**Plan de Pruebas**

**1. Cómo escribir un caso de prueba**

Un caso de prueba es un conjunto de condiciones o variables bajo las cuales un probador determinará si una aplicación, sistema, o una de sus características está funcionando como se espera. Aquí están los pasos para escribir un caso de prueba efectivo:

* **Identificación del Caso de Prueba**: Asignar un ID único al caso de prueba para facilitar su rastreo.
* **Título del Caso de Prueba**: Escribir un título claro y conciso que resuma lo que el caso de prueba está verificando.
* **Descripción**: Describir brevemente el propósito del caso de prueba y qué funcionalidad está probando.
* **Precondiciones**: Listar cualquier condición que deba cumplirse antes de ejecutar el caso de prueba (por ejemplo, el usuario debe estar logueado).
* **Datos de Prueba**: Especificar los datos que se necesitan para ejecutar el caso de prueba.
* **Pasos**: Detallar los pasos que deben seguirse para ejecutar el caso de prueba. Los pasos deben ser claros, secuenciales y detallados.
* **Resultados Esperados**: Describir el resultado que se espera en cada paso o al final de la ejecución del caso de prueba.
* **Estatus**: Indicar si el caso de prueba pasó o falló basándose en la comparación entre el resultado real y el esperado.

**2. Cómo reportar un defecto**

Cuando se encuentra un defecto durante las pruebas, es crucial reportarlo de manera clara y detallada para que los desarrolladores puedan entenderlo y resolverlo. Aquí están los pasos para reportar un defecto:

* **ID del Defecto**: Asignar un ID único al defecto para su rastreo.
* **Título**: Escribir un título claro y conciso que describa el problema.
* **Descripción**: Describir detalladamente el defecto, incluyendo en qué contexto ocurre y qué es lo que falla.
* **Pasos para Reproducir**: Enumerar los pasos exactos que llevaron al defecto. Esto ayuda a los desarrolladores a replicar el problema.
* **Resultado Esperado**: Describir lo que debería haber sucedido si no hubiera habido un defecto.
* **Resultado Real**: Describir lo que realmente sucedió cuando se encontró el defecto.
* **Severidad**: Indicar la gravedad del defecto (Crítica, Alta, Media, Baja).
* **Prioridad**: Indicar la prioridad del defecto para su resolución (Alta, Media, Baja).
* **Entorno de Pruebas**: Especificar el entorno donde se encontró el defecto (Sistema operativo, navegador, versión del software).
* **Adjuntos**: Incluir capturas de pantalla, registros o cualquier evidencia que ayude a entender el defecto.

**3. Criterio para incluir un caso de prueba en una suite de humo**

La suite de pruebas de humo incluye casos de prueba que validan la funcionalidad básica y crítica de la aplicación. Estos casos son ejecutados para asegurarse de que las funciones esenciales del sistema funcionan antes de proceder a pruebas más detalladas. Un caso de prueba debe cumplir con los siguientes criterios para ser incluido en una suite de humo:

* **Verifica Funcionalidades Críticas**: El caso de prueba debe validar características o funciones que son esenciales para el funcionamiento de la aplicación (ej. inicio de sesión, navegación básica).
* **Prueba de Camino Feliz**: Se enfoca en escenarios donde el usuario sigue el flujo esperado sin desviaciones.
* **Alta Prioridad**: Debe ser un caso de prueba que cubra áreas que, si fallan, impedirían la ejecución de pruebas adicionales.
* **Rápido de Ejecutar**: Debe poder ejecutarse rápidamente para proporcionar retroalimentación inmediata sobre la estabilidad del sistema.

**4. Criterio para incluir un caso de prueba en una suite de regresión**

La suite de pruebas de regresión está diseñada para verificar que las modificaciones en el código no han introducido nuevos defectos en áreas previamente funcionales. Un caso de prueba debe cumplir con los siguientes criterios para ser incluido en una suite de regresión:

* **Cobertura de Funcionalidades Existentes**: El caso de prueba debe cubrir funcionalidades que han sido previamente probadas y validadas como correctas.
* **Áreas Afectadas por Cambios Recientes**: El caso debe incluirse si la funcionalidad que cubre ha sido modificada o es susceptible de ser impactada por cambios recientes en el código.
* **Historial de Defectos**: Casos de prueba que han revelado defectos en el pasado son candidatos ideales para la suite de regresión.
* **Impacto Crítico**: Casos de prueba que verifican funcionalidades críticas o de alto impacto en la aplicación deben ser considerados para la suite de regresión.
* **Alta Prioridad**: Casos que aseguren la estabilidad y confiabilidad de las funciones principales del sistema.