

CICLO FORMATIVO DE GRADO SUPERIOR DESARROLLO DE APLICACIONES WEB

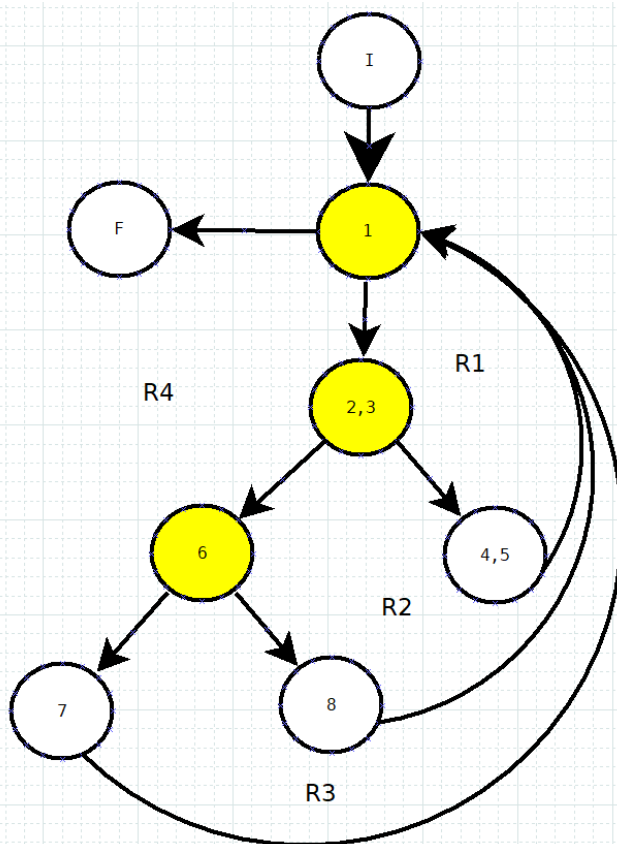
CURSO 2022/2023

**Módulo : Entornos de Desarrollo
Práctica : Pruebas del Software**



Victor Stala.

Primero: Grafo 1



Nodos:8

Aristas:10

Regiones:4

Nodos-predicado:3

Complejidad ciclomática:4

Secuencia de nodos:

-Camino 1: I,1,F

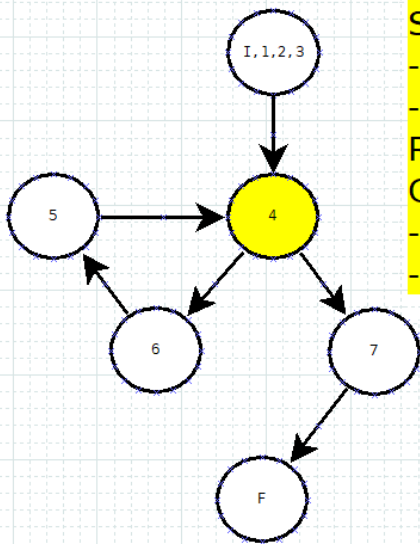
-Camino 2: I,1,2,3,4,5,1,F

-Camino 3: I,1,2,3,6,7,1,F

-Camino 4: I,1,2,3,6,8,1,F

Segundo: Grafo2. Factorial

```
(I) public static int factorial(int n) {  
    (1) int resultado;  
    (2) resultado = 1;  
    for ((3)int i = 2; (4)i <= n; (5)i++) {  
        (6)resultado = resultado * i;  
    }  
    (7)return resultado;  
(F)}
```



Complejidad ciclomática:2

Secuencia de nodos:

-Camino 1: 1,1,2,3,4,7,F

-Camino 2: 1,1,2,3,4,6,5,4,7,F

Riesgo: Muy bajo, 2 caminos.

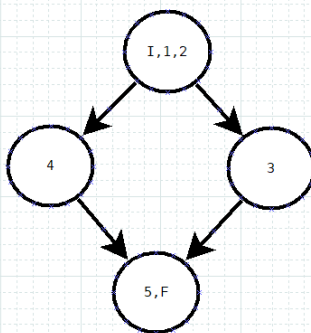
Casos de prueba:

-Camino 1: Entrada = 0, resultado esperado = 1.

-Camino 2: Entrada = 3, resultado esperado = 6.

Tercero: Grafo3. Divisible

```
(I) public boolean divisible(int multiplo, int divisor) {  
    (1) boolean resultado;  
    (2) if (multiplo % divisor == 0) {  
        (3) resultado = true;  
    } else {  
        (4) resultado = false;  
    }  
    (5) return resultado;  
(F) }
```



Complejidad ciclomática:2

Secuencia de nodos:

-Camino 1: I,1,2,3,5,F

-Camino 2: I,1,2,4,5,F

Riesgo: Muy bajo, 2 caminos.

Casos de prueba:

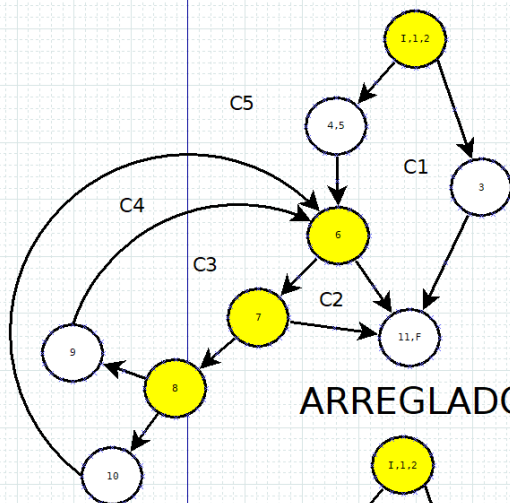
-Camino 1: Entrada = 4,2, resultado esperado = TRUE.

-Camino 2: Entrada = 4,3, resultado esperado = FALSE.

Cuarto: Grafo4. esPrimo

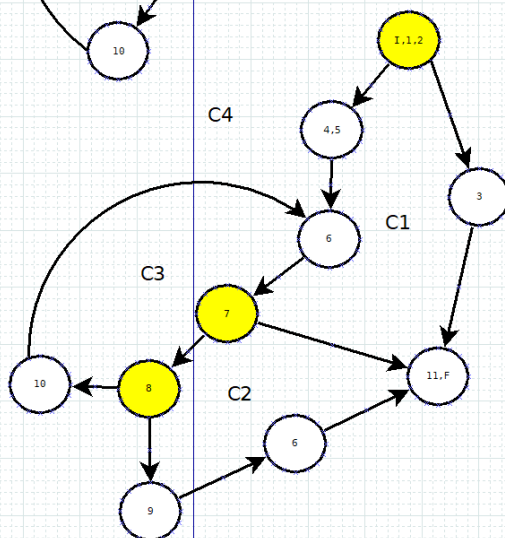
```
(1) public boolean esPrimo(int n) {  
    (1) boolean primo;  
    (2) if (n <= 1) {  
        (3) primo = false;  
    } else {  
        (4) primo = true;  
        (5) int i = 2;  
        (6 y 7) while (primo && i <= n / 2) {  
            (8) if (divisible(n, i)) {  
                (9) primo = false;  
            } else {  
                (10) i++;  
            }  
        }  
    }  
    (11) return primo;  
(F)}
```

ORIGINAL



Complejidad ciclomática: 5
 Secuencia de nodos:
 -Camino 1: 1,1,2,3,11,F
 -Camino 2: 1,1,2,4,5,6,F(imposible)
 -Camino 3: 1,1,2,4,5,6,7,11,F
 -Camino 4: 1,1,2,4,5,6,7,8,9,6,11,F
 -Camino 5: 1,1,2,4,5,6,7,8,10,6,11,F

ARREGLADO



Complejidad ciclomática: 4
 Secuencia de nodos:
 -Camino 1: 1,1,2,3,11,F
 -Camino 2: 1,1,2,4,5,6,7,11,F
 -Camino 3: 1,1,2,4,5,6,7,8,9,6,11,F
 -Camino 4: 1,1,2,4,5,6,7,8,10,6,7,11,F
 Riesgo: Muy bajo
 Casos de prueba:
 -Camino 1: Entrada = 1 resultado esperado = FALSE
 -Camino 2: Entrada = 2 resultado esperado = TRUE
 -Camino 3: Entrada = 4 resultado esperado = FALSE
 -Camino 4: Entrada = 5 resultado esperado = TRUE

Quinto: Clases de equivalencia

factorial() →

Caso de prueba	Entrada	Salida esperada
Caso 1	0	1
Caso 2	1	1
Caso 3	2	2
Caso 4	3	6
Caso 5	4	24
Caso 6	5	120
Caso 7	6	720
Caso 8	10	3628800
Caso 9	-1	Error

divisible() →

Caso de prueba	Entrada	Salida esperada
Caso 1	10, 2	true
Caso 2	10, 3	false
Caso 3	0, 10	true
Caso 4	10, 0	Error
Caso 5	-10, 2	true
Caso 6	10, -2	true
Caso 7	-10, -2	true

esPrimo() →

Caso de prueba	Entrada	Salida esperada
Caso 1	2	true
Caso 2	3	true
Caso 3	4	false
Caso 4	5	true
Caso 5	10	false
Caso 6	11	true
Caso 7	13	true
Caso 8	17	true
Caso 9	19	true
Caso 10	20	false