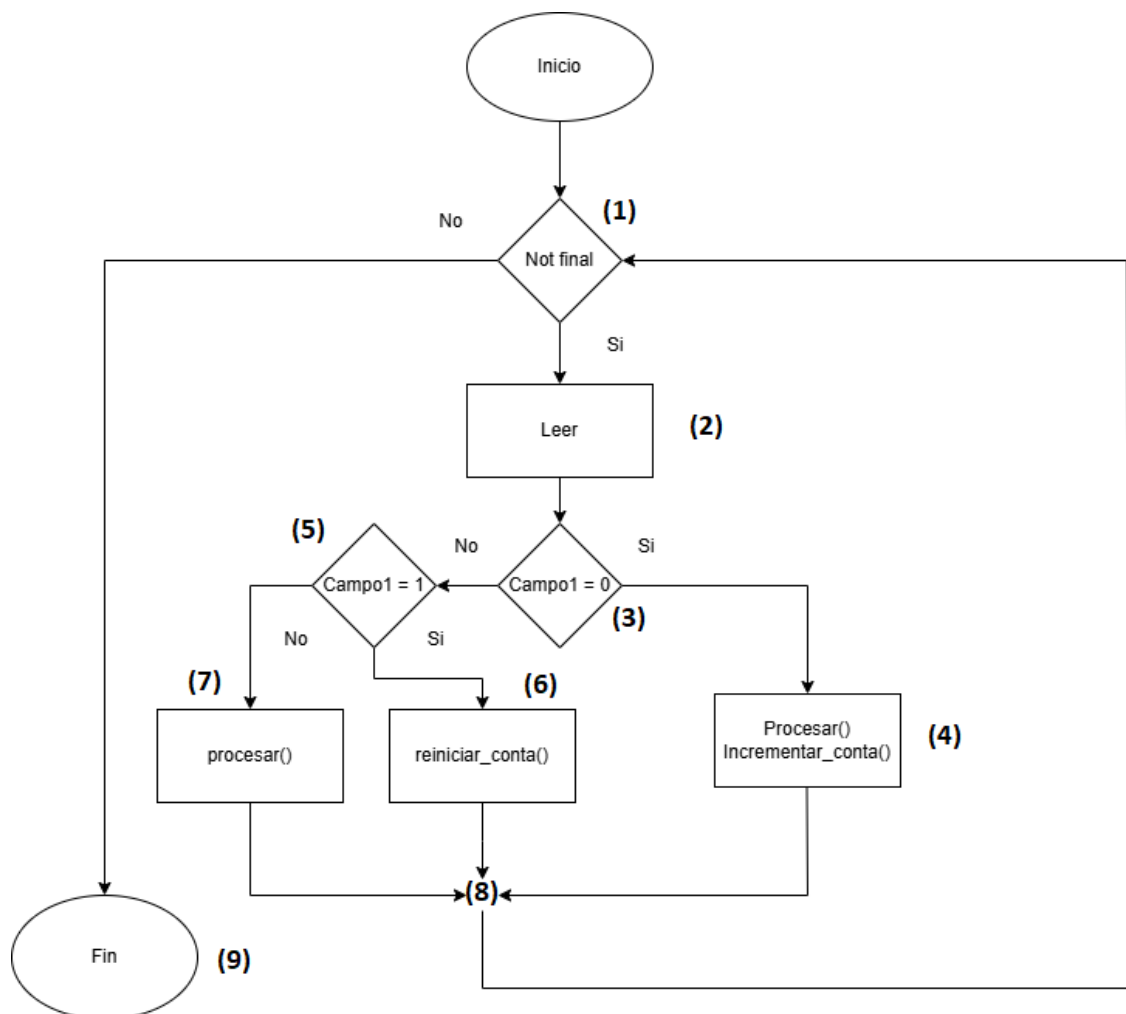


Pac Desarrollo de Entornos de desarrollo: Javi Escobar Fernández.

Pseudocódigo:

```
WHILE NOT final DO
  leer
  IF campo1=0 THEN
    procesar()
    incrementar_conta()
  ELSE IF campo1=1 THEN
    reiniciar_conta()
  ELSE
    procesar()
  END IF
END WHILE
```

Diagrama de flujo (no solicitado, pero me ayuda para hacer mejor el grafo, así que lo incluyo, en la siguiente página se incluye el grafo de flujo)



Grafo de Flujo:

- **Número total de aristas:** 11
- **Número total de nodos predicado:** 3 (los nodos 1, 3 y 5) Los nodos predicado son los nodos que contienen más de una salida (bifurcaciones)
- **Número total de nodos:** 9
- **Número total de regiones:** 4

Complejidad ciclomática: Tenemos 3 formas de calcular la complejidad

- 1- **$V(G) = \text{número de regiones}$:** 4 regiones que tiene el nodo, cada región representa un fragmento del grafo, delimitadas por las aristas y nodos, donde también contamos el exterior como una región más (la región 4 en este caso).
- 2- **$V(G) = \text{aristas} - \text{nodos} + 2$:** (11 aristas – 9 nodos) + 2 = 4
- 3- **$V(G) = \text{nodos predicado} + 1$:** (3 nodos predicado) + 1 = 4

Como podemos ver, en todas las fórmulas obtenemos el valor 4, por lo que hemos realizado de forma adecuada dicho cálculo de la complejidad ciclomática.

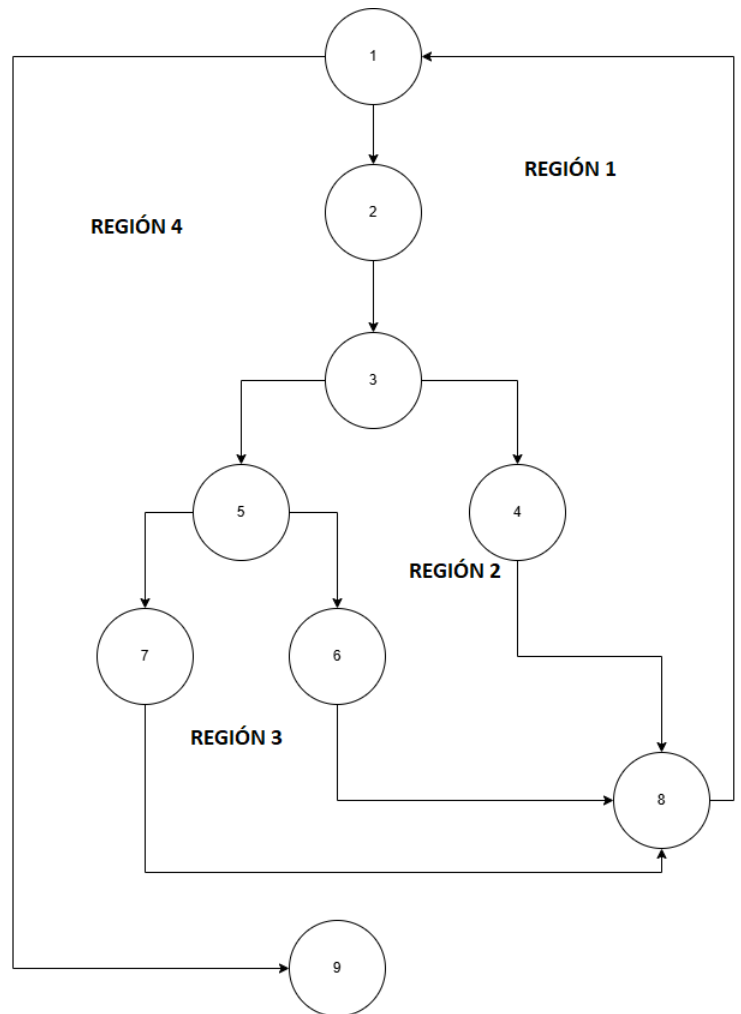
El valor 4 correspondería a programas o métodos sencillos, sin mucho riesgo.

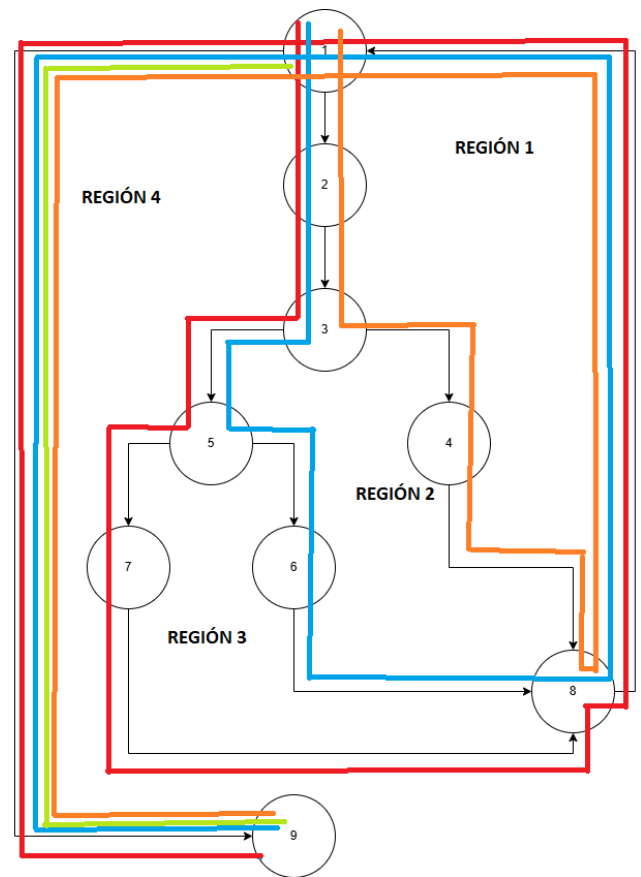
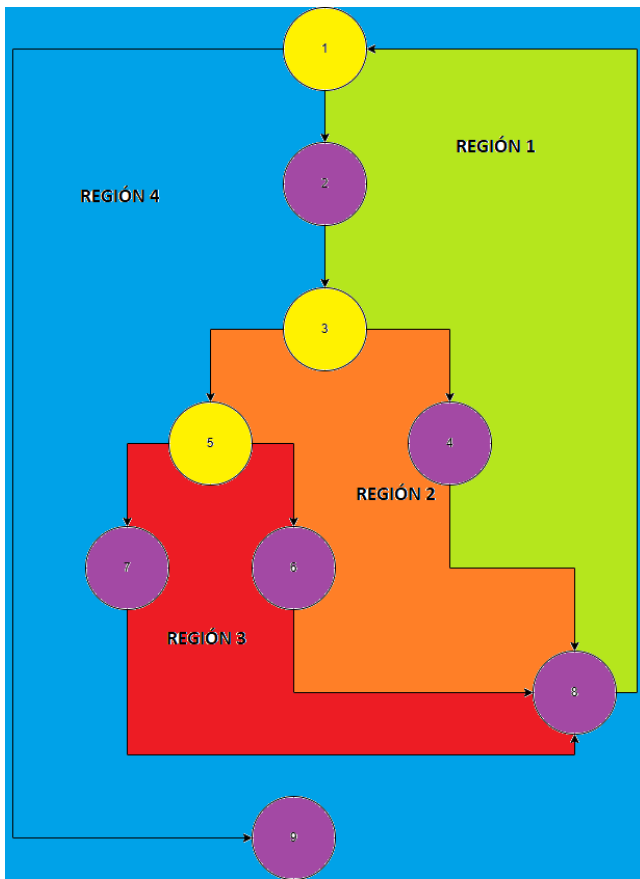
Camino mínimo: La cantidad de caminos mínimos posible nos lo va a proporcionar el valor obtenido en la complejidad ciclomática. Representará el número de pruebas a realizar.

- **Camino 1:** 1 - 2 - 3 - 4 - 8 - 1 - 9
- **Camino 2:** 1 - 2 - 3 - 5 - 6 - 8 - 1 - 9
- **Camino 3:** 1 - 2 - 3 - 5 - 7 - 8 - 1 - 9
- **Camino 4:** 1 - 9

Me parece una forma sencilla y eficaz de determinar las pruebas que habría que realizar para analizar todas las sentencias del código.

A continuación se añade una página extra con la muestra utilizando colores de cada una de las regiones, nodos, nodos predicado y caminos posibles y una explicación detallada de cada camino.





Nodos en color morado, **nodos predicados en amarillo**, **Región 1 en verde**, **Región 2 en naranja**, **Región 3 en rojo**, **Región 4 en azul**. Se pueden apreciar perfectamente lo que abarca cada región y la diferencia entre los nodos y nodos predicados, que estos últimos siempre tienen más de una salida.

En cuanto al número de caminos, tenemos los siguientes:

- **Camino 1: 1 - 2 - 3 - 4 - 8 - 1 - 9 (Naranja):** El camino 1 pasa por la condición verdadera del bucle while (1), realiza la instrucción Leer (2), entra al condicional (3) que comprueba si `campo1 = 0`, pasando por la parte verdadera, realizando por tanto las instrucciones (4) Procesar() e Incrementar_conta() y finalizando el bucle while (8) (volviendo al nodo 1) y saliendo del bucle, llegando a su fin de programa en el nodo 9
- **Camino 2: 1 - 2 - 3 - 5 - 6 - 8 - 1 - 9 (Azul):** El camino 2 pasa por la condición verdadera del bucle while (1), realiza la instrucción Leer (2), entra al condicional (3) que comprueba si `campo1 = 0`, pasando por la parte falsa, entrando en el segundo condicional (5) que comprueba si `Campo1 = 1`, entrando por la condición verdadera y por tanto, ejecutando la instrucción (6) Reiniciar_conta() y finalizando el bucle while (8) (volviendo al nodo 1) y saliendo del bucle, llegando a su fin de programa en el nodo 9
- **Camino 3: 1 - 2 - 3 - 5 - 7 - 8 - 1 - 9 (Rojo):** El camino 3 pasa por la condición verdadera del bucle while (1), realiza la instrucción Leer (2), entra al condicional (3) que comprueba si `campo1 = 0`, pasando por la parte falsa, entrando en el segundo condicional (5) que comprueba si `Campo1 = 1`, entrando por la condición falsa y por tanto, ejecutando la instrucción (7) procesar() y finalizando el bucle while (8) (volviendo al nodo 1) y saliendo del bucle, llegando a su fin de programa en el nodo 9
- **Camino 4: 1 - 9 (Verde):** El camino 4, el más corto, pasa por la condición falsa del bucle while (1) llegando a la finalización del mismo y del programa en el nodo 9.