas****:*

* ***Registro de Ventas****: El usuario debe poder registrar ventas de productos con facilidad, asegurando que los reportes reflejen las transacciones de forma precisa.*
* ***Reportes Diarios de Ventas****: Validar que el usuario pueda generar reportes diarios de ventas, incluyendo detalles como el total vendido y productos destacados.*

***Notificaciones y Alertas****:*

* ***Alertas de Productos por Vencer****: Se probará la funcionalidad que avisa cuando los productos están próximos a vencer, ajustable por el usuario.*
* ***Tiempo de Alerta Configurable****: Confirmar que los usuarios puedan definir el tiempo de notificación para los productos.*

***Experiencia del Usuario****:*

* ***Interfaz Intuitiva****: Pruebas de la nueva interfaz diseñada para una mejor experiencia del usuario, incluyendo botones, navegación y presentación visual.*

***Configuración de Seguridad****:*

* ***Roles y Permisos****: Probar la asignación y validación de roles específicos, garantizando que los usuarios solo puedan acceder a funciones autorizadas.*
* ***Integridad de Datos****: Verificar que la confidencialidad y la integridad de los datos se mantengan en todo momento.*

***Pruebas de Regresión***

***Módulos de Reportes****:*

* *Probar la funcionalidad de generación de reportes para asegurarse de que los nuevos desarrollos no afecten el sistema de reportes preexistente.*

***Gestión de Usuarios****:*

* *Validar que los cambios realizados en la seguridad y gestión de usuarios no interfieran con funcionalidades ya existentes, como la creación y edición de perfiles.*

***Actualización de Inventario****:*

* *Asegurarse de que la actualización de inventario sigue funcionando correctamente incluso después de la implementación de nuevas alertas y notificaciones.*

***Interfaz de Usuario Preexistente****:*

* *Revisar que las partes de la interfaz no modificadas sigan funcionando y siendo accesibles de manera fluida.*

***Funcionalidades a No Probar***

1. ***Integraciones Externas****:*
   * *No se probarán las conexiones con sistemas externos o integraciones de software, ya que no forman parte del alcance actual.* ***Riesgo****: Potencial impacto si se integran en futuras versiones sin pruebas previas.*
2. ***Optimización de Desempeño a Gran Escala****:*
   * *No se realizarán pruebas de estrés o carga a gran escala debido a limitaciones de recursos.* ***Riesgo****: Desempeño no óptimo bajo alta carga, especialmente si el negocio crece rápidamente.*
3. ***Funcionalidades Obsoletas****:*
   * *Elementos o módulos identificados como obsoletos o destinados a ser reemplazados no se incluirán en las pruebas.* ***Riesgo****: Posibles problemas si estos módulos son utilizados por error.*

***Enfoque de Pruebas (Estrategia)***

***Tipos de Pruebas****:*

* ***Pruebas Funcionales****: Validación de que cada funcionalidad del software se comporta como se espera.*
* ***Pruebas de Interfaz de Usuario****: Asegurar que la interfaz sea amigable y sin errores visuales.*
* ***Pruebas de Seguridad****: Verificar que la autenticación y permisos se manejen adecuadamente.*
* ***Pruebas de Desempeño Básico****: Realizar pruebas de carga y respuesta para funciones clave, limitadas a las herramientas disponibles.*

***Configuraciones a Probar****:*

* *Pruebas en múltiples navegadores y dispositivos para garantizar la compatibilidad.*
* *Verificación con subconjuntos de datos reales y simulados para observar diferentes escenarios.*

***Criterios de Aceptación o Rechazo***

***Criterios de Aceptación o Rechazo***

* *Completar el 100% de los casos de prueba funcionales con éxito.*
* *Tener un mínimo del 90% de cobertura de código con pruebas unitarias.*
* *Corregir al menos el 95% de los defectos críticos antes de la liberación.*
* *Validación de la experiencia del usuario con comentarios positivos en las pruebas de usabilidad.*

***Criterios de Suspensión***

* *Si se descubre un defecto crítico que impida continuar con las pruebas de otros componentes.*
* *Si más del 40% de los casos de prueba funcionales fallan.*
* *Problemas de acceso o disponibilidad del entorno de pruebas por más de 48 horas.*

***Criterios de Reanudación***

* *Una vez que se solucionen los defectos críticos y el equipo de desarrollo los valide.*
* *Cuando el entorno de pruebas esté completamente funcional y accesible.*
* *Aprobación del líder de pruebas tras una revisión de las condiciones y el estado del sistema.*

***Entregables***

* *Análisis del caso*
* *Squad y responsabilidades*
* *Mapa mental*
* *Mapa de actores*
* *Visión del proyecto +4 pilares*
* *Epicas*
* *Historias de usuario*
* *Impact mapping*
* *Product backlog priorizado*
* *User story mapping*
* *Sprint planning*
* *Sprint backlog*
* *Scrumboard*
* *Daily meeting*
* *Registro de impedimentos*
* *Burndown chart*
* *Release*
* *Sprint Review*
* *Sprint Retrospective*
* *Retrospectiva del proyecto*

***Recursos***

***Requerimientos de Entornos – Hardware***

***Servidores****: Servidor de aplicaciones y servidor de bases de datos configurados con las especificaciones requeridas.*

***Estaciones de Trabajo****: Computadoras con mínimo 8 GB de RAM y procesadores de última generación para los testers.*

***Conectividad****: Acceso seguro a la red y al entorno de pruebas, con todas las configuraciones de firewall necesarias.*

***Herramientas****: Disponibilidad de herramientas de pruebas automatizadas y manuales instaladas en las estaciones de trabajo.*

***Requerimientos de Entornos – Software***

***Sistemas Operativos****: Windows 10 o superior para todos los dispositivos de prueba, con las últimas actualizaciones de seguridad instaladas.*

***Bases de Datos****: Acceso a una base de datos de pruebas que imite el entorno de producción, con datos de muestra representativos y restaurables en caso de errores.*

***Entornos de Pruebas****:*

* *Un entorno de desarrollo separado que permita ejecutar las pruebas sin afectar el sistema en producción.*
* *Configuración y acceso a los sistemas de gestión de inventario y ventas en un entorno de simulación.*

***Software Adicional****:*

* *Navegadores actualizados (Google Chrome, Mozilla Firefox, Microsoft Edge) para pruebas de compatibilidad.*
* *Instalación de software de pruebas, como gestores de casos de prueba y herramientas de seguimiento de errores.*

***Herramientas de Pruebas Requeridas***

***Automatización de Pruebas****: Herramientas como Selenium o Cypress para automatizar las pruebas funcionales y de interfaz.*

***Gestión de Pruebas****: JIRA o un software de gestión similar para el seguimiento y planificación de las pruebas, integrado con nuestro tablero ágil.*

***Seguimiento de Errores****: Un sistema como Bugzilla o el módulo de seguimiento de errores en JIRA para reportar y gestionar incidencias.*

***Análisis de Código****: Herramientas de revisión de código como SonarQube para asegurar la calidad y detectar posibles vulnerabilidades.*

***Pruebas de Carga****: Uso de JMeter para pruebas de desempeño y carga, especialmente en funcionalidades críticas como el registro de ventas.*

***Personal***

* *Leonardo Aguilera -*
* *Bastian Navarrete -*
* *Danny Betancourt -*

***Entrenamiento***

***Sistema de Gestión****: Capacitación básica en el uso de nuestro software de gestión de inventario y ventas, enfocándose en las nuevas funcionalidades.*

***Herramientas de Pruebas****: Entrenamiento práctico en herramientas como Selenium, JIRA, y JMeter para el equipo de testers, incluyendo talleres sobre cómo automatizar y gestionar pruebas.*

***Metodologías Ágiles****: Refuerzo de las metodologías ágiles para asegurar que todo el equipo esté alineado con la forma de trabajo, facilitando la colaboración y la entrega continua.*

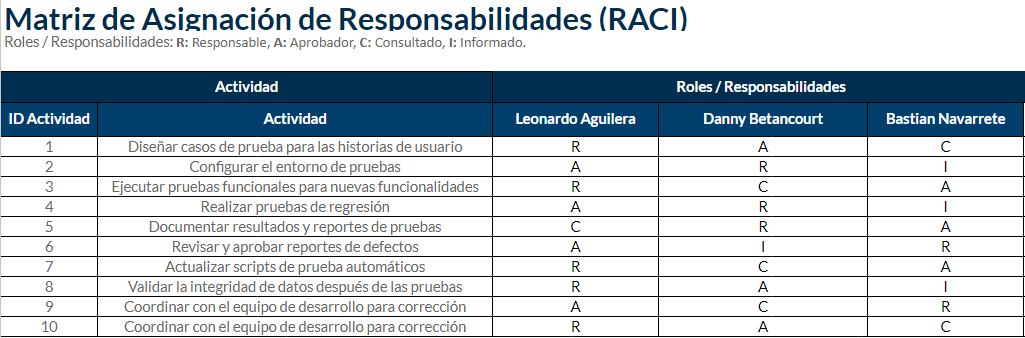
***Planificación y Organización***

***Procedimientos para las Pruebas***

*La metodología a utilizar será Scrum, una metodología ágil lo que permitirá la entrega continua de funcionalidades y ajustes de manera temprana según el feedback del usuario. Las etapas incluirán:*

* *Análisis de requerimientos: Reuniones con los stakeholders para definir las funcionalidades necesarias.*
* *Diseño del sistema: Creación de diagramas UML para representar la arquitectura del sistema.*
* *Desarrollo: Implementación de módulos utilizando frameworks de desarrollo modernos.*
* *Pruebas: Ejecución de pruebas unitarias y de integración para garantizar la calidad del software.*
* *Validación: Evaluación del sistema en un entorno simulado y ajuste según el feedback.*

***Matriz de Responsabilidades***

**

***Cronograma***

| ***Actividad*** | ***Fase 1*** | | | | ***Fase 2*** | | | | | | | | | | | | ***Fase 3*** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***S 1*** | ***S 2*** | ***S 3*** | ***S 4*** | ***S 5*** | ***S 6*** | ***S 7*** | ***S 8*** | ***S 9*** | ***S 10*** | ***S 11*** | ***S 12*** | ***S 13*** | ***S 14*** | ***S 15*** | ***S 16*** | | ***S 17*** | ***S 18*** |
| *Inicio* | ***X*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Análisis de requerimientos* |  | ***X*** | ***X*** |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Diseño del sistema* |  |  |  | ***X*** | ***X*** | ***X*** |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Implementación* |  |  |  |  |  |  | ***X*** | ***X*** | ***X*** | ***X*** | ***X*** | ***X*** |  |  |  |  | |  |  |
| *Pruebas* |  |  |  |  |  |  |  |  |  |  |  |  | ***X*** | ***X*** | ***X*** |  | |  |  |
| *Validación final* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ***X*** | | ***X*** |  |

***Premisas***

***Limitaciones de Tiempo****: Se ha estimado un tiempo específico para cada sprint de pruebas, asumiendo que no habrá interrupciones significativas. La disponibilidad de los recursos es crítica, y se espera que los desarrolladores y testers trabajen de manera coordinada según la metodología ágil.*

***Disponibilidad de Recursos****: Se cuenta con un equipo de testers que tiene acceso a todos los recursos tecnológicos y herramientas necesarias para ejecutar las pruebas. Sin embargo, se asume que el entorno de pruebas estará disponible y será estable durante todo el ciclo de pruebas.*

***Metodología****: Las pruebas se llevarán a cabo siguiendo principios ágiles, con ciclos iterativos que permiten la revisión y ajuste constante de los casos de prueba y resultados. Se espera que todos los participantes del proyecto estén familiarizados con esta metodología.*

***Herramientas de Prueba****: Se asume el uso de herramientas específicas para la automatización y gestión de pruebas, como Selenium, JIRA, y JMeter, que estarán completamente configuradas y operativas para el equipo.*

***Soporte y Comunicación****: Se espera una comunicación fluida entre los equipos de desarrollo y pruebas para resolver rápidamente los problemas que surjan. Esto incluye el apoyo técnico necesario para cualquier imprevisto.*

***Dependencias y Riesgos***

***Dependencias****:*

* ***Dependencia con Desarrollos****: El éxito de las pruebas depende de la finalización oportuna de las funcionalidades por parte del equipo de desarrollo. Cualquier retraso en las entregas afectará el calendario de pruebas.*
* ***Dependencia con Otros Proyectos****: Si hay proyectos paralelos que comparten recursos, como infraestructura o personal, podría haber conflictos en la asignación de estos.*
* ***Disponibilidad de Recursos****: La falta de disponibilidad de recursos humanos o técnicos en momentos clave puede retrasar las pruebas.*
* ***Restricciones de Tiempo****: La ejecución de las pruebas está limitada por el calendario del proyecto, lo cual puede dejar poco margen para pruebas extensivas en caso de que se identifiquen problemas significativos.*

***Riesgos****:*

* ***Cambio en los Requerimientos****: Si se produce un cambio en los requisitos del sistema durante las pruebas, habrá que replanificar las actividades, lo que impactará los tiempos y recursos asignados.*
  + *Mitigación: Establecer revisiones frecuentes con los stakeholders para asegurar que los requisitos estén alineados y aprobados.*
* ***Entorno de Pruebas Inestable****: Si el entorno no está disponible o falla durante las pruebas, se pueden perder datos y tiempo.*
  + *Mitigación: Planificar pruebas de respaldo y realizar copias regulares de seguridad.*
* ***Defectos Mayores****: La aparición de errores críticos que bloqueen el avance de las pruebas podría retrasar las entregas.*
  + *Contingencia: Priorización inmediata de defectos críticos para la corrección y coordinación con desarrollo.*
* ***Inexperiencia del Equipo****: El personal podría carecer de suficiente conocimiento sobre las herramientas de prueba, afectando la eficiencia.*
  + *Mitigación: Planificar sesiones de formación previas y soporte continuo.*

***Referencias***

* ***Plan de Proyecto****: Documentación general del proyecto que define la línea de tiempo, los hitos y las entregas.*
* ***Especificaciones de Requerimientos****: Documento que detalla los requisitos funcionales y no funcionales del sistema.*
* ***Diseño General y Detallado****: Planos y diagramas del sistema, que incluyen estructuras y componentes.*
* ***Procedimientos y Estándares de Desarrollo****: Normas que deben seguirse durante la construcción del software.*
* ***Procedimientos y Estándares de Pruebas****: Directrices que rigen la manera de ejecutar y documentar las pruebas.*
* ***Metodologías y Procedimientos Corporativos****: Referencias a las mejores prácticas y estándares establecidos por la organización para el desarrollo y pruebas de software.*

***Glosario***

1. ***Metodología Ágil****: Un enfoque iterativo y colaborativo para la gestión de proyectos de desarrollo de software que prioriza la entrega incremental y la colaboración continua entre equipos. Incluye la adaptación y el ajuste constantes según las necesidades cambiantes del proyecto.*
2. ***Sprint****: Un ciclo de trabajo breve y definido, generalmente de una a cuatro semanas, durante el cual se desarrolla y entrega un incremento de software funcional. Es la unidad de trabajo básica en Scrum, una de las metodologías ágiles.*
3. ***Backlog del Producto****: Lista priorizada de funcionalidades, mejoras y correcciones pendientes, que el equipo debe desarrollar. En el contexto de tu proyecto, incluye las historias de usuario que deben ser probadas y desarrolladas.*
4. ***Backlog del Sprint****: Conjunto de tareas y actividades seleccionadas del Backlog del Producto para completarse durante un Sprint. Aquí se incluyen las historias de usuario que se planean probar en un ciclo específico.*
5. ***Historia de Usuario****: Una descripción breve y simple de una funcionalidad desde la perspectiva del usuario final o cliente. Las historias de usuario en tu proyecto guían el desarrollo y las pruebas para garantizar que se cumplan los requisitos del cliente.*
6. ***Pruebas Funcionales****: Verificación de que cada funcionalidad del software cumple con los requisitos especificados. Se centran en la interacción del usuario con el sistema y aseguran que las funciones trabajen según lo esperado.*
7. ***Pruebas de Regresión****: Conjunto de pruebas diseñadas para asegurarse de que los cambios recientes no afecten las funcionalidades existentes del software. Son fundamentales en tu proyecto para confirmar que las actualizaciones no generen nuevos defectos.*
8. ***Pruebas No Funcionales****: Pruebas que evalúan aspectos como el rendimiento, la seguridad y la usabilidad del software, en lugar de las funcionalidades específicas. Por ejemplo, las pruebas de carga y estrés podrían ser necesarias en tu proyecto.*
9. ***Definición de Hecho (Definition of Done - DoD)****: Criterios claros que deben cumplirse para que una historia de usuario o una tarea se considere completada y lista para entrega. Incluye pruebas exitosas y revisión por parte del equipo.*
10. ***Definición de Listo (Definition of Ready - DoR)****: Condiciones que deben cumplirse para que una historia de usuario esté lista para ser trabajada durante un Sprint. Asegura que los elementos del Backlog tengan los detalles necesarios para la ejecución.*
11. ***Tareas de Prueba****: Actividades específicas asignadas para evaluar el software, como la creación de casos de prueba, la ejecución de pruebas y la documentación de resultados. En tu proyecto, estas tareas están organizadas por Sprints y asignadas a los miembros del equipo.*
12. ***Casos de Prueba****: Conjunto de condiciones y acciones que se utilizan para verificar que una funcionalidad del software funcione correctamente. Cada caso de prueba incluye los pasos, entradas y resultados esperados.*
13. ***RACI****: Matriz de asignación de responsabilidades que define quién es Responsable, Aprobador, Consultado e Informado en cada tarea. En tu proyecto, se asignan roles específicos a Leonardo Aguilera, Bastian Navarrete y Danny Betancourt.*
14. ***Herramientas de Pruebas****: Software utilizado para planificar, ejecutar y gestionar actividades de prueba. Puede incluir herramientas de gestión de casos de prueba, automatización y seguimiento de defectos.*
15. ***Integración Continua****: Práctica de desarrollo ágil en la que el código se integra con frecuencia en un repositorio compartido, seguido por la ejecución de pruebas automáticas. Facilita la detección temprana de errores en el software.*