

Actividad | #3 | Plan de pruebas y reglas para el versionado e incidencias

Aseguramiento de Calidad

Ingeniería en Desarrollo de Software

TUTOR: Jessica Hernández Romero

ALUMNO: Alejandra Ibarra Carmona

FECHA: 18/05/2025

Índice

Introducción…………………………………………………..3

Descripción…………………………………………………..3

Justificación………………………………………………….3

Desarrollo……………………………………………………4

Conclusiones……………………………………………….5

Referencias………………………………………………..5

Introducción

El análisis orientado a objetos es considerado una tecnica y métodos para entender un problema, estas técnicas y métodos nos ayudan a entender nuestro problema que deseamos resolver. El análisis orientado a objetos nos implementa dos métodos principales que nos ayudan a plantear la estructura del problema los cuales son el modelo de casos de uso y el modelo de dominio.

Los modelos de caso de uso nos ayudan a identificar la interacción entre el usuario y el sistema que también a su vez nos van describiendo el rol que debe tener cada usuario, mientras que en el modelo del dominio se encarga de generalizar el vocabulario implementado en el sistema lo que nos permite tener todas las definiciones del sistema. Después de pasar por el AOO se inicia el proceso de desarrollo de software en donde ya se implementa el método orientado a objetos, a partir de este se construye un modelo de objetos que se implementa en las tarjetas CRC.

Descripción

También existe el diseño de pruebas basado en escenarios, los resultados de prueba basados en errores no capturan dos tipos principales de errores, primero las especificaciones incorrectas y segundo interacciones entre subsistemas, y finalmente hay dos métodos de prueba que son aplicables en el nivel de clases. En el caso de las aplicaciones web además de las pruebas tradicionales se evalúan otros seis elementos:

* Contenido
* Interfaz
* Navegación
* Configuración
* Seguridad
* Rendimiento

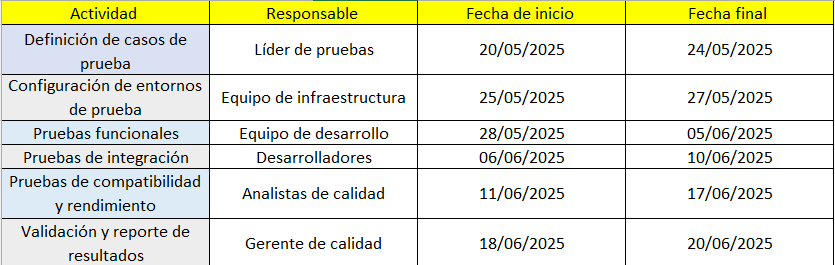
La prueba de contenido tiene como objetivos descubrir errores sintácticos que es la gramática y la ortografía, errores semánticos es la exactitud de la información presentada y la consistencia entre los objetos de contenido y por ultimo los errores de organización o estructura del contenido que presenta el usuario final. La prueba de interfaz en esta el usuario tiene como objetivo descubrir los errores que resultan de una pobre implementación de interacción, omisiones e inconsistencias.

Justificación

La prueba de navegación busca garantizar que todos los mecanismos que permiten al usuario navegar a través de la aplicación web y que con esto funcione de manera correctamente, mas sin en cambio las pruebas de configuración sr aplican para descubrir errores de desempeño que se presentan debido a la falta de recursos de lado del servidor por el ancho de banda de red inapropiados por las capacidades inadecuadas de la base de datos y por las debilidades del sistema operativo. Las pruebas de seguridad están diseñadas para probar las vulnerabilidades del lado del cliente, así como las comunicaciones de red que ocurren mientras se pasan los datos de servidor del cliente y viceversa, finalmente las pruebas de rendimiento sirven para validar y verificalos atributos de la calidad del sistema, comprar dos sistemas para saber cual es mejor y funciona mejor, medir que partes del sistema o de carga de trabajo provocan que el conjunto rinda mal.

Desarrollo

* Plan de pruebas integral
* Objetivo: Garantizar la calidad y correcto funcionamiento del sistema multiplataforma de control de inventario y clientes, asegurando su accesibilidad desde cualquier dispositivo con acceso web.
* Alcance
* Pruebas sobre la funcionalidad del sistema en diversos dispositivos y navegadores.
* Evaluación de rendimiento, seguridad y compatibilidad
* Verificación de gestión de usuarios, clientes e inventario
* Cronograma de Actividades



* Roles y Responsabilidades
* Líder de Pruebas: Diseña estrategias de prueba y coordina actividades.
* Analistas de Calidad: Ejecutan pruebas y documentan incidencias.
* Desarrolladores: Realizan corrección de errores y pruebas de integración.
* Equipo de Infraestructura: Configura y mantiene entornos de prueba.
* Gerente de Calidad: Evalúa resultados y define mejoras.
* Estándares de calidad
* ISO 25010: Evaluación de calidad del software.
* IEEE 829: Documentación de pruebas.
* OWASP: Seguridad en aplicaciones web.
* Buenas prácticas Agile: Aplicables si el equipo usa metodologías ágiles.
* Uso de herramientas automatizadas para pruebas de regresión y estrés.
* Reglas para Versionado y Manejo de incidencias
* Uso de Git como sistema de control de versiones.
  + Nomenclatura basada en SemVer (ejemplo: v1.0.0).
  + Creación de ramas feature/ para nuevas funcionalidades, hotfix/ para correcciones urgentes.
  + Implementación de pull requests y revisiones de código antes de integrar cambios.
* Manejo de incidencias
* Registro de incidencias en una herramienta como JIRA o Azure DevOps.
* Priorización de errores según criticidad (alta, media, baja).
* Asignación de responsables y seguimiento de resolución.
* Pruebas de regresión tras la corrección de errores antes de su despliegue en producción.

Conclusión

El plan de prueba es un documento detallado que describe la estrategia de prueba, objetivos, cronograma, estimación, entregables y los recursos necesarios para realizar pruebas de un producto de software, el plan de prueba nos ayuda a determinar el esfuerzo necesario para validar la calidad de la aplicación bajo prueba. La importancia de elaborar un documento del plan de prueba tiene múltiples beneficios, ayudar a personas ajenas al equipo prueba, como lo son desarrolladores, gerentes comerciales y clientes, con el plan de prueba guias el pensamiento, ya que es como un libro de reglas que hay que seguir con aspectos importantes como la estimación de la prueba, alcance de prueba, estrategia de prueba que se están documentado en el plan de pruebas.

Los beneficios son extensamente diversos beneficios con la planificación efectiva, la planificación de pruebas de software es un componente esencial para garantizar la calidad y el éxito de cualquier proyecto de desarrollo de software. Al invertir tiempo en una planificación cuidadosa y estratégica, puedes identificar problemas de manera temprana, optimizar recursos y reducir riesgos, lo que resulta en un producto final de alta calidad que cumple con las expectativas del usuario.

Referencias

(Hamilton, 2025)