

Testing de caja blanca

- Cobertura de enunciados o caminos básicos
- Cobertura de sentencias
- Cobertura de decisión
- Cobertura de condición
- Cobertura de decisión/condición
- Cobertura múltiple
- Etc

Cobertura: Estrategias para recorrer los distintos caminos que nuestro código nos provea para desarrollar una funcionalidad

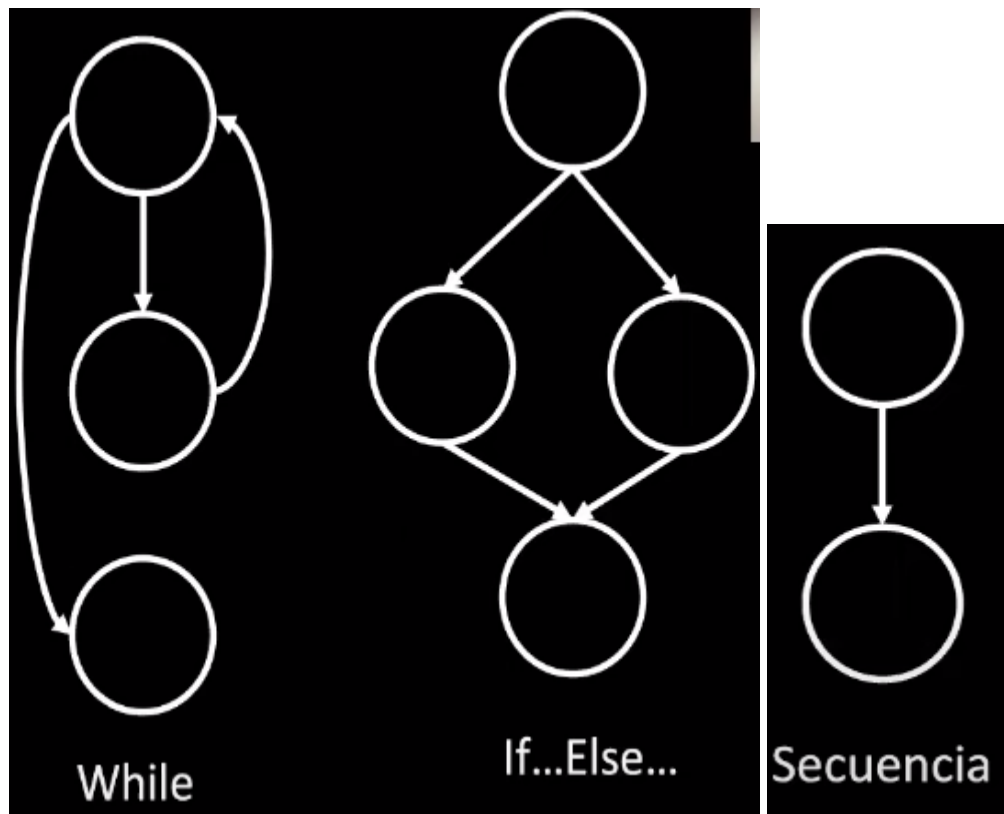
Caminos básicos: Todos los caminos independientes de nuestra funcionalidad los recorremos una vez

Cobertura de enunciados o caminos básicos

- Propuesto por McCabe
- Permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución

Para la prueba del camino básico:

- Se requiere poder representar la ejecución mediante grafos de flujo
- Se calcula la complejidad ciclomática
- Dado un grafo de flujo se pueden generar casos de prueba



M = Complejidad ciclomática.

E = Número de aristas del grafo

N = Número de nodos del grafo

P = Número de componentes conexos,
nodos de salida

$$M = E - N + 2 * P$$

$$M = \text{Número de regiones} + 1$$

- Es una métrica de software que provee una medición cuantitativa de la complejidad lógica de un programa
- Usada en el contexto de testing, define el número de caminos independientes en el conjunto básico y entrega un limite inferior para el número de casos necesarios para ejecutar todas las instrucciones al menos una vez

Complejidad Ciclomática	Evaluación del Riesgo
1-10	Programa Simple, sin mucho riesgo
11-20	Más complejo, riesgo moderado
21-50	Complejo, Programa de alto riesgo
50	Programa no testeable, Muy alto riesgo

Pasos del diseño de pruebas mediante el camino básico

- Obtener el grafo de flujo
- Obtener la complejidad ciclomática del grafo de flujo
- Definir el conjunto básico de caminos independientes
- Determinar los casos de prueba que permitan la ejecución de cada uno de los caminos anteriores
- Ejecutar cada caso de prueba y comprobar que los resultados son los esperados

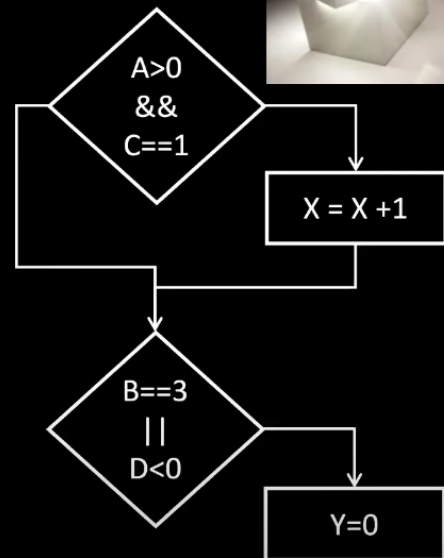
Cobertura de sentencias: Buscamos la cantidad mínima de casos de prueba que nos permiten ejecutar o evaluar todas las sentencias

Caja Blanca

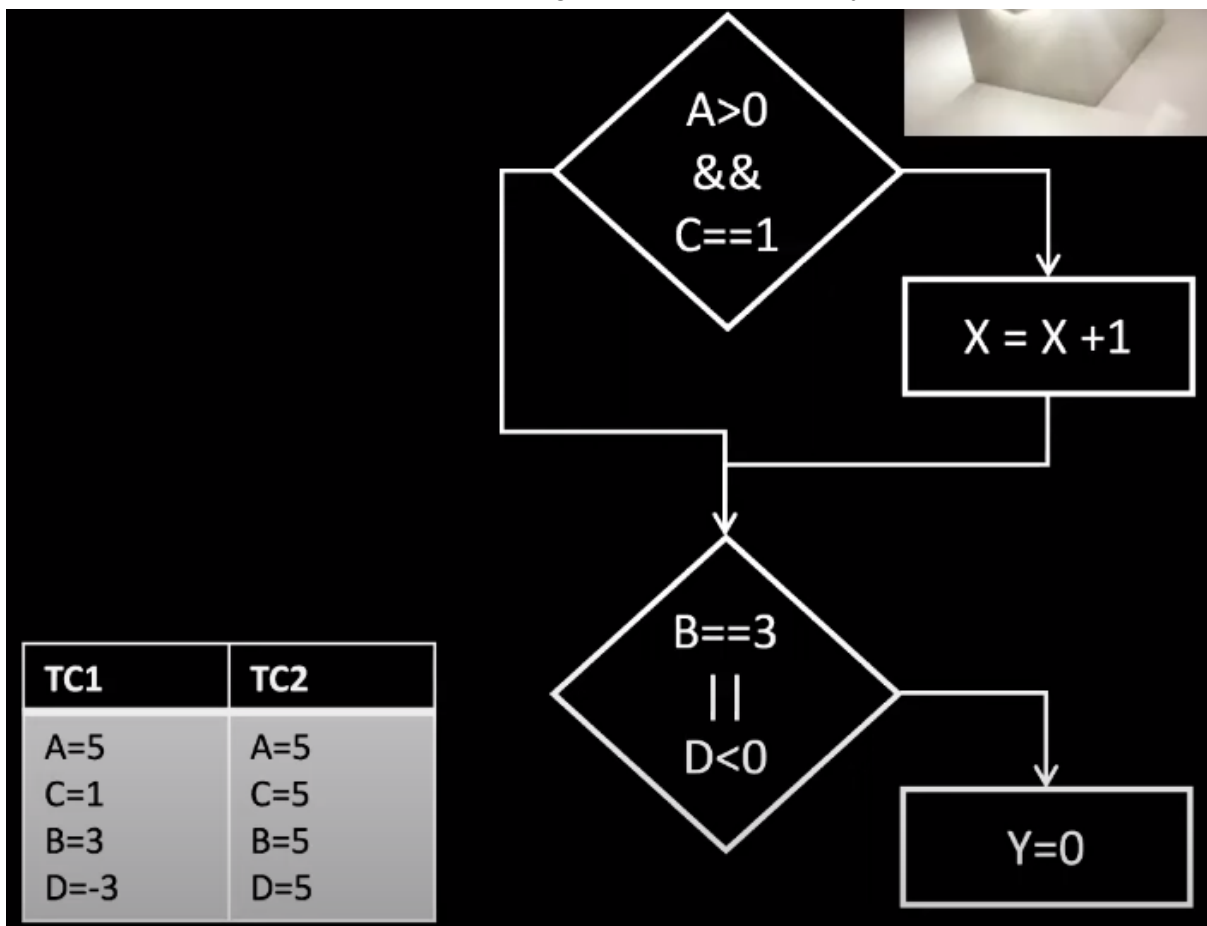
Cobertura de sentencias

```
IF (A>0 && C==1)
    X = X + 1
IF (B==3 || D<0)
    Y=0
END
```

TC1
A=5
C=1
B=3
D=-3



Cobertura de decisión: Una decisión es una estructura de control completa, cada estructura es cada uno de los rombos. Busco que las decisiones vayan por donde tienen que ir. No importa cómo valore los condicionales. Tengo que pasar por True y por False

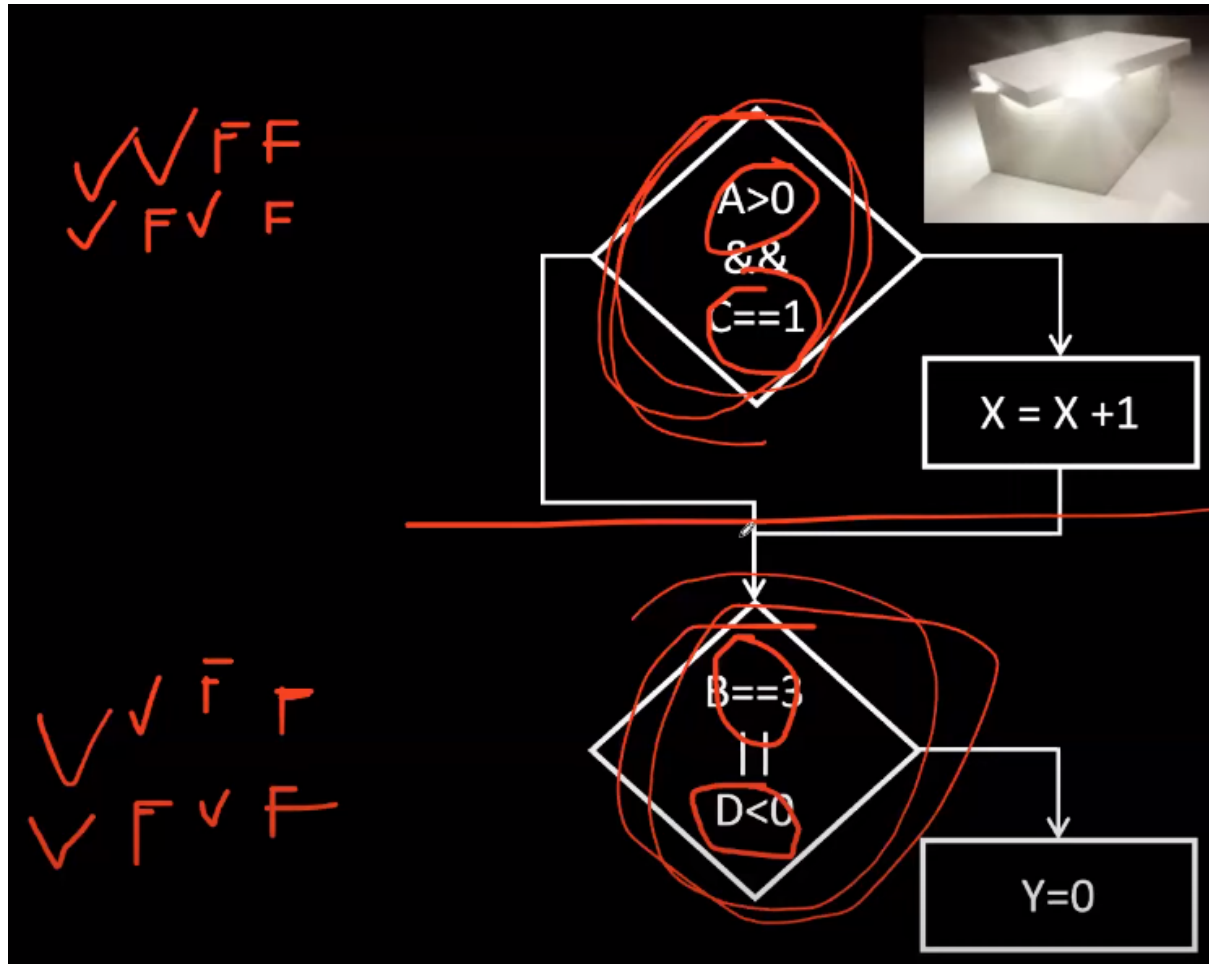


TC1	TC2
A=5	A=5
C=1	C=5
B=3	B=5
D=-3	D=5

Cobertura de condición: Busca la menor cantidad de casos de prueba que nos permiten valorar cada una de las condiciones tanto en su valor verdadero como su valor falso, independientemente de por donde salga la decisión. Obviamos si es Or o And

Cobertura decisión/condición: Busca valor tanto las decisiones como las condiciones en su valor verdadero y su valor falso

Cobertura múltiple: Busca valor el combinatorio de todas las condiciones en todos sus valores de verdad posibles



Ejercicio

Cobertura: De sentencias	Cantidad de casos de prueba: 5
Datos de los casos de prueba:	
TC1 <ul style="list-style-type: none"> BusquedaNumeroChapa = True Chapa a filtrar Se encontró numero de chapa = True Estado = "Ocupado" 	
TC2 <ul style="list-style-type: none"> BusquedaNumeroChapa = True Se encontró numero de chapa = False ... 	
TC3 <ul style="list-style-type: none"> BusquedaNumeroChapa = True Se encontró numero de chapa = True Estado = "Libre" 	
TC4 <ul style="list-style-type: none"> BusquedaNumeroChapa = True Se encontró numero de chapa = True Estado = "Solicitado" 	
TC5 <ul style="list-style-type: none"> BusquedaNumeroChapa = True Se encontró numero de chapa = True Estado = "Fuera de servicio" 	
Precondiciones:	

