ENTORNOS DE DESARROLLO UD-5

PRUEBAS DE CAJA BLANCA

ÍNDICE

[INTRODUCCIÓN 3](#_Toc130539583)

[CASO PRÁCTICO DE CAJA BLANCA 3](#_Toc130539584)

[A. OBTENCIÓN DEL GRAFO 4](#_Toc130539585)

[B. COMPLEJIDAD DE MCCABE O CICLOMÁTICA. 5](#_Toc130539586)

[C. CAMINOS DE PRUEBA 6](#_Toc130539587)

# INTRODUCCIÓN

Su función es comprobar que se van a ejecutar todas las instrucciones del programa, que no hay código no usado, comprobar que los caminos lógicos del programa se van a recorrer, etc.

Este tipo de pruebas, se basan en unos criterios de cobertura lógica, cuyo cumplimiento determina la mayor o menor seguridad en la detección de errores.

Los criterios de cobertura son:

* **Cobertura de sentencias**. Se han de generar casos de pruebas suficientes para que cada instrucción del programa sea ejecutada al menos una vez.
* **Cobertura de decisiones**. Se trata de crear los suficientes casos de prueba para que cada opción, resultado de una comprobación lógica del programa, se evalúe al menos una vez a true y otra a false.
* **Cobertura de condiciones**. Se trata de crear los suficientes casos de prueba para que cada elemento de la condición se evalúe al menos una vez a false y otra a true.
* **Cobertura de condiciones y decisiones**. Cumplir simultáneamente las dos anteriores.
* **Cobertura de camino de prueba**. Se pueden realizar dos variantes, una indica que cada bucle se debe ejecutar sólo una vez, ya que hacerlo más veces no aumenta la efectividad de la prueba y otra que recomienda que se pruebe cada bucle tres veces: la primera sin entrar en su interior, otra ejecutándolo una vez y otra más ejecutándolo al menos dos veces.

# CASO PRÁCTICO DE CAJA BLANCA

La técnica para determinar los casos de prueba de caja blanca que garantiza cobertura de sentencias, decisiones/condiciones y de caminos.

Se realiza completando los siguientes pasos:

1. Crear un grafo que represente el código a probar.
2. Calcular la complejidad ciclomática o de McCABE del grafo obtenido.
3. Determinar tantos caminos (recorridos del grafo) como la complejidad ciclomática calculada.
4. Generar un caso de uso por cada camino, determinando sus datos de entrada y los resultados esperados.
5. Lanzar una ejecución del programa por cada caso de uso y comparar los resultados obtenidos con los esperados para comprobar la corrección del código.

## OBTENCIÓN DEL GRAFO

Se trata de crear un grafo en base al tipo de instrucciones que vayamos encontrando en nuestro código. Los tipos de estructuras principales que aparecen en los programas son secuencias de instrucciones, condiciones e iteraciones. Éstas se representan como sigue en el grafo.

Diagrama

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Los grafos se construyen a partir de **nodos** y **aristas**. Los nodos representan secuencias de instrucciones consecutivas donde no hay alternativas en la ejecución o condiciones a evaluar, que en función del resultado hará que la ejecución siga una dirección u otra.

Las aristas son las encargadas de unir los nodos. En el caso de que las decisiones tengan múltiples condiciones, habrá que separar cada condición en un nodo como sigue:

Diagrama

Descripción generada automáticamente

## COMPLEJIDAD DE MCCABE O CICLOMÁTICA

La **complejidad de McCabe o ciclomática** es una medida del software que aporta una valoración cuantitativa de la **complejidad lógica** de un programa.

A partir del grafo se determina su complejidad ciclomática. Es posible hacerlo por tres métodos diferentes, pero todos ellos han de dar el mismo resultado.

* + V(G) = a - n + 2, siendo a el número de arcos o aristas del grafo y n el número de nodos.
  + V(G) = r, siendo r el número de regiones cerradas del grafo (incluida la externa).
  + V(G) = c + 1, siendo c el número de nodos de condición.

Diagrama

Descripción generada automáticamente

## CAMINOS DE PRUEBA

El número de caminos de prueba debe ser igual a la complejidad calculada. Consiste en hacer recorridos desde el inicio hasta el final del método con el que se esté trabajando. Se irán registrando los nodos por los que va pasando la ejecución del camino.

Cada nuevo camino deberá aportar el paso por nuevas aristas/nodos del grafo. La definición de caminos se hará desde los más sencillos a los más complicados.

Texto

Descripción generada automáticamente