Módulo: Entornos de Desarrollo

Unidad 5: Diseño y realización de Pruebas

Pruebas de caja blanca

Durante esta unidad se han estudiado los distintos tipos de pruebas software, a las que se puede someter la aplicación software desarrollada, para practicar las pruebas software con un enfoque estructural o de caja blanca, se propone la realización del siguiente supuesto práctico.

Diseñar las pruebas de caja blanca, para el siguiente método:

```
public static int getMax(int [] lista){

        int indice, max=Integer.MAX_VALUE;

        for(indice=0; indice<lista.length-1; indice++){
            if(lista[indice]<max){
                max=lista[indice];
            }
        }

        return max;

}
```

Se pide someter el método anteriormente, a unas pruebas software de caja blanca utilizando la técnica del cubrimiento de caminos básicos, para ello se tendrán que llevar a cabo las siguientes acciones:

- Identificar los caminos válidos
- Identificar los caminos inválidos
- Comprobar que caminos se ejecutan realmente y cuáles no.

Objetivos:

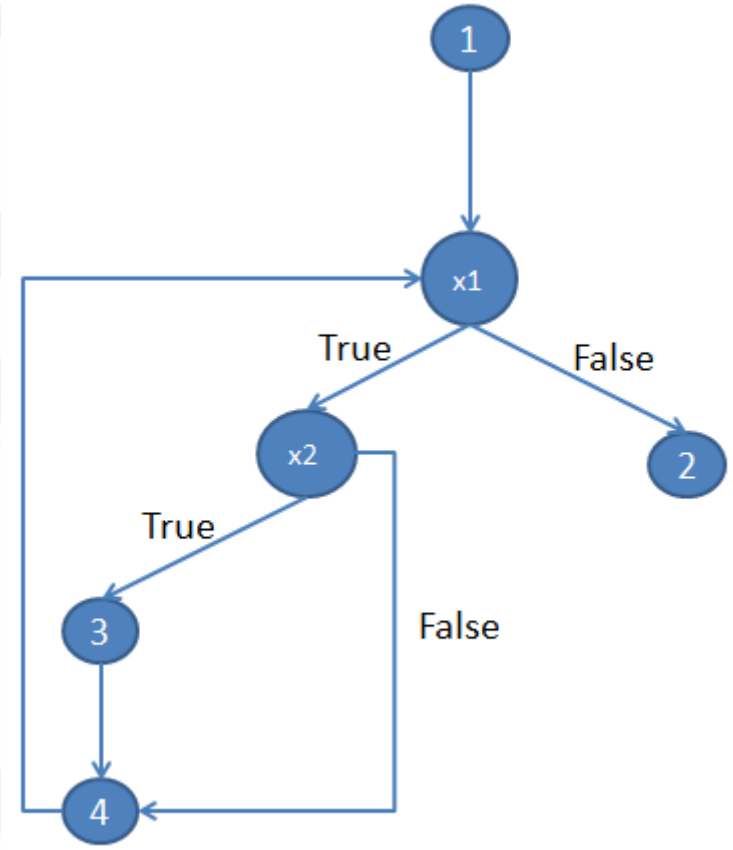
- Identificar los diferentes tipos de pruebas.
- Definir casos de prueba.
- Documentar el plan de pruebas

Recursos:

- Acceso a Internet.
- Procesador de texto

Resolución:

```
public static int GetMax(int[] lista){
        int indice, max = Int32.MaxValue;
        for(indice=0;indice<lista.Length-1;indice++){
                if (lista[indice]>max){
                    max = lista[indice];
                }
        }
                return max;
}
```



Definición de Caminos Válidos

Camino 1: 1-X1-2

Camino 2: 1-X1-X2-4-X1-2

Camino 3: 1-X1-X2-3-4-X1-2

El método debe recibir una lista de números enteros.

Casos de prueba:

1) Se le manda una lista vacía: debe devolver el valor inicial de Max y ejecutar sólo el camino 1, el caso se pasa con éxito

2) Se le manda una lista con 1 elemento: debe devolver el valor del elemento único y ejecutar el camino3 una vez. El caso falla, devuelve 2147483647, sólo se ejecuta el camino1

3) Se le manda una lista n elementos: debe devolver el elemento mayor y ejecutar los caminos 2 o 3 n veces. El caso falla, devuelve 2147483647, el camino2 se ejecuta n-1 veces, y luego se ejecuta el camino1, el camino3 no se ejecuta.

En el caso de prueba 2 no se ejecuta el camino3 ni el 2 porque para una lista de longitud 1, la condición X1 no se cumple; podríamos poner <= o quitar el -1 en la condición X1.

En el caso de prueba 3, el bucle no recorre la lista por completo porque la condición X1 está mal, para una lista de n elementos, se ejecuta n-1; para solucionarlo habría que poner <= o quitar el -1 en la condición X1.

En los casos de prueba 2 y 3, además, devuelve ese número tan grande porque la condición X2 no se cumple nunca, puesto que los elementos de la lista son enteros, y ninguno de ellos será nunca mayor que el valor máximo de un entero, que es lo que se asigna a Max en el bloque de código 1. Debido a esto el bloque de código 3 no se alcanza nunca, y el valor de Max es siempre el inicial. La solución podría ser cambiar Max por Max=int32.MinValue, de esa manera cualquier elemento de la lista sería o igual al valor mínimo para un entero (con lo cual devolveríamos Max directamente) o mayor, y en ese caso sí se cumpliría la condición X2 y se ejecutaría el bloque de código 3.