
Entornos de Desarrollo

PRUEBAS DE CAJA NEGRA



DEFINICIÓN

Se pretende verificar la funcionalidad de cada componente de la aplicación.

NO se pretende comprobar el código desarrollado, ni las estructuras de datos utilizadas, ni la eficiencia, ni partes de código innecesarias.

Se trata de comprobar si dados unos valores de entrada, las salidas son las esperadas.

Habitualmente se denominan **pruebas funcionales**.

TIPOS DE PRUEBAS FUNCIONALES

- **Particiones equivalentes:** se trata de generar el menor número posible de casos de prueba, por lo que cada caso de prueba tiene que abarcar el mayor número posible de entradas diferentes. Se crea un conjunto de clases de equivalencia, donde la prueba de un valor representativo sería extrapolable para cualquier valor de la clase.
- **Análisis de valores límite:** para cada caso de prueba, se eligen como valores de entrada los que se encuentran en el límite de las clases de equivalencia.
- **Pruebas aleatorias:** consiste en generar valores aleatorios de entrada para cada caso de prueba, habitualmente utilizando generadores automáticos.
- **Conjetura de errores:** se generan casos de prueba que la experiencia ha demostrado que habitualmente detectan errores.

TÉCNICA DE DISEÑO

Para el diseño de los **casos de prueba** que garanticen las pruebas funcionales, se deberían realizar los siguientes pasos:

- Determinar las **clases de equivalencia**.
- Realizar el **análisis de valores límite**.
- Añadir las **conjeturas de errores**.
- Generar los **casos de prueba** necesarios para probar las clases de equivalencia válidas y no válidas, determinando los datos de entrada y resultados esperados.
- Lanzar una **ejecución del programa** por cada caso de uso y **comparar los resultados obtenidos con los esperados** para comprobar el código.

PASO 1: clases de equivalencia

Se basa en identificar las **condiciones de entrada al programa**, es decir las restricciones de formato o valor de los datos de entrada.

A partir de dichas condiciones identificadas, definir clases de equivalencia para **datos válidos** y para **datos no válidos**, a partir de las siguientes reglas:

		Clases de equivalencia	
Tipo de dato	Ejemplo	Válida	No Válida
Rango de valores	La edad está comprendida entre 18 y 100 años	Valor entre 18 a 100	Valor menor de 18 y Valor mayor de 100
Número finito y consecutivo de valores	Una encuesta puede tomar valores 0, 1, 2 y 3.	Valor entre 0 y 3	Valor menor de 0 y Valor mayor de 3
Verdadero / Falso	Ser mayor de edad	Valor ≥ 18	Valor < 18
Múltiples valores	Opción de menú con 'A'=alta, 'B'=baja y 'S'=salir	Cualquiera de los tres valores ('A', 'B' o 'S')	Valor = 'X'

PASO 2: análisis de valores límite (AVL)

Se diferencia del anterior en que para los rangos de valores y los números finitos y consecutivos de valores, realizaremos la prueba con los valores límite.

		Valores límite	
Tipo de dato	Ejemplo	Válida	No Válida
Rango de valores	La edad está comprendida entre 18 y 100 años	Caso 1: 18 Caso 2: 100	Caso 3: 17 Caso 4: 101
Número finito y consecutivo de valores	Una encuesta puede tomar valores 0, 1, 2 y 3.	Caso 1: 0 Caso 2: 3	Caso 3: -1 Caso 4: 4

PASO 3: conjetura de errores

La experiencia dice que existen ciertos valores de entrada que habitualmente generan errores, y en ocasiones, pasan desapercibidos en las técnicas de clases de equivalencia y de valores límite.

Consiste en definir nuevos casos de prueba para esos datos de entrada, que conoceremos de una **forma intuitiva a través de la experiencia**.

Por ejemplo, probar el valor 0 para datos numéricos por si pudiera participar como denominador en alguna división durante la ejecución del software.

PASO 4: casos de prueba

Se definirán tantos **casos de prueba** como clases de equivalencia y valores límite, tanto validos como no válidos, y añadiendo también las conjeturas de errores.

En este proceso, se intentarán **agrupar** los casos de prueba tanto como sea posible y se definirán los **resultados previstos**.