

CAJA BLANCA

CHRISTIAN MILLÁN SORIA

1º DAW TARDE

1. Si tenemos el siguiente algoritmo, que averigua si un número es primo o no:

```
algoritmo primos
  escribir "Numero: "
  leer num

  cont<-2;
  swPrimo<-'V'

  mientras((cont<=(num/2))y(swPrimo == 'V'))hacer
    si((num mod cont)==0)entonces
      swPrimo<-F
    fin si

    cont<-cont+1
  fin mientras

  si(swPrimo=='V')entonces
    escribir num, " SI es primo"
  sino
    escribir num, " NO es primo"
  fin si
fin algoritmo
```

Indica varios casos de prueba posibles, así como algún procedimiento de prueba.

Algunos casos de prueba posibles podrían ser, por ejemplo:

- Caso de prueba válido: Ingresar el número 2, que es el número primo más pequeño.
- Caso de prueba válido: Ingresar el número 7, que es un número primo.
- Caso de prueba inválido: Ingresar el número 10, que no es un número primo.
- Caso de prueba válido: Ingresar el número 23, que es un número primo.
- Caso de prueba válido: Ingresar el número 29, que es un número primo.

Un posible procedimiento de prueba para verificar que el algoritmo funciona correctamente sería el siguiente (por pasos):

- Paso 1: Ingresar un número primo como entrada, por ejemplo, 7.
- Paso 2: Ejecutar el algoritmo.
- Paso 3: Verificar que el algoritmo indique que el número ingresado es un número primo.
- Paso 4: Repetir los pasos 1 a 3 con otros números primos para asegurarse de que el algoritmo los identifique correctamente.

- Paso 5: Ingresar un número que no sea primo, por ejemplo, 10.
- Paso 6: Ejecutar el algoritmo.
- Paso 7: Verificar que el algoritmo indique que el número ingresado no es un número primo.
- Paso 8: Repetir los pasos 5 a 7 con otros números que no sean primos para asegurarse de que el algoritmo los identifique correctamente.
- Paso 9: Asegurarse de que el algoritmo no tenga errores de lógica para cualquier número de entrada mediante el análisis del código fuente y la verificación de los cálculos realizados en el código.

2. ¿Qué diferencia existe entre las pruebas de caja blanca y las de caja negra? En el ejemplo anterior, ¿en qué consistirían?

Las pruebas de caja blanca y las de caja negra son dos enfoques diferentes para realizar pruebas de software.

Las pruebas de caja negra se basan en probar el software desde el punto de vista del usuario final, sin conocer los detalles internos de cómo se implementa el software. En este enfoque, el software se trata como una "caja negra" y se prueban sus entradas y salidas sin considerar su funcionamiento interno. Las pruebas de caja negra se centran en evaluar si el software cumple con los requisitos funcionales y no funcionales, y si el software es fácil de usar y comprender.

Por otro lado, las pruebas de caja blanca se enfocan en evaluar el software desde su funcionamiento interno y se basan en el conocimiento detallado de la implementación del software. En este enfoque, el tester conoce la estructura interna del código fuente y de la aplicación, y se asegura de que el software se ejecute según lo previsto. Las pruebas de caja blanca se centran en la cobertura de código, en la verificación de la lógica interna del software, y en la identificación de posibles problemas de rendimiento y seguridad.

En el ejemplo anterior, las pruebas de caja blanca implicarían examinar el código fuente del algoritmo para diseñar casos de prueba que cubran todas las posibles rutas de ejecución, lo que se puede lograr mediante pruebas de cobertura de código para asegurarse de que todas las declaraciones y ramas de código se hayan ejecutado al menos una vez. Por otro lado, las pruebas de caja negra implicarían proporcionar diferentes entradas al algoritmo y verificar si las salidas corresponden a las salidas esperadas.

3. ¿En qué consiste la cobertura o cubrimiento? Busca en el ejemplo 1 los diferentes tipos de coberturas que existen, si no encuentras ninguno, plantea un ejemplo adicional.