

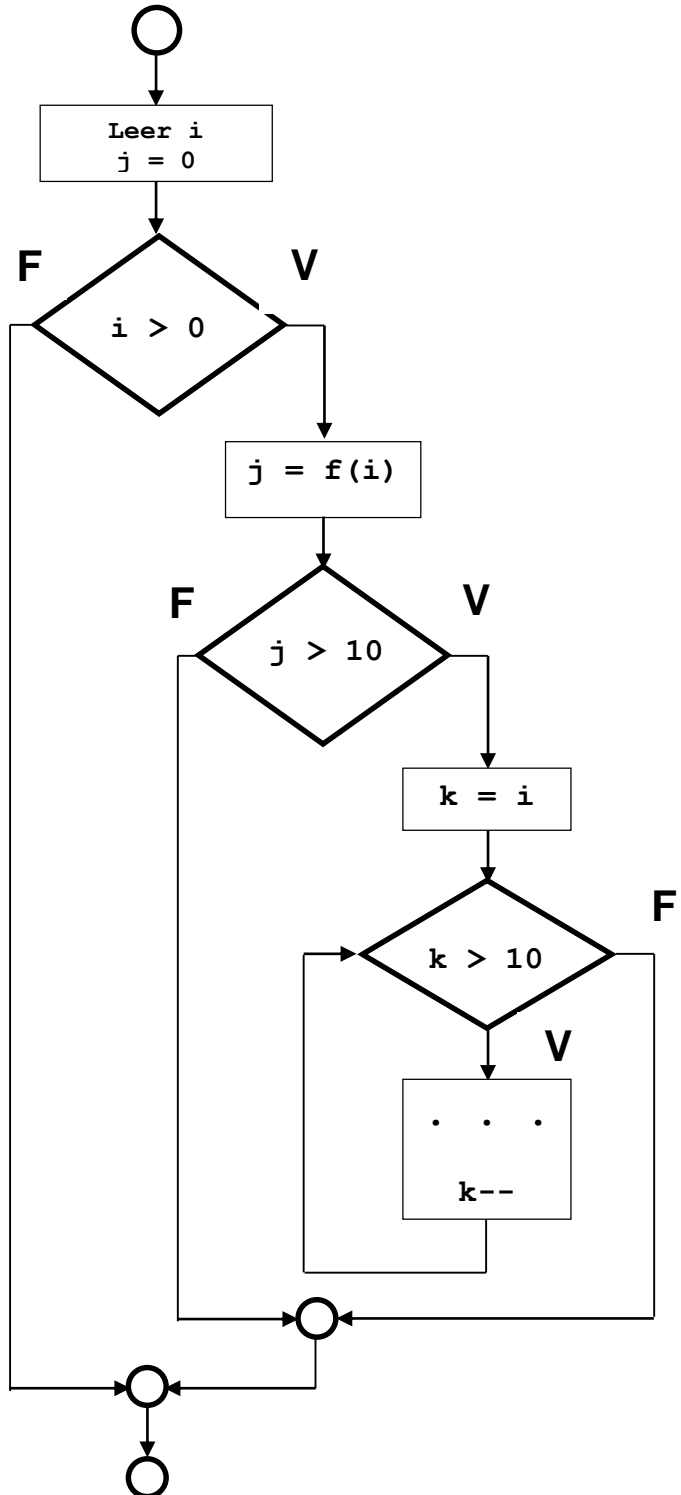
**CAPÍTULO 3. PRUEBAS DE SOFTWARE.**  
**3.2. Pruebas Unitarias. Enfoque de Caja Blanca**  
**3.2.1. Coberturas Lógicas**

**1. COBERTURA DE SENTENCIA (STATEMENT COVERAGE)**

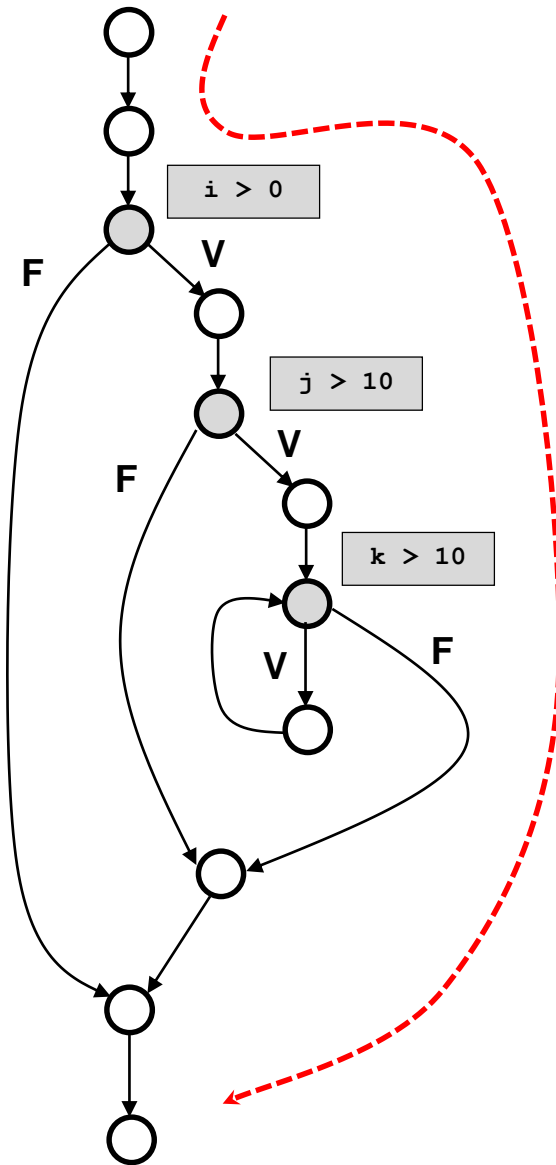
*Ejemplo 1. Se evalúa el siguiente segmento:*

```

INICIO
Leer i
j = 0
if(i>0)
{
j = f(i);
if (j>10)
{
for (k=i; k>10; k--)
{
. . .
}
}
}
FIN
    
```



Y que se puede representar mediante este grafo:



**Se aprecia que:**

Contiene 2 sentencias "if" y un bucle "mientras" el cual es interno al segundo "if".

**Hay tres "caminos" diferentes a través del segmento de programa**

El primer "if" permite DOS direcciones. La dirección de la derecha del primer "if" se divide nuevamente debido al segundo "if".

**Todas las sentencias de este segmento de programa pueden alcanzarse (o cubrirse) haciendo uso de este camino a la derecha.**

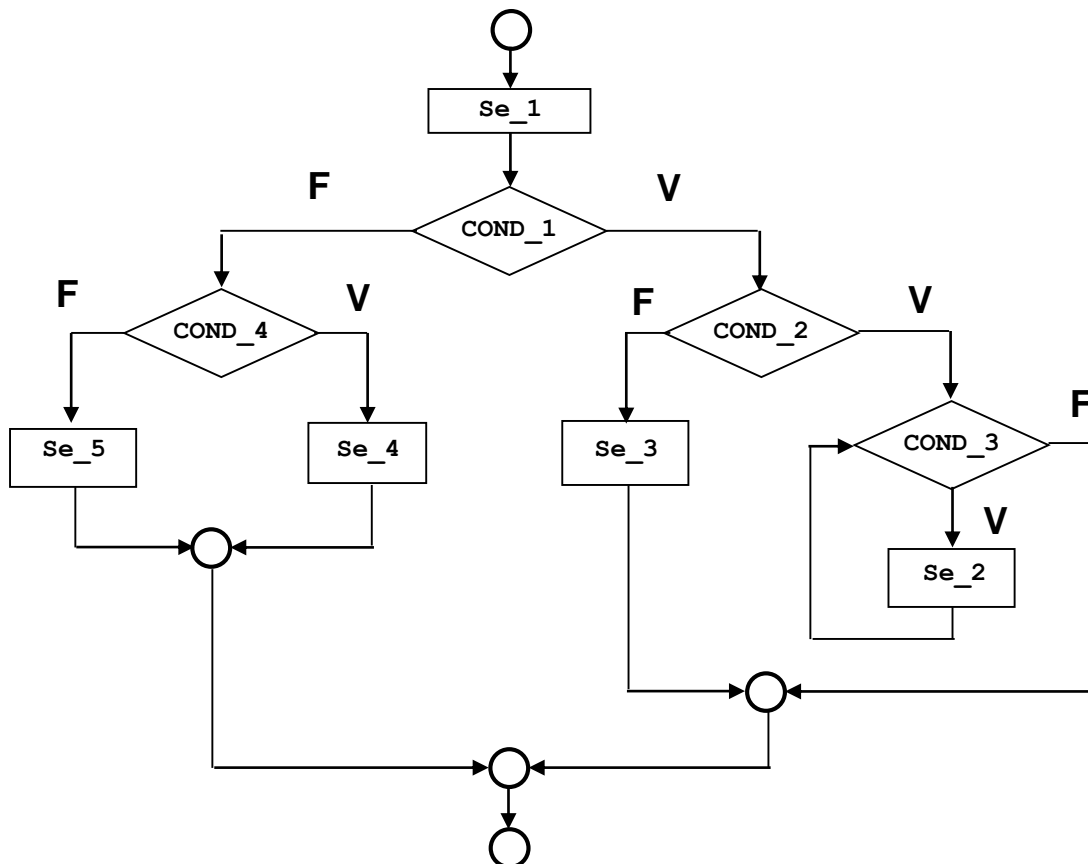
Por consiguiente, será suficiente un solo caso de prueba para alcanzar el 100% de cobertura de sentencia.

Aquí la clave es:  
**"CUBRIR NODOS"**

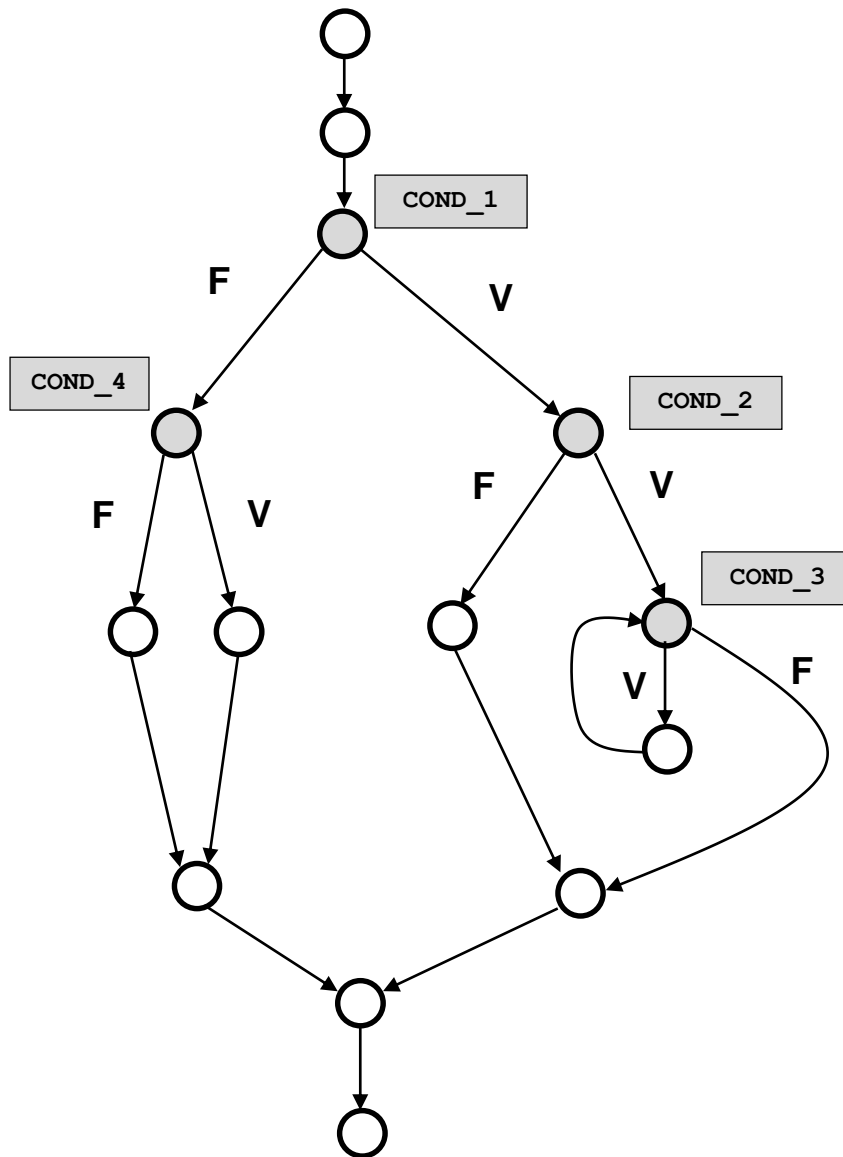
**Ejemplo 2. Se evalúa el siguiente segmento:**

```

INICIO
  Sentencia_1
  SI (COND_1) entonces
    SI (COND_2) entonces
      MIENTRAS (COND_3) hacer
        Sentencia_2
      FIN_MIENTRAS
    SI_NO
      Sentencia_3
    FIN_SI
  SI_NO
    SI (COND_4) entonces
      Sentencia_4
    SI_NO
      Sentencia_5
    FIN_SI
  FIN_SI
FIN
    
```



Y que se puede representar mediante este grafo:



**Se aprecia que:**

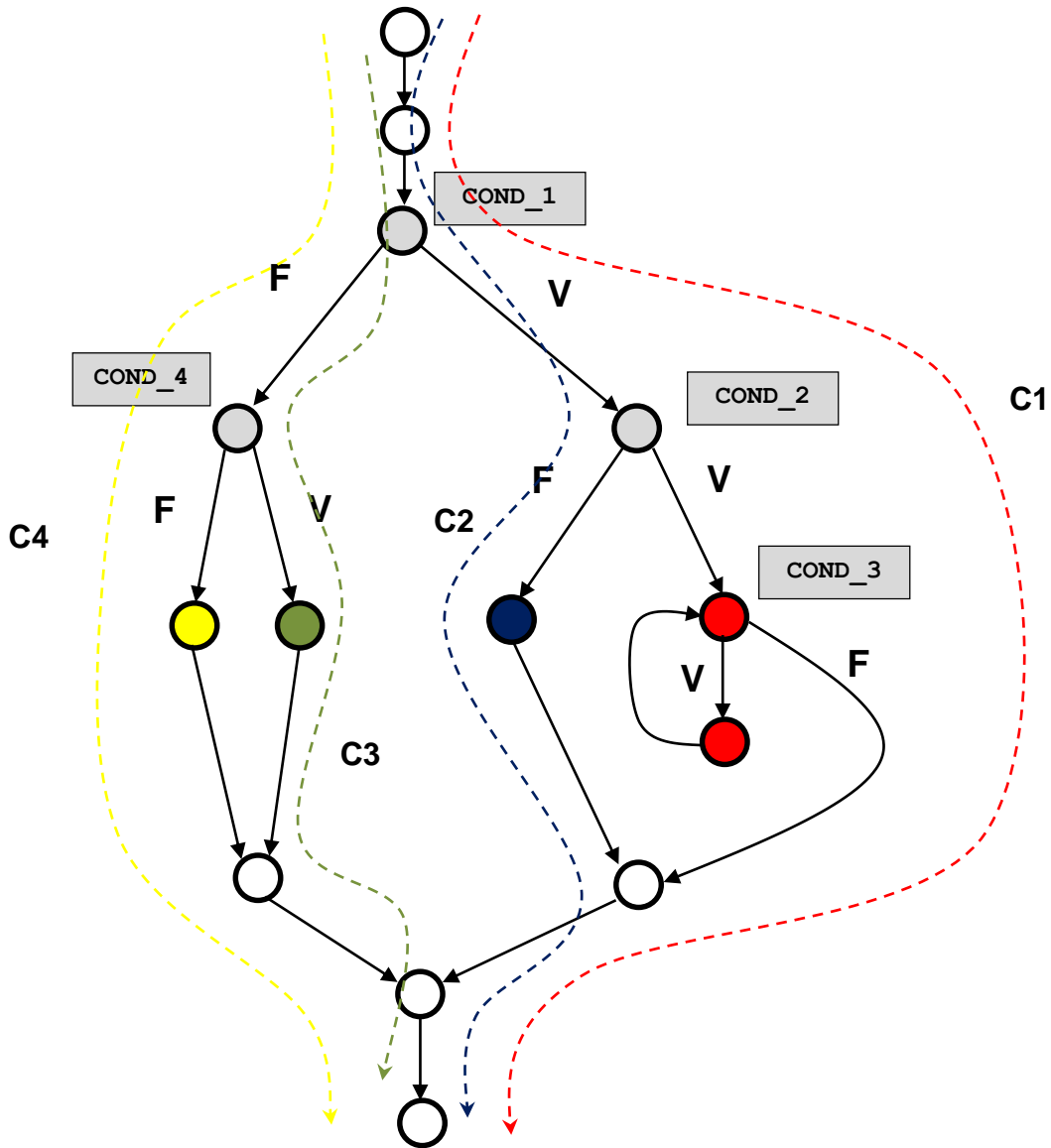
Este segmento de programa contiene sentencias “if” y un bucle dentro de uno de esos “if” (COND\_2).

**Cuatro caminos diferentes conducen a través de este segmento de programa:**

- El primer “if” permite DOS direcciones.
- En cada rama de la sentencia “if”, se tiene otra sentencia “if” que permite nuevamente DOS direcciones diferentes.
- Para una cobertura de sentencia al 100% aquí se requieren CUATRO casos de prueba.

Recuerde que aquí la clave es: “CUBRIR NODOS”

Los casos se pueden resaltar así:



**SUMARIO:**

Esta cobertura requiere que se ejecute por lo menos una vez cada sentencia del programa.

Es útil para detectar código muerto, es decir, código constituido por sentencias que nunca se ejecutan. Y si hay código muerto en un programa, no se logrará cobertura del 100%

Este criterio es necesario, pero no suficiente

Es un criterio débil: **No siempre se comprueban ambas vertientes de las condiciones.**

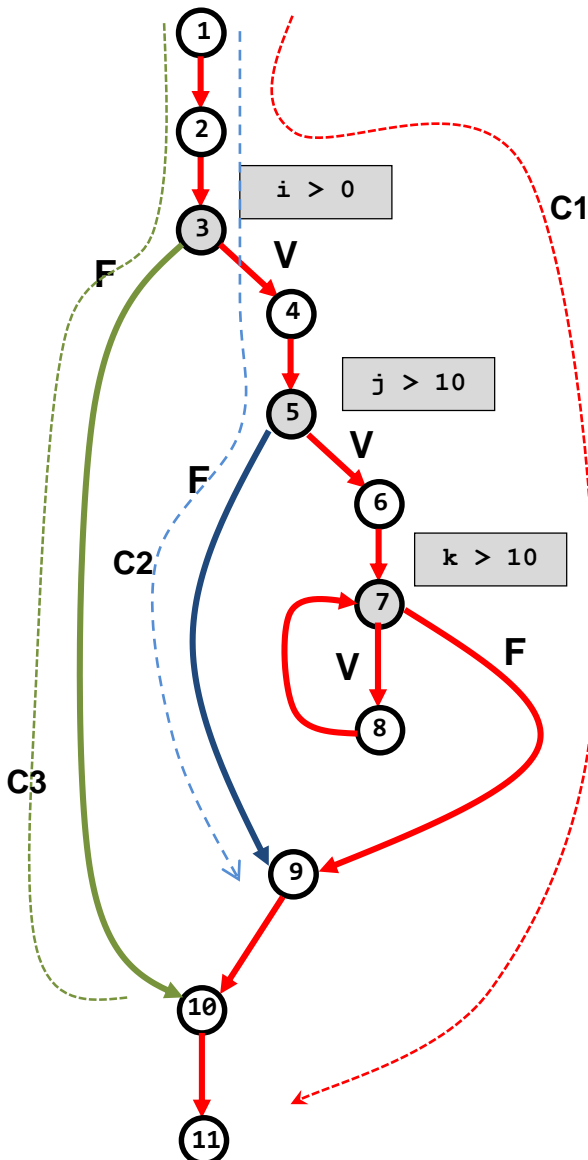
## 2. COBERTURA DE DECISIÓN (DECISION COVERAGE)

En este caso, se centra en el flujo de control en un segmento del programa (es decir, aquí se trata de "CUBRIR ARISTAS" en lugar de cubrir los nodos del grafo que representa el flujo de control). También se le conoce como cobertura de rama (branch coverage). Esto significa que TODAS las aristas del grafo deben ser cubiertas al menos UNA vez. El propósito de esta técnica (criterio de salida) es lograr la cobertura de un porcentaje específico de todas las decisiones, denominado cobertura de decisión.

Cobertura de Decisión (Caso i) =  $(\text{N}^\circ \text{ de decisiones ejecutadas} / \text{N}^\circ \text{ total de decisiones}) * 100\%$

Cobertura de Rama (Caso i) =  $(\text{N}^\circ \text{ de ramas ejecutadas} / \text{N}^\circ \text{ total de ramas}) * 100\%$

**Ejemplo 3. Se evalúa el siguiente segmento (el del ejemplo 1):**



**Se aprecia que:**

**Hay tres "caminos" diferentes a través del segmento de programa**

- El primer "if" permite DOS direcciones.
- Un camino del primer "if" se divide nuevamente en dos caminos diferentes, uno de los cuales contiene un bucle.
- Solamente se puede alcanzar todas las aristas a través de una combinación de los tres caminos posibles.

El camino C1 (rojo) cubre 11 de las 13 aristas.  
 [1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 7 – 9 – 10 – 11]  
 $C1 = 11/13 = 84.6\%$

El camino (C2) cubre 7 de las 13 aristas.  
 [1 – 2 – 3 – 4 – 5 – 9 – 10 – 11]  
 $C2 = 7/13 = 53.8\%$

El camino (C3) cubre 4 de las 13 aristas.  
 [1 – 2 – 3 – 10 – 11]  
 $C3 = 4/13 = 30.8\%$

C1 tiene 5 aristas exclusivas. C2 y C3 tienen 1 arista exclusiva cada uno.

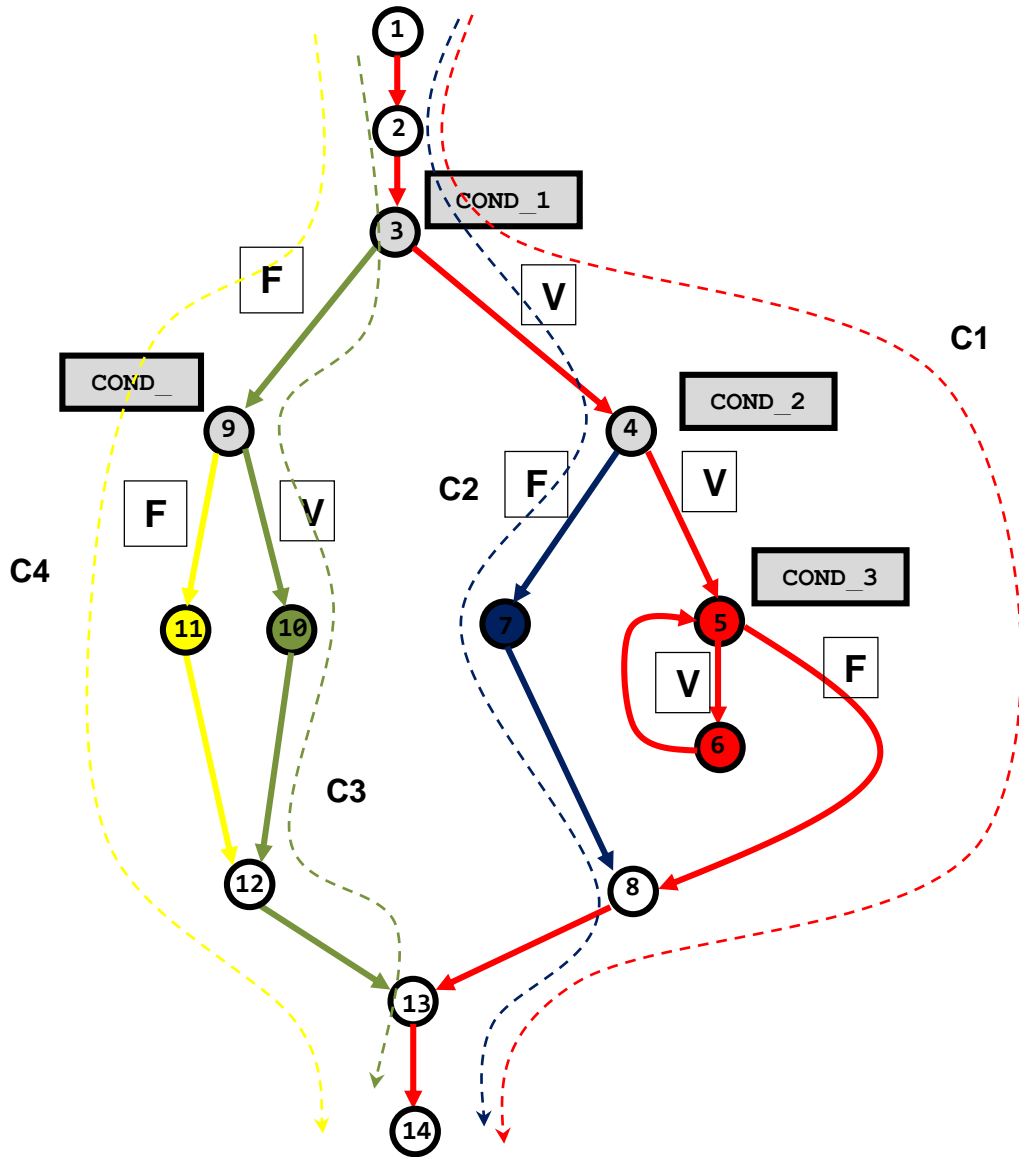
Tomando C1 como base: **C1 cubre en total 11 aristas + 1 exclusiva de C2 + 1 exclusiva de C3 = 13 aristas.**

Por ello se requieren TRES casos de prueba para lograr una cobertura de decisión del 100%.

**Aquí la clave es: "CUBRIR ARISTAS"**

**Ejemplo 4. Se evalúa el siguiente segmento (el del ejemplo 2):**

Los casos se pueden resaltar así:



**SUMARIO:**

- El camino (C1) cubre 9 de las 17 aristas. [1 – 2 – 3 – 4 – 5 – 6 – 5 – 8 – 13 – 14].  $C1 = 9/17 = 52.9\%$
- El camino (C2) cubre 7 de las 17 aristas. [1 – 2 – 3 – 4 – 7 – 8 – 13 – 14].  $C2 = 7/17 = 41.2\%$
- El camino (C3) cubre 7 de las 17 aristas. [1 – 2 – 3 – 9 – 10 – 12 – 13 – 14].  $C3 = 7/17 = 41.2\%$
- El camino (C4) cubre 7 de las 17 aristas. [1 – 2 – 3 – 9 – 11 – 12 – 13 – 14].  $C4 = 7/17 = 41.2\%$

C1 y C3 tienen 4 aristas exclusivas. C2 y C4 tienen 2 aristas exclusivas cada uno.

Tomando C1 como base:

**C1 cubre en total 9 aristas + 4 exclusivas de C3 + 2 exclusivas de C2 + 2 exclusivas de C4 = 17 aristas.**

Por ello se requieren CUATRO casos de prueba para lograr una cobertura de decisión del 100%.

Aquí la clave es: **“CUBRIR ARISTAS”**

## **GENERALIDADES DE ESTA TÉCNICA:**

Lograr una cobertura de decisión del 100% requiere, como mínimo, los mismos casos de prueba que requiere la cobertura de sentencia; de hecho, en la mayoría de situaciones se necesitarán más casos de prueba en cobertura de decisión que en cobertura de sentencia.

Una cobertura de decisión del 100% siempre incluye una cobertura de sentencia del 100%.

Desventajas:

- No es suficiente para probar expresiones condicionales compuestas; solo tiene en cuenta el resultado final de la decisión.
- No es suficiente para probar bucles de forma extensiva.
- No considera las dependencias entre bucles.



### 3. COBERTURA DE CONDICIÓN

Aquí se tiene en cuenta la complejidad de una condición que esté constituida por múltiples condiciones atómicas.

El objetivo es detectar errores (OJO) que pudieran resultar de la implementación de condiciones múltiples (combinadas).

Las condiciones múltiples están constituidas por condiciones atómicas, que se combinan con el uso de operadores lógicos como OR, AND, XOR, etc.

Ejemplo:  $((a > 2) \text{ OR } (b < 6))$

Recuerden que las condiciones atómicas no contienen operadores lógicos, solo contienen operadores relacionales y el operador NOT ( $=$ ,  $>$ ,  $<$ , etc.).

Existe cobertura de condición SIMPLE y cobertura de condición MÚLTIPLE.

**COBERTURA DE CONDICIÓN SIMPLE:** Cada sub-condición atómica de una sentencia condicional combinada tiene que tomar, al menos una vez, los valores lógicos VERDADERO, así como el FALSO.

Considerar la siguiente condición: <b><math>((a &gt; 2) \text{ OR } (b &lt; 6))</math></b>		
Los casos de prueba para la cobertura de condición <b>simple</b> podrían ser, por ejemplo:		
$a = 6$ (VERDADERO)	$b = 9$ (FALSO)	$((a > 2) \text{ OR } (b < 6))$ : (VERDADERO)
$a = 1$ (FALSO)	$b = 2$ (VERDADERO)	$((a > 2) \text{ OR } (b < 6))$ : (VERDADERO)

Solo se necesitan DOS casos de prueba para lograr una cobertura de condición simple.

Cada sub-condición ha tomado los valores VERDADERO y FALSO.

Sin embargo, el resultado final es VERDADERO para ambos casos.

**COBERTURA DE CONDICIÓN MÚLTIPLE:** Todas las combinaciones que puedan crearse utilizando permutaciones de las sub-condiciones atómicas deben formar parte de las pruebas.

Considerar la siguiente condición: <b><math>((a &gt; 2) \text{ OR } (b &lt; 6))</math></b>		
Los casos de prueba para la cobertura de condición <b>múltiple</b> podrían ser, por ejemplo:		
a = 6 (VERDADERO)	b = 2 (VERDADERO)	$((a > 2) \text{ OR } (b < 6))$ : (VERDADERO)
a = 6 (VERDADERO)	b = 9 (FALSO)	$((a > 2) \text{ OR } (b < 6))$ : (VERDADERO)
a = 1 (FALSO)	b = 2 (VERDADERO)	$((a > 2) \text{ OR } (b < 6))$ : (VERDADERO)
a = 1 (FALSO)	b = 9 (FALSO)	$((a > 2) \text{ OR } (b < 6))$ : (FALSO)

Se necesitan CUATRO casos de prueba para lograr una cobertura de condición múltiple.

Se han creado todas las combinaciones de los valores VERDADERO y FALSO.

Por consiguiente, se han logrado todos los posibles resultados de la condición múltiple.

El número de casos de prueba se incrementa de forma potencial:

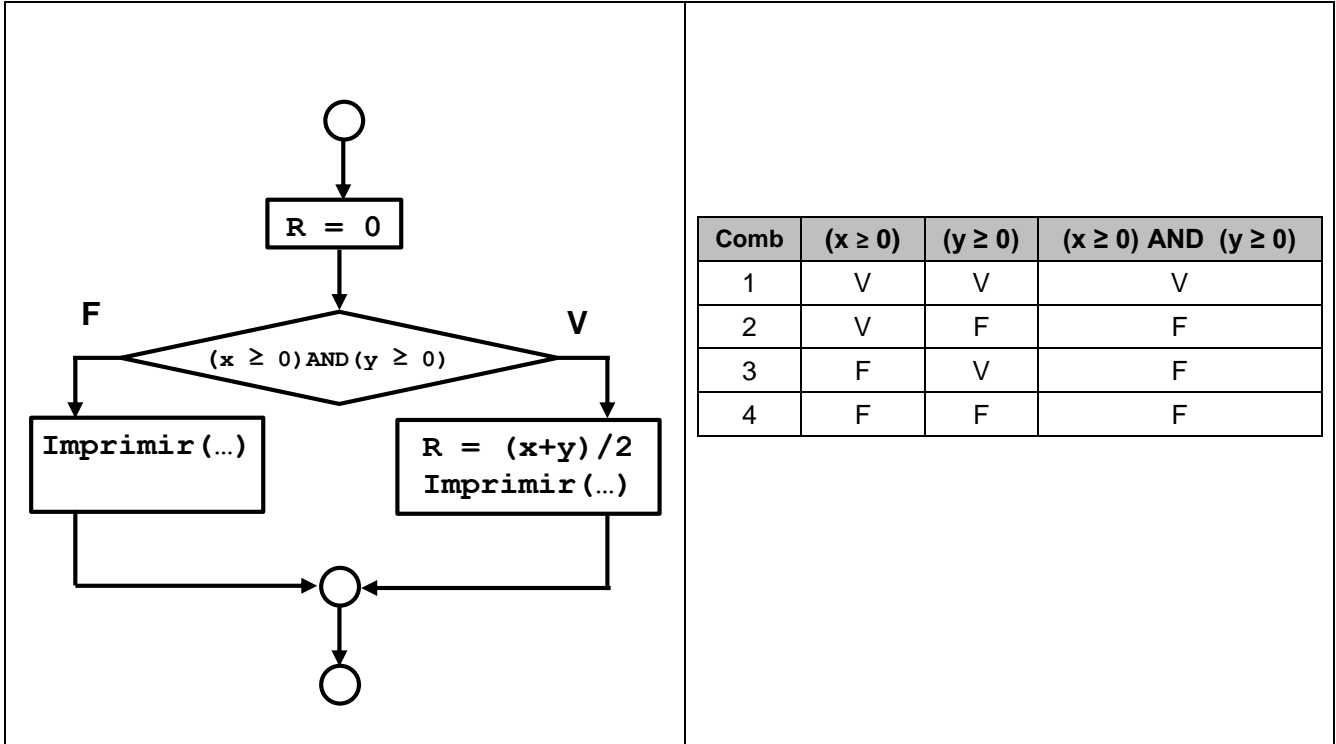
Si **n** representa la cantidad de condiciones atómicas,

la cantidad MÁXIMA de casos de prueba será de  $2^n$ .

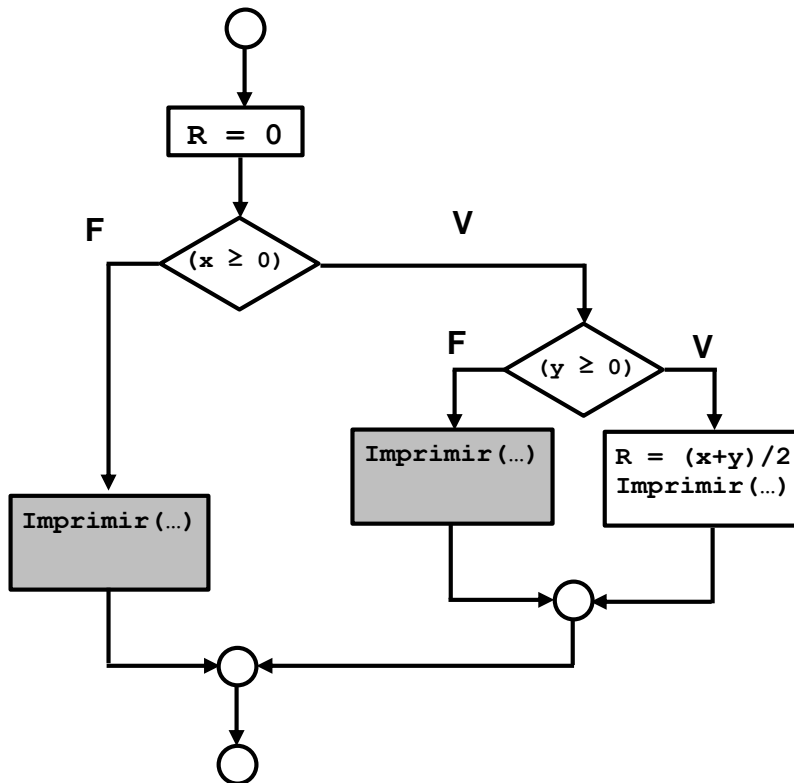
Sin embargo, se busca que la cantidad de casos de prueba requeridos en la práctica sea MENOR que ese valor máximo.

Veamos el siguiente ejemplo:

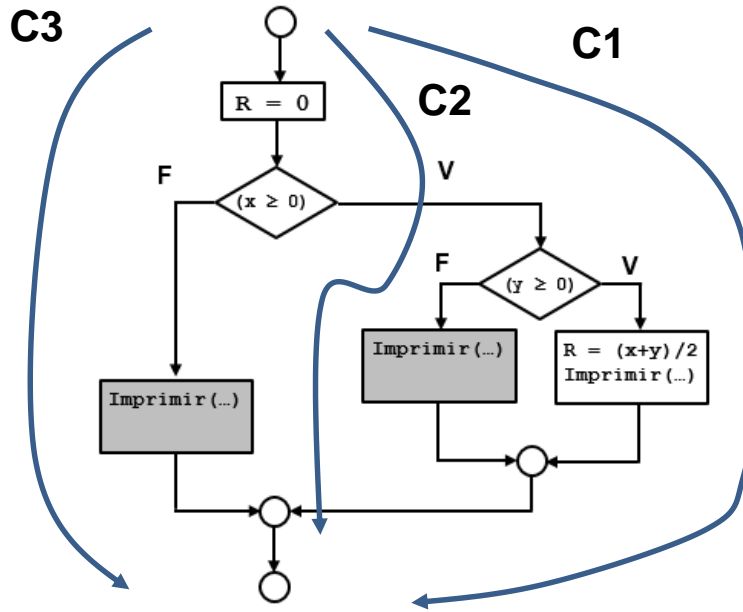
Dado este segmento:



Si descomponemos la condición múltiple, tenemos:



Y en consecuencia, solo se necesitan TRES CASOS DE PRUEBA, para cubrir el diagrama:



Comb.	(x ≥ 0)	(y ≥ 0)	(x ≥ 0) AND (y ≥ 0)	Camino
1	V	V	V	<b>1</b>
2	V	F	F	<b>2</b>
3	F	V	F	<b>3</b>
4	F	F	F	

CASOS DE PRUEBA	
C1: (x≥0):VE AND (y≥0):VE (Comb. 1)	[x=8] [y=0]
C2: (x≥0):VE AND (y≥0):FA (Comb. 2)	[x=0] [y=-1]
C3: (x≥0):FA AND y[Indiferente] (Comb. 3,4)	[x=-1] [y=...]

----- FIN DEL DOCUMENTO