**Plan de pruebas de software**

***Roberto L. Mora***

***Fecha: [09/04/2021]***

**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 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** |
| **20/03**  **2021** |  | **Roberto L. Mora** |  | **Primera entrega del trabajo con el código del programa** |
| **08/04**  **2021** |  | **Roberto L. Mora** |  | **Creación de pruebas JUnit y documentación Java Doc** |
| **09/04**  **2021** |  | **Roberto L. Mora** |  | **Desarrollo del plan de pruebas y alojamiento en la página de github** |
|  |  |  |  |  |

# Información del proyecto

|  |  |
| --- | --- |
|  |  |
| Crear una calculadora sencilla con test Junit implementados, documentación java doc y plan de pruebas |  |
|  |  |
|  |  |

# Aprobaciones

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

# 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.

Es un plan de pruebas detallado sobre el proyecto \*Calculadora”

que muestra sus funcionalidades y características, criterios, pruebas…

# 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.

La calculadora se ha realizado en el entorno de desarrollo Eclipse y se ha utilizado la página web de GitHub y Microsoft Teams para los apuntes.

## 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.

La calculadora cuenta con las características y funciones de una básica; los números del 0 al 9, los signos de suma, resta, división y multiplicación y CE para borrar.

## 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.

Las funciones que hacen los botones y ventanas de esta, que no son directamente el software pero son funcionales y hay que tenerlos en cuenta.

## 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.

Funciones como qué pasaría al dividir 0 entre otros números o añadir demasiados números y otras operaciones no válidas no se prueban. Se asume el riesgo de que falle y colapse el programa .

## 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.

Las pruebas unitarias JUnit son exitosas al 100%, aunque no todos los métodos del

Código se incluye en estas.

## 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.

No tiene defectos que impida la ejecución

## Criterios de reanudación

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

Se reanudan con el criterio de verificar el correcto funcionamiento de todo

# 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.

* Programa en Java con la calculadora (main y métodos asociados)
* Documentación JavaDoc generada en Html
* Plan de pruebas en Word
* Implementación de Test Junit

# 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.

Requerido un Pc con un entorno de desarrollo y conexión a la red.

## 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.

Requerimiento de un sistema operativo estable que pueda ejecutar el programa

donde desarrollamos el codigo

## 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.

No se utilizaron herramientas de gestión de pruebas.

## 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.

Trabajo totalmente personal, con un único usuario desarrollador (Roberto L.)

## Entrenamiento

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

Entrenamiento adquirido durante las sesiones de clase de Entornos.

# 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).

----

1 único responsable del proyecto.

## 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.

Los recursos utilizados son libres para cada uno, tanto apuntes de clase como de la web. Limitación de tiempo de entrega de dos semanas para entregar el trabajo

## 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.

Los problemas y riesgos principales encontrados fueron el poco conocimiento previo sobre un programa como este y tener que crearlo y trabajar solo en vez de en pareja

# 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.