

Plan de pruebas de software

[Calculadora]

Fecha: [20/03/2021]

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	5
Pruebas de Regresión	6
Funcionalidades a No Probar	6
Enfoque de Pruebas (Estrategia)	6
Criterios de Aceptación o Rechazo.....	7
Criterios de Aceptación o Rechazo	7
Criterios de Suspensión.....	7
Criterios de Reanudación	8
Entregables.....	8
Recursos.....	9
Requerimientos de Entornos – Hardware	9
Requerimientos de Entornos – Software.....	9
Herramientas de Pruebas Requeridas	10
Personal	10
Entrenamiento	10
Planificación y Organización	10
Procedimientos para las Pruebas.....	10

Matriz de Responsabilidades.....	11
Cronograma	11
Premisas.....	11
Dependencias y Riesgos	12
Referencias.....	13
Glosario.....	13

Historial de versiones

Fecha	Versión	Autor	Organización	Descripción
20/03/2021	1	José Antonio Blancas Rodríguez		Prueba de calculadora de interfaz y funcionamiento.

Información del proyecto

Empresa / Organización	Cesur
Proyecto	Calculadora
Fecha de preparación	
Cliente	Bernat
Patrocinador principal	Cesur
Gerente / Líder de proyecto	José Antonio Blancas Rodríguez
Gerente / Líder de pruebas de software	José Antonio Blancas Rodríguez

Aprobaciones

Nombre y Apellido	Cargo	Departamento u organización	Fecha	Firma
José Antonio Blancas Rodríguez	Alumno	Desarrollo de aplicaciones multiplataforma	20/03/2021	JoseA

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.

Crear una calculadora para un trabajo de clase que tenga buena interfaz, fácil de leer y entender y tenga un funcionamiento básico.

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

Funcionalidad de los botones de la calculadora, el panel donde se muestran los números y/o el resultado final de las operaciones.

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.

Se prueba si los botones funcionan, es decir, al pulsar un numero dicho numero aparezca en el panel de texto y al pulsar cualquier tecla de operación (suma, resta, multiplicación y/o división) realice dicha operación.

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.

Comprobar que la suma, resta, multiplicación y división se realiza de forma correcta con varios tamaños de números ya sean números pequeños por ejemplo entre el 1-99 o números grandes entre el 1000-9999 y con números decimales.

Funcionalidades a no probar

Listado de las funcionalidades que no se van a probar. Debe incluir información de las razones por las cuales no se van a probar y los riesgos que se están asumiendo.

Operaciones complejas en la calculadora con números elevados, paréntesis, porcentajes...

Enfoque de pruebas (estrategia)

La estrategia de pruebas puede definirse como un documento por separado, 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.

.

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.

Funcionamiento correcto de las funciones básicas del programa.

Criterios de suspensión

Establece claramente bajo qué condiciones se detienen un conjunto de casos de pruebas, por ejemplo en caso de existir defectos que impidan la ejecución de más casos de pruebas, cierto porcentaje de casos fallidos, o cualquier otro que se especifique.

Fallo de los botones de operaciones simples y fallo en el resultado de dichas operaciones en la calculadora.

Criterios de reanudación

Luego de haber suspendido las pruebas, aquí se establece bajo qué criterios se reanudarán.

Correcto funcionamiento de las operaciones que fallaban para llevar a cabo la suspensión de las pruebas del producto.

Entregables

Establece que se entregará como parte de la ejecución del plan, por ejemplo: Documento de plan de pruebas, casos de pruebas, especificación de diseño de casos, logs de errores, reportes de incidencias, evidencias de pruebas, reportes emitidos por herramientas de pruebas y cualquier otro que se establezca.

Se entregará el documento de pruebas, el JavaDOC, el proyecto para poder ser ejecutado y ser probado.

Recursos

Requerimientos de entornos – Hardware

Lista de los requerimientos de equipos, hardware y red necesarios para completar las actividades del plan de pruebas de software. Incluye servidores de aplicación, bases de datos, equipos de PC que necesitan los Testers, conectividad a la red (incluyendo accesos), entre otros.

Requiere el poder ejecutar Java con su respectivo JDK.

Requerimientos de entornos – Software

Lista de los requerimientos de software necesarios para completar las actividades de prueba, puede incluir accesos a Sistemas (en entorno de pruebas) y bases de datos, así como instalación de software en los Computadores asignados a los Testers.

Puede usar los programas como Eclipse o Visual Studio Code, el programa de la calculadora ha sido desarrollado con Visual Studio Code.

Herramientas de pruebas requeridas

Especifica las herramientas de software, metodologías o técnicas especiales empleadas en las pruebas, por ejemplo herramientas de automatización de pruebas, software de gestión de pruebas, entre otros.

Personal

Lista del personal necesario para completar las actividades de pruebas, especificando sus roles, por ejemplo: Un (1) líder de pruebas, cinco (5) analistas de pruebas (Testers), dos (2) especialistas en automatización de pruebas, entre otros.

**Es un trabajo de clase individual por lo que solo he participado yo en este proyecto:
José Antonio Blancas Rodríguez**

Entrenamiento

Necesidades de entrenamiento en el sistema o aplicación, así como en las herramientas de prueba a utilizar.

Planificación y organización

Procedimientos para las pruebas

Especifica los procedimientos o metodología de pruebas a emplear durante la ejecución del plan de pruebas de software.

Matriz de responsabilidades

Lista cada una de las personas integrantes del equipo de QA y sus responsabilidades. Se puede hacer uso de una matriz RACI (responsable, aprobador, consultado, informado).

Cronograma

Debe estar basado en estimaciones de actividades realizadas por el equipo de prueba. En él se identifican los hitos relevantes en las pruebas de software, se establecen las dependencias (actividades predecesoras) y demás aspectos componentes de un cronograma.

Premisas

Las premisas relacionadas con las tareas de pruebas de software, incluyendo limitaciones de tiempo, disponibilidad de recursos que se asumen, uso de una metodología de pruebas, uso de una herramienta, entre otros.

Disponibilidad de recursos limitado, tiempo limitado, no se ha tenido el suficiente tiempo ni medios necesarios para desarrollar de buena forma el proyecto.

Dependencias y Riesgos

Aquí se listan los riesgos asociados con el proceso de pruebas de software, por ejemplo, algunas fuentes de riesgos suelen ser:

- Dependencias con desarrollos.
- Dependencias con otros proyectos.
- Disponibilidad de recursos.
- Restricciones de tiempo.
- Premisas que resulten no ser ciertas.

Los riesgos se pueden clasificar en función de su probabilidad e impacto, cada uno debe contemplar un plan de mitigación para evitar que ocurra o plan de contingencia cuando el riesgo no puede mitigarse y tiene que aceptarse.

Referencias

Lista de todos los documentos que pueden citarse como apoyo o para ampliar el contenido del plan de pruebas. Algunos ejemplos de lo que se puede hacer referencia aquí son:

- Plan de proyecto.
- Especificaciones de requerimientos.
- Diseño general.
- Diseño detallado.
- Procedimientos y estándares de desarrollo.
- Procedimientos y estándares de pruebas.
- Metodologías, procedimientos y estándares corporativos.

Glosario

Definiciones de términos usados en la documentación, y general sobre el área de pruebas.