TODI STORE

**DOCUMENTO DE PLAN DE PRUEBAS DE SOFTWARE PARA EL DASHBOARD DE SEGUIMIENTO DE ESTUDIOS DE VIDEOJUEGOS INDEPENDIENTES (DSEVI)**



|  |  |  |
| --- | --- | --- |
| **HISTÓRICO DE CAMBIOS** | | |
| **Versión** | **Fecha** | **Cambios realizados** |
| 01 | 20/12/2022 | Emisión Inicial |

|  |  |  |
| --- | --- | --- |
| **Elaboró** | **Revisó** | **Aprobó** |
| * Calderón Bermudo | * Villacis Alvear | * Torres Domenack |

**PLAN DE PRUEBAS**

**PROYECTO DSEVI**

1. Introducción

El cliente Todi Store requiere un sitio web confiable, que pase todas las pruebas unitarias y de integración. Ya que cuenta con una pasarela de pagos y un módulo del para el registro de usuarios.

El Plan de Pruebas ha sido creado para facilitar la comunicación dentro de los miembros del equipo. Este documento describe los enfoques y metodologías que se aplicarán a la unidad, la integración y las pruebas del Sistema de Comercio Electrónico Todi Store. Incluye los objetivos, las responsabilidades de las pruebas, los criterios de entrada y salida, el alcance, el calendario de los principales hitos, los criterios de entrada y salida y el enfoque.

1. ALCANCE

El propósito de este documento es proporcionar la información y el marco requerido para planificar y desarrollar las actividades del proceso de pruebas del Sistema de Comercio Electrónico Todi Store.

* 1. RUTAS POR PROBAR (pruebas de integración)
* Método GET, POST, PUT Y DELETE del módulo de categorías
* Método GET, POST, PUT Y DELETE del módulo de productos
* Método GET, POST, PUT Y DELETE del módulo de usuarios
* Método GET, POST, PUT Y DELETE del módulo de ordenes
  1. Funciones por probar (pruebas unitarias)
* Función manejadora de errores
* Función validadora de categorías
* Función validadora de IDs
* Función validadora de órdenes, estado de órdenes y checkout
* Función validadora de productos
* Función validadora de usuarios y de inicio de sesión

1. OBJETIVOS DE CALIDAD
   1. Objetivos primarios

Un objetivo principal de las pruebas es: asegurar que el sistema cumpla todas las historias de usuarios establecidas en el documento de Product Backlog. Al final del ciclo de desarrollo del proyecto, el usuario debe encontrar que el proyecto ha cumplido o superado todas sus expectativas como se detalla en las historias de usuarios.

Cualquier cambio, adición o eliminación al Product Backlog, se documentará y probará con el más alto nivel de calidad permitido dentro del tiempo restante del proyecto y dentro de la capacidad de la prueba.

* 1. Objetivos secundarios

Los objetivos secundarios de las pruebas serán: identificar y exponer todos los problemas y riesgos asociados, comunicar todos los problemas conocidos al equipo del proyecto y garantizar que todos los problemas se aborden de manera adecuada antes de la liberación. Como objetivo, esto requiere pruebas cuidadosas y metódicas de la aplicación para garantizar primero que todas las áreas del sistema se examinen y, en consecuencia, todos los problemas (errores) encontrados se traten adecuadamente.

1. ENFOQUE DE PRUEBA

El enfoque que se utiliza es, por lo tanto, Analítico, de acuerdo con la estrategia basada en requisitos, donde un análisis de la especificación de requisitos forma la base para planificar, estimar y diseñar pruebas. Los casos de prueba se crearán durante las pruebas exploratorias. Todos los tipos de prueba se determinan en la estrategia de prueba.

El equipo también debe utilizar las pruebas basadas en la experiencia y la suposición de errores, utilizar las habilidades y la intuición de los evaluadores, junto con su experiencia con aplicaciones o tecnologías similares.

El proyecto está utilizando un enfoque ágil, con iteraciones semanales. Al final de cada semana, los requisitos identificados para esa iteración se entregarán al equipo y se probarán.

1. criterios de entrada y salida
   1. Criterios de entrada

* Debe estar disponible toda la documentación necesaria, el diseño y la información de los requisitos que permitirán a los testers operar el sistema y juzgar el comportamiento correcto.
* Todas las herramientas de software estándar, incluidas las herramientas de prueba, deben haberse instalado con corrección y funcionar correctamente.
* Los datos de prueba adecuados están disponibles.
* El entorno de prueba, el software y el soporte de administración del sistema, debe estar listo.
* Los recursos de control de calidad han entendido completamente los requisitos.
* Los recursos de control de calidad tienen un sólido conocimiento de la funcionalidad.
* Escenarios de prueba revisados, casos de prueba.
  1. Criterios de salida

Se ha alcanzado un cierto nivel de cobertura de los requisitos.

No quedan pendientes errores graves o de alta prioridad.

Costo: cuando se ha gastado el presupuesto.

El calendario se ha cumplido

Todas las zonas de alto riesgo han sido completamente probadas, quedando pendientes solo riesgos residuales menores

1. CRITERIOS DE SUSPENSIÓN Y REQUISITOS DE REANUDACIÓN
   1. Criterios de suspensión

* La construcción contiene muchos defectos graves que limitan seriamente o limitan el progreso de las pruebas.
* Significativo cambio en el Product Backlog propuesto por el cliente
* Problemas de software
* Los recursos asignados no están disponibles cuando lo necesita el equipo de pruebas.
  1. Criterios de reanudación

La reanudación solo se producirá cuando se hayan resuelto los problemas que causaron la suspensión causada.

1. ESTRATEGIA DE PRUEBA
   1. ROL DE QA EN el proceso de prueba

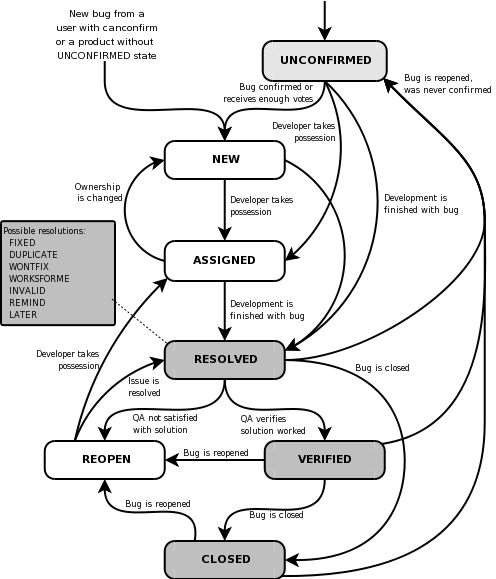
* Comprensión de las historias de usuario:
  + Las especificaciones de las funcionalidades serán evaluadas con el cliente.
  + El control de calidad se encargará de la comprensión de las funcionalidades.
* Preparación de casos de prueba:
  + El control de calidad preparará casos de prueba basados en las pruebas exploratorias. Esto cubrirá todos los escenarios para las funcionalidades.
* Preparación de la matriz de pruebas:
  + El control de calidad preparará la matriz de prueba que asigna los casos de prueba al requisito respectivo. Esto asegurará la cobertura de las funcionalidades.
* Revisión de casos de prueba y matriz:
  + El líder de control de calidad llevará a cabo una revisión por pares para los casos de prueba y la matriz de prueba.
  + Cualquier comentario o sugerencia sobre los casos de prueba y la cobertura de la prueba será proporcionado por el revisor respectivo Autor de Caso de prueba y Matriz de prueba.
  + Las sugerencias o mejoras serán reelaboradas por el autor y se enviarán para su aprobación.
  + Las mejoras reelaboradas serán revisadas y aprobadas por el revisor.
* Crear Prueba Datos:
  + Los datos de prueba serán creados por el control de calidad respectivo en el sitio de desarrollo / prueba del cliente en función de escenarios y casos de prueba.
* Ejecución de casos de prueba:
  + Los casos de prueba serán ejecutados por el control de calidad respectivo en el sitio de desarrollo / prueba del cliente en función de los escenarios diseñados, los casos de prueba y los datos de prueba.
  + El resultado de la prueba (Resultado real, Aprobado/Reprobado) se actualizará en el documento del caso de prueba Registro e informes de defectos: El control de calidad registrará los defectos/errores en un documento de Word que se encuentren durante la ejecución de los casos de prueba. Después de esto, el control de calidad informará al desarrollador respectivo sobre el defecto o los errores.
* Pruebas de reevaluación y regresión:

La repetición de las pruebas de errores corregidos se realizará por el control de calidad respectivo una vez que sea resuelto por el desarrollador respectivo y el estado del error / defecto se actualizará en consecuencia. En ciertos casos, se realizarán pruebas de regresión si es necesario.

* Implementación/Entrega:
  + Una vez que se hayan corregido todos los errores / defectos informados después de completar las pruebas y no se encuentren otros errores, el informe se implementará en el sitio de prueba del cliente por PM.
  + Una vez que el control de calidad realizará la ronda de pruebas en el sitio de prueba del cliente, si es necesario, el informe se entregará junto con la salida de la muestra por correo electrónico al cliente potencial respectivo y al grupo de informes
  + QA enviará la copia impresa completa del comprobante de entrega al desarrollador respectivo.
  + Una vez que el cliente potencial reciba la copia impresa del comprobante de entrega llenado por QA y el desarrollador, enviará el correo electrónico de entrega del informe al cliente.
  1. Ciclo de vida del error:

Todos los problemas encontrados durante las pruebas se registrarán en el documento de Word.

El ciclo de vida del error para este proyecto es el siguiente:



* 1. Tipos de pruebas
* Pruebas unitarias:

Son un método de pruebas de software que se realizan escribiendo fragmentos de código que testeará unidades de código fuente. El objetivo es asegurar que cada unidad funciona como debería de forma independiente.

* Pruebas de integración:

Se examinan las interfaces entre grupos de componentes o subsistemas para asegurar que son llamados cuando es necesario y que los datos o mensajes que se transmiten son los requeridos.

* 1. Gravedad de errores y definición de prioridad

Los campos Gravedad del error y prioridad son muy importantes para categorizar los errores y priorizar si se corregirán y cuándo. Los niveles de gravedad y prioridad del error se definirán como se describe en las siguientes tablas a continuación. Las pruebas asignarán un nivel de gravedad a todos los errores. El líder de prueba será responsable de asegurarse de que se asigne un nivel de gravedad correcto a cada error.

El Líder de control de calidad, el Líder de desarrollo y el Gerente de proyecto participarán en las reuniones de revisión de errores para asignar la prioridad de todos los errores actualmente activos. Esta reunión se conocerá como "Reuniones de clasificación de errores". El líder de control de calidad es responsable de organizar estas reuniones de forma rutinaria para abordar el conjunto actual de errores nuevos y existentes, pero no resueltos.

**Lista de gravedad**

|  |  |  |
| --- | --- | --- |
| **ID de gravedad** | **Severidad** | **Descripción de la gravedad** |
| 1 | Crítico | El módulo/producto falla o el error provoca condiciones no recuperables. Bloqueos del sistema, fallas de GP, daños en la base de datos o archivos, o posible pérdida de datos. |
| 2 | Alto | Componente principal del sistema inutilizable debido a fallas o funcionalidad incorrecta. Estos errores causan problemas graves como la falta de funcionalidad, o mensajes de error insuficiente o poco claro que pueden tener un gran impacto para el usuario, impiden probar otras áreas de la aplicación, etc. Pueden tener una solución, pero la solución es inconveniente o difícil. |
| 3 | Medio | Funcionalidad incorrecta del componente o proceso. |
| 4 | Menor | Errores de documentación o errores de gravedad 3. |

**Lista de prioridades**

|  |  |  |
| --- | --- | --- |
| Prioridad | Nivel de prioridad | Descripción de la prioridad |
| 1 | Debe arreglar | Este error debe corregirse de inmediato; el producto no se puede enviar con este error. |
| 2 | Debería arreglar | Estos son problemas importantes que deben solucionarse lo antes posible. |
| 3 | Arreglar cuando tenga tiempo | El problema debe solucionarse en el tiempo disponible, si el error no retrasa la fecha de envío. |
| 4 | Baja prioridad | No es importante (en este momento) que se aborden estos errores. Solucione estos errores después de que se hayan solucionado todos los demás errores. |

1. NECESIDADES DE RECURSOS Y MEDIO AMBIENTE
   1. Herramientas de prueba

SonarQube: Esta herramienta nos permite realizar un análisis estático de código. Lo que hace es identificar los puntos susceptibles de mejora, que facilitarán la obtención de métricas necesarias para la optimización del código.

* Realiza revisiones automáticas con análisis de código estático detectando problemas que afectan la calidad del código.
* Facilita informes ofreciendo información objetiva de la calidad actual de los proyectos utilizando métricas y gráficos de prueba de calidad avanzados. Esto incluye datos sobre código duplicado, estándares de codificación, pruebas unitarias, cobertura de código, complejidad del código, errores potenciales, comentarios, diseño y arquitectura.
  1. Entorno de prueba

|  |  |
| --- | --- |
| Procesador de Sistema: | Intel® Core™ i5 |
| Memoria de Sistema: | 16GB RAM |
| Espacio Libre en Disco: | 20 GB |
| Sistema Operativo: | Windows 10 |
| Acceso Red: | LAN |