Ejemplo de caja blanca

Código a probar:

```
public static double contarLetras(char[] cadena, char letra) {
    int contador = 0;
    int n = 0;
    int lon = 0;
    lon = cadena.length;
    if(lon > 0){
        do {
            if (cadena[contador] == letra) {
                n++;
            }
            contador++;
            lon--;
        } while (lon > 0);
    }
    return n;
}
```

Paso 1: Analizar el Código:

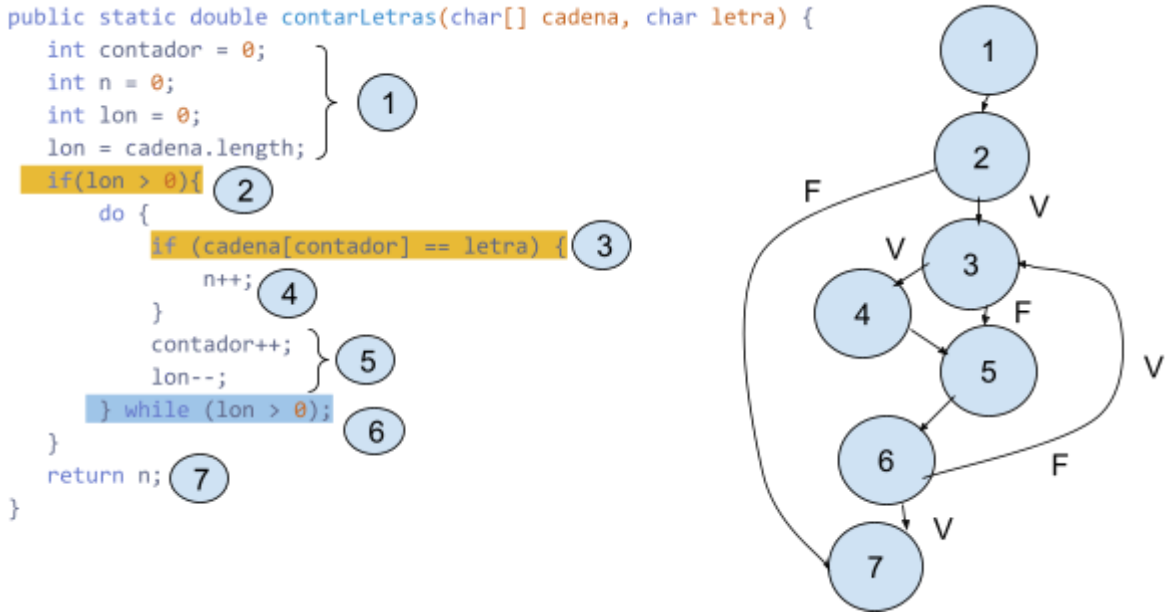
Descripción del Programa:
El método contarLetras recibe un array de caracteres (cadena) y una letra (letra) como entrada. Su propósito es contar cuántas veces aparece letra en el array.

Bloques Principales:

1. **Inicialización:** Se inicializan las variables contador, n (contador de coincidencias), y lon (longitud de la cadena).
2. **Condición inicial:** Si lon > 0, ejecuta un bucle do-while.
3. **Bucle do-while:**
    - Compara el carácter actual con letra.
    - Si coinciden, incrementa n.
    - Incrementa contador y decrementa lon.
4. **Retorno:** Devuelve n (el número de coincidencias).

```java
public static double contarLetras(char[] cadena, char letra) {
   int contador = 0;
   int n = 0;
   int lon = 0;
   lon = cadena.length;
   if(lon > 0){
       do {
           if (cadena[contador] == letra) {
               n++;
           }
           contador++;
           lon--;
       } while (lon > 0);
   }
   return n;
}
```

Paso 2: Realizar el diagrama de flujo de control:

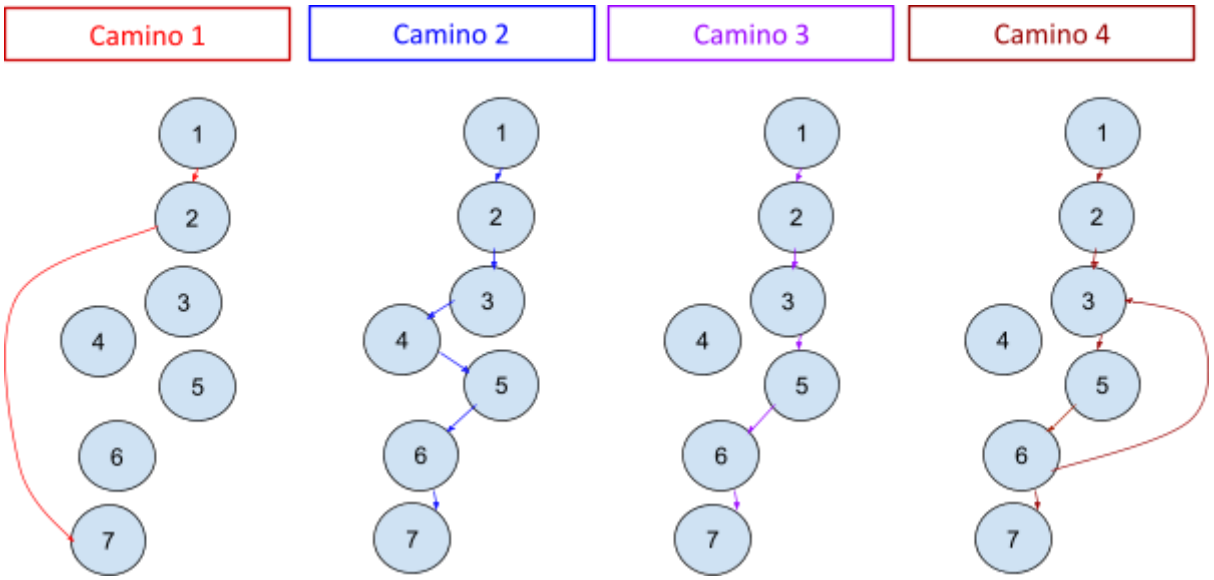


Paso 3: Cálculo de la complejidad ciclomática

$M = E - N + 2.$

M = 9 – 7 + 2 = 4

Paso 4: Identificar los caminos independientes



Atención: El camino 4 no es único, se podría haber obtenido la arista 6-3 pasando por 4.

Paso 5: Diseñar los casos de prueba

Camino 1:

- **Descripción: Cadena vacía**
- **Entrada: {}**
- **Salida esperada: 0**

Camino 2:

- **Descripción: Contiene la letra**
- **Entrada: {'a'}**
- **Salida esperada: 1**

Camino 3:

- **Descripción: No contiene la letra**
- **Entrada: {'b'}**
- **Salida esperada: 0**

Camino 4:

- **Descripción: Contiene varias letras pero ninguna es "a"**
- **Entrada: {'b', 'b'}**
- **Salida esperada: 0**

Paso 6: Ejecutar y validar

```
public static void main(String[] args) {
        char[] caso1 = {};
        char[] caso2 = {'a'};
        char[] caso3 = {'b'};
        char[] caso4 = {'b', 'b'};

        ArrayList<char[]> casosPrueba = new ArrayList<>();
        casosPrueba.add(caso1);
        casosPrueba.add(caso2);
        casosPrueba.add(caso3);
        casosPrueba.add(caso4);

        for (char[] caso : casosPrueba) {
            System.out.println("El caso a analizar es " + Arrays.toString(caso));
            System.out.println("El resultado es " + contarLetras(caso, 'a'));
        }

    }
```

```
El caso a analizar es []
El resultado es 0.0
El caso a analizar es [a]
El resultado es 1.0
El caso a analizar es [b]
El resultado es 0.0
El caso a analizar es [b, b]
El resultado es 0.0
```