

Adrián Tirado Ramos

1º De DAW

Entorno Desarrollo

CASOS DE PRUEBA Y DISTINTOS
TIPOS DE PRUEBA

ÍNDICE

| | |
|--|----------|
| 1. CASOS DE PRUEBA..... | 2 |
| 2. PRUEBAS FUNCIONALES Y NO FUNCIONALES | 3 |
| PRUEBAS FUNCIONALES..... | 3 |
| PRUEBAS NO FUNCIONALES | 3 |
| 3. PRUEBAS ESTRUCTURALES | 4 |
| 4. PRUEBAS DE REGRESIÓN | 4 |
| 5. PRUEBAS DE CAJA BLANCA..... | 4 |
| 6. PRUEBAS DE CAJA NEGRA | 5 |

1. Casos de prueba

Los casos de prueba son conjuntos de condiciones o variables que se diseñan para verificar la correcta ejecución de una funcionalidad específica de un sistema o aplicación.

Cada caso de prueba se centra en una parte específica del software y describe los pasos a seguir, los datos de entrada y el resultado esperado.

El objetivo es evaluar si el software cumple con los requisitos establecidos.

Ejemplo: Supongamos que estamos probando una función de inicio de sesión en una aplicación. Un caso de prueba podría ser:

- Paso 1: Abrir la aplicación.
- Paso 2: Introducir un nombre de usuario válido.
- Paso 3: Introducir una contraseña válida.
- Paso 4: Hacer clic en el botón de inicio de sesión.

Resultado esperado: El sistema debe permitir el acceso y mostrar la pantalla principal.

2. Pruebas funcionales y no funcionales

Pruebas funcionales

Evalúan las funciones específicas del sistema y aseguran que cumplan con los requisitos funcionales.

Se centran en lo que hace el sistema.

Ejemplo:

Para una aplicación de comercio electrónico, una prueba funcional podría ser verificar si se pueden agregar productos al carrito y completar el proceso de compra.

Pruebas no funcionales

Evalúan aspectos no relacionados directamente con las funciones específicas del sistema, como el rendimiento, la seguridad y la usabilidad.

Se centran en cómo se realiza una función.

Ejemplo:

Una prueba no funcional podría ser evaluar el tiempo de carga de una página en la aplicación de comercio electrónico.

3. Pruebas estructurales

Las pruebas estructurales, también conocidas como pruebas de caja blanca, se centran en evaluar la estructura interna del software.

Se analiza el código fuente para garantizar que todas las partes sean ejecutadas y que se cumplan las condiciones lógicas.

Ejemplo:

Supongamos que tenemos una función en un programa que realiza una operación matemática compleja.

Una prueba estructural podría involucrar la revisión del código para asegurarse de que todas las ramas condicionales se ejecuten correctamente.

4. Pruebas de regresión

Las pruebas de regresión se realizan para asegurar que las nuevas modificaciones en el software no afecten negativamente a las funcionalidades existentes.

Se ejecutan pruebas previamente realizadas para detectar posibles regresiones o errores introducidos por cambios recientes.

Ejemplo:

Si se introduce una nueva función en una aplicación de procesamiento de texto, las pruebas de regresión se asegurarían de que las funciones de edición y guardado existentes sigan funcionando como se espera después de la actualización.

5. Pruebas de caja blanca

Las pruebas de caja blanca implican el examen directo del código fuente y la lógica interna del software.

Los casos de prueba se diseñan en base a la estructura interna del código para asegurar una cobertura completa.

Ejemplo:

En una función que realiza una operación matemática, una prueba de caja blanca podría involucrar la evaluación de las variables internas y la verificación de que los cálculos se realizan correctamente.

6. Pruebas de caja negra

Las pruebas de caja negra se centran en evaluar la funcionalidad del software sin conocer su estructura interna.

Los casos de prueba se diseñan con base en los requisitos y la entrada esperada, sin considerar cómo se implementa el código.

Ejemplo:

En una prueba de caja negra para una aplicación de procesamiento de imágenes, podríamos ingresar diferentes tipos de imágenes y verificar si la salida es la esperada sin necesidad de conocer la lógica interna del procesamiento de imágenes.