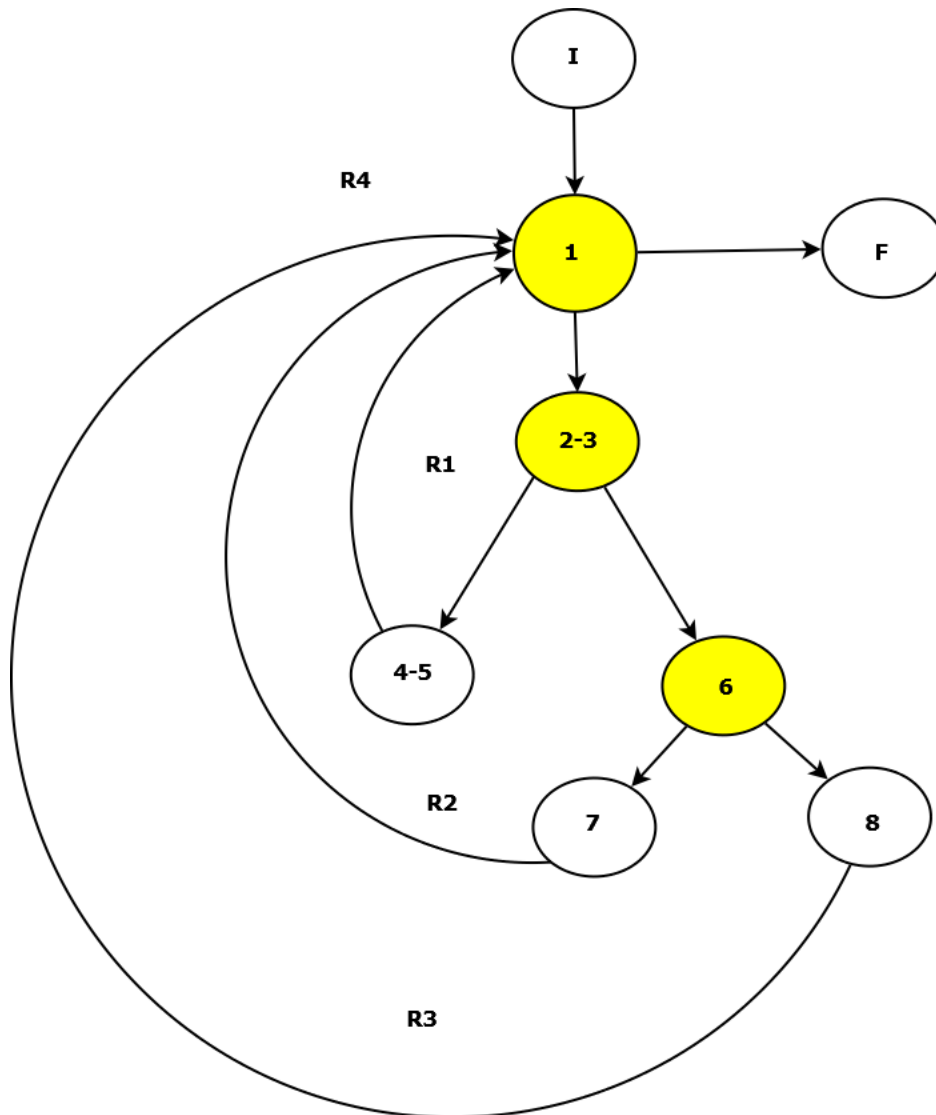


Tema 3: Pruebas del software

Práctica 1: Pruebas del software

Primero: Grafo1



Número de nodos: 8

Número de aristas: 10

Número de regiones: 4

Número de nodos predicado: 3

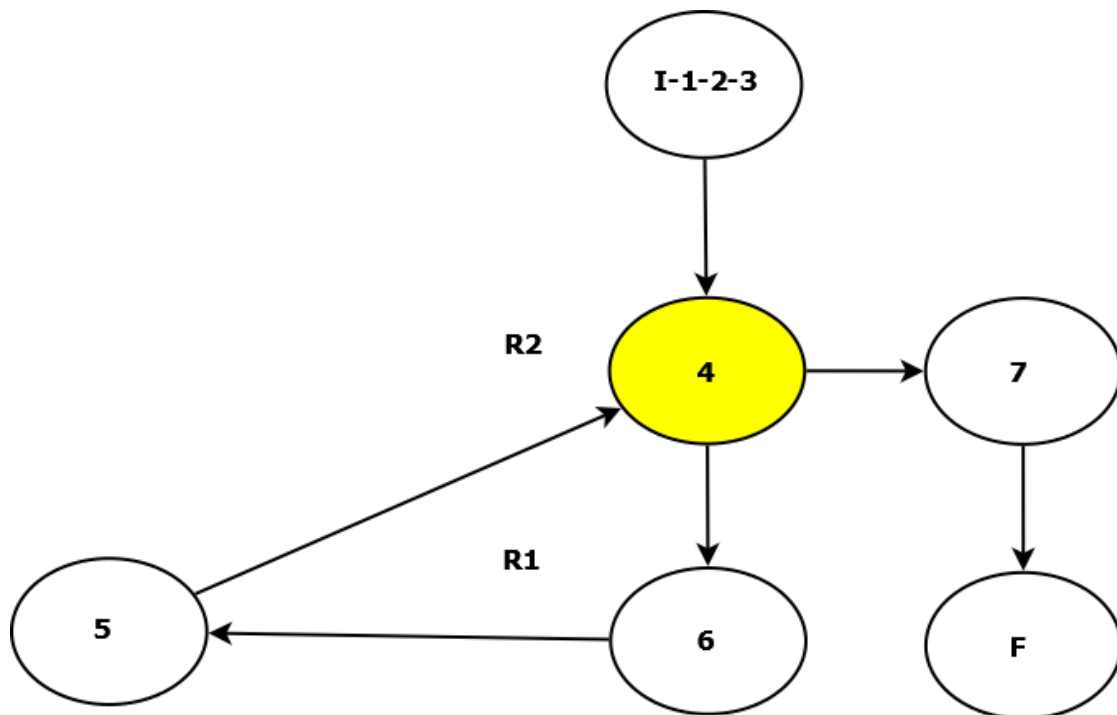
Complejidad ciclomática: 4

Secuencia de nodos:

- 1º Secuencia: I,1,F
- 2º Secuencia: I,1,2,3,4,5,1,F
- 3º Secuencia: I,1,2,3,6,7,1,F
- 4º Secuencia: I,1,2,3,6,8,1,F

Segundo: Grafo2. Factorial

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



Complejidad ciclomática: 2

Riesgo: Muy bajo

Conjunto de caminos:

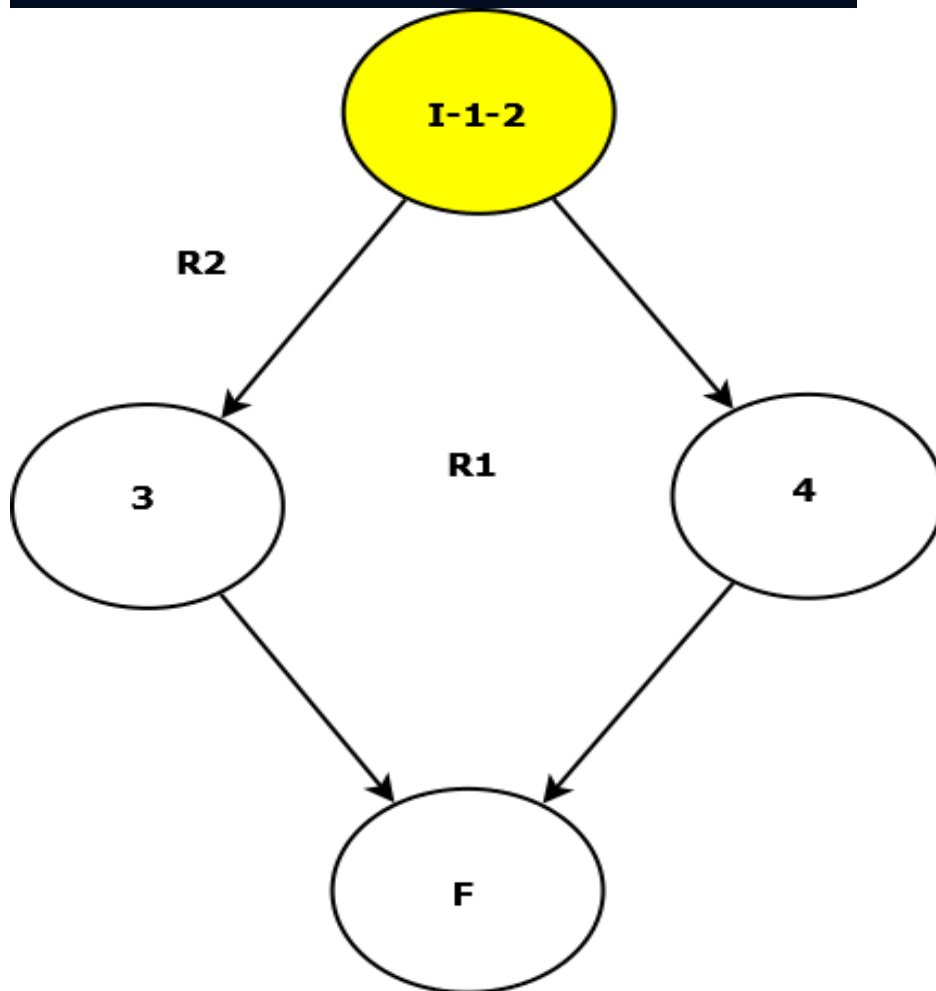
- Camino 1: I,1,2,3,4,6,5,4,7,F
- Camino 2: I,1,2,3,4,7,F

Casos de prueba:

Valor	Resultado
n = 1	1
n = 3	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;  
        }  
(F) }
```



Complejidad ciclomática: 2

Riesgo: Muy bajo

Conjunto de caminos:

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

Casos de prueba:

Valor	Resultado
Múltiplo = 12 Divisor = 4	true
Múltiplo = 8 Divisor = 7	false

Cuarto: Grafo4. esPrimo

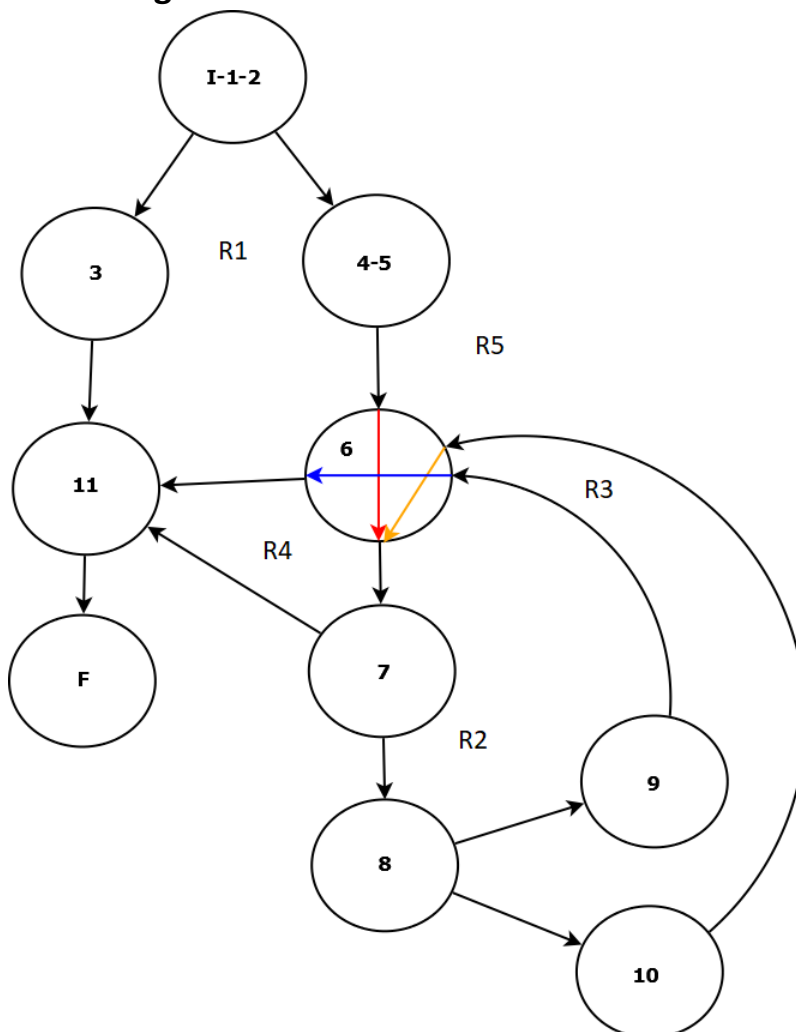
```

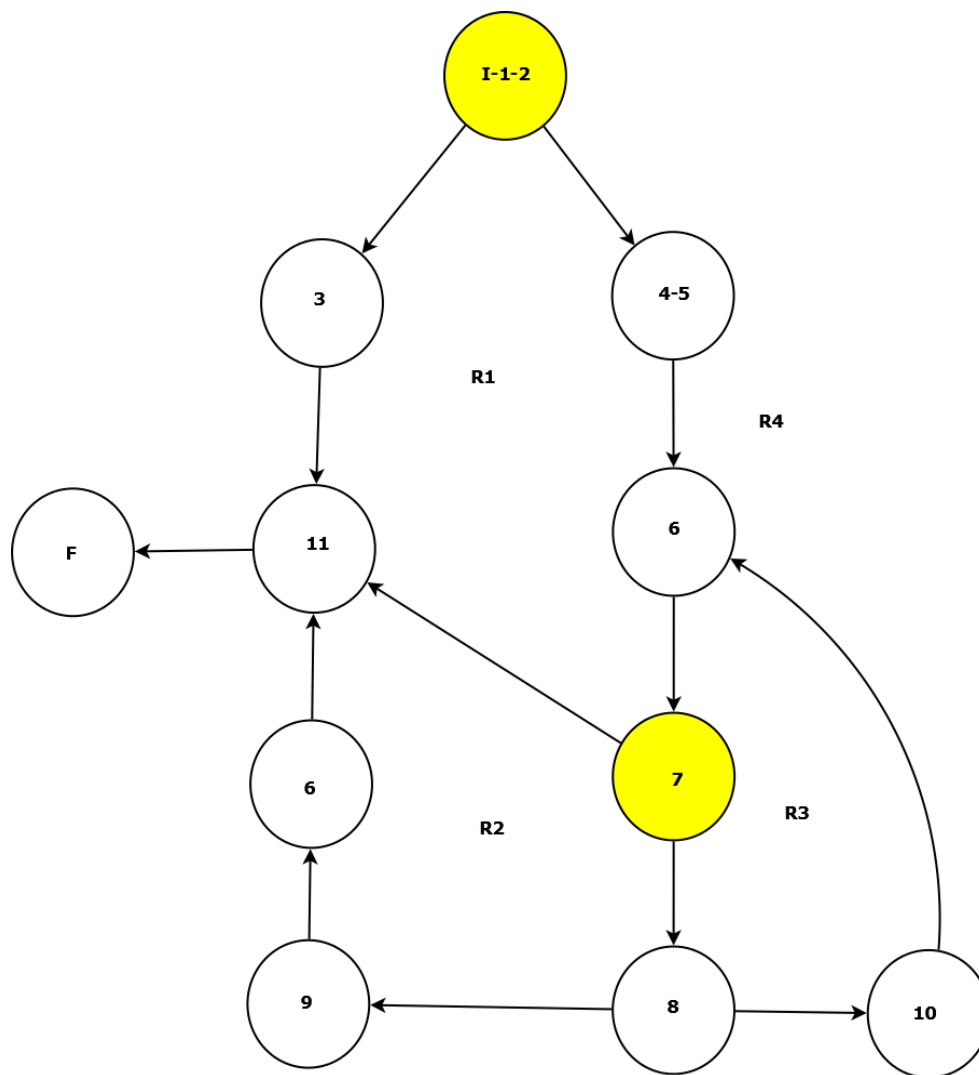
(I) public boolean esPrimo(int n) {
(1)   boolean primo;

(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) }

```

Grafo Original



Grafo Corregido

Complejidad ciclomática: 4

Riesgo: Muy bajo

Conjunto de caminos:

- Camino 1: I,1,2,3,11,F
- Camino 2: I,1,2,4,5,6,7,11,F
- Camino 3:
I,1,2,4,5,6,7,8,9,6,11,F
- Camino 4:
I,1,2,4,5,6,7,8,10,6,7,11,F

Casos de prueba:

Valor	Resultado
n = 1	false
n = 2	true
n = 11	true

Quinto: Clases de equivalencia

Tabla de casos de **Factorial**:

Caso	Entrada de prueba	Equivalencia	Salida esperada
1	$n = -1$	Valor no válido	Excepción
2	$n = 0$	Clase 1	1
3	$n = 1$	Clase 2	1
4	$n = 2$	Clase 3	2
5	$n = 5$	Clase 4	120
6	$n = 10$	Clase 5	3628800
7	$n = \text{Valor máximo de Int (2147483647)}$	Valor límite superior	Excepción

Tabla de casos de **Divisible**:

Caso	Entrada de prueba	Equivalencia	Salida esperada
1	multiplo = 6 divisor = 2	Múltiplo es divisible por divisor	true
2	multiplo = 7 divisor = 2	Múltiplo no es divisible por divisor	false
3	multiplo = 0 divisor = 5	Múltiplo es igual a cero	true
4	multiplo = 7 divisor = 0	Divisor es igual a cero	Excepción
5	multiplo = -6 divisor = 2	Múltiplo es negativo	true
6	multiplo = 6 divisor = -2	Divisor es negativo	true
7	multiplo = -6 divisor = -2	Múltiplo y divisor son negativos	true
8	multiplo = Valor máximo de Int (2147483647) divisor = 1	Múltiplo es el valor máximo de un entero	true
9	multiplo = Valor mínimo de Int (-2147483648) divisor = 1	Múltiplo es el valor mínimo de un entero	true

10	multiplo = 15 divisor = Valor máximo de Int (2147483647)	Divisor es el valor máximo de un entero	false
11	multiplo = 15 divisor = Valor mínimo de Int (-2147483648)	Divisor es el valor mínimo de un entero	False

Tabla de casos de **esPrimo**:

Caso	Entrada de prueba	Equivalencia	Salida esperada
1	n = 2	Número primo más pequeño	true
2	n = 3	Número primo mayor que 2	true
3	n = 1	Número igual a 1	false
4	n = 0	Número igual a cero	false
5	n = -7	Número negativo	false
6	n = 4	Número compuesto	false
7	n = 19	Número primo	true
8	n = 97	Número primo mayor que 50	true
9	n = Valor máximo de Int (2147483647)	Número primo máximo para un entero	false
10	n = Valor mínimo de Int (-2147483648)	Número primo mínimo para un entero	false