

# PAC 1. UF2.

*Diseño y realización de pruebas.*

## **1 - ¿Qué diferencias existen entre pruebas de caja blanca y pruebas de caja negra?**

**Las pruebas de caja blanca** son aquellas pruebas estructurales donde se lleva a cabo un examen sobre los detalles procedimentales del código. En este tipo de pruebas obtendremos casos donde se asegure la ejecución de todos los caminos de cada módulo y las sentencias al menos una vez, las decisiones lógicas se ejecutarán una vez al menos en parte verdadera y false, los bucles serán ejecutados en sus límites y las estructuras de datos internas se usaran para asegurar su validez.

**Las pruebas de caja negra** son pruebas de comportamiento, donde sin necesidad de conocer la estructura del programa ni como funciona se llevará a cabo una prueba sobre la interfaz. Aquí buscaremos obtener errores de interfaz y del acceso a los datos o bases de datos externas, y las funcionalidades erróneas en el inicio o fin del programa.

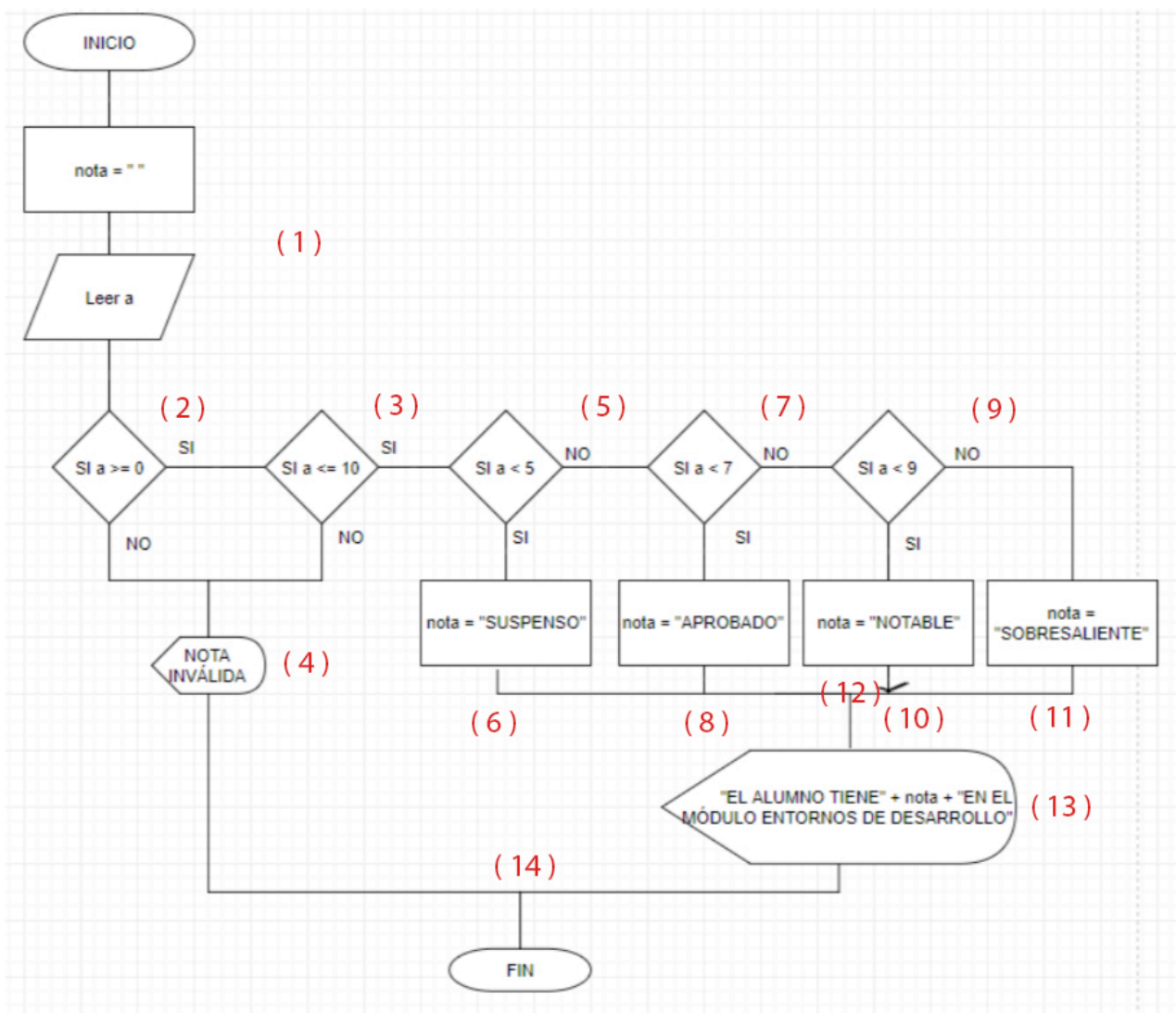
## **2 - Comenta alguna característica de cada una de las estrategias de pruebas.**

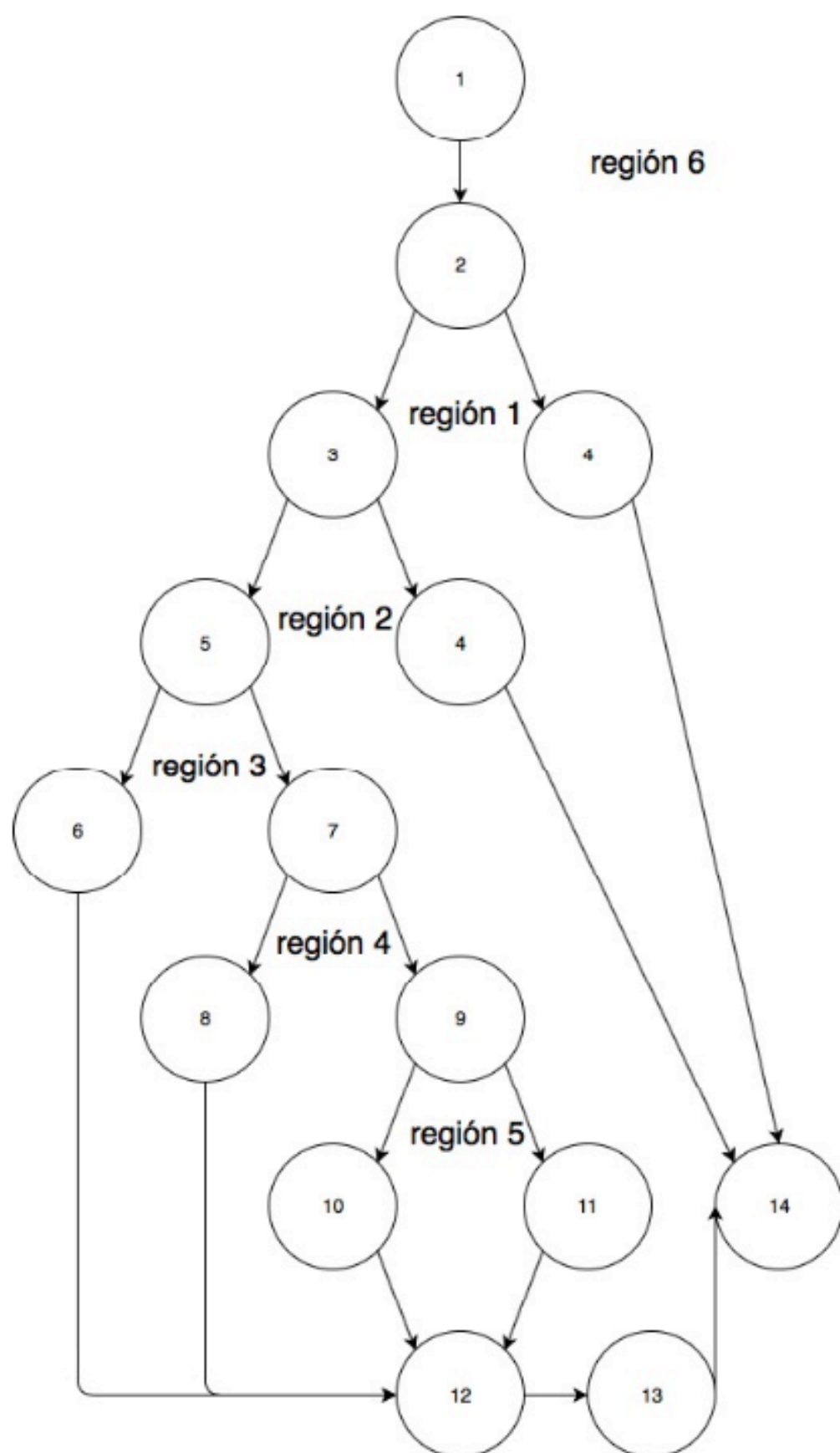
Los diferentes tipos de prueba dentro de las estrategia de prueba son:

- Prueba de unidad: ésta comprueba cada módulo para eliminar cualquier tipo de error en la interfaz, aquí se utilizan tanto la prueba de caja negra como la de prueba de caja blanca. Se llevan a cabo pruebas de interfaz del módulo, estructura de datos locales y las condiciones límite.
- Prueba de integración: aquí se comprobará la interacción de los distintos módulos del programa. Se podrá realizar mediante integración no incremental o incremental.
- Prueba de validación: se realiza la comprobación de los requisitos del software o ERS. Se llevan a cabo pruebas de caja negra que o bien son prueba alfa o prueba beta.

- Prueba del sistema: su misión es ejercitar en profundidad el software a través de la prueba de recuperación, la prueba de seguridad y la prueba de resistencia.

**3 - Realiza el grafo de flujo para el siguiente diagrama de flujo. Indica el número de nodos, aristas, regiones y nodos predicado. Calcula la complejidad ciclomática y muestra los caminos independientes.**





- **Número de nodos : 15**
- **Número de aristas: 19**
- **Número de regiones del grafo: 6**
- **Número de nodos predicado: ((2),(3),(5),(7),(9)) : 5**

#### **A. Complejidad ciclomática:**

- 1 - Número de regiones del grafo: 6
- 2 - Aristas - Nodos + 2:  $19 - 15 + 2 = 6$
- 3 - Nodos predicado + 1:  $5 + 1 = 6$

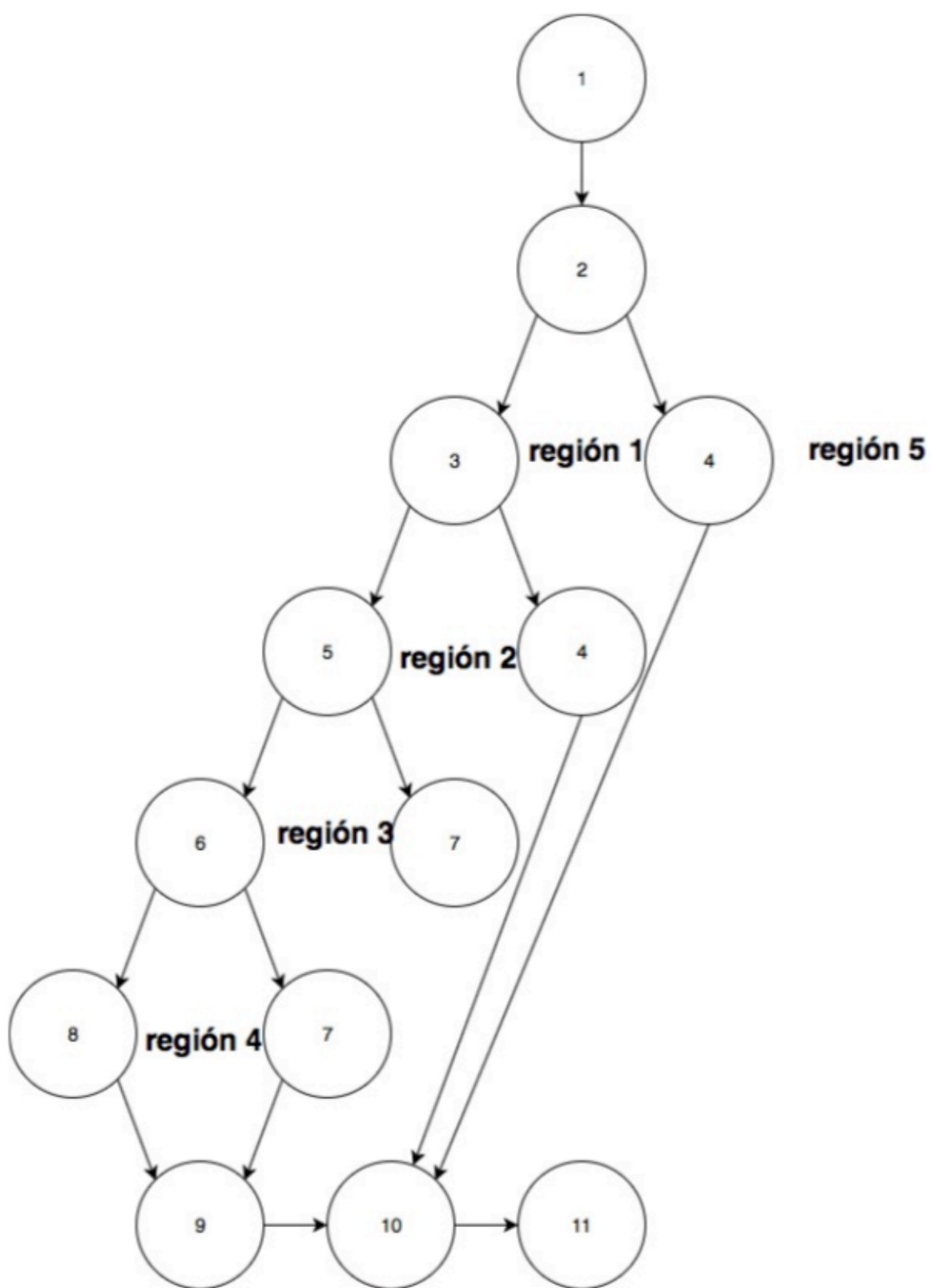
Observamos que la complejidad ciclomática está comprendida entre 1 y 10 por lo que el programa es sencillo o no entraña demasiado riesgo.

#### **B. Caminos independientes:**

- 1-2-4-14
- 1-2-3-4-14
- 1-2-3-5-6-12-13-14
- 1-2-3-5-7-8-12-13-14
- 1-2-3-5-7-9-10-12-13-14
- 1-2-3-5-7-9-11-12-13-14

4 - Realiza el grafo de flujo para el siguiente código. Indica el número de nodos, aristas, regiones y nodos predicado. Calcula la complejidad ciclomática y muestra los caminos independientes.

```
INICIO
Escribe importe de compra
Leer importe ( 1 )
Escribe mes
Leer mes ( 2 )
SI ((mes = "ENERO") OR (mes = "JULIO")) ENTONCES ( 3 )
    total <- precio * 0'70 ( 4 )
SINO ( 5 )
    SI ((mes = "FEBRERO") OR (mes = "AGOSTO")) ENTONCES ( 6 )
        total <- precio * 0'50 ( 7 )
    SINO
        total <- precio ( 8 )
    FIN SI ( 9 )
FIN SI ( 10 )
Mostrar total ( 11 )
FIN
```



- **Número de nodos : 13**
- **Número de aristas: 16**
- **Número de regiones del grafo: 5**
- **Número de nodos predicado: ((2),(3),(5),(6)) : 4**

#### **A. Complejidad ciclomática:**

- 1 - Número de regiones del grafo: 5
- 2 - Aristas - Nodos + 2:  $16 - 13 + 2 = 5$
- 3 - Nodos predicado + 1:  $4 + 1 = 5$

En este segundo caso, se dan las mismas circunstancias que en el anterior, la complejidad ciclomática está comprendida entre 1 y 10 por lo que el programa es sencillo o no entraña demasiado riesgo.

#### **B. Caminos independientes:**

- 1-2-4-10-11
- 1-2-3-4-10-11
- 1-2-3-5-7-9-10-11
- 1-2-3-5-6-7-9-10-11
- 1-2-3-5-6-8-9-10-11



**5 - Realiza el grafo de flujo para un programa que recibe tres números por teclado y muestra el menor de los números. Indica el número de nodos, aristas, regiones y nodos predicado. Calcula la complejidad ciclomática y muestra los caminos independientes.**

Entorno:

Real numero1, numero2, numero3, numeroMayor

INICIO

Escribe numero1;

Leer numero1;

Escribe numero2;

Leer numero2;

Escribe numero3;

Leer numero3;

SI  $n1 > n2$  ENTONCES

SI  $n1 > n3$  ENTONCES

mayor=numero1;

SINO

mayor=numero3;

SINO

SI  $n2 > n3$  ENTONCES

mayor=numero2;

SINO

mayor=numero3;

FINSI;

FINSI;

Mostrar mayor;

FIN

Entorno:

Real numero1, numero2, numero3, numeroMayor

INICIO

Escribe numero1;

Leer numero1;

Escribe numero2; ( 1 )

Leer numero2;

Escribe numero3;

Leer numero3;

( 2 ) ( 3 )

SI  $n1 > n2$  ENTONCES

SI  $n1 > n3$  ENTONCES ( 5 )

mayor=numero1;

SINO ( 6 )

mayor=numero3;

SINO ( 4 )

SI  $n2 > n3$  ENTONCES

mayor=numero2; ( 7 )

SINO

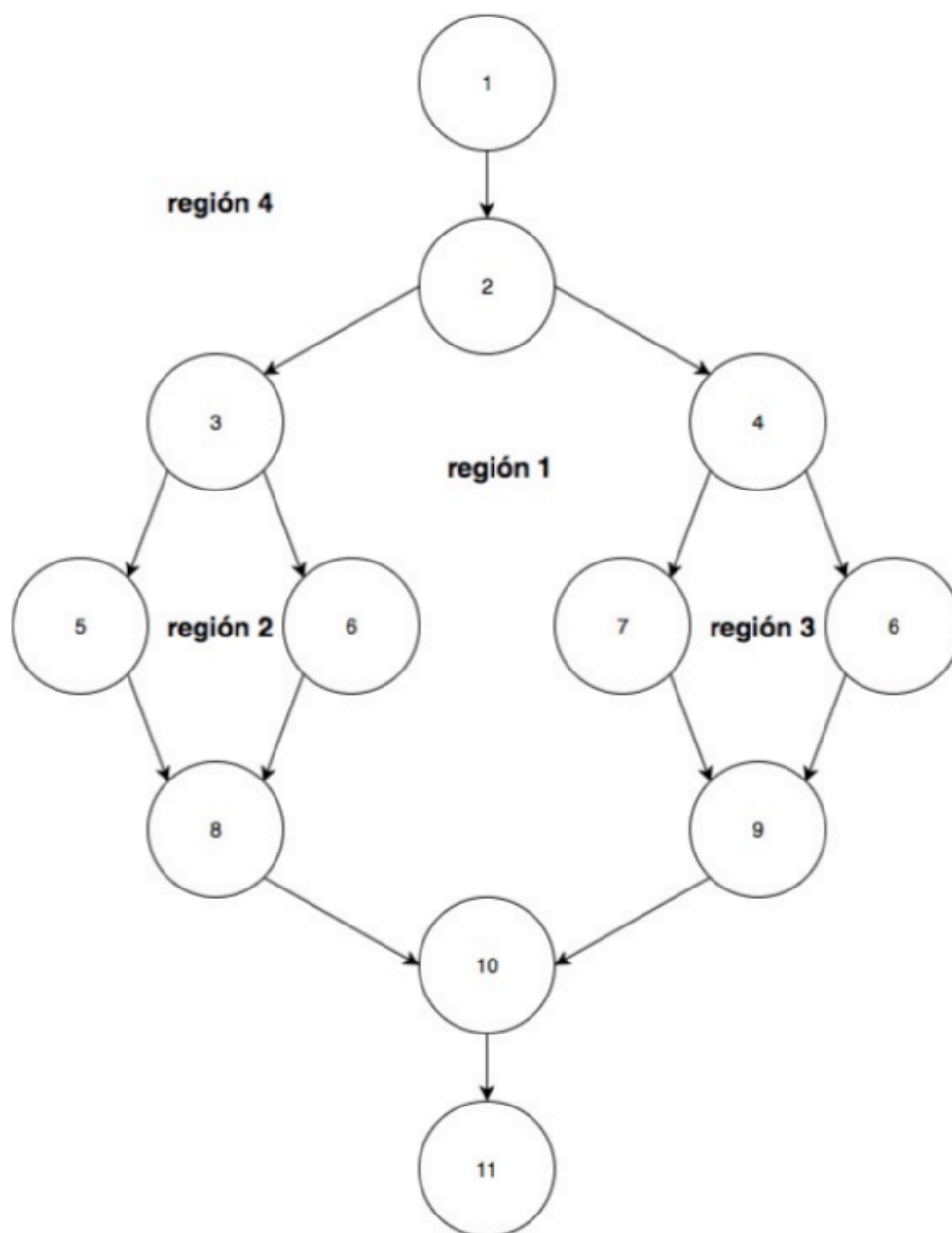
mayor=numero3;

FINSI; ( 8 )

FINSI; ( 9 )

Mostrar mayor; ( 10 )

FIN ( 11 )



- **Número de nodos : 12**
- **Número de aristas: 14**
- **Número de regiones del grafo: 4**
- **Número de nodos predicado: ((2),(3),(4)) : 3**

#### **A. Complejidad ciclomática:**

- 1 - Número de regiones del grafo: 4
- 2 - Aristas - Nodos + 2:  $14 - 12 + 2 = 4$
- 3 - Nodos predicado + 1:  $3 + 1 = 4$

En este tercer y último caso, se dan las mismas circunstancias que en los casos anteriores, la complejidad ciclomática está comprendida entre 1 y 10 por lo que el programa es sencillo o no entraña demasiado riesgo.

#### **B. Caminos independientes:**

- 1-2-3-5-8-10-11
- 1-2-3-6-8-10-11
- 1-2-4-6-9-10-11
- 1-2-4-7-9-10-11