Pruebas de software

La prueba de software es un método para verificar si el producto de software real coincide con los requisitos esperados y para garantizar que el producto de software no tenga defectos. Implica la ejecución de componentes de software / sistema utilizando herramientas manuales o automatizadas para evaluar una o más propiedades de interés. El propósito de las pruebas de software es identificar errores, lagunas o requisitos faltantes en contraste

¿Por qué las pruebas de software son importantes?
La prueba de software es importante porque si hay algún error o error en el software, se puede identificar temprano y se puede resolver antes de la entrega del producto de software. El producto de software debidamente probado garantiza fiabilidad, seguridad y alto rendimiento, lo que se traduce en ahorro de tiempo, rentabilidad y satisfacción del cliente.

Las pruebas son importantes porque los errores de software pueden ser costosos o incluso peligrosos. Los errores de software pueden causar pérdidas monetarias y humanas, y la historia está llena de ejemplos de este tipo.

¿Cuáles son los beneficios de las pruebas de software?
Estos son los beneficios de utilizar las pruebas de software:

Rentable: es una de las ventajas importantes de las pruebas de software. Probar cualquier proyecto de TI a tiempo le ayuda a ahorrar dinero a largo plazo. En caso de que los errores se detecten en la etapa anterior de la prueba del software, cuesta menos corregirlos.
Seguridad: es el beneficio más vulnerable y sensible de las pruebas de software. La gente busca productos de confianza. Ayuda a eliminar riesgos y problemas antes.
Calidad del producto: Es un requisito esencial de cualquier producto de software. Las pruebas garantizan que se entregue un producto de calidad a los clientes.
Satisfacción del cliente: El objetivo principal de cualquier producto es dar satisfacción a sus clientes. Las pruebas de UI / UX garantizan la mejor experiencia de usuario.

Tipos de pruebas de software
Normalmente, las pruebas se clasifican en tres categorías.

Pruebas funcionales
Pruebas no funcionales o pruebas de rendimiento
Mantenimiento (regresión y mantenimiento)

Definición Pruebas Funcionales
Las pruebas funcionales se definen teniendo como fuente los requisitos del sistema, estas pruebas validan y verifican que el producto cumple con lo especificado y hace lo que debe y cómo lo tiene que hacer dando también una idea del grado de calidad del software.

Definición Pruebas No Funcionales
Las pruebas No Funcionales se centran en aspectos muy importantes del comportamiento del producto pero que no están relacionados con las funciones que realiza el sistema.

Nos centraremos el tipo de pruebas funcionales :

Las pruebas unitarias
son a bajo nivel (cercanas al código fuente de nuestra aplicación).
Este tipo de testing consiste en probar de forma individual las funciones y/o métodos (de las clases, componentes y/o módulos que son usados por nuestro software).

Debido a lo específicas que son, generalmente son las pruebas automatizadas de menor coste, y pueden ejecutarse rápidamente por un servidor de continuous integration (integración continua).

Más detalles acerca de las pruebas unitarias:

Idealmente, cuando planeamos y escribimos pruebas unitarias, debemos aislar la funcionalidad hasta un punto en que no se pueda desglosar más, y entonces escribir pruebas a partir de ello. Justamente, el nombre de este tipo de testing hace referencia a una "unidad de código", que es independiente del resto.

Estas pruebas verifican que el nombre de la función o método sea adecuado, que los nombres y tipos de los parámetros sean correctos, y así mismo el tipo y valor de lo que se devuelve como resultado.

Dado que las pruebas unitarias no deben tener ningún tipo de dependencia, se suele reemplazar los llamados a APIs y servicios externos por funcionalidad que los imite (para que no exista interacción que vaya más allá de la unidad que está siendo probanda).

En muchos casos inclusive se suele reemplazar las consultas a bases de datos, de modo que la prueba se enfoque en operar a partir de los valores de entrada, sin depender de ninguna fuente externa.

Si no es factible aislar el uso de bases de datos de nuestras pruebas unitarias, será importante tener en cuenta el rendimiento y buscar optimizar nuestras consultas.

Esto es importante, porque si nuestras pruebas unitarias son de larga duración, resultará incómodo ejecutarlas y ralentizará significativamente los tiempos de desarrollo.

Las pruebas de integración verifican que los diferentes módulos y/o servicios usados por nuestra aplicación funcionen en armonía cuando trabajan en conjunto.

Las pruebas de regresión verifican un conjunto de escenarios que funcionaron correctamente en el pasado, para asegurar que continúen así.

Las pruebas de humo son pruebas que verifican la funcionalidad básica de una aplicación.

Se pretende que sean pruebas rápidas de ejecutar,
y su objetivo es asegurar que las características más importantes del sistema funcionan como se espera

Formato de casos de pruebas funcionales

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No.	*<Número del caso de prueba constituido [número del caso de uso]-[Numero del caso de prueba]>*	**VERSIÓN DE EJECUCIÓN**	*<Versión diligenciado por el analista de pruebas en el momento de ejecutarla. Este número se incrementa de 1 en 1>*
		FECHA EJECUCIÓN	*<Fecha de ejecución diligenciado por el analista de pruebas>*
CASO DE USO:	*<Identificación del caso de uso objeto de la prueba>*	**MODULO DEL SISTEMA**	*<Nombre del modulo al que corresponde el caso de uso objeto de la prueba>*
Descripción del caso de prueba:	*<Descripción de lo que se pretende probar en el caso de prueba>*		

1. CASO DE PRUEBA

a. Precondiciones

<Lista de precondiciones que deben cumplirse para realizar la prueba>

b. Pasos de la prueba

<Pasos secuenciales que deben ser ejecutados por el analista de pruebas o usuario, ante el sistema para ejecutar la prueba>

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	**VALOR**	**TIPO ESCENARIO**		**SI**	**NO**	
<Descripción del dato de entrada>	*<Valor que debe ser suministrado en la prueba para el dato de entrada>*	*<Tipo de escenario que pretende probarse: Correcto/Incorrecto>*	*<Respuesta que se espera de la aplicación>*			*<Respuesta que se obtuvo de la aplicación en el momento de la ejecución de la prueba>*

c. Post condiciones

<Lista de pos condiciones que deben cumplirse después de realizar la prueba>
2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	**Veredicto**
<Lista de defectos o desviaciones encontrados por el analista o usuario al ejecutar la prueba>	☐ **Paso** ☑ **Falló**

Observaciones	**Probador**
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	**Firma:** **Nombre:** **Fecha:**

.

Identificador	*CP - <<Número de identificación del CP>>*	***Versión***	*<<Número de versión del CP>>*
Responsable	*<Nombre del funcionario que realiza la prueba>*		
Nombre del caso de prueba	*<Escriba el nombre del caso de prueba>*		
Módulo		***Submódulo***	
<Escriba aquí el módulo al cual se realiza la prueba>		*<Escriba aquí el submódulo al cual se realiza la prueba>*	
Formulario			
<Escriba aquí el formulario en el cual se realiza la prueba>			
Descripción de la prueba			
<Escriba el procedimiento (pasos) en detalle de la prueba que va a realizar>			
Resultados esperados			
<Escriba aquí el resultado ideal de la prueba>			
Resultados reales			
<Escriba aquí el resultado real de la prueba realizada>			
Error			

<Describa el error que se encontró luego de realizar la prueba>
Imagen
<Pegue aquí la(s) imagen(es) del error encontrado>

Formato 1 Plan de pruebas de usuario

Resultados posibles de las pruebas

Nombre del resultado de la prueba	*Descripción*	*Impacto*
Correcta o superada	*El sistema funciona de acuerdo a lo solicitado en los requerimientos iniciales.*	
Con no conformidades de diseño	*El sistema presenta inconsistencias en el diseño del formulario y/o informes. P.eje. ubicación de un campo, líneas, sombreados, tamaño de la letra, etc.*	*Medio*
Con no conformidades de lógica	*El sistema no funciona como se especificó en los requerimientos iniciales del sistema.*	*Alto*
Requerimientos nuevos	*Requerimientos nunca solicitados al proveedor, que afectan el funcionamiento del sistema.*	*Alto*

1. CRONOGRAMA DE IMPLEMENTACION DE PRUEBAS

<Cronograma de las pruebas>

2. RESPONSABLES DE LAS PRUEBAS

<Relacione los responsables de las pruebas incluyendo los responsables funcionales y el grupo de pruebas>